2073

CASSINI/RPWS Instrument

V2.6.5/SOI

Users Guide and Software Operations Manual

The University of Iowa
Department of Physics & Astronomy
700 Van Allen Hall
Iowa City, Iowa
52242

Revision 1.16

March 9, 2004

William Robison
Terrance Averkamp
Robert Brechwald

Custodian: William Robison

Revision Notes

Flight Software Revision 1.6 Feb 1997
Flight Software Revision 1.7/2.0 May 1997
Flight Software Revision 2.1 Jun 1997
Flight Software Revision 2.2 Jly 1997
Flight Software Revision 2.3 Jly 1998
Flight Software Revision 2.4 Dec 1999
Flight Software Revision 2.5 Dec 2001
Flight Software Revision 2.6 Dec 2002
Flight Software Patch 2.6.5 August 2003

Document Revision 1.0 contains last pre launch updates
Document Revision 1.2 contains post-deploy updates
Document Revision 1.3 contains ICO updates
Document Revision 1.4 contains post-ICO updates
Document Revision 1.5 contains Venus-2/Earth updates
Document Revision 1.6 contains post V2/E1 updates
Document Revision 1.7 is for Flight SoftwareV2.4
Document Revision 1.8 is for Flight SoftwareV2.5
Document Revision 1.9 is for Flight SoftwareV2.6
Document Revision 1.10 is for Flight SoftwareV2.6
Document Revision 1.11-1.14 is for Flight SoftwareV2.6/SOI
Document Revision 1.15 is for Flight SoftwareV2.6.5/SOI
Document Revision 1.16 is for Data Archiving Effort

Original Document edited with
StarOffice 6.0
and saved in SXW format
spica://opt/project/cassini/devel/document/ug_som.sxw

RPWS Archiving Effort Notes

This manual has been saved in several formats as part of the Cassini/RPWS archive effort.

The original work, as noted on the previous page, is in the native Staroffice-6 format. As the archive volume is produced, we attempt to produce several additional formats for the convenience of the target audience.

The SXW file, the native Staroffice-6 format, is copied to the archive volume with only a name change. NO other alterations have been made to the native format of the document.

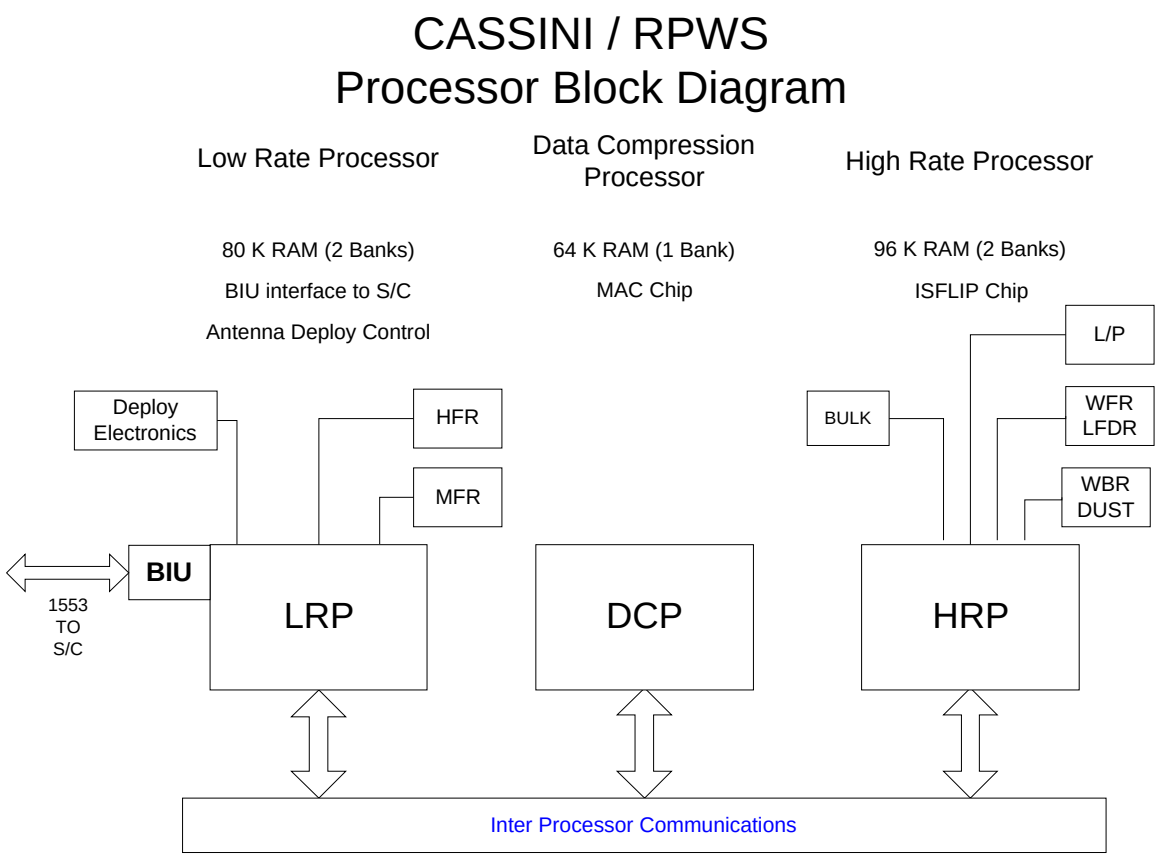
Staroffice-6 has a capability to produce an HTML file that is used to produce the RPWSUG.HTM that appears on the archive volumes. Natively, Staroffice-6 produces *GIF* images for any graphical figures in the document. A name is derived using the name of the parent document and appending hexadecimal numeric string. As this is done on a Unix system, it is all lowercase. As the HTML files are moved to the archive volume, several alterations are made to better suit the target environment. The primary document is renamed to RPWSUG.HTM and placed in an appropriate directory on the target volume. All of the *GIF* files are translated to *PNG* format as they are copied and their names are changed to uppercase. In addition, they are placed in a subdirectory to reduce clutter in the directory where several versions of the parent document are stored. As expected, the HTML is altered to reflect the placement and casechange applied to the *GIF/PNG* files.

In order to produce the PDF file, we start with an intermediate PostScript file (which Staroffice-6 will produce) and convert to PDF using a utility called ps2pdf13. This produces a version 1.3 PDF file that appears to work with Adobe Acrobat. As with all of the version of the user guide, it is name changed to RPWSUG.PDF.

The PDF file should reproduce the best hardcopy. It should reproduce very close to what we produce at Iowa with tables and graphics intact.

1 Instrument overview

The CASSINI instrument may be broken down into several functional parts. The primary control and interface with the spacecraft is performed by a set of 8 bit processors (8085) designed and built at Iowa. Data communications with the spacecraft are directly controlled using a 1553 protocol controller supplied by JPL. The sensor and signal conditioning (i.e. the receivers) hardware was designed and built by the science teams from Iowa, France, Sweden, and England. In addition to the hardware used to perform science there is additional hardware used to deploy the electric antenna mechanism.

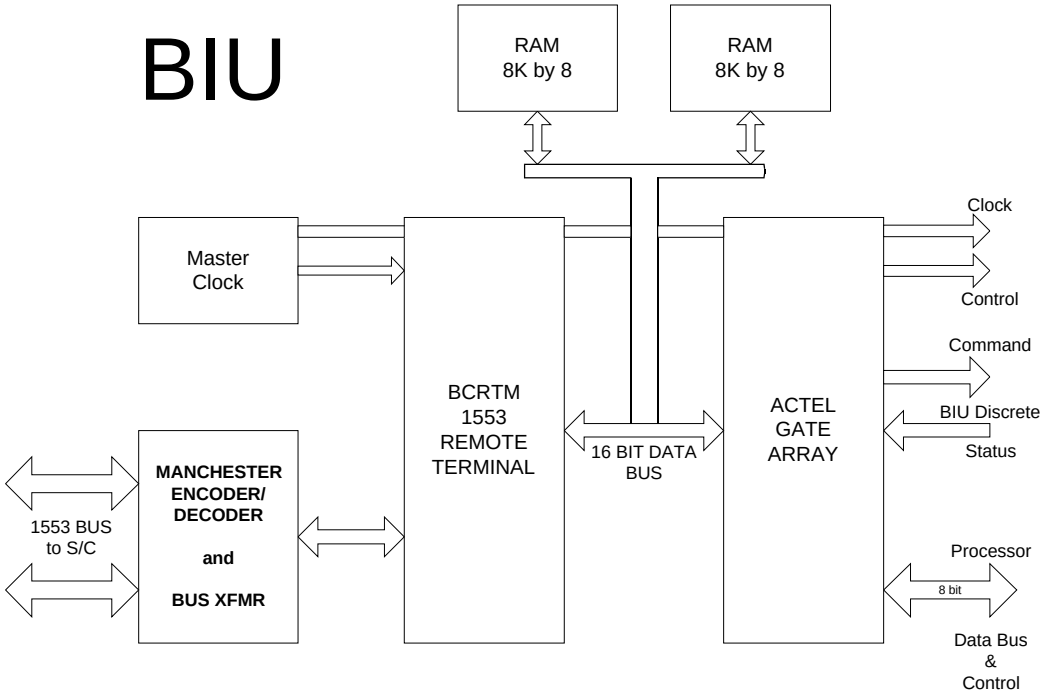


1.1 BIU

The BIU is the sole data channel used for communications between the instrument and the spacecraft. All command and control, data, and housekeeping are delivered through the BIU. The BIU uses a single physical line (the 1553 bus) to present a group of logical circuits (1553 sub address) to the DPU allowing commands, data, and housekeeping to be handled in a logically separate fashion.

The logical circuits (the telemetry sub address) can be further broken down into data channels (high rate science and low rate science) with each data channel delivering a separate flow of data. (This division applies to commands as well)

The DPU is responsible for managing the resources (i.e. memory allocation) within the BIU.



1.2 DPU

The DPU consists of 3 **8085** processors interconnected with an 8 bit parallel communications channel. Two of the processors use more than 64K bytes of memory that the **8085** architecture allows through the use of bank select hardware using the **8085** ***SOD*** signal. The third processor treats the ***SOD*** signal in the same manner to maintain compatibility.

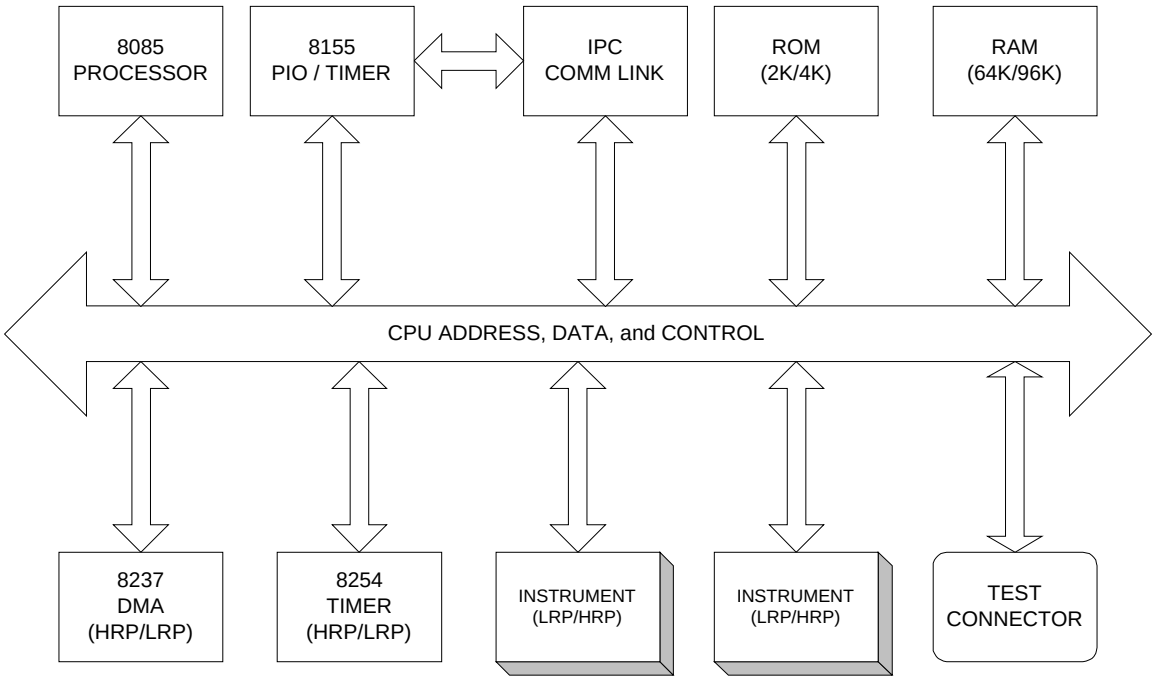
The memory architecture is effectively identical on all processors. A banking mechanism allows the processor to make use of more than 64K bytes of memory. The LRP with 80K bytes has the BIU present in the top of memory with the area from 32K to 48K banked. The HRP with 96K bytes banks all of the memory above 32K. The DCP with 64K memory implements the bank select mechanism in the same manner as the LRP/HRP although it has no effect.

Each processor has several peripheral chips that are configured identically on each processor to allow use of common software. An **8155** parallel port with timer, an **8254** counter/timer and an **8237** DMA controller are present on **LRP** and **HRP** while the **DCP** has only the **8155**.

The communications path between the processors is a half duplex 8 bit bus. Although logically identical on all processors the physical implementation differs on the three processors. The DCP has the control circuit implemented in discrete logic while both the LRP and HRP implement the control logic within a gate array. The port addressing differs between DCP and HRP/LRP due to the radically different I/O requirements although this has no impact as the IPC driver is not generic enough to take advantage of code duplication at the ALF level due to identical hardware (although the entire IPC driver does reuse most code at the sources level).

The driver can not be generic as the implementation is rather asymmetric in order to support a special high speed channel from the HRP to the LRP. The source code, however, is essentially identical for the most part with small parts of the code that is unique to each processor. The activity required to build the driver causes the 3 individual components to be assembled with the required code fragments enabled.

COMMON DPU CONFIGURATION



1.3 Antenna Control

Following successful deploy on 25 October, 1997 no additional details will be added to the users guide describing deploy activities(antenna temperature details are the only information that will be updated from this point on).

Antenna control consists of an A.C. drive circuit and necessary switching to allow any one of the three antenna mechanisms to be deployed. Several levels of interlocks are present to avoid the possibility of moving an antenna element at an unintended time.

The LRP is used as a final level of *intelligence* in controlling the antenna mechanism. No hardware interlocks are provided to monitor limit switches, motor current, position or temperature. The control electronics is used to generate the AC waveform required by the motors.

As one might expect, the antenna control routines are not typically loaded into the LRP. We would expect to purge any deploy software resident on the spacecraft once the antenna elements have been deployed (In normal circumstances the deploy software is not loaded into BULK MEMORY so a processor reset is sufficient to purge the deploy software from memory).

The antenna control hardware consists of a 2 phase AC signal generator and amplifier along with relays used to route the AC drive signal to any one of the antenna deploy mechanisms. Directional control is achieved by exchanging the signals delivered to the amplifier. Signals from the BIU are used to enable relay commands to prevent software problems from activating the antenna mechanism.

1.4 Bulk Memory

The block of indirectly accessible memory attached to the HRP is referred to as *BULK memory*. This memory block is addressed by the HRP using a pair of address ports and a data port. A 16 bit address is loaded into the 2 address registers to allow access to a single 8 bit storage location in the *BULK Memory*. The 16 bit address provides access to all 65,536 locations of *BULK memory*.

The ROM makes use of this memory to implement a fault recovery mechanism. When the ROM code gains control of the processor, it will attempt to read the contents of the BULK memory in 44 byte records. The records will be processed as if they were download records from the spacecraft. If the records are validated the contents will be loaded into memory.

The BULK memory may be loaded from the spacecraft using the normal memory download mechanism. Each download record contains a bit that is used to target the memory image for one of the three processors and the BULK memory (See the section on *memory download* for details).

The us of BULK memory as part of the fault recovery mechanism is controlled by a bit in the memory download record which. The download image may be updated, as needed, by the ground. The BULK memory may, therefore, be used for other purposes if required at a later time (i.e. the initial download image is loaded into the BULK memory and will be reloaded into memory if the processor is reset by the watch dog timer).

1.5 Power Supply

The power supply is designed to allow the DPU to control power to each of 3 receivers independently.

The Langmuir probe is switched both in the power supply (analog electronics) and on the HRP (digital electronics) where the data interface exists. The Langmuir probe power switching is interlocked in hardware to prevent power from being applied to the analog section when the digital section is not powered.

Although the WBR and WFR analog converters are switched separately from the analog electronics, there is no internal requirement that they be switched on in any particular order (although external requirements may dictate the order of internal power switching).

There is no internal requirement for the order in which power is switched, any dependencies are interlocked in hardware to prevent latch-up etc. Switching transients will, however, cause problems on the spacecraft if not sequenced correctly. The operating software provides a means to sequence power correctly.

Also note that the instrument is capable of presenting power transients to the spacecraft if power is applied carelessly. In most cases it is required that all of the receivers be powered down prior to application of power in order to allow the receivers to be switched on in the correct order. The HFR, for example, will cause RPWS to exceed power allocation if it is switched on when either the L/P or WBR/WFR/MFR are powered.

1.5.1 BIU power requirements

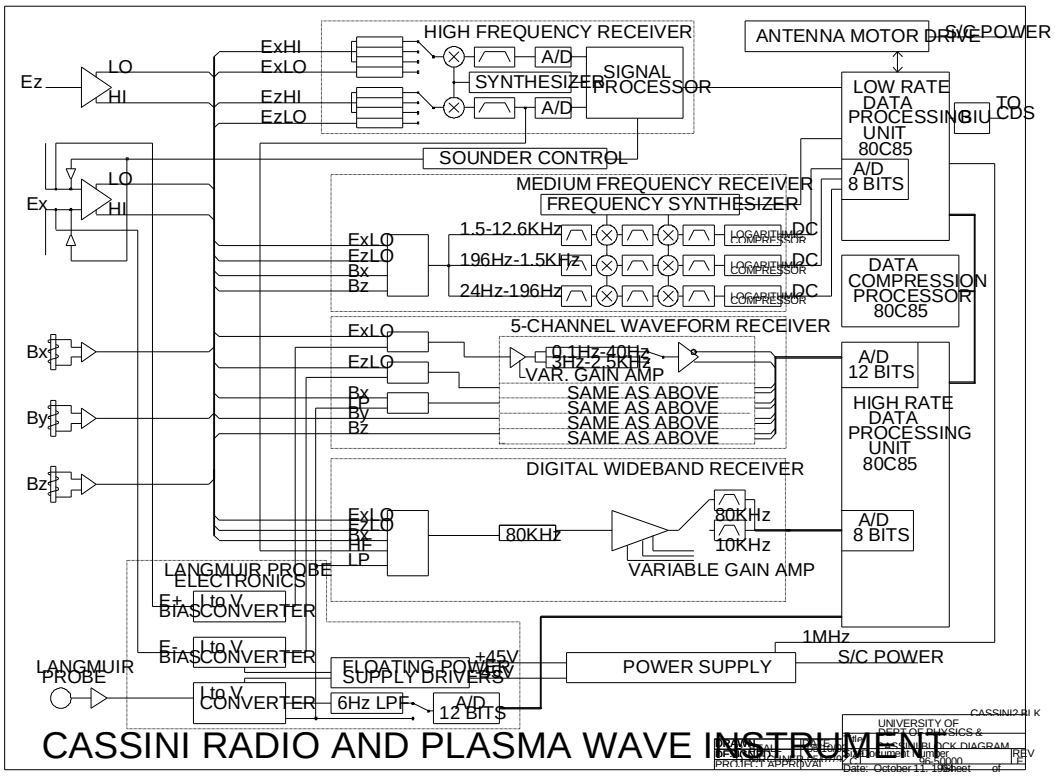
The BIU requires a noticeable increase in power during transmission. The delta current on the 5 volt supply seems to be on the order of a 150mA-200mA increase.

The time constant in the power supply monitor circuit is rather short so BIU operation is visible in the housekeeping channel that monitors digital logic current. BIU pickups and housekeeping interact in unexpected ways.

1.6 Receivers

The receivers perform the analog signal processing required to allow the A/D system within the DPU to acquire data. Power to the receivers is switched and under the control of the LRP.

The receivers are switched in 3 groups: the Meudon HFR, the Upsala Langmuir probe, and the Iowa MFR, WFR, and WBR.



1.6.1 MFR

The MFR interface is resident on the LRP and consists of the following parts.

1. 3 Channel mixer clock (dedicated 8254)
2. 3 Channel A/D converter (shared with housekeeping channels)
3. Antenna Control bits

Control of the MFR is shared by hardware and software. The software driver controls sweeping of the instrument by triggering the A/D converter and reprogramming the 8254 clock. Timing is derived from the operation of the BIU and is, therefore, slaved to the data collection cycle of the Spacecraft.

The MFR shares the A/D converter with other housekeeping functions. The A/D must be properly handled to allow for proper settling of the various multiplexers. The MFR handler and the Housekeeping handler share operating procedures designed to provide proper multiplexer settling allowances. Sharing the A/D resources imposes some timing restrictions that limit the peak rate at which MFR can operate as well as the peak rate with which housekeeping data can be collected.

1.6.2 HFR

The HFR interface is resident on the LRP and consists of the following parts.

1. Command channel (3 control lines, shared port)
2. Data channel, 8 bit (uses 8237 channel)
3. Data timing control (in ACTEL gate array)

The HFR is an autonomous system and is controlled by the issue of serial commands. The HFR driver commands the HFR using a serial command interface with the shift register implemented in software.

Data retrieval from the HFR is assisted by the use of one channel of the 8237.

The timing of data collection activities is controlled by an edge on one of the signals of the command interface. This line is under control of the HFR driver software and is synchronized with the data collection cycle of the spacecraft to within approximately 10 mS.

1.6.3 WBR

The WBR interface is resident on the HRP and consists of the following parts.

1. Single Channel A/D converter
2. Antenna Control bits
3. Hardware compression chip.
4. Timing control (ACTEL gate array)

The WBR control is primarily hardware. Data movement is supervised by an 8237. Antenna and filters are selected using a dedicated control port.

The timing of data collection activities may be controlled by a signal generated on the BIU that is controlled by the data collection cycle of the spacecraft. Timing accuracy is limited by the spacecraft (i.e. no software latency within the instrument). Data collection may also be run unsynchronized to achieve higher data rates.

There are *quirks* to obtaining proper operation when using the compression chip.

1.6.4 WFR

The WFR interface is resident on the HRP and consists of the following parts.

1. 5 Channel A/D system (2 converters and associated multiplexer)
2. Antenna Control bit
3. Hardware compression chip.
4. Timing control (ACTEL gate array)

The WFR control is primarily hardware. Data movement is supervised by an 8237. Antenna and filters are selected using a dedicated control port. The hardware controls multiplexing 5 analog inputs through the two available A/D converters.

The timing of data collection activities is controlled by a signal generated on the BIU that is controlled by the data collection cycle of the spacecraft. Timing accuracy is limited by the spacecraft (i.e. no software latency within the instrument).

The compression chip is shared with the WBR. Only one analog converter may be attached to the compression hardware at any time.

1.6.5 L/P

The L/P interface is resident on the HRP and consists of the following parts.

1. 8 bit data channel (command and data) using shared 8237 channel.
2. programmable clock (timer in 8155)

The L/P control may be assumed by either hardware or software. Data movement may be supervised either by the 8237 or under software control. The 8237 is used for higher sampling rates with the switch to software required for sample rates that cannot be accommodated by the timer in the 8155.

Control bits are implemented within the L/P electronics, external to the HRP. These control bits are accessed by writing to the appropriate address that accesses the L/P data channel.

No attempt has been made in the L/P driver to exercise high quality timing control. Scheduling accuracy is limited to the basic system time reference which is 1/8 second. Note that very low sample rates are achieved using software control, resulting in some jitter of the sample clock (i.e. when not using the 8155 for timing).

2 Software Overview

The DPU contains minimal software in ROM. The ROM provides minimal capabilities, basically sufficient capability to operate without interfering with the spacecraft (i.e. the 1553 controller is configured in a minimal manner to allow software downloads and provide valid ancillary data to the S/C). For any science data to be gathered, an appropriate software load must be supplied to the instrument.

The downloaded software supplied at launch consists of a ***science load,*** a ***deploy load,*** and possibly a ***maintenance load.*** The science load, as it's name implies is intended to operate the receivers, collect and deliver scientific data. The deploy load is used to operate the antenna mechanism without ever powering the receivers (or collecting science data). The maintenance load is intended to address some of the shortcomings of the ROM software (allows biasing the Langmuir Probe to be biased to any voltage).

The deploy software is used during the early portion of the mission for antenna deploy activities. Once the antenna elements have been deployed, this software is no longer required and should be removed from the SSR to prevent any possibility of moving the antenna elements.

The maintenance software will be used at several points during the cruise phase of the mission. This load may be considered expendable in the event that a new science software load is to be loaded onto the SSR (the maintenance software should reside in the non-default partition of the SSR).

The science software is, obviously, required for any science activities as the ROM and other loads do not allow for telemetering science data (both deploy and maintenance software make use of the housekeeping channel only).

There is nothing that precludes the creation and use of any future software. Any download that meets the restrictions imposed by the ROM memory architecture should be able to be used in the instrument.

2.1 Real Time Kernel

This is the operating system for the instrument. The kernel mediates access to the hardware resources. The kernel is identical on all processors (as is the core hardware complement).

The kernel mediates access to most of the system resources such as the CPU, memory buffers, interprocess communications, hardware and interrupts. The primary user of these resources are application programs that are referred to as a process. Each process has associated with it a descriptor that is used by the kernel to manage the process. There is usually no reason to access fields within the process descriptor although some degree of command and control from the ground may be exercised through the fields within the process descriptor.

The kernel services include process control in the form of prioritized process management (i.e. a priority sensitive dispatcher), queue management, interrupt control (flag services), time control (S/C time field maintenance), some utility routines, and an exclusive access control mechanism.

At the beginning of each RTI period, the kernel receives control through the RTI interrupt handler. The interrupt handler implements an access mechanism that allow application programs to gain control of the CPU at interrupt level, updated the system time field, scans the delay list for processes that are ready to use the CPU and finally makes a pass through the dispatcher to schedule any processes that are ready to use the CPU.

Other interrupts may return control directly to the interrupted process or through the dispatcher. As an example, the 8237 interrupt handler will return control to the interrupted process when no process is waiting on one of the 8237 interrupts, but return control to the dispatcher when a process in a flag wait state on one of the 8237 flags. This will allow higher priority activities to gain control of the CPU when an interrupt causes a process to become unblocked, but at the same time minimize the number of CPU cycles devoted to process context switching.

A queueing system provides a means to pass messages, such as data and commands, between processes. The kernel provides for run-time binding of the queues so that processes may be built independently (the address of the queue control structures are bound at run time, not link time). Queue activities can also cause the dispatcher to run for the same reason. A process that sends data to a queue may unblock a process waiting on data and activate the dispatcher. Again, there is an effort made within the kernel to keep the number of CPU cycles devoted to process context switching to a minimum.

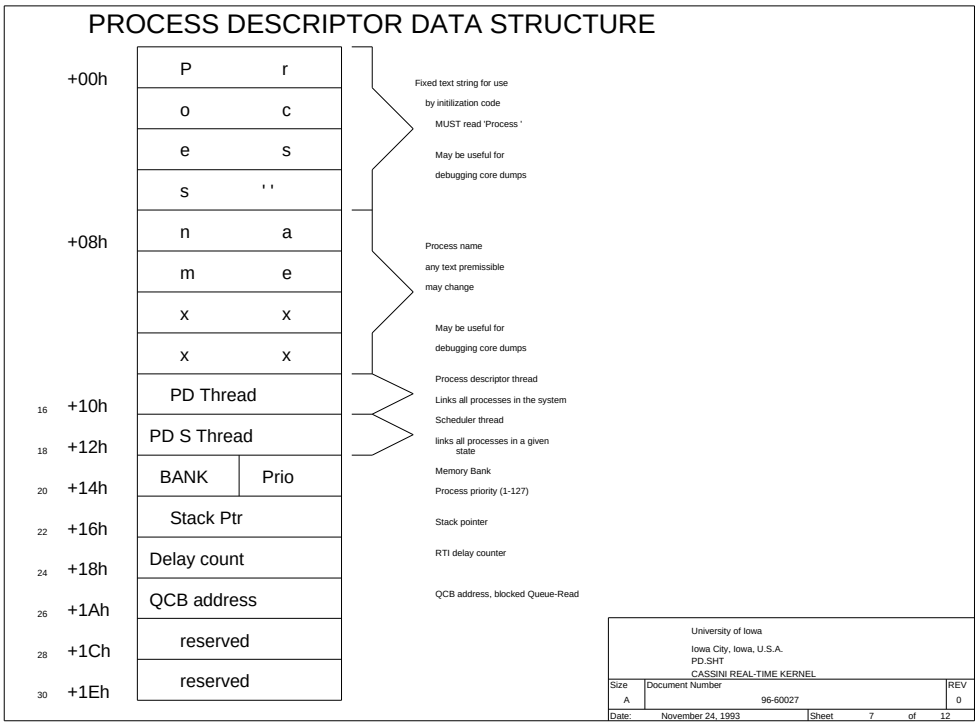
Interrupts are abstracted through the use of a mechanism referred to as a flag. A process that needs to wait for an interrupt to occur (typically waiting for a DMA activity to complete) releases the CPU by calling the *Flag Wait* system service. The corresponding interrupt service routine may then release the blocked process for execution by invoking the *Flag Set* system service. Application code will not contain interrupt handlers and need not attach to the interrupt system of the 8085 in a direct fashion.

Shared resources, other than the CPU, are managed using a set of system services referred to as *MX Flags*. Any process that requires exclusive use of shared resource must acquire the appropriate flag using the *MX Flag* call before using the shared resource and the corresponding release following use of the resource. In the event that a 2nd process attempts to acquire the same resource, it will be blocked until the 1st process releases the resource.

2.1.1 Process Descriptor

Fields marked in the ***Crit***ical column should not be altered as this would, in most cases, cause the software to crash.

Process Descriptor			
Crit	Offset	Size	Description
	0	8 characters	Identification string "Process "
	8	4 characters	Process Name
	12	4 characters	Process Version
*	16	16 bits	Process Thread
*	18	16 bits	Scheduler Thread
	20	8 bits	Bank Select
	21	8 bits	Process Priority
*	22	16 bits	Stack Pointer
	24	16 bits	Delay Count
*	26	16 bits	QCB Pointer



2.2 BIU Handler

Bus Interface Unit. The BIU handler manages the 1553 interface electronics and protocol controller chip. The BIU handler is responsible for moving data into the BIU memory buffer and removing command/status information for the BIU memory buffer. Command buffers are delivered to the 1st level command decoder.

The BIU Handler also manages and formats telemetry data. Data delivered directly to the BIU from the HRP is managed by the BIU handler although the BIU handler does minimal work on this data (this is referred to as *High Rate Science* data). All other science telemetry is formatted by the BIU handler for delivery to the spacecraft data system.

All of the telemetry data products managed by the BIU handler are formatted into fixed size blocks before being released for delivery to the S/C data system. Each block is used completely (i.e. no more than 32 bits of wasted space) before being released to the S/C. This behavior must be kept in mind when performing diagnostics on the instrument and when operating at very low bit rates. As an example, when the MFR is the only instrument powered and running (producing data at 56 bits/second), data is released to the S/C every 2 ½ minutes even though the instrument produces a data set every 32 seconds.

This packetizing is particularly evident when performing diagnostics on the instrument when the receivers a powered off. In this scenario there is not a source of recurring data to push the diagnostic data (such as MRO packets) through the BIU.

With the introduction of the V2.5 flight software, the BIU handler specifically avoids allowing the 8237 to remain active when the BIU handler is writing to BCRTM registers during the dead time handler.

2.3 Command Subsystem

We can break commands down into several broad classes based on where and how they are processed. The **73ALF** commands are used by the ROM software to load the operating software into the instrument. They are distinct in that they have parity that is inappropriate for the command decoder within the science software. In addition, science software recognizes the unique properties of the **73IEB_LOAD** command (this command is the only command available to the science software with a length encoding scheme that encodes a total length of more than 64 bits).

1. 73ALF commands (wrong parity)
2. 73IEB_LOAD commands (wrong length)
3. All remaining commands (proper parity and length)

Science commands are decoded using a two level decoding scheme. Embedded in the first word of each command are four destination bits and three length bits. The destination field allows for a 1 of 16 select. The length field allows commands of from one to eight words. The initial word of a command contains 2 parity bits that are checked as the command is extracted from the incoming command buffer. Invalid parity causes the remaining contents of a command buffer to be discarded (which implies that multiple commands may arrive from the spacecraft in any RTI).

73ALF commands are simply discarded by the science command decoder as they have invalid parity. The science software does not, then, have an effective method for loading software patches other than the 73MEM_TWEAK command, which is capable of loading 8 or 16 bits of memory.

73IEB_LOAD commands must arrive alone (i.e. 1 command per RTI period) as these commands do not follow the normal length encoding method.

The presence of the length bits in the remaining commands allows 1st. level command decoding to separate the incoming commands for delivery to the 2nd level decoding. The IPC mechanism is used for transporting commands between 1st and 2nd level decoding. As might be expected, the 1st level decode is performed on the LRP within the BIU handler.

2nd. Level decoding is performed by the process that actually makes use of the command. As such, the 2nd. Level command decoder is distributed across all of the processors.

Note that the 1st level decoder performs a dual parity check on the 1st word of the command and requires 7 bits for destination/length leaving 7 bits for use by the 2nd. Level decoding. Although not a strict requirement, the 2nd. Level decoders tend to use 2 or 3 bits as a function code to implement individual commands.

As mentioned above, when the 1st. level command decoder encounters a parity error, the remainder of the command buffer is discarded. This can result in additional lost commands when multiple commands are delivered in a single RTI period.

A block diagram of the RPWS command format is show in the **RPWS Commands** section that follows.

Also note that the 73ALF command index, 0010, could be assigned to a class of commands for use with the science software. Not only do we have a good idea of when commands arrive at the instrument (i.e. little danger of sending an ambiguous command when the instrument is in ROM vs. Science), but the reversed parity keeps ALF commands from being executed by the science software and a science command being executed by the ROM software.

2.3.1 Command/IEB Flow

A discussion of the command/IEB system follows. Use the following diagram to better understand how the systems are interconnected.

BIU handler accepts commands from the spacecraft and takes care of updating the command byte counter. The command byte counter, therefore, only registers commands that are delivered to the instrument from the spacecraft.

CMD, the command decoder,

2.4 IPC Handler

Inter Processor Communications. This is the half duplex, 8 bit communications channel that connects the 3 processors. As the channel is half duplex and connected to all processors, only two of the processors may use the bus at any time.

The software driver completes the IPC subsystem. The driver handles incoming and outgoing datagrams, performing the required routing tasks. The IPC handler provides a generic data transport system.

The command decoding and delivery mechanism makes use of the IPC driver to deliver commands from the 1st level command decoder to the various handlers. The IPC driver will route traffic to any of the three processors (including the local processor).

2.5 IEB Handler

Instrument Expanded Block. Internal data collection scheduling.

This handler allows operating sequences to be downloaded into the instrument in order to significantly reduce the volume of commands to be delivered to the instrument to accomplish an observation.

This handler processes what are, effectively, memory downloads in the form of **73IEB_LOAD** commands. These loads are placed in an area of memory reserved for holding these sequences.

The handler interprets the sequence and delivers the embedded commands when commanded using the **73IEB_TRIGGER** commands.

2.6 Watch Dog Timer

The watchdog timer is a combination of hardware and software. The hardware portion is located on the LRP and physically resides in the Actel gate array. The software is spread across all three processors. The interface between hardware and software occurs as part of the *time lock* process on the LRP.

The hardware consists of a 10 bit counter that is clocked at 7.629 Hz. When the top bit goes from a 0 to a 1, the 8085 is reset. The timeout period for the watchdog timer is about 67 seconds. The timer is reset by writing to a pair of I/O addresses (on the LRP, of course). As long as both of the ports are accessed (prior to the 8085 being reset) the timer is reset and the 8085 will continue to operate without interference. The HRP has a similar circuit but the code constantly keeps it reset, effectively disabling it.

The software portion consists of a regularly scheduled process that accesses the reset port (this occurs in the *time lock* process on the LRP), and a means to include the other processors in the loop so that the failure of any of the three processors will force a software reload (the mem tweak process already passes commands from LRP to HRP to DCP and back to LRP, so it was a good candidate for assisting)..

The hardware interface, as mentioned earlier, resides in the *time lock* process. This process is responsible for keeping DCP and HRP time updated and runs on a regular basis (i.e. more frequently than the hardware timer). There is an override flag (that may be altered by using a ground command) that disables the watch dog timer (by continuously accessing the reset ports), and a count-down timer that suppresses writes to the reset port when it reaches zero (followed, in good time, by a reset pulse to the 8085).

The software makes use of the *IPC handler* and *mem tweak* process to test all three processors. The time lock process, on a regular basis, sends a time update to DCP and HRP causing a packet to circulate through all processors and return to the LRP where the housekeeping page is updated along with resetting the counter used in the time lock process to continue to write to the watch dog timer reset registers.

IPC handler simply routes traffic between the 3 processors. Failure of this handler inhibits delivery of traffic. Mem tweak commands are passed through all processors and back to LRP with each IPC buffer causing a field in the housekeeping to increment (loop count). The counter gives the ground a little visibility into the health of the processor complex. This loop counter is used by the watch dog thread (in the time lock process) to reload the software timer. As long as ***73mem_tweak*** commands continue to circulate through all processors, and the loop count in housekeeping increments, the hardware watch dog timer continues to be accessed to prevent the reset from occurring.

2.6.1 WatchDog Timer Event 2002-271T07:09:50

The first watch-dog timer trip occurred on **2002-271T07:09:50** and was caused by an attempted transition from a high rate trigger (with 80Khz WBR) to a low rate trigger (Trigger 28, Interplanetary Cruise). With FSW release V2.6, we began to realize that the method used to shut down high rate science data on the HRP was basically flawed and was susceptible to causing the HRP to hang. Finally, at the above time, the HRP did succumb and stopped processing IPC traffic. As a result, the LRP no longer received looper traffic and stopped retriggering the watch dog timer on a regular basis. The LRP reset was triggered shortly after this point. This event demonstrates that the watch dog timer is effective in recovering from a processor failure. What has not been demonstrated in flight is the capability to autonomously recover from a failure without assistance from the ground. This particular event was addressed by sending a set of patches to address 3 issues followed by a power-up command and a trigger.

The patches for the V2.5 software that was in use at the time fix the following bugs:

1. WBR AGC holdoff. Patch in a missing instruction.

2. WFR/LFDR Mx when WBR High Band active. Patch a pair of addresses that were in error.

3. Patch out the command that caused the WDT trip.

Version V2.6 software (and subsequent IEB loads) do not contain the offending instruction, although sending **73IEB_HALT, IDLE** to the instrument will cause the problem sequence to occur.

Version V2.7 removes the problem code sequence entirely. The **73POWER_CNTL, PAUSE** command is processed but no hardware actions are taken as a result of this command. **73IEB_HALT, IDLE** command also has had the **73POWER_CNTL, PAUSE** command removed.

2.7 Instrument Handlers

Each instrument is managed by a unique handler (or group of handlers). The handler is responsible for controlling the hardware interface to the instrument, accepting commands for the instrument, gathering data from the receiver, and finally formatting the data for delivery through the BIU. Some of the hardware used by the instruments on the HRP is shared and the handlers make use of kernel services to mediate access to the shared resource.

An instrument handler may consist of any number of processes located on any of the processors. As an example the WBR/WFR handler consists of eleven processes on the HRP and 3 additional processes on the DCP. The MFR handler consists of two processes on the LRP. Communications between these processes may rely on the IPC handler or make use of local communications methods.

Mx/Flag Comments	***Number***	***LRP***	***HRP***	***DCP***
Common to all processors		(parenthesis) indicates unused Mx/Flag		
IPC Mx	0	IPC	IPC	IPC
IPC Int, Receive	1	IPC	IPC	IPC
IPC Int, Transmit	2	IPC	IPC	IPC
8155 Interrupt	3	IPC	L/P	IPC
DMA Int Ch0	4	m/m move	WBR & L/P	(n/a)
DMA Int Ch1	5	m/m move	WFR	(n/a)
DMA Int Ch2	6	IPC	IPC	(n/a)
DMA Int Ch3	7	HFR	L/P	(n/a)
8155 Mx	8	IPC	L/P	IPC
	9		MM move	
	10	BIU	DCC	AGC
	11		WFR Mx	
	12	HSK2	WBR Mx	MAC

Mx/Flag Comments	*Number*	*LRP*	*HRP*	*DCP*
	13	RTI-0	L/P 0	LFDR
	14	HSK	L/P 1	Read
	15	HFR	L/P 2	Send

2.8 Memory Allocation Scheme

Code space and dedicated buffer space are statically allocated (when an application is assembled on the ground). A module, such as the handler for a given receiver, may consist of any number of processes. This makes managing memory on the ground rather simple and straightforward as each module has a block of memory to use. This also makes building the download for the instrument manageable while making the job of dump analysis much more difficult. Dump analysis is covered in the ***IEB Internals*** section.

There is also a dynamic memory allocation scheme implemented within the kernel that is used to facilitate interprocess communications. An area of 6K to 12K is divided into 256 byte blocks to be managed by the queueing system. This pool of free space is used to pass data and commands around the instrument (this is the data entity processed by the IPC driver). This poll of free space is referred to as **F5 elements** or simply **F5** in other parts of this document.

These allocation methods all make use of the queueing facilities provided by the kernel. Data moving through the IPC driver typically make us of global free space while data moving within an instrument handler would make use of local free space lists (i.e. the WBR/WFR use several free space pools tailored to the needs of the instrument).

Allocations:

1. LRP 5100-7EFF 11.5K
   46 buffers in the F5 queue

2. HRP 3E00-52FF 5.25K
   21 buffers in the F5 queue

3. DCP 5100-7EFF and B000- BAFF 14.25K
   57 buffers in the F5 queue

2.9 Version Control Information

Each version of software has some means of unique identification. Starting with Version 2.2 of the flight software the 1st ALF record contains version control information in the data portion of the record. The instrument does not make use of this data (i.e. it is not loaded into memory) so it may contain any pattern useful on the ground. As of version 2.2 of the flight software, 3 of the 16 words are defined.

The kernel also has a version and date string embedded in the cold start code that is updated as the kernel is modified and new loads are submitted. Although the version string is rather static (the kernel has been at version 2.4 for some time), the data string is modified with each submission. This information may be accessed through a **73MRO** command if needed.

In some of the software loads (i.e. early science loads), a stim control packet may be obtained (using the **73STIM** command) . This packet contains a date code that mat be used to identify the version of the software that generated the data (this appears in the science telemetry).

The housekeeping data stream may also be used to identify the software currently executing in the processors. There are 3 unique CCSDS patterns associated with the housekeeping. The CCSDS header can be used to verify that the operating software was, in fact, downloaded, and if it was downloaded from the correct partition.

2.9.1 Version verification procedure, on-line, prior to load

Obtain a dump of the 1st SSR record for both the default and the non-default SSR partition. This should be record 7067 (as of launch). The data words may be compared with the tables that follow to determine the release version and the purpose of the load (i.e. science, deploy, or maintenance). Be particularly observant of the maintenance vs. Deploy word to guarantee that the appropriate load is on the SSR.

The instrument can now be downloaded, as required, and the housekeeping data inspected to determine if the correct load was delivered. If the default/non-default selection is reversed, the housekeeping CCSDS header will indicate a problem.

Recovery is as simple as sending a reset (**73RT_RESET**) and loading using the alternate partition.

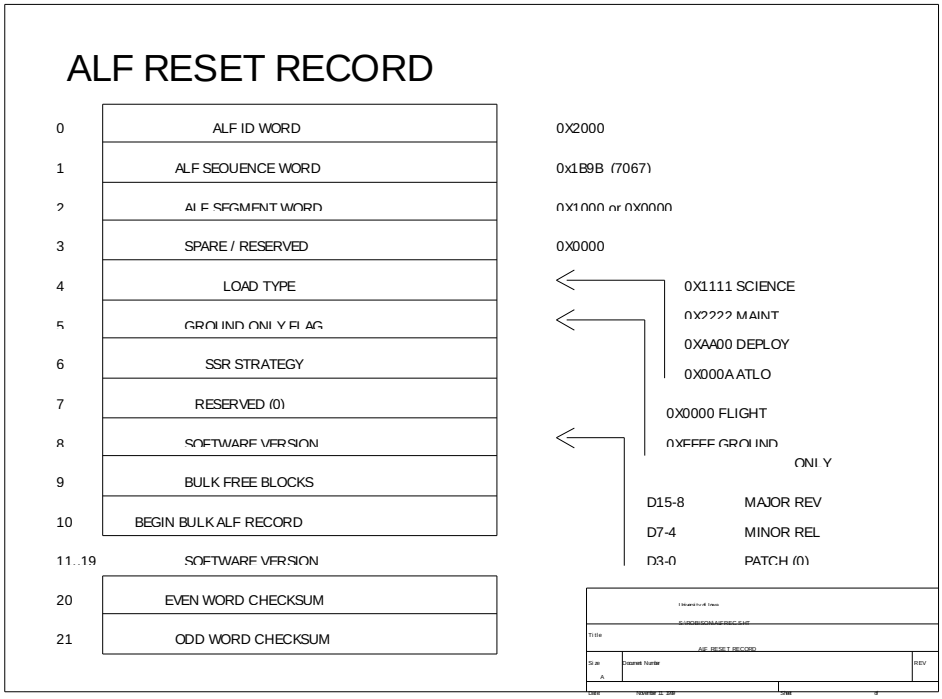
2.9.2 ALF Reset record

The 1st ALF record of any load consists of an ALF record with all of the destination bits cleared. This record is used to identify the beginning sequence number of an ALF load (starting sequence information is **NOT** contained in the ROM). In addition to the 1st. reset record, version 2.4 and later software may contain a 2nd. reset record that delimits the beginning of memory image data.

V2.4/V2.5/V2.6 load contain IEB images at the beginning of the ALF load and require a reset record between the IEB image and memory image to allow an internal reload to occur properly. Typically the 2nd. reset record would contain the software version number in all data locations.

The only information within a reset record that is used by the ROM is the sequence and segment fields. Although the checksum is validated before being used, the data area of the record may contain any desired pattern as long as the checksum is valid.

The data area of the 1st ALF record is available to be used by Spacecraft or Ground as any data pattern is acceptable in this area.



2.9.2.1 1st ALF_RESET Data Word

The 1st. ALF data word contains an indication of the function this load is intended to perform. Currently there are only 3 software loads that will be used for flight activities with a 4th. Being used for ground testing.

Version	Release Date	1st ALF Data Word	Comment
Science	N/A	0x1111	Indicates default partition data
Maintenance	N/A	0x2222	Indicates non default partition data
Deploy	N/A	0xAA00	Indicates Antenna Deploy Software
Deploy ATLO	N/A	0x000A	Indicates special Deploy Software For ground use only

2.9.2.2 2nd ALF_RESET Data Word

The 2nd. ALF data word is a flag that indicates **ground use only** when set. This field is intended to be used as a quick sanity check of the software loaded into the SSR.

Version	Release Date	2nd ALF Data Word	Comment
Flight	N/A	0x0000	May be used for flight
Ground	N/A	0xFFFF	**NOT** to be used for flight

2.9.2.3 3rd ALF_RESET Data Word

Version	Release Date	3rd ALF Data Word	Comment
2.3	May 1998	0x0011	No special SSR loading strategy (may be placed in all partitions?)
2.4	Summer 1999	0x0011	No special SSR loading strategy (may be placed in all partitions?)
2.5	Winter 2001	0x0011	No special SSR loading strategy. (may be placed in all partitions)
2.6	Fall 2002	0x0011	No special SSR loading strategy. (may be placed in all partitions)
2.7	Fall 2003	0x0011	AS always, nothing special here

Data word 3 is intended for use in the identification of the SSR partition the load belongs in. As strategies emerge, a table of values will appear here.

Currently, the build file for the 2.3 and 2.4 loads place a value of 0x0011 in this field.

2.9.2.4 4th ALF_RESET Data Word

Data word 4 is currently reserved and should appear as zero.

2.9.2.5 5th ALF_RESET Data Word

Data word 5 contains the version number. The version number may be duplicated through to the end of the data area of the ALF record.

Version	Release Date	5th/8th ALF Data Word	Comment
2.2	July 1997	0x0220 0x0220	1st release with version information contained in the ALF RESET record.
2.3	May 1998	0x0230 0x0230	1st. Post-Launch release Misc Fixes MiniPKT non-blocking
2.4	March 2000	0x0240 0x0240	Cruise updates Misc fixes IEB loader changes
2.5	Winter 2001	0x0250 0x0251	WPV violation problem found & fixed Other misc. fixes.
2.6	Fall 2002	0x260 0x260	Misc Fixes Burst WBR / Toggle WFR Telemetry mode updates
2.7	Fall 2003	0x270 0x270	RST-5 Fix Removal of 73POWER_CNTL implementation

Word 8 is also documented here. This is the first free data word as of version V2.5 software and has been used to indicate an intermediate update of the IEB load. Finding 0x250 in the 8th. Word would have been an indication that an inappropriate internal IEB is present in the flight software load.

2.9.2.6 6th ALF_RESET Data Word

Remaining BULK memory records.

This should contain the number of ALF records that remain in BULK memory. For lack of a better place to store the information, this word provides a convenient place to save this count.

2.9.2.7 7th ALF_RESET Data Word

1st. ALF Record in BULK Memory

This is an indication of where the IEB image ends and memory-image begins. When using the ALF mechanism to load IEB memory, we must place IEB image before memory-image so that IEB is not placed in BULK memory (to avoid overflow).

Again, note that the 2nd. reset record will nominally have all data words filled with the software version number.

2.9.2.8 8th ALF_RESET Data Word

IEB Update Indicator. This is a psuedo-version to indicate a build, with no kernel changes, that contains an updated IEB image.

2.9.2.9 9th ALF_RESET Data Word

IEB Version string. This should contain the Phase and release number from the Base IEB. We see a version string ***C352*** in the V2.6 base IEB and this word will contain the value **0x0352**.

2.9.2.10 Remaining ALF_RESET Data Words

All of the remaining words will contain the version number.

The first of this group of words may contain update information, for example V2.5 software contains 0251 in the first of these remaining words to indicate that this is the correct build as some IEB updates were applied after the FSW load had undergone several hundred hours of testing. Access to these remaining data words is available through the build files used to create the first ALF records in the load, in this case the **MAPIEB.BLD** control file.

Additional version unique identifiers may appear in these remaining words by placing the desired values in the **MAPIEB.BLD** control file using the /RESET directive. The line should appear as follows for the V2.5 flight load:

```
/RESET=0x0000,0x1111,0,0x0011,0,*,25,7233,0x0251
```

The 8nth item (0x251) has been appended to this load to differentiate it from previous ALF loads that contain identical flight software but differing internal IEB loads. Although the IEB load contains identifying information that is readily available in the next ALF record, we add the information at this point to allow verification using only the initial ALF record record in the load.

2.9.3 Version verification procedure, off-line, using science telemetry

If the STIM handler is present and the instrument has been controlled using 73IEB_TRIGGER commands, the science telemetry may contain STIM packets that will contain an 16 bit identifier. This identifier may be compared with the tables that follow to determine the version of the science software that generated the data.

2.9.3.1 STIM Packet Data Word

Version information from the running science software is presented in a ***Stimulus Control Packet*** that appears only as a result of a ***Stimulus Control Command***. Much of the internal IEB activities generate *stim control packets* that also contain information about the step number within the IEB **that is currently active.** The last 16 bits of the packet are the **Stim pkt ID** and must be converted to decimal for use with this table (this number is simply the day-of-year when the stim handler was last assembled).

Stim Packet Data Word

Version	Release Date	Stim pkt ID	Comment
1.0	February 1996	N/A	Initial release
1.1	April 1996	N/A	PFR fixes
1.2	June 1996	191	PFR fixes
1.3	July 1996	191	Introduction of STIM packets
1.4	January 1997	357	Crashes due to interrupt problem
1.5	not released		
1.6	March 1997	041	First fully functional HRS mode
1.7 (2.0)	June 1997	108	Small PFR/bugfix introduction of DUST detection
2.1	June 1997	150	PFR (HFR Hang) fix
2.2	July 1997	195	PFR (SLEEP power problem) fix
2.3	June 1998	100	Misc fixes MiniPKT non-blocking/fast delivery IPC deadlock fixed
2.4	March 2000	306	HSK, LOCK, TWEK changes IEB handler changes 73POWER_CNTL ? (for WBR)
2.5	March 2002	349	WPV Fix HRS sequence fix
2.6	October 2002	75	Misc fixed WBR Burst & WFR Toggle Telemetry mode updates
2.6	April 2003	76	Patch in BASE 2.6.5 IEB Fixe3s glitch caused by WBR
2.7	Fall 2003	303	RST-5 Fix Removal of 73POWER_CNTL

2.9.4 Version verification procedure, on-line, post load

The housekeeping data will already indicate which load is executing (i.e. ROM, Science or Deploy/Maintenance). Recovery at this point involves a reload from the appropriate partition.

To determine the version of the software, use the tables below to send a **73MRO** command to dump the kernel memory that contains the version and date strings. As this is a short string, it is probably reasonable to send the data to the housekeeping stream, particularly if real time data is available.

2.9.4.1 Kernel Version String

The kernel also contains a string used for version identification. There is a string that identifies the version of the kernel that may be used to locate the kernel source files as well as a string that indicates the date on which the kernel source was assembled.

The *Kernel Date String* should change with each release of the instrument software (the *Kernel Version String* is the kernel version, not the release version).

In the event that the flight software is submitted without change to the code area (i.e. when IEB commands are the only changes) the *kernel version string* and *kernel date string* remain unchanged. In this case, the patch level in the reset records would be update, for example from 0x240 to 0x241.

Version	Kernel Version String	Kernel Date String	Version Key Location	HSK Dump Location	Notes
2.1	V2.3	97171	0x07BE	0x07C6	Kernel is actually V2.4
2.2	V2.4	97195	0x07B9	0x07C1	
2.3	V2.5	98065	0x07B9	0x07C1	No kernel changes
2.4	V2.5	98220	0x07B9	0x07C1	No kernel changes
2.5	V2.5	01349	0x07B9	0x07C1	No kernel changes
2.6	V2.5	02240	0x07B9	0x07C1	No kernel changes
2.7	V2.6	03003	0x07B7	0x07B9	Expanded Flag Array

When dumping data through the housekeeping telemetry stream, the *HSK Dump Location* column should be used to obtain all of the useful version information in a single MRO record (each MRO record in housekeeping has 10 bytes of dump data).

2.10 Software delivery procedure

Seems we forget how this works every time we do it, so here goes...

Contact the contact person and request that the directory be opened for access and note the files that are present in the directory from the last delivery, the files should match those listed in this document.

Remove the old files from the directory and FTP the current files. The tar archive, being around 600Mb is size, will take several minutes to transfer.

Reformat the ALF file (using something like 'dos2unix -ascii <infile> <outfile>') and remove the incorrectly formatted file.

Notify JPL that the files are in place and schedule a DCM/SRCR teleconference.

2.10.1 JPL contacts

Version	*Delivery Date*	*DCM Date*	*Contact Person*
V2.3			
V2.4			
V2.5			Bill Lackey
V2.6	10/02/02	10/25/02	Bill Lackey

2.10.2 Delivered files

The following files are copied to a directory on the SOPC.

1. RDDs (release description documents)
    - srcr*.rdd
    - cover*.rdd
    - scien*.rdd
    - change*.rdd
    - testr*.rdd
2. MAP files (memory usage maps)
    - mapieb.map
    - mapkrnl.map
    - maplrp.map
    - mapdcp.map
    - maphrp.map
3. 73ALF load image
    - map.alf
4. File listing
    - cassini_fs_1.file_list
5. source archive (tar archive)
    - cassini_fs_1.tar
6. ALF file (UNIX text file)
    - rpws_science_alf_*.text

Notes

1. The ALF file is the only file that **MUST** follow UNIX text file conventions (i.e. Line termination is 0x0A only; and the file does NOT terminate with a 0x1A).

2. The version number is present in the RDDs and the ALF file (represented by the asterisk in the above lists).

2.11 Spacecraft Time

The spacecraft maintains and distributes a 32 bit field representing seconds from some defined epoch. This time is distributed to all instruments once each second during the 6th RTI period (RTI period 0 occurs as the second changes). The instrument notices the arrival during RTI period 7 and moves the upcoming second field from BIU memory into the new time field located in the system data area and sets a flag to indicate a time update has arrived. During the next RTI period, the processor moves the new time to the current time and clears the RTI field to zero to indicate this is RTI 0.

During each RTI interrupt the RTI field in the system data area is incremented. In the event that S/C fails to deliver a time update, the current seconds field is incremented when the RTI field overflows. The LRP should have an accurate idea of time **if** the S/C is delivering time information without discontinuities. If the S/C delivers discontinuous time, serious scheduling problems can occur with the HFR and MFR. Note that is not essential for the S/C to deliver a time update every second, (a single accurate time along with an accurate 1/8 second RTI pulse should be sufficient to set an accurate time).

In order to keep DCP and HRP up to date, a 2 level mechanism is employed. The IPC mechanism is used to deliver upper bits (coarse) of the current time, while a status signal is generated to indicate the lower bits (fine) are all zero. During RTI-0 when the lower 8 bits of the seconds field is zero, a signal is delivered from LRP to both DCP and HRP to signal the event. DCP and HRP notice the signal while processing the RTI interrupt and clear the lower time bits (8 bits of seconds and RTI). Within two RTI periods, the LRP generates a pair of MEM_TWEAK commands to set the upper 24 bits of time and sends the commands on to HRP and DCP to synchronize the coarse time.

This combination of a hardware assist for the fine time and IPC communications for coarse time allow DCP and HRP to establish the correct value for time. The RTI signal processed by the three processors is identical and the time synchronization status signal is synchronized with the RTI period in hardware.

This synchronization occurs every 256 seconds. If the spacecraft time is inaccurate the DCP and HRP will reflect a bad time until the next update occurs(i.e. the next time S?C delivers a time with LSB equal to zero). Also note that the LRP is essentially slaved to the time delivered from the spacecraft(LRP time field is updated each second when S/C is operating normally). If the S/C is delivering time each second and delivering it correctly, we can expect DCP and HRP to fall into synch with LRP within a few minutes. If, on the other hand, S/C is delivering bad time, and delivering them regularly, then we can expect DCP and HRP to have bad time.

If the S/C is delivering good time, but missing delivery windows, the LRP will free-wheel through lost time updates in the same manner that DCP and HRP maintain their time during the 256 second update cycle. When LRP notices that the LSB of the time field is zero, it will trigger the RTI-0 pulse and deliver the upper 24 bits of time as usual.

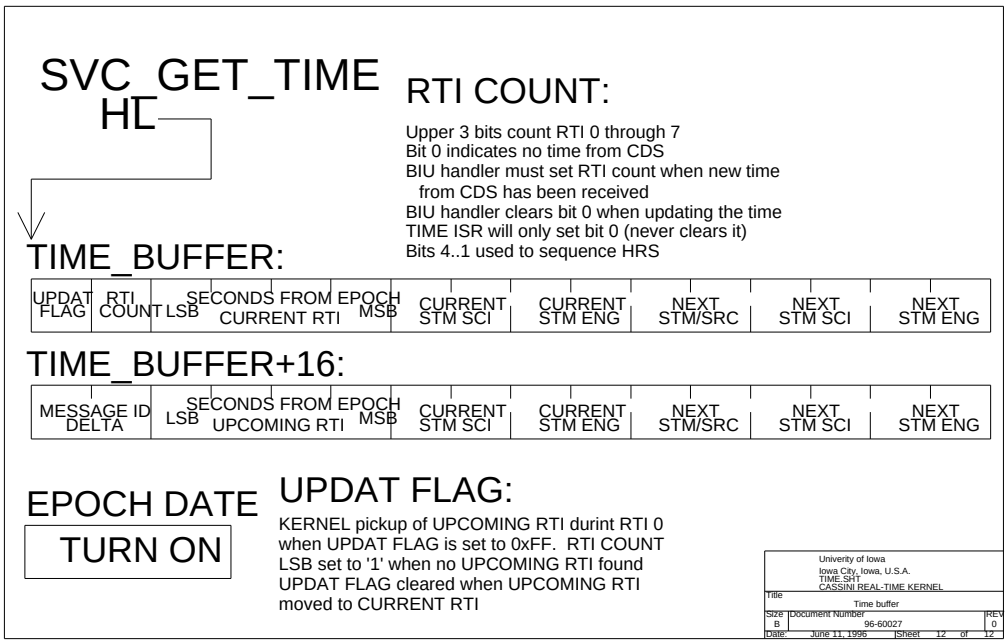
2.11.1 Failure mechanisms

There are several problems that can lead to inaccurate time or to scheduling problems within the instrument. Bear in mind that the assumption is made throughout the architecture of the time system that S/C will deliver accurate time and RTI signals to the instrument.

RTI period drifts with respect to the time field. The RTI signal is assumed to be exactly 1/8 second. Any deviation will cause RTI periods to be lost resulting in a time slip between LRP and DCP/HRP that may take up to 256 seconds to correct. Obviously if there is significant drift in the RTI signal, the time fields will be constantly be corrected resulting in scheduling anomalies (particularly evident in the WBR/WFR data).

Occasional bad times may also result in scheduling anomalies. There is no mechanism in place to validate the time delivered by the S/C. If a bad time is delivered with the seconds field set to zero, a time update will occur on the DCP/HRP that may not be corrected for up to 256 seconds. This will, again, by most evident in the way WBR and WFR data acquisition is scheduled.

In the event that S/C fails to deliver a time update (or if the message is lost/garbled), the instrument should automatically update the time as the RTI field overflows. This is an expected event and should cause no problems. If the update indicates an mode change, the BIU handler will notice any change in the number of science telemetry packets collected and adjust the formatting activities (and flow control to HRP) to match.



2.12 SLEEP Management

It seems that we need to mention the method used to manage bringing the instrument in and out of the low power ***SLEEP*** mode. The management of sleep is distributed over the LRP and HRP. This is required to simplify the hardware design and, we hope, to increase reliability.

2.12.1 ROM

In ROM mode, all control functions are coded as simple control loops. On the LRP, one pass is made through the control loop each RTI period with the BIU discrete bit being checked at the same point in the loop. If the SLEEP Discrete has changed, the control bit to DCP and HRP is changed accordingly. The only power control is for the Langmuir Probe and this power switch is disabled when the sleep bit is encountered in an active state.

The DCP has no power switch and is essentially oblivious to SLEEP so there is no special handling required.

The HRP has a power switch for the Langmuir Probe, similar to the switch on the LRP which is deactivated whenever the sleep status line is asserted.

2.12.2 Science & Special Maintenance

The downloaded software provides support for multitasking so we eliminate the simple control loop found in the ROM code. This allows the work to be broken down into individual tasks to make management simpler, at least that's the plan.

Sleep may be asserted using either the discrete bit or a command to the instrument. Either command may be used, but the instrument command is treated with less respect when exiting sleep mode (see **73POWER_CNTL**).

On the LRP, the BIU handler monitors the sleep discrete. When the sleep discrete bit is asserted, the BIU handler clears the power switches on the LRP and asserts the sleep status line to DCP and HRP.

The DCP software ignores the sleep state as it has no power switches to control.

Starting with V2.2 the HRP monitors the sleep status using the IPC Watch Dog Timer context. When sleep status is active, it commands both the L/P and the A/D power switches to the off state. This monitoring occurs once during each RTI period. The code fragment happens to be located within the L/P handler. Note that none of the activities that occur in the context of the IPC WDT are allowed to use kernel services that can block the CPU (in order to prevent deadlock situations). This particular implementation results in minimal memory and CPU overhead.

Starting with V2.3, the HRP handles SLEEP in 2 stages. During the first RTI that SLEEP is presented to the HRP, the timing control processes for L/P, WBR, WFR, LFDR, and DUST are stopped by writing a zero to the appropriate control bytes. This stops subsequent data acquisitions and will, eventually, cause the HRP to stop delivering data. When RTI remains active for more than 4 consecutive RTI periods, the power control ports are cleared to remove power from the L/P digital electronics and from the analog converters on the HRP. Any subsequent activity on the SLEEP line has no effect as the HRP is executing a Halt instruction for most of the RTI period (in other words, power levels will not increase when SLEEP is de-asserted)

The V2.3 software also continuously clears the two power switches on the HRP when SLEEP is active. If, for any reason, a 00PORT_TWEAK command is issued to attempt to power up the WBR/WFR analog converters or to power on the Langmuir probe digital electronics, the sleep management module will clear the power switch at the start of the next RTI period. It should not be possible to increase the power dissipation on the HRP for more than a single RTI period if an attempt is made to activate either of the switches.

This handling of the sleep function is intended to allow the instrument to transition into sleep within five RTI interrupts or less than 3/4 second.

The status returned in the housekeeping telemetry that indicates that the instrument is in sleep does not indicate the instrument has exited the sleep state until power is applied to the receivers. In other words, the housekeeping telemetry will continue to indicate that the instrument in sleep until a power command is received. Note that power commands are ignored when the sleep discrete is asserted.

2.12.3 Exiting Sleep

When exiting sleep mode, the LRP will not command the DCP/HRP clock control to full speed until a power command is received. Since no data is acquired when the power switches are off, there is no reason to have DCP/HRP clocks running at full speed.

This may lead to some confusion when observing the housekeeping display as the clock rate indication in the housekeeping telemetry is, effectively, synonymous with sleep. Some of the GSE display take the clock rate status bit as an indication that the processor is in sleep state, so it may appear that the instrument has missed a SLEEP ACTIVE command. This apparent error will be corrected when the 1st power command is processed by the instrument.

If it becomes necessary to switch power on the HRP immediately following a SLEEP state, either make use of the **73IEB_TRIGGER, MASK** command or send **73POWER_CNTL, SLEEP, ACTIVE** to release HRP/DCP from the sleep state.

2.12.4 Power Control Hardware Summary

Just a quick review of what the hardware that is connected to the sleep discrete bit.

Note from this, that the 73POWER_CNTL, PAUSE command should not be used when the Langmuir Probe is powered as it will remove power from the L/P momentarily and then re-apply the power 2 RTI periods later. Although the L/P electronics are relatively low power, switching both sections at the same time will cause power spikes to overlap, possibly exceeding peak power available to RPWS.

2.12.4.1 Power Control: LRP

1. MFR Clocks are disabled during sleep HARDWARE INTERLOCK
2. L/P analog enable is disabled when HRP power switch is disabled (interlocked with L/P digital on HRP)
3. L/P, ME02, and HFR power are under local software control only (no hardware interlock).
4. Sleep status to HRP/DCP are under local software control only (no hardware interlock).
5. Processor clock speed is under local software control only.

Note the purpose of the L/P interlock is to guarantee proper sequencing of power to L/P.

2.12.4.2 Power Control: HRP

1. L/P power is interlocked with the sleep status, it is also under local software control (i.e. either source may assert sleep, in other words, both must be in the correct state to apply power).
2. HRP analog section is under local software control only (no hardware interlock).
3. Processor speed is controlled by LRP.

Note the purpose of the SLEEP interlock on the L/P is to force proper sequencing of power to L/P.

2.12.4.3 Power Control: DCP

1. Processor speed is controlled by LRP.

3 Deploy Operations

This section describes the operations and software for the antenna mechanism. The deploy software is used only once and then removed from processor memory. It is also expected that once a deploy has been successfully performed that the deploy software would be removed from the SSR (i.e. spacecraft memory).

The antenna elements were successfully deployed in October 1997 and the deploy software was overwritten shortly thereafter. This section of the users manual is included for reference, particularly with respect to the temperature monitor.

3.1 Real Time Kernel

Deploy software runs under the same real-time kernel used by the science software. The same BIU handler and Command decoder used by science software are also used here. Most of the commands that the instrument would normally accept would simply be discarded by the IPC handler as the software components that would use the commands are not loaded.

The kernel may be loaded into the DCP and HRP to reduce power consumption (by causing the DCP/HRP to execute a **HALT** instruction) although no application software would be loaded.

Note that having the kernel loaded on the DCP/HRP may mean that IPC activity is not possible (due to lack of IPC drivers on DCP and HRP). Although the LRP will not crash when an attempt is made to deliver traffic to DCP/HRP, this may cause the IPC driver on the LRP to block subsequent traffic as the driver performs it's error recovery activities (on the order of several seconds for each packet). ***73MEM_TWEAK*** commands, in particular, will attempt to deliver traffic to DCP/HRP.

When the kernel in not loaded on DCP/HRP, the ROM code will accept IPC traffic although at a slightly reduced rate. Although IPC traffic is delivered to DCP/HRP none is ever generated (the protocol used in ROM mode is slightly different in that IPC traffic is always initiated by the LRP).

In either case, DCP/HRP is missing the software necessary to originate traffic so the loop counter in the housekeeping page never increments.

3.2 Antenna Control Software

The control software is responsible for controlling the bits that select the antenna element, it's direction, and for monitoring static operating limits.

The static monitoring checks temperature, position, current, and limits switches against pre-set limits. If any limits are exceeded the software will clear the power control bits and wait for the next command.

3.3 Antenna Commands

In addition to the commands documented in CAS 3-291 the antenna software recognizes several undocumented commands. These commands are listed with the antenna commands with descriptions of their use.

The undocumented commands are not expected to be used during the deploy operation and were not supplied to JPL for inclusion in the 3-281 document.

3.4 Antenna Monitoring

The monitoring software performs dynamic monitoring. This consists of watching the position potentiometer to insure element movement. If the position pot does not change value for approximately 8 seconds, the monitoring software will issue a 00ANT_HALT command.

In addition to the monitoring operation, this software keeps a log of position and current information that is delivered to the ground.

The monitoring software also keeps a running record of the deploy activity with a time resolution of a few seconds. There is sufficient memory to record the entire deploy of a single element. This record is not disturbed by the science software as long as the HFR remains powered off. The data is continuously presented in the housekeeping. In the event that a housekeeping record is lost, the data it contains will be presented in the housekeeping when the data pointer wraps around

Although the monitored data is continuously (and repeatedly) presented in the housekeeping data, it is possible to perform a memory readout to obtain the data in significantly less time. This method makes use of an extended MRO command and requires that the time synchronization activity be suspended for the duration of the dump (as you may expect, no other MRO or MEM_TWEAK commands may be issued during the dump). This group of commands may be issued to any of the downloaded software before the HFR is powered (the HFR makes use of the same area of memory used to store the antenna readings). The last MRO is present to flush data out of the BIU and may not be necessary in all cases. This can be accomplished using the following commands.

00:00	**73MEM_TWEK, LRP, BYTE, 0x60, 0x00, LOCK**
00:05	**73MEM_TWEAK, LRP, BYTE, 0X14, 0X40, TWEK**
00:10	**73MRO, LRP, TLM, 8000, BFFF**
05:00	**73MEM_TWEAK, LRP, BYTE, 0X14, 0X40, TWEK**
05:05	**73MEM_TWEK, LRP, BYTE, 0x60, 0xFF, LOCK**
05:10	**73MRO, LRP, TLM, 0000, 03FF**

00:00	**73WRAP, (3862, 0060, 0000, 4F4C, 4B43)**
00:05	**73WRAP, (3862, 0014, 0040, 5754, 4B45)**
00:10	**73WRAP, (3483, 8000, BFFF)**
05:00	**73WRAP, (3862, 0014, 0040, 5754, 4B45)**
05:05	**73WRAP, (3862, 0060, 00FF, 4F4C, 4B43)**
00:10	**73WRAP, (3483, 0000, 03FF)**

3.5 Antenna BIU Discrete Bits

Three of the eight BIU discrete bits are used to enable the latching relays within the antenna control electronics. One discrete bit is dedicated to each antenna. In order for the relay that controls power to the antenna mechanism to be switched on, the corresponding discrete bit must be enabled. The control signal, generated by the processor, is gated with the discrete bit to control the latching relay.

Note that the BIU discrete bit is used to control a **latching** relay. This means that removing the BIU discrete does not directly remove power from the antenna mechanism. The software monitors the BIU discrete during the deploy operation and removes power from the antenna mechanism within 1 RTI period if the discrete bit becomes inactive.

3.6 Antenna Software memory dump

In order to perform a post mortem analysis on the deploy software the following memory dump may be used. This dump assumes that science telemetry in excess of 1500 bits/second is available.

```
00:00   73MEM_TWEAK, LRP, BYTE, 0X14, 0X40, TWEK
00:05   73MRO, LRP, TLM, 0000, 7FFF
06:00   73MRO, LRP, TLM, 8000, FFFF
12:00   73MEM_TWEAK, LRP, BYTE, 0X14, 0XC0, TWEK
12:05   73MRO, LRP, TLM, 8000, BFFF
15:00   73MEM_TWEAK, LRP, BYTE, 0X14, 0X00, TWEK
15:05   73MRO, DCP, TLM, 8000, 83FF
```

3.7 Antenna Status

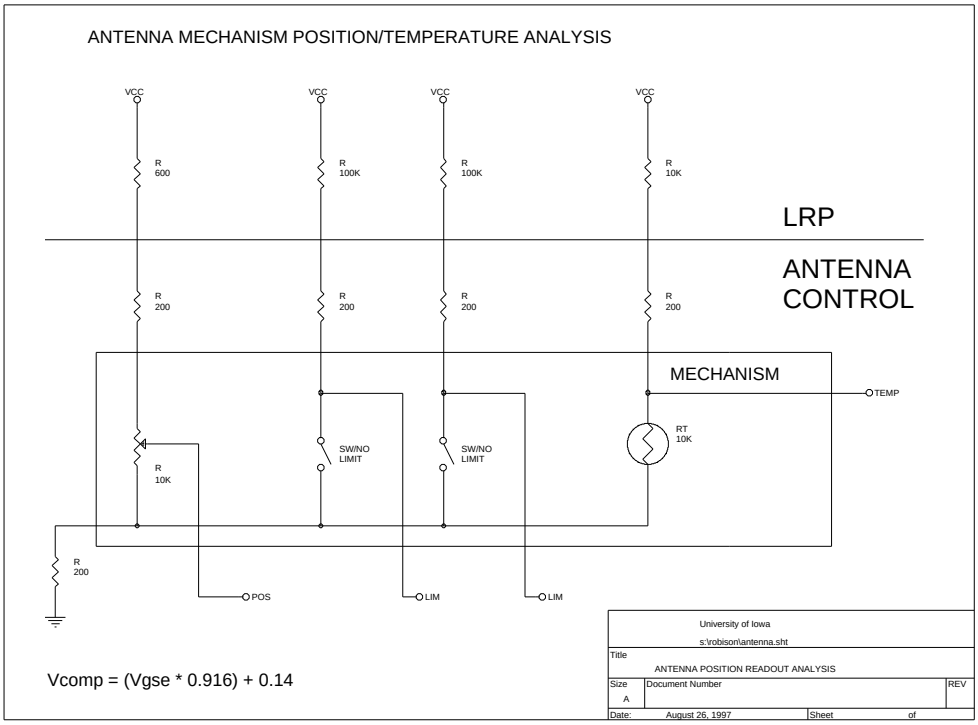
Status values are present in the housekeeping and are updated upon reception of a command or completion of a commanded operation. The housekeeping section discusses details of the information available in the housekeeping.

Two items worth mentioning here concern the *Command Pattern* field and the *Reason Code* field.

The Command Pattern field contains the 1st. 16 bits of the last command received. In most cases the command is generated from the ground so this field will not change by itself. When the antenna is moving, however, the antenna monitoring software may decide to stop antenna element movement for one of several reasons. This antenna monitor routine does not directly control the registers used to control the deploy electronics. It accomplishes the task of stopping movement by sending an **antenna hold** command to the antenna control routine. The command is delivered directly to the antenna control software, past the point where parity checking is performed. The **73ANT_HOLD** command has the parity bit inverted to distinguish it from commands that originate on the ground. The actual patterns are documented in the housekeeping section of this document.

The *Reason Code* field contains an indication of the reason that the software made the transition to an idle state. There are, however, several status codes that must be used to indicate that the antenna element is moving. In the nominal case, the element will be retracted until the retract limit switch is tripped followed by a full extension in order to release the retaining pin that keeps the element from extending due to vibration during launch. There is no distinct status used to indicate the first step is in progress or has been successful as the operation requires less than 2 seconds. The text with the status message that is expected during a normal extension indicates that an initial retract was requested prior to extending the element (as opposed to an extend continuation that would not attempt to perform a short retract prior to extension).

3.8 Antenna Position & Temperature Analysis



Equivalent Circuit for antenna mechanism.

Each of the antenna mechanisms contains the above circuit to monitor the position and temperature of the deploy mechanism. Note that the limit switches are closed only when the element is fully retracted or extended and may be ignored when performing the analysis to convert from the GSE readings to those expected to appear in the housekeeping telemetry during deploy.

3.9 Antenna Position Table

The position measured on the GSE are available for the full deploy. These measurements were taken when the antenna element was installed. It may be necessary to make note of the initial element position prior to the beginning of the deploy operation as the position pot may have been reset during handling or when the antenna element was trimmed.

The TLM values are calculated from the measured GSE values and are approximately the value that will be displayed using the GSE display software. The TLM value represents the voltage that appears at the input to the A/D converter on the LRP.

		Installed as EZ			Installed as EX -			Installed as EX +	
8/97		Serial Number 005			Serial Number 006			Serial Number 007	
Len		GSE Value	TLM Value		GSE Value	TLM Value		GSE Value	TLM Value
0m		059	0.74		094	1.06		089	1.02
1m		091	1.04		123	1.33		118	1.28
2m		122	1.32		153	1.60		148	1.56
3m		154	1.61		183	1.88		178	1.83
4m		186	1.91		212	2.14		208	2.08
5m		218	2.20		243	2.43		237	2.37
6m		250	2.49		273	2.70		267	2.65
7m		281	2.78		303	2.98		297	2.92
8m		313	3.07		333	3.25		326	3.19
9m		345	3.36		363	3.53		356	3.46
10m		378	3.66		393	3.80		388	3.76

3.10 Antenna Contingency Commands

The following table may be used to specify commands that may be used to alter dynamic monitoring tables in the antenna deploy software in the event that the default timer value of 900 seconds (15 minutes) is inappropriate for conditions at the time of the deploy operation.

It appears from the data gathered during assembly of the antenna mechanisms that a combination of low bus voltage and low temperature can result is a slowed deploy operation resulting in a time expiration prior to full element deploy. These commands may be used to alter the timers prior to deploying the antenna elements.

Time Sec	**EX +**	**EX-**	**EZ**
900	73WRAP, 1643, 00, 00, 1C20	73WRAP, 1645, 00, 00, 1C20	73WRAP, 1649, 00, 00, 1C20
950	73WRAP, 1643, 00, 00, 1DB0	73WRAP, 1645, 00, 00, 1DB0	73WRAP, 1649, 00, 00, 1DB0
1000	73WRAP, 1643, 00, 00, 1F40	73WRAP, 1645, 00, 00, 1F40	73WRAP, 1649, 00, 00, 1F40
1050	73WRAP, 1643, 00, 00, 20D0	73WRAP, 1645, 00, 00, 20D0	73WRAP, 1649, 00, 00, 20D0
1100	73WRAP, 1643, 00, 00, 2260	73WRAP, 1645, 00, 00, 2260	73WRAP, 1649, 00, 00, 2260
1050	73WRAP, 1643, 00, 00, 23F0	73WRAP, 1645, 00, 00, 23F0	73WRAP, 1649, 00, 00, 23F0

Another method that may be employed to overcome a sluggish deploy operation is to simply resend the deploy command, although this requires the use of a special deploy command variant that suppresses the initial retract operation that is required to release the caging pin.

	EX +	**EX-**	**EZ**
	73WRAP, 1002	73WRAP, 1004	73WRAP, 1008

If it is deemed necessary, the timer may be set to an appropriate value from the following table.

Time Sec	**EX +**	**EX-**	**EZ**
50	73WRAP, 1643, 00, 00, 0190	73WRAP, 1645, 00, 00, 0190	73WRAP, 1649, 00, 00, 0190
100	73WRAP, 1643, 00, 00, 0320	73WRAP, 1645, 00, 00, 0320	73WRAP, 1649, 00, 00, 0320
150	73WRAP, 1643, 00, 00, 04B0	73WRAP, 1645, 00, 00, 04B0	73WRAP, 1649, 00, 00, 04B0
200	73WRAP, 1643, 00, 00, 0640	73WRAP, 1645, 00, 00, 0640	73WRAP, 1649, 00, 00, 0640

The following table may be used to alter the cage pin retract timer. The default retract period is one second or until the limit switch is triggered. Changing this location will affect the maximum period of time that will be spent retracting the antenna in search of the retract limit switch. The time period is expressed in RTI periods with the default value being 8 (for a time-out of 1 second).

Version	**Command**
2.2	73MEM_TWEAK, LRP, BYTE, 0x2755, nn
?	73MEM_TWEAK, LRP, BYTE, 0x????, nn

4 Maintenance Operations

A low impact maintenance capability is built into the ROM in the instrument. In addition to providing ALF download capability, the ROM will allow the Langmuir Probe to be biased at approximately 10 volts. Although this voltage falls within some specification it is not optimal for scouring the sphere during the early portion of the mission where the S/C is close to the sun.

If a higher bias voltage is desired it becomes necessary to download the maintenance mode software. The maintenance mode download makes use of many parts of the science software load to allow the Langmuir Probe to be switched on and biased using the BIU discrete bit that is used to activate maintenance mode.

The Langmuir Probe module used in science mode along with a unique module used to monitor the maintenance command from the BIU provides control of the Langmuir Probe. This approach allows reuse of a working module without change. As a side-effect, it is possible to operate the Langmuir probe in a more-or-less normal manner when the maintenance bit is asserted (this being the only way to power the L/P electronics).

The special maintenance mode handler does **not** provide any support for ***73POWER_CNTL*** commands, so it is not possible to enable power on any of the instruments using the power control command.

Many of the utility functions present in the science load are, however, available. The ***MEM_TWEAK*** and ***MRO*** commands are functional.

With the release of Version **V2.3** science software, maintenance may be run using **73IEB_TRIGGER, MASK, 0, 3**. Due to hardware and software limitations, this maintenance mode does not set the associated discrete status bit (similar to the **V2.2** maintenance software).

4.1 Periodic Instrument Maintenance

Periodic Instrument Maintenance may be performed using either the ROM or the downloaded maintenance software. **PIM** is controlled by the BIU discrete bit using the **73RT_MAINTENANCE** command.

The purpose of downloading code to perform maintenance is to allow the Langmuir Probe to be biased to a higher voltage. Failure to download the maintenance software will result in the L/P being biased to 10 volts rather than the level specified in the download. This difference in voltage, although not optimal, is **NOT** fatal to the **PIM** activity.

The following section contains the standard PIM sequence for RPWS and the rationale for the choice of commands, their order and timing.

4.1.1 PIM Commands

Sample PIM commands. The final command may be followed by a **73RPWS_POWER, OFF** command to shut the instrument off, if required.

```
00:00    73RPWS_POWER, ON

01:00    73RT_RESET, RELEASE
01:05    73RT_RESET, RESET
01:10    73RT_RESET, RELEASE
01:15    73RT_SLEEP, ACTIVE
07:45    6EXT_MEM_LOAD, , ,RPWS

10:00    73RT_MAINT, ON
. . .
XX:XX    73RT_MAINT, OFF

XX:30    73RT_SLEEP, SLEEP
```

Sample PIM commands when using V2.3 science software. The final command may be followed by a **73RPWS_POWER, OFF** command to shut the instrument off, if required. This sequence of commands will not result in inadvertent application of power to the other receivers.

```
00:00    73RPWS_POWER, ON

01:00    73RT_RESET, RELEASE
01:05    73RT_RESET, RESET
01:10    73RT_RESET, RELEASE
01:15    73RT_SLEEP, ACTIVE
07:45    6EXT_MEM_LOAD, , ,RPWS

10:00    73IEB_TRIGGER, MASK, 0, 3
. . .
XX:XX    73IEB_TRIGGER, MASK, 0, 4   (bias to 0 volts, L/P power on)

XX:30    73RT_SLEEP, SLEEP
```

4.1.2 Power control

In the event that S/C has performed some safeing activity, the power command I included. If the instrument has been left in sleep from a previous activity, this command is redundant, but should not produce any power glitches for the instrument.

4.1.3 Memory loading philosophy

This is an attempt to maximize the availability of the instrument in the event that problems with the SSR or instrument occur.

The **73RT_RESET** should not be required if everything is working exactly as planned. In the event that the instrument has suffered some software problem, the reset is expected to restore the processor to an operating condition. It is placed before the **73RT_SLEEP, ACTIVE** command to match the nominal order of commands in other sequences.

The processor is immediately placed into an active state to bring all processor clocks to full speed. A six minute delay allows the internal bulk memory to load the instrument. The bulk memory integrity check is through the use of the ALF checksum. If bulk memory contains a valid load, it will cause the instrument to begin execution of the maintenance download. If the bulk memory has not been loaded or the contents corrupted, the internal load will fail at some point and the ROM will remain in control and sensitive to **73ALF** commands.

As is the case, when software has been downloaded, the instrument is no longer sensitive to **73ALF** commands and will ignore any loads that subsequently occur. Any **73ALF** commands that are received will be logged as invalid commands as the parity bit is incorrectly set. This is as expected and should not be detrimental to the instrument (note that this may raise alarms that may be safely dismissed).

If the internal load fails, the SSR download should then occur as planned with a new copy being placed into bulk memory.

We allow a download from bulk memory to occur to cover loss or corruption of the memory load located in the SSR. If either bulk or SSR contains a corrupted copy of the download, this allows the other copy to act as a backup (with the SSR actually providing up to 4 copies of the memory image). If the SSR is allowed to load first, this will effectively migrate any corrupted image from SSR to bulk memory (as each **73ALF** record is processed and stored in bulk memory, the next ALF image in bulk memory is marked as invalid in order to prevent bulk memory from ever being able to attempt execution of a split image in the event of a download failure).

4.1.4 Memory Load Timing Issues

When the Maintenance Software is loaded into the instrument, it performs a cleanup operation on the Langmuir Probe. Commands are issued to initialize the hardware even though it would not be powered up. This is a side-effect of the method used to detect changes in the maintenance bit, but this also causes the status table to be initialized such that housekeeping will correctly reflect that status of the L/P hardware and power switches.

This also allows the memory load to be accomplished when the maintenance bit is asserted. Following a successful download, the software initialization activity will power down all hardware within the instrument (including the Langmuir Probe). The sequence of commands to apply power to the Langmuir Probe is timed to require about 20 to 25 seconds to execute to allow any voltages within the L/P digital section to decay in order to avoid a latch-up condition in the A/D chip.

Following the turn-on delay, the probe will be biased to 32 volts, as dictated by the maintenance software.

4.1.5 Maintenance of Langmuir Probe

The maintenance bit is honored by biasing the Langmuir Probe to approximately 32 volts. Since the standard Langmuir Probe science module is used to control the hardware, any Langmuir Probe dac 0, Langmuir Probe relay, and Langmuir Probe multiplexer commands may be used to change the bias.

In the event that Langmuir Probe bias is altered, cycling the maint3enance bit off and back on will return to the 32 volt bias setting. In other words, whenever a transition into maintenance mode occurs, the probe will be biased at 32 volts.

Changes to the DAC, relays, and multiplexer are overridden when the maintenance bit transitions form off to on.

It is possible to send the commands required to enter maintenance mode independent of the maintenance bit.

4.1.6 End of maintenance activity

A transition from the active state to the inactive state will remove bias from the Probe by removing power from the Langmuir Probe electronics.

If, for some reason, **73LP** commands were used to bias the probe with the maintenance bit off, sending an additional **73RT_MAINT, OFF** command is ineffective (such as when using the **V2.3** science software).

Sending the 73RT_SLEEP, SLEEP command will, without regard to how the Langmuir Probe was controlled, remove power and place the instrument into a low power state.

4.2 PIM capability with Science Software

The Langmuir Probe can be biased with Science Software loaded into the instrument. (Version **V2.3** provides a specific trigger for this activity that controls power as well as biasing the sphere).

It probably won't work well if one tries to collect data at the same time. Specifically, the density mode measurement will not show much interesting when the sphere is biased to 32 volts.

Power Commands

```
00:00    73POWER_CNTL, SLEEP, ACTIVE
00:05    00PORT_TWEAK, HRP, 3, 1 (73WRAP,(0x32C8, 0x0103))
00:07    73POWER_CNTL, PROBE, ON
```

Bias to 32 volts

```
00:10    73LP_VOLT_CNTL, BOTH, 0xFF, 0x80
00:11    73LP_MUX0_CNTL, BIT0, OFF
00:12    73LP_MUX0_CNTL, BIT1, OFF
00:13    73LP_MUX0_CNTL, BIT2, OFF
00:14    73LP_MUX0_CNTL, BIT3, OFF
00:15    73LP_MUX0_CNTL, BIT4, ON
00:16    73LP_MUX0_CNTL, BIT5, OFF
00:17    73LP_MUX0_CNTL, BIT6, ON
00:18    73LP_MUX0_CNTL, BIT7, OFF
00:19    73LP_RELAY_CNTL, RELAY1, COIL_A
00:20    73LP_RELAY_CNTL, RELAY2, COIL_A
00:21    73LP_RELAY_CNTL, RELAY3, COIL_A
00:22    73LP_RELAY_CNTL, RELAY4, COIL_A
00:23    73LP_RELAY_CNTL, RELAY5, COIL_B
```

Bias to 0 volts (highlighted commands)

```
00:00    73LP_VOLT_CNTL, BOTH, 0x80, 0x80
00:01    73LP_MUX0_CNTL, BIT0, OFF
00:02    73LP_MUX0_CNTL, BIT1, OFF
00:03    73LP_MUX0_CNTL, BIT2, OFF
00:04    73LP_MUX0_CNTL, BIT3, OFF
00:05    73LP_MUX0_CNTL, BIT4, OFF
00:06    73LP_MUX0_CNTL, BIT5, OFF
00:07    73LP_MUX0_CNTL, BIT6, OFF
00:08    73LP_MUX0_CNTL, BIT7, OFF
00:09    73LP_RELAY_CNTL, RELAY1, COIL_A
00:10    73LP_RELAY_CNTL, RELAY2, COIL_A
00:11    73LP_RELAY_CNTL, RELAY3, COIL_A
00:12    73LP_RELAY_CNTL, RELAY4, COIL_A
00:13    73LP_RELAY_CNTL, RELAY5, COIL_B
```

Note the use of **73POWER_CNTL, SLEEP, ACTIVE** to bring the instrument completely out of sleep state. When the sleep discrete bit is used to bring the instrument to a low power state, the DCP and HRP clocks are slowed and HRP power switches are disabled. Following assertion of the sleep discrete (i.e. exiting the sleep state), the DCP and HRP clocks continue to operate at reduced rates until the first power command is processed by LRP.

Since the 00PORT_TWEAK command is not processed by the power control handler, the HRP and DCP clocks rates are not changed and any attempts to apply power on the HRP (to either L/P or the WBR/WFR A/D converters) are suppressed at the next RTI. The **73POWER_CNTL, SLEEP, ACTIVE** command is used to inform the power subsystem that the instrument is about to be commanded and HRP/DCP clocks should be switched to normal speed. If the instrument has not been in a sleep state, the **73POWER_CNTL, SLEEP, ACTIVE** command is not, strictly, necessary (this would be the case if the science software has just been downloaded in preparation for entering maintenance mode).

4.3 Maintenance ROM: commands

The following commands are functional when the instrument is powered and software has not been downloaded. All commands, other than ALF loads are ignored and treated as invalid commands.

4.3.1 73RT_SLEEP

The sleep discrete is always honored. Asserting this bit will cause the instrument to revert to it's lowest power mode.

4.3.2 73RT_MAINTENANCE, ON/OFF

This command powers and biases the Langmuir Probe Sphere to approximately 10 volts. Switching into and out of maintenance mode will disrupt a memory download as the HRP takes about 30 seconds to perform the switch.

4.3.3 73ALF

Memory downloads will be accepted when the instrument is operating in maintenance mode. Although the transition to/from maintenance will disrupt downloading, once the instrument has fully transition to maintenance mode, memory downloads will be accepted.

4.4 Science download: commands

The following commands are functional when the instrument is powered and the science software has been downloaded.

4.4.1 73RT_SLEEP

The sleep discrete is always honored. Asserting this bit will cause the instrument to revert to it's lowest power mode.

4.4.2 73RT_MAINTENANCE, ON/OFF

This command is currently ignored by the science software. If implemented, when the controlled bit transitions to an active state, the command **73IEB_TRIGGER, MASK, 0, 3** will be issued. When the bit transitions to an inactive state, the command **73IEB_TRIGGER, MASK, 0, 4** will be issued.

4.4.3 73IEB_TRIGGER, MASK, 0, 3

This trigger applies power to the Langmuir Probe electronics in the same order and with the same timing as the **73IEB_TRIGGER, MASK, 0, 0** command (without switching ME02, the HFR or the WBR/WBR A/D circuits). This timing is faster than that used with either the ROM or the maintenance software so care must be exercised that this command ***not*** be issued immediately following other power commands.

Also, it would be prudent to use this command only when the instrument is fully powered or when the instrument has just exited sleep (73RT_SLEEP) to prevent undesired power spikes. Allow enough time in sleep state for the power supply voltages in the Langmuir probe electronics to fully decay (30 seconds is sufficient). This is to prevent driving the A/D converter in the L/P digital section into a latch-up condition (power sequencing is interlocked in the hardware to prevent power up in the wrong order).

4.5 Maintenance download: commands

The following commands are functional when operating the special maintenance download. Any command that is not functional within the special maintenance download will be ignored. As with all other downloads, ALF commands are flagged as invalid as they are processed by the ROM software only (parity checking on the ALF records differs between ROM and RAM software).

4.5.1 73RT_SLEEP

The sleep discrete is always honored. Asserting this bit will cause the instrument to revert to it's lowest power mode. Downloading the maintenance software will result in a slightly lower power level than when operating out of the ROM.

4.5.2 73RT_MAINTENANCE, ON/OFF

This command powers and biases the Langmuir Probe Sphere to an alternate voltage level.

4.5.3 73IEB_TRIGGER, MASK, 0, 2

Applies power to the HFR electronics for a period of ***TBD*** minutes. At the end of the period the HFR will be powered down automatically.

4.5.4 73LP commands

Once the instrument has entered maintenance mode, the configuration of the Langmuir Probe can be changed using any of the commands associated with the Langmuir Probe. If the software is idle (i.e. not sweeping) it is necessary to set both DAC's.

4.5.5 73MEM_TWEAK

The memory tweak handler is required for internal communications so it's function is available for use from the ground.

4.5.6 73MRO

Same comment as 73MEM_TWEAK.

4.6 Maintenance Download: command counts

The special maintenance mode software mimics the function of the ROM although the implementation is quite different. The normal science Langmuir Probe handler is used for hardware control with the logical control resting with software resident on the LRP. The allows use of a well tested module on the HRP.

As a result, normal L/P commands are used to communicate with the HRP. The control module, located on LRP, packages commands and delivers them to the 1st. level command decoder for distribution. This results in valid command counters registering the activity as changes in the counters.

Command Count: 0

Valid Commands: 8

Invalid Commands: 0

The *Valid Command Count* field is also incremented as the **73RT_MAINTENANCE** is processed. The maintenance download uses normal L/P commands to power and bias the sphere when maintenance mode is requested.

Also of interest is the time required for the special maintenance software to organize itself to a point where commands will be correctly handled. The software requires approximately 30 seconds to issue all of it's commands to the *Langmuir Probe handler* before any commands will be correctly processed. In addition, the maintenance bit should not be asserted during this initialization activity. Failure to meet this settling time can result in commands being overwritten by internally generated commands (appears as though a command was lost as the intended action, although taken, is quickly overwritten by the internal command).

4.7 Maintenance Download: memory dump

Several methods are presented below to perform a diagnostic memory readout on the 3 processors.

The first set of commands are timed to avoid overloading the internal communications resources. Enough time is allowed between successive commands to allow any blocked traffic to be cleared.

The second command set may be used if the instrument is in a quiet state with no other command or data activities in progress. Note that the time synchronization activity is suspended for the duration of the dump.

The third set of commands operates all three processors in parallel to achieve an elevated delivery rate. Normally the MRO activity is throttled to achieve a delivery rate of around 1000 bits of MRO data each second (it is actually 1 MRO record every 9 RTI periods). By commanding all 3 processors, in the correct order, to perform their memory dump, all three processors delivery MRO data at their 1 Kb/sec rate for an aggregate rate of 3 Kb/s.

In all cases there is an additional group of MRO records that are generated to flush any data of interest out of the BIU.

The critical timing for these command sets are the gap following the **73MRO** command. The delta is critical to avoid overflowing internal buffers and to avoid exceeding target data delivery rates.

- Conservative

Data rate less than 1000 bits / second. Does not interfere with time update activity. Allows 30 seconds for each **73MRO** command.

- Alternate

Data rate less than 1000 bits/second. Interferes with time update, so time update suspended. Allows more than 5 minutes for each group.

- Fast

Data rate less than 3000 bits/second. Interferes with time update, so time update suspended. Allows more than 5 minutes for each group of three commands. All three processors active at one time.

The timing and order of commands for the *Fast* dump are critical. Altering timing or order of commands can result in resource exhaustion and loss of the memory dump. A resource exhaustion can cause the processor to crash, requiring a reset to recover.

4.7.1 Conservative memory dump command sequence

```
00:00   73MEM_TWEAK, HRP, BYTE, 0X14, 0X40, TWEK
00:05   73MRO, HRP, TLM, 8000, 8BFF
00:35   73MRO, HRP, TLM, 8C00, 97FF
01:05   73MRO, HRP, TLM, 9800, A3FF
01:35   73MRO, HRP, TLM, A400, AFFF
02:05   73MRO, HRP, TLM, B000, BBFF
02:35   73MRO, HRP, TLM, BC00, C7FF
03:05   73MRO, HRP, TLM, C800, D3FF
03:35   73MRO, HRP, TLM, D400, DFFF
04:05   73MRO, HRP, TLM, E000, EBFF
04:35   73MRO, HRP, TLM, EC00, F7FF
05:05   73MRO, HRP, TLM, F800, FFFF
05:30   73MEM_TWEAK, HRP, BYTE, 0X14, 0XC0, TWEK
05:35   73MRO, HRP, TLM, 8000, 8BFF
06:05   73MRO, HRP, TLM, 8C00, 97FF
06:35   73MRO, HRP, TLM, 9800, A3FF
07:05   73MRO, HRP, TLM, A400, AFFF
07:35   73MRO, HRP, TLM, B000, BBFF
08:05   73MRO, HRP, TLM, BC00, C7FF
08:35   73MRO, HRP, TLM, C800, D3FF
09:05   73MRO, HRP, TLM, D400, DFFF
09:35   73MRO, HRP, TLM, E000, EBFF
10:05   73MRO, HRP, TLM, EC00, F7FF
10:35   73MRO, HRP, TLM, F800, FFFF
11:00   73MEM_TWEAK, HRP, BYTE, 0X14, 0X00, TWEK
11:05   73MEM_TWEAK, DCP, BYTE, 0X14, 0X40, TWEK
11:10   73MRO, DCP, TLM, 8000, 8BFF
11:40   73MRO, DCP, TLM, 8C00, 97FF
12:10   73MRO, DCP, TLM, 9800, A3FF
12:40   73MRO, DCP, TLM, A400, AFFF
13:10   73MRO, DCP, TLM, B000, BBFF
13:40   73MRO, DCP, TLM, BC00, C7FF
14:10   73MRO, DCP, TLM, C800, D3FF
14:40   73MRO, DCP, TLM, D400, DFFF
15:10   73MRO, DCP, TLM, E000, EBFF
15:40   73MRO, DCP, TLM, EC00, F7FF
16:10   73MRO, DCP, TLM, F800, FFFF
16:35   73MEM_TWEAK, DCP, BYTE, 0X14, 0X00, TWEK
16:40   73MEM_TWEAK, LRP, BYTE, 0X14, 0X40, TWEK
16:45   73MRO, LRP, TLM, 8000, 8BFF
17:15   73MRO, LRP, TLM, 8C00, 97FF
17:45   73MRO, LRP, TLM, 9800, A3FF
18:15   73MRO, LRP, TLM, A400, AFFF
18:45   73MRO, LRP, TLM, B000, BBFF
19:15   73MRO, LRP, TLM, BC00, BFFF
19:40   73MEM_TWEAK, LRP, BYTE, 0X14, 0XC0, TWEK
19:45   73MRO, LRP, TLM, 8000, 8BFF
20:15   73MRO, LRP, TLM, 8C00, 97FF
20:45   73MRO, LRP, TLM, 9800, A3FF
21:15   73MRO, LRP, TLM, A400, AFFF
21:45   73MRO, LRP, TLM, B000, BBFF
22:15   73MRO, LRP, TLM, BC00, C3FF
22:40   73MEM_TWEAK, LRP, BYTE, 0X14, 0X00, TWEK
```

4.7.2 Alternate memory dump command sequence

```
00:00    73MEM_TWEAK,  LRP, WORD, 0x60, 0x00, LOCK
00:05    73MEM_TWEAK, HRP, BYTE, 0X14, 0X40, TWEK
00:10    73MRO, HRP, TLM, 8000, FFFF
05:35    73MEM_TWEAK, HRP, BYTE, 0X14, 0XC0, TWEK
05:45    73MRO, HRP, TLM, 8000, FFFF
11:05    73MEM_TWEAK, HRP, BYTE, 0X14, 0X00, TWEK
11:10    73MEM_TWEAK, DCP, BYTE, 0X14, 0X40, TWEK
11:15    73MRO, DCP, TLM, 8000, FFFF
16:40    73MEM_TWEAK, DCP, BYTE, 0X14, 0X00, TWEK
16:45    73MEM_TWEAK, LRP, BYTE, 0X14, 0X40, TWEK
16:50    73MRO, LRP, TLM, 8000, BFFF
19:45    73MEM_TWEAK, LRP, BYTE, 0X14, 0XC0, TWEK
19:50    73MRO, LRP, TLM, 8000, C3FF
22:45    73MEM_TWEAK, LRP, BYTE, 0X14, 0X00, TWEK
22:50    73MEM_TWEAK,  LRP, WORD, 0x60, 0xFF, LOCK
```

4.7.3 Fast memory dump command sequence

```
00:00   73MEM_TWEAK,  LRP, WORD, 0x60, 0x00, LOCK
00:05   73MEM_TWEAK, ALL, BYTE, 0X14, 0X40, TWEK
00:10   73MRO, DCP, TLM, 8000, BFFF
00:15   73MRO, HRP, TLM, 8000, BFFF
00:20   73MRO, LRP, TLM, 8000, BFFF
02:50   73MRO, HRP, TLM, C000, FFFF
05:20   73MEM_TWEAK, ALL, BYTE, 0X14, 0XC0, TWEK
05:25   73MRO, DCP, TLM, C000, FFFF
05:30   73MRO, HRP, TLM, 8000, BFFF
05:35   73MRO, LRP, TLM, 8000, BFFF
08:05   73MRO, HRP, TLM, C000, FFFF
10:35   73MEM_TWEAK, ALL, BYTE, 0X14, 0X00, TWEK
10:40   73MRO, LRP, TLM, C000, C7FF
11:00   73MEM_TWEAK,  LRP, WORD, 0x60, 0xFF, LOCK
```

OR

```
00:00   73MEM_TWEAK,  LRP, WORD, 0x60, 0x00, LOCK
00:05   73MEM_TWEAK, ALL, BYTE, 0X14, 0X40, TWEK
00:10   73MRO, DCP, TLM, 8000, FFFF
00:15   73MRO, HRP, TLM, 8000, FFFF
00:20   73MRO, LRP, TLM, 8000, BFFF
05:20   73MEM_TWEAK, ALL, BYTE, 0X14, 0XC0, TWEK
05:30   73MRO, HRP, TLM, 8000, FFFF
05:35   73MRO, LRP, TLM, 8000, BFFF
10:35   73MEM_TWEAK, ALL, BYTE, 0X14, 0X00, TWEK
10:40   73MRO, LRP, TLM, C000, C7FF
11:00   73MEM_TWEAK,  LRP, WORD, 0x60, 0xFF, LOCK
```

4.8 Maintenance Download: Internals

The maintenance download consists of 2 unique modules, several modifications to existing science modules and several unaltered science modules. There is a short discussion of the changes to each module listed below.

There is a potential dead-lock situation that occurs in connection with the "F5" queue which might be triggered if the DCP or HRP fail (if LRP fails, it's all over anyway). This potential dead-lock is addressed by discarding messages within the Storage Manager.

4.8.1 HFR Venus operation

This activity is triggered using **73IEB_TRIGGER, 0, 2** and requires approximately 3 hours to perform the full data acquisition. During this period the HFR is repeatedly sent an analysis command that take approximately 12 seconds to execute and the results are stored onboard the instrument for relay to the ground. Due to the volume of data that is presented in housekeeping, it is necessary to leave the instrument on for almost a full day following the operation. As the status information is saved in memory within the instrument it is not essential that housekeeping is collected during the checkout activity. There is a desire, however, to relay the data to the ground as soon as possible to avoid the loss of data caused by any unforeseen event (i.e. SEU of unplanned power cycle).

The BIU discrete status bits used for antenna deploy activities are used for a similar purpose during the HFR checkout operation. BIU Discrete status bit 5 is set when the HFR acquisition cycle operation completes successfully. BIU Discrete bit 7 is set in the event that the HFR Checkout acquisition cycle is not able to complete as expected.

The HFR Venus mode must **NOT** be initiated immediately following any maintenance or download operation. The instrument should be allowed to ***settle*** **for a period of several minutes** before the **73IEB_TRIGGER** is sent to the instrument to allow power supply switching to complete. In addition it will be necessary to leave the instrument powered on until the status information is retrieved from housekeeping. If the telemetry link to the ground is of poor quality (i.e. data dropouts occur), the housekeeping may be collected for a longer period of time in an attempt to recover lost frames (the data from the HFR Venus operation is repeated every 1024 housekeeping frames). It is also acceptable to collect housekeeping from the instrument at an elevated rate. The software will deliver new data approximately every 15 seconds so a housekeeping collection schedule may be changed by a factor of four. Faster collection schedules will not cause problems, this is simply all the faster new data can be placed into the housekeeping buffer.

Dumping the full status area requires a minimum of 19 hours of normal housekeeping collection. The memory dump commands, above, may be used to reduce the time required to perform the memory dump when science telemetry is available at rates above 4,000 bits/sec.

Keep in mind that the 73MRO commands in the previous sections are also available to dump the stored data although ground processing may not be immediately available for this format of data.

4.8.1.1 Venus observation commands

The following commands may be used to initiate the Venus lightning observation. Delta between commands are minimum required delta. Note, in particular, that the instrument requires approximately two minutes following the maintenance download to configure itself into an idle state. The configuration period is used to deliver commands from LRP to the L/P handler on the HRP. No other commands should be delivered to the instrument during this configuration period.

```
00:00:00   73RPWS_POWER, ON

00:00:05   73RT_RESET, RELEASE
00:00:10   73RT_RESET, RESET
00:00:15   73RT_RESET, RELEASE
                                        ←Allow reset activities to finish
00:00:20   73RT_SLEEP, ACTIVE
00:00:25   6EXT_MEM_LOAD, , ,RPWS
                                        ←Delta of about two minutes
           here
00:02:30   73IEB_TRIGGER, MASK, 0, 2
                                        ←Delta of about 150 minutes
           here
02:32:30   73RT_SLEEP, SLEEP
```

4.8.1.2 Housekeeping during Venus observation

During the Venus observation, the observation data is placed into housekeeping for delivery to the ground. Data is recovered in an asynchronous manner from the internal data storage area. The recovery mechanism is, more or less, using the same addressing as the data production code, although the two step will typically be out of synchronization with each other.

This lack of synchronization may result in buffers being presented in housekeeping before data has been loaded into the buffer. Since the housekeeping handler expects a valid micro-packet header, uninitialized data may produce unexpected results. Once the data collection activity passes the data delivery activity, the micro packets presented in the housekeeping will contain valid data.

4.8.1.3 Venus observation data

The HFR is placed into a mode that generates data records that contain less than 122 bytes of data to allow a compressed packet to occupy a 128 byte record in the *ramdisk* The *ramdisk* will be discussed in the following paragraphs). The overhead for each record that is kept is the minipackt identification/length field, the time tag field, and the segmentation field. A checksum is also generated and included with the data to make bit errors visible as the spacecraft is operating close to the Sun and is more susceptible to bit errors.

The HFR status words that indicate the mode the instrument is in are removed prior to saving the data. This information does not change and is replaced on the ground.

4.8.1.4 Timetags for the Venus observation

The timetag placed on the Venus observation data is not time tagged in quite the same manner as data that is collected during the remainder of the mission. We know, going in to the observation, that the 16 bit RTI field is marginally adequate to cover the time period. The observation is expected to take about 2.25 hours (8100 seconds) which is only about three minutes short of the time resolution of the RTI field in the minipacket. Loss of a portion of the observation could make rebuilding the time information more difficult. By redefining the RTI filed in the minipacket to be a seconds counter (rather than an RTI counter), any ambiguity about when a data packet was taken by the HFR is removed.

Since the 16 bit field now covers a little more than 18 hours and the MRO is scheduled to occur within minutes of the end of the observation, there will be enough time information present in each CDS record to accurately time tag the data to within 1 second.

We also know that the HFR will cycle every 7.75 seconds (62 RTI periods). Assuming that the HFR cycles at this rate, we can further refine the time information that is included with each record to provide an accuracy of 250 mSec (2 RTI periods). This is accomplished by replacing the RTI field in each minipacket with an estimated value. The estimated value for each successive record is decremented by two. When a correct starting point is chosen (0, 2, 4, or 6) the result will be a delta of 62 between successive timetags (using any of the other choices will result in the delta value being wrong).

4.8.1.5 Recovery of Venus observation data, HSK

Not a planned method, although software exists to recover the data from the housekeeping buffers.

4.8.1.6 Recovery of Venus observation data, MRO

Data recovery from the science telemetry, in the form of MRO records, is accomplished using a GSE program called **venus**. This program reformats the MRO records, replacing the missing status bytes and rebuilding the time information. In addition a time sort may be performed on the data to place it time order (it is recovered from memory in a rather jumbled manner).

4.8.2 New Modules

Storage Manager
Ram Disk Manager

These modules are used to manager the complement of on-board memory that is not actively used by the maintenance software. Specifically, none of the maintenance code makes use of the upper memory available on the 3 processors. These modules provide access to the upper memory areas on the 3 processors. This memory are is logically presented to the application programs as a contiguous area of memory that is addressable on 32 bit boundaries. The logical address space is 256 K bytes with either 128K or 192K being physically available to the user.

The logical to physical mapping is somewhat contorted to place the last logical memory block in a region of memory on the LRP that is corrupted during a reset operation. The HRP memory is located in the bottom of logical address space. The DCP is located in the middle of address space followed by the LRP that is located at the top of address space.

The implementation used to support the Venus observation does not include bulk memory in the logical address space. The handler is logically prepared to deal with the bulk memory, but due to proximity to the Sun, this memory is used to hold a backup copy of the operating software.

Maintenance Mode Manager

The maintenance mode manager monitors the BIU discrete command bit that is used to place the Langmuir Probe into maintenance mode. When the instrument is commanded into the maintenance mode (i.e. the BIU discrete bit is asserted) the maintenance manager sends commands to the Langmuir Probe handler and issues 00PORT_TWEAK commands to the power control ports on both HRP and LRP to bias the sphere to the desired voltage.

The BIU discrete command bit that controls ***Sleep*** is also monitored, In the event that this discrete bit goes active (i.e. to a logic 0), the DCP and HRP sleep control lines are asserted and the internal power control bits are all cleared. This brings the instrument into a low power state in the RTI period following the arrival of the sleep indication.

The maintenance mode manager does not accept any commands (other than the 2 BIU discrete bits mentioned) so it is not possible to use the **73POWER_CNTL** command to switch power or enter a sleep state.

The maintenance mode manager in version 2.2 contains a flaw that prevents the maintenance status bit, that is presented in one of the BIU discrete status bits, from being properly asserted when the instrument in maintenance mode. The maintenance activity is performed correctly, in spite of the indication in the BIU discrete status bit. Version 2.3 addresses this issue.

4.8.3 Modified Modules

Langmuir Probe handler

This handler is linked such that all memory allocations are made within the bottom half of the processor address space. Other that the change in the memory map, this module is identical to the science module. The bias voltage applied to the sphere is determined by sending normal **73LP** commands to the *Langmuir Probe handler*.

Version 2.2 software introduces a slight change into the handler and the method used to handle SLEEP. This change is reflected in the special maintenance download and should not have any visible effect on maintenance operations as the L/P is never operated in a data acquisition mode. The L/P module is provided in the special maintenance download as a means to control the L/P sphere bias voltage.

IPC driver, DCP

Some F5 buffers are allocated in the top half of processor address space, these buffers are not present in the special maintenance load. As the DCP is not used to perform data compression there is no need for these additional free space buffers.

HFR handler

Changes to accommodate data acquisition at Venus. The modification alters the size and destination of the HFR minipackets. Minipacket size is reduced to 128 bytes to allow the data to be routed through the variable area in the housekeeping packet. In addition, some of the header bytes with fixed values are stripped from the minipacket prior to delivery. And finally, a CRC is generated and placed at the end of the minipacket as an integrity check of the data (may have been stored on the instrument for some time and been affected by an SEU).

IEB Handler

The ability to store and process **IEB_LOAD** commands has been effectively removed. The memory area used to hold the IEB is located in the top half of processor address space and has been moved to unused BIU memory for this load. As a reset clears all of BIU memory, the **IEB_LOAD** is lost when the instrument is reset.

The **73IEB_TRIGGER, MASK, 0, 2** is unique to this load as well. This trigger is included to perform the HFR Venus activity. This trigger also applies power to the HFR prior to sending commands for the Venus data acquisition.

4.8.4 Unchanged Modules

Kernel

The kernel should not require any alterations to accommodate any downloaded software (there is no *mission/task specific* code in the kernel).

IPC driver, LRP and HRP

The free space buffers on LRP and HRP all reside in the bottom half of processor address space and do not interfere with the memory manager.

Memory Tweak

Identical module used with science, deploy and special maintenance.

BIU handler

Identical module used with science, deploy and special maintenance.

5 Science Operations

How data is collected.

5.1 Process Priority.

All processes running within the instrument are assigned a priority that is used to allocate the CPU resource. The priority values run from 1 through 127 with 1 being highest priority and 127 being the lowest. In order to streamline the process dispatcher it is mandatory that at least one process always exist that uses the CPU. This last position is typically occupied by the *Idle Process* that is assigned a priority of 126.

Because the *Idle Process* is not using a priority value of 127, it is possible to block any process by simply altering it's priority to 127. Once a process has it's priority set to 127 it will be moved to the end of the compute list the next time the dispatcher runs (typically at the start of the next RTI period).

When the dispatcher is activated, it takes the current process and moves it to a position in the computable list just ahead of any lower priority jobs (this accomplishes round-robin scheduling when more than 1 process use the same priority). A side-effect of this action being that the idle process will always be placed just ahead of any tasks with a priority of 127. The dispatching activity keeps the computable process list in order by priority if left undisturbed. In addition, when a process alters it's own priority (either using the system call or directly altering its process descriptor), it will be moved to the appropriate place in the computable list the next time that the dispatcher runs.

If priority is altered using the **73MEM_TWEAK** command there is a condition that can be triggered that prevents the altered process from regaining access to the CPU when more than 1 process has been idled through modification of the process priority.

Consider that 2 process's have been idled using **73MEM_TWEAK** to a lower priority of 127(i.e. below Idle process priority). Typically the compute list will consist of the idle process followed by process ***A*** and then by process ***B***. If we attempt to activate process ***B*** by returning the priority to it initial value, the dispatcher will never bring to the head of the compute list as the idle process will always be placed immediately before process A (the dispatcher aborts the scan of the computable list when it encounters any process with a priority below that of the current process). Process ***B***, in other words, effectively disappears from the system, at least until process ***A*** receives the attention of the dispatcher.

The solution, of course, is to bring all inactive process's to a priority above the idle process.

A final note on priority. Simply stated, **the highest priority process on the compute list receives the CPU•** (Emphasis on the period there) Having a process priority slightly lower does not imply slightly less CPU cycles are allocated. A given process will be given all the CPU cycles it requires, at the expense of lower priority processes. I/O activity will typically relinquish control of the CPU as a side effect of a wait (for data from a queue, for data from an acquisition, for an MX protected resource, or using the Delay system service). **A compute bound processes must have the appropriate priority•** (there's that emphatic period again) Releasing to the scheduler (using the S_Dispatch system service) will not allow any lower priority process to gain control of the CPU (it just don't work that way).

Using a priority of 0x7E will probably result in using about half of the remaining CPU cycles, sharing them with the Idle process.

Process Priorities

Process Name	*Processor*	*Process Priority*	*Process Address*	*Comments*
IPCD	**ALL**	**0x1E**	**1200**	IPC handler Watchdog timer
IPCR	**ALL**	**0x1F**	**1180**	IPC handler Receive
IPCX	**ALL**	**0x20**	**1280**	IPC handler Transmit
BIUH	**LRP**	**0x23**	**1B00**	BIU Handler
TWEK	**ALL**	**0x32**	**1300**	Memory Tweak / Memory Readout
CMD	**LRP**	**0x32**	**1B80**	Instrument command decoder
LOCK	**LRP**	**0x32**	**1900**	Time Lock
SOND	**LRP**	**0x32**	**2800**	HFR Sounder
LP_C	**HRP**	**0x32**	**1600**	Langmuir Probe command
CMPX	**DCP**	**0x32**	**2700**	Compression Read
MFRC	**LRP**	**0x33**	**2980**	MFR Command decode
FBC_ hi	**HRP**	**0x63**	**2100**	WBR/WFR command decode
MFRI	**LRP**	**0x64**	**2900**	MFR data acquisition loop.
LP_I	**HRP**	**0x64**	**1700**	Langmuir Probe Input
STIM	**DCP**	**0x64**	**3E00**	Stimulus Echo (IEB trigger echo)
LP_O	**HRP**	**0x65**	**1680**	Langmuir Probe Output
W08I	**HRP**	**0X65**	**21C0**	8 bit Data Acquisition
W12J	**HRP**	**0X65**	**2160**	12 bit Data Acquisition
WBRC	**HRP**	**0x66**	**24D0**	WBR Control
DSTC	**HRP**	**0x66**	**2540**	DUST Control
LFDC	**HRP**	**0x67**	**2458**	LFDR Control
WFRC	**HRP**	**0x67**	**23E0**	WFR Control
FBC_ lo	**HRP**	**0x68**	**2100**	WBR/WFR command fetch
WBRX	**HRP**	**0x69**	**2300**	WBR/DUST LRS Delivery
WFRX	**HRP**	**0x6A**	**2220**	WFR LRS Delivery
LFDX	**HRP**	**0x6A**	**2290**	LFDR LRS Delivery
DIR_	**HRP**	**0x6B**	**2370**	WBR/WFR HRS Delivery

Process Name	*Processor*	*Process Priority*	*Process Address*	*Comments*
IEBC	**LRP**	**0x78**	**3300**	IEB Command Decode
SSR_	**LRP**	**0x78**	**3C00**	SSR bit management
CMPX	**DCP**	**0x78**	**2700**	Compression Compute
HSK_	**LRP**	**0x7A**	**1600**	Housekeeping
AGC_	**DCP**	**0x7A**	**3D00**	AGC for WFR/LFDR
LFDR	**DCP**	**0x7A**	**1600**	LFDR FFT analysis
HFRI	**LRP**	**0x7B**	**1D00**	HFR Input
HFRO	**LRP**	**0x7C**	**1D80**	HFR Output
DUST	**DCP**	**0x7E/7D**	**2770**	DUST Analysis V2.6/V2.7
IDLE	**ALL**	**0x7E**	**1100**	Idle Process

LOCK/SOND on LRP

These processes do not interfere as neither use significant CPU cycles. LOCK has a several minute window in which to complete it's activities.

DUST on DCP

DUST process priority is altered to 0x7D when using C39 and later IEB's.

This process attempts to consume all remaining CPU cycles on the DCP. Version 2.6 and prior had the priority incorrectly set to the same as the IDLE process, potentially starving Dust of CPU cycles. Apply the following tweak to correct this:

73MEM_TWEAK, DCP, BYTE, 0x15, 0x7C, DUST

W08I/W12J on HRP

In most cases WBR is scheduled to start on the next RTI (WFR is always scheduled on the upcoming RTI) so it makes little difference if the 8 bit setup or the 12 bit setup occurs first. Resource conflicts are resolved automatically through the use of Mx semaphores.

Process Names

Process Name	*Processor*	*Process Priority*	*Process Address*	*Comments*
AGC_	**DCP**	**0x7A**	**3D00**	AGC for WFR/LFDR
BIUH	**LRP**	**0x23**	**1B00**	BIU Handler
CMD	**LRP**	**0x32**	**1B80**	Instrument command decoder
CMPX	**DCP**	**0x78**	**2700**	Compression Compute
CMPX	**DCP**	**0x32**	**2700**	Compression Read
DIR_	**HRP**	**0x6B**	**2370**	WBR/WFR HRS Delivery
DSTC	**HRP**	**0x66**	**2540**	DUST Control
DUST	**DCP**	**0x7E/7D**	**2770**	DUST Analysis V2.6/V2.7
FBC_ hi	**HRP**	**0x63**	**2100**	WBR/WFR command decode
FBC_ lo	**HRP**	**0x68**	**2100**	WBR/WFR command fetch
HFRI	**LRP**	**0x7B**	**1D00**	HFR Input
HFRO	**LRP**	**0x7C**	**1D80**	HFR Output
HSK_	**LRP**	**0x7A**	**1600**	Housekeeping
IDLE	**ALL**	**0x7E**	**1100**	Idle Process
IEBC	**LRP**	**0x78**	**3300**	IEB Command Decode
IPCD	**ALL**	**0x1E**	**1200**	IPC handler Watchdog timer
IPCR	**ALL**	**0x1F**	**1180**	IPC handler Receive
IPCX	**ALL**	**0x20**	**1280**	IPC handler Transmit
LFDC	**HRP**	**0x67**	**2458**	LFDR Control
LFDR	**DCP**	**0x7A**	**1600**	LFDR FFT analysis
LFDX	**HRP**	**0x6A**	**2290**	LFDR LRS Delivery
LOCK	**LRP**	**0x32**	**1900**	Time Lock
LP_C	**HRP**	**0x32**	**1600**	Langmuir Probe command
LP_I	**HRP**	**0x64**	**1700**	Langmuir Probe Input
LP_O	**HRP**	**0x65**	**1680**	Langmuir Probe Output
MFRC	**LRP**	**0x33**	**2980**	MFR Command decode
MFRI	**LRP**	**0x64**	**2900**	MFR data acquisition loop.

Process Name	*Processor*	*Process Priority*	*Process Address*	*Comments*
SOND	**LRP**	**0x32**	**2800**	HFR Sounder
SSR_	**LRP**	**0x78**	**3C00**	SSR bit management
STIM	**DCP**	**0x64**	**3E00**	Stimulus Echo (IEB trigger echo)
TWEK	**ALL**	**0x32**	**1300**	Memory Tweak / Memory Readout
W08I	**HRP**	**0X65**	**21C0**	8 bit Data Acquisition
W12J	**HRP**	**0X65**	**2160**	12 bit Data Acquisition
WBRC	**HRP**	**0x66**	**24D0**	WBR Control
WBRX	**HRP**	**0x69**	**2300**	WBR/DUST LRS Delivery
WFRC	**HRP**	**0x67**	**23E0**	WFR Control
WFRX	**HRP**	**0x6A**	**2220**	WFR LRS Delivery

Process Address

Process Name	*Processor*	*Process Priority*	*Process Address*	*Comments*
IDLE	**ALL**	**0x7E**	**1100**	Idle Process
IPCR	**ALL**	**0x1F**	**1180**	IPC handler Receive
IPCD	**ALL**	**0x1E**	**1200**	IPC handler Watchdog timer
IPCX	**ALL**	**0x20**	**1280**	IPC handler Transmit
TWEK	**ALL**	**0x32**	**1300**	Memory Tweak / Memory Readout
LP_C	**HRP**	**0x32**	**1600**	Langmuir Probe command
LFDR	**DCP**	**0x7A**	**1600**	LFDR FFT analysis
HSK_	**LRP**	**0x7A**	**1600**	Housekeeping
LP_O	**HRP**	**0x65**	**1680**	Langmuir Probe Output
LP_I	**HRP**	**0x64**	**1700**	Langmuir Probe Input
LOCK	**LRP**	**0x32**	**1900**	Time Lock
BIUH	**LRP**	**0x23**	**1B00**	BIU Handler
CMD	**LRP**	**0x32**	**1B80**	Instrument command decoder
HFRI	**LRP**	**0x7B**	**1D00**	HFR Input
HFRO	**LRP**	**0x7C**	**1D80**	HFR Output
FBC_ lo	**HRP**	**0x68**	**2100**	WBR/WFR command fetch
FBC_ hi	**HRP**	**0x63**	**2100**	WBR/WFR command decode
W12J	**HRP**	**0X65**	**2160**	12 bit Data Acquisition
W08I	**HRP**	**0X65**	**21C0**	8 bit Data Acquisition
WFRX	**HRP**	**0x6A**	**2220**	WFR LRS Delivery
LFDX	**HRP**	**0x6A**	**2290**	LFDR LRS Delivery
WBRX	**HRP**	**0x69**	**2300**	WBR/DUST LRS Delivery
DIR_	**HRP**	**0x6B**	**2370**	WBR/WFR HRS Delivery
WFRC	**HRP**	**0x67**	**23E0**	WFR Control
LFDC	**HRP**	**0x67**	**2458**	LFDR Control
WBRC	**HRP**	**0x66**	**24D0**	WBR Control
DSTC	**HRP**	**0x66**	**2540**	DUST Control

Process Name	*Processor*	*Process Priority*	*Process Address*	*Comments*
CMPX	**DCP**	**0x32**	**2700**	Compression Read
CMPX	**DCP**	**0x78**	**2700**	Compression Compute
DUST	**DCP**	**0x7E/7D**	**2770**	DUST Analysis V2.6/V2.7
SOND	**LRP**	**0x32**	**2800**	HFR Sounder
MFRI	**LRP**	**0x64**	**2900**	MFR data acquisition loop.
MFRC	**LRP**	**0x33**	**2980**	MFR Command decode
IEBC	**LRP**	**0x78**	**3300**	IEB Command Decode
SSR_	**LRP**	**0x78**	**3C00**	SSR bit management
AGC_	**DCP**	**0x7A**	**3D00**	AGC for WFR/LFDR
STIM	**DCP**	**0x64**	**3E00**	Stimulus Echo (IEB trigger echo)

5.1.1 Priority Inversion

Processes on the DCP that make use of the MAC chip are susceptible to priority inversion problems. Care should be exercised when building fas LRS modes that depend on DCP. They typical symptom we observe when this occurs is intermittent loss of data.

In particular, both the AGC and LFDR analysis processes make use of the MAC chip. An MX Flag is used to mediate access to the chip. Since MX_Acquire requests are served on a first-come first-served basis, it is normal for the low priority process to cause the higher process to be blocked when both require the MAC chip. As the MAC chip is not a bottleneck, as long as CPU cycles are available to route data through the chip, no deadlocks can occur. If, however, other activities on the DCP (such as Walsh/Rice compression) dominate the available CPU cycles, the common use of the MAC chip by AGC and LFDR may succeed in blocking the wrong process.

This condition is not fatal, however, it will simply result in some of the data analysis steps being skipped (i.e. lost data records). Once the demand on CPU cycles abates, the waiting process will regain access to the CPU, completing it's task and releasing the MAC chip.

Lack of available CPU cycles is simply an indication that the workload on the DCP is excessive.

5.2 Basic Scheduling

The basic scheduling period for activities within the instrument is based on the data collection schedule used by the spacecraft. This period, called the ***RTI period*** is 125mS long and assumed to be reasonably accurately timed by the spacecraft. The spacecraft begins each data collection interval by sending a command to all terminals on the bus indicating the beginning of the period. This message generates a signal within the DPU's that causes an interrupt that is used to drive the process scheduler.

As part of the scheduling activity a delay list is updated and any processes on the list that require the CPU are scheduled for execution.

5.2.1 Fixed Schedules

The MFR operates on a fixed schedule. The lower 5 bits (D0 through D4) of the time field supplied by the S/C is used to control the sweep of the MFR with the next bit (D5) used to toggle the antenna.

The MFR operation is fixed and cannot be altered without modifications to the software that manages the MFR.

Another example of a fixed schedule is the time update activity. This task maintains the S/C time field on the HRP and DCP at regular (fixed) intervals.

5.2.2 Pickup based schedules

The housekeeping process is scheduled based on the rate CDS collects housekeeping data. Once housekeeping has completed a collection cycle (i.e. updated all of the fields in the housekeeping packet), it waits for the BIU handler to signal that CDS has picked up the updated packet (using a *flag-wait* system call). Housekeeping consumes the CPU and A/D resources in proportion to the pickup rate. We typically use a pickup rate of 24 bits/second (one HSK packet every 64 seconds) on the spacecraft and a pickup rate of 192 bits/second (one HSK packet every 8 seconds) on the bench.

Pickup rates as high as 1536 bits/second have been used on the bench, which is faster than the housekeeping process cycles through a collection cycle. In this case, housekeeping will run continuously while managing to update the housekeeping packet in about 6 seconds. I the cae of elevated pickup rates, housekeeping requires only a few CPU cycles each RTI period as it waits for the A/D system (using the *delay* system service).

Another example of a pickup based schedule is operating the WBR in a free-run mode. Here, the WBR is allowed to produce as much data as the HRS data stream can hold. Buffer resources are held by a formatting process until there is room to move them to the BIU before being returned to the WBR process for collection of more data.

5.2.3 *Mod* based schedules

All of the instrument timing is based on a ***SCLK*** used in a ***mod()*** function. ***SCLK*** is divided by the specified schedule period and the data acquisition is scheduled when the remainder of the division matches the specified remainder.

This scheduling method can cause conflicts within the instrument when several activities are scheduled at conflicting times. The conflicts are resolved by delaying some of the activities. The resolution does not delete an activity, but simply delays until the first activity completes.

Additional conflicts can occur within the antenna selection matrix. The signal multiplexer on the MFR, WBR, and WFR are all connected to the outputs of a single set of input amplifiers. Some glitching may occur whenever the multiplexers are switched. The MFR, for example operates on a 32 second collection schedule and may be programmed to switch antennas on every cycle. This switching will, in most cases, cause a glitch on the signal line associated with the antenna that was just selected by the MFR.

The switching transients are most evident on the WBR and WFR.

Another point to bear in mind with this scheduling method is that the activity being scheduled relinquishes use of the processor rather than continuously polling the system time. This reduces the CPU cycles required to perform the scheduling but can have interesting side-effects when it becomes necessary to change the scheduling interval. A good example may be made of the WFR which typically schedules at around a 600 second interval. Changing to a short sample interval may require up to 5 minutes to take effect (i.e. the process is completely dormant for the period between samples, it will not have access to the CPU to reschedule any activities until the next scheduled sample time)

5.2.4 IEB Based Scheduling

A higher level of scheduling may be achieved through the use of **IEB** loads. The IEB interpreter provides looping and timing constructs that may be used to trigger data acquisitions using a scheduling scheme not present in the instrument handlers.

5.2.5 WFR/LFDR configuration

Please refer to the *Instrument Handlers* section when reviewing this section. Some detailed knowledge of the process structure on HRP that manage WFR and LFDR is necessary to understand the following discussions.

The WFR/LFDR handlers provide buffer management, synchronization and data routing functions on the HRP and may be configured in what might be considered and unusual manner to accomplish some unusual data gathering tasks. Version 2.6 provides some additional clues to routines that run on DCP that perform compression, AGC, and Analysis of the 12 bit data. In particular, we now propagate the AGC enable bit through to DCP so that AGC processing may be skipped when not required (in prior versions DCP would perform the AGC analysis whenever data happened to come through DCP, even when AGC was turned off).

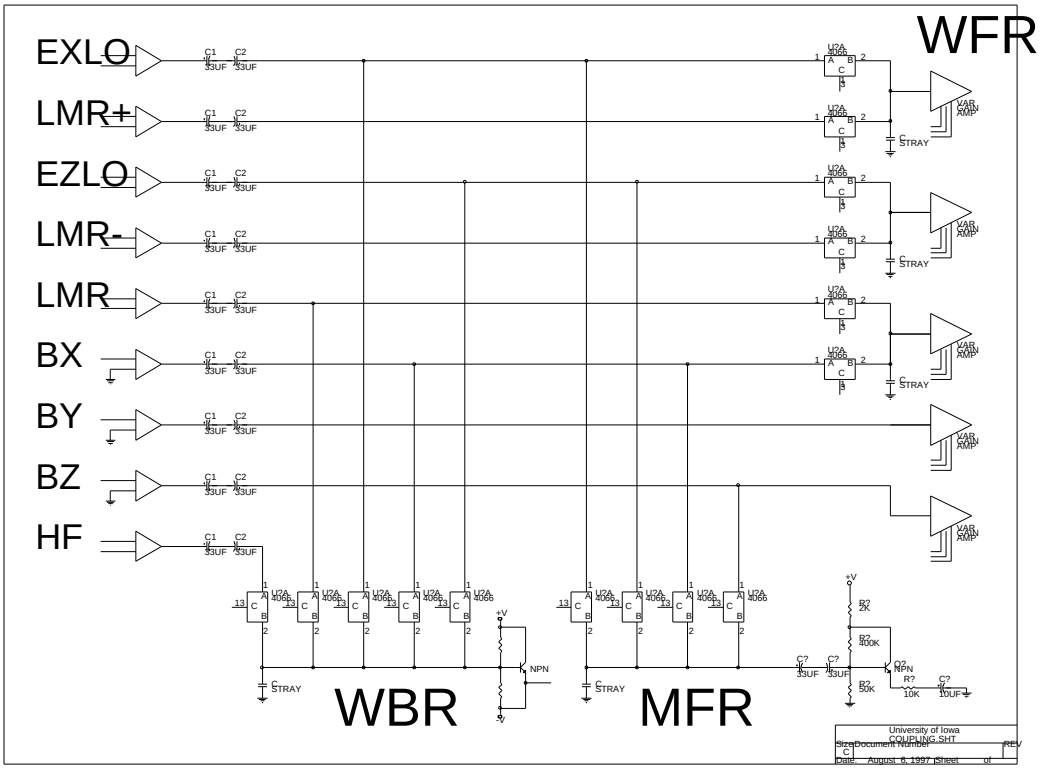
We can, therefore, route both WFR and LFDR through the same delivery process on HRP to the same compression process on DCP, provided that we can keep the gain response straight. WFR and LFDR use a common gain control process with the two sources kept separate knowing on which input queue (on DCP) the data arrived on. If we mix LFDR and WFR data on the WFR queue, it becomes necessary to disable gain control on the LFDR (assuming the WFR is in a 5 channel mode and we want to manage all 3 gain controls).

(We may want to route in this manner to provide a little mode temporal resolution on the electric channels, as an example)

5.3 Antenna Switching

Antenna inputs are not buffered between the differential amplifiers and the inputs to the various receivers. Analog switches are used to connect the output of the differential amplifier to the buffer amp on the input to each of the receivers. Switching on one of the analog switches will typically cause a transient to occur on the signal line. Although this is of little consequence for the receiver that is being switched, as it is not acquiring data, the other receivers may be active and capture the glitch. This behavior must be allowed for when designing data acquisition schedules.

As an example, the MFR, when in toggle mode, switches on a regular 32 second boundary that is naturally aligned with the S/C time field. DUST captures are scheduled when CPU cycles on the DCP are available and are delayed during MFR antenna switches. WFR should be scheduled such that data captures do not occur when MFR switches. High rate WBR activities may experience these switching transients.



5.3.1Antenna Switching and AGC (WBR)

Antenna switching can affect AGC operation on the WBR as it makes use of a signal integrator (as opposed to a software AGC scheme as used on the WFR where we look at the waveform and make gain change decisions). With the WBR, any transients caused by changing antenna inputs are integrated along with the signal from the selected antenna and used to determine when the gain needs to be changed.

The antenna selected between datasets is may be controlled by the 8 bit data acquisition process. In most cases the most recent WBR or DUST mode control command will set the antenna multiplexor control field in the acquisition process. This means that the mode control command that specifies the desired antenna mux setting should occur last in a sequence of setup commands. The antenna selection may also be explicitly altered by using a memory tweak command.

Operating both WBR and DUST at the same time can present a problem when attempting to utilize the AGC function as the antenna switching transients may cause an artificially high signal level to be indicates in the AGC reading. We can correct this by operating both logical instruments in manual gain mode, of course, but this is somewhat lewss than optimal.

The acquisition process is intended to operate in a default mode where the antenna selection is set prior to a data set and not changed. This method would be appropriate when WBR is operating in a burst mode, we would see a gain change (or perhaps two) when the burst occurs (and possibly when DUST begins acquiring data following the burst), but the selected gain should settle on an appropriate level.

5.4 Low Rate Science

Low rate science may be produced at any time and will be marked for delivery to the spacecraft whenever enough data has been placed in the outgoing buffer area to completely fill a science telemetry packet.

All of the receivers may be active when the instrument is in an LRS mode. In other words, LRS does not refer to data from a specific receiver, rather it refers to the logical path the data takes out of the instrument and into the spacecraft.

Bandwidth using the *Low Rate Science stream* is architecturally limited to approximately 30Kbit/sec. This limit is calculated using a packet pickup every other RTI periods as the *Low Rate Science stream* is not double buffered. One RTI period is required for CDS pickup and a second RTI is required for LRP to move data into the BIU memory (depending on CPU availability, more than 1 RTI may be required for data movement, further limiting the peak LRS data rate).

5.5 High Rate Science

This mode of operation requires that the spacecraft collect data at least every other **RTI** period. This mode is required to make use of telemetry bandwidth in excess of 30Kb/s.

Only instruments located on the **HRP** are capable of delivering data using **HRS**. Currently only the **WBR** and **WFR** receivers are capable of delivering to **HRS.** Also note that only the WBR is capable of producing sufficient data to allow data rates in excess of 100Kb/s to be sustained (there are hardware limitations that limit the WFR data rate).

Prior to V2.5, packet sequencing was normally added to the HRS packets by the BIU handler (process ***BIUH***) on the LRP. The process on the HRP, DIR_, also sequences the HRS packets although this sequence field is overwritten on the LRP. As the packet sequence is always placed in the CCSDS header on the HRP, it will become *visible* on the ground if the sequencing performed by the LRP is suppressed. With the introduction of the V2.5 flight software, sequencing on the LRP is normally suppressed to avoid a problem with improper HRS sequencing. This change, along with other changes in the BIU handler closes an ISA.

Version V2.6 adds several enhancements to WBR, LDFR, and LFDR capabilities. WBR adds a burst mode that allows a programmable number of WBR data sets to be acquired in a burst. This capability allows the instrument to be configured to collect high rate data but leaving the WBR initially inactive. As data bursts are desired, they may be activated using a single command without the need to worry about getting WBR stopped at a later time (this has a favorable impact on IEB memory requirements and also provides a simple means to trigger these collection activities directly from the spacecraft).

The WFR and LFDR have been enhanced with a band toggle capability. This allows the 12 bit receivers to switch between high band and low band with each data set acquisition. The data acquisition process, on HRP, provides AGC support, but the AGC code is located on the DCP. If the DCP code is not updated to accommodate the toggle mode, it is necessary to operate with AGC disabled to avoid problematic situations (where the secondary band data is analyzed causing a gain level change on the primary band).

The gain control commands for both WFR and LFDR have been enhanced to provide control over the new toggle capability.

5.5.1 73MEM_TWEAK, LRP, BYTE, 0x32, n, BIUH

High Rate Science Control

This field must be set to 0xFF in order to allow high rate science activities to occur.

5.5.2 73MEM_TWEAK, LRP, BYTE, 0x33, n, BIUH

Sequencing control, prior to Version 2.5

- 0x00 LRP sequences HRS data

  When the flag is set to zero, the LRP will update the sequence field within each CCSDS record that is to be passed on to the spacecraft. Any data that is lost between HRP and LRP will not be evident to the spacecraft or ground system.

- 0xFF LRP passes HRS unchanged

  When the flag is set to any non-zero value LRP will ignore the sequence field within each CCSDS record that is to be passed on to the spacecraft. LRP will calculate the appropriate number of CCSDS records to deliver without changing the records in any way. Any data that is lost between HRP and LRP will not become evident to the spacecraft and ground system.

Version 2.5 alters the default sequencing of HRS data by inverting the meaning of the flag. No commanding, up to this point in time, has attempted to change the default behavior so we simply altered the meaning of the default to disable LRP sequencing. This is part of a fix to eliminate a problem with duplicate packets that appeared when telemetry modes changed at the same time the RPWS modes changed.

This change also make packet loss between HRP and LRP visible.

Sequencing control, Version 2.5

- 0xFF LRP sequences HRS data

  When the flag is set to zero, the LRP will update the sequence field.

- 0x00 LRP passes HRS unchanged

  When the flag is set to any non-zero value LRP will ignore the sequence field within each CCSDS record that is to be passed on to the spacecraft.

The remainder of the CDS Packet formatting is performed by a single process on the HRP. This process is called BIU Direct and has an internal name of "**DIR_**". The formatting may be performed in several distinct ways.

5.5.3 73MEM_TWEAK, HRP, BYTE, 0x60, n, DIR_

Scheduling mode.

0 Programmed move

Data movement is accomplished using program loop. Peak data rates are on the order of 100Kb/sec. The only resource required to support this method of movement is the processor.

1 Simple DMA

Data movement is accomplished using the 8237 channel 0 and 1. Exclusive access to the 8237 is obtained by requesting the mutual exclusion flag associated with the 8237. The WFR, particularly when operating in low-band will, interfere with this data movement scheme (as the 8237 is occupied for long periods of time).

2 DMA/Programmed Mx dependent

This is a combination of the first 2 methods. When a data movement is necessary, the mutual exclusion flags for channels 0 and 1 are inspected and if both are free the data movement is performed using the 8237. If either/both mutual exclusions indicate a channel in use, the move is performed using a program loop. This method attempts to strike a balance between speed and resource requirements.

This is a high data rate mode designed to accommodate the WFR/LFDR operating in low-band (i.e. 10mSec sample period). When WFR is used at the low sample rate, the associated 8237 channel is locked for a period of seconds. In order to avoid blocking WBR activity there are hardware assists designed to synchronize access to the 8237 during WFR/LFDR activities.

In this mode, when WBR data needs to be moved, each WFR/LFDR sample causes an interrupt to be generated. The interrupt service routine examines a register that indicates the time remaining before the next WFR/LFDR sample is to be taken. If sufficient time is available, the 8237 word count and address register for the WFR/LFDR are saved and then the 8237 is programmed to move a block of data. Following the block move, the original word count and address are re-loaded and the 8237 then continues with the WFR/LFDR activity.

This is workable only when the WFR/LFDR is in low band where the sample rate is 10 milliseconds. It is desirable, although not necessary, that the L/P be inactive when operating in this mode as it will block both WBR and the data movement activity (i.e. It will depress the bit rate).

When WFR is operating in high band, the code reverts to using a programmed move, which is quite slow. The test for WFR operation is performed once for each group of data movements, so the slowdown should be temporary.

4 Burst DMA

This defers the data movements to allow the mutual exclusion system service request to be used to control access to the 8237 for an entire group of moves. The system service calls to gain exclusive access to the 8237 consume as much CPU time as the data movement operation, so reducing the number of system calls relative to the number of data movement operations should result in a noticeable performance improvement. This mode should be interchangeable with **mode 1**.

5.5.473MEM_TWEAK, HRP, BYTE, 0x70, n, DIR_

RST-5 interrupt time remaining point.

When operating in mode 3 (using the LFSR sample clock as an interrupt source to perform DMA memory moves) this is the value used to determine if there is adequete time remaining in the sample interval to perform the DMA move.

The hardware register that is read is a 16 bit register that is clocked with the 6Mhz system clock. It is preset to 60,000 at the start of each LFDR sample and is primarily used to generate the next data sample (i.e. Every 10 milliseconds). The upper 8 bits of this counter is made available to the processor (i.e. The HRP). The register is not latched so the processor can read the register as a carry ripples through the counter so it is necessary to take steps to insure that the register contents are valid (i.e. multiple reads). The lowest visible bit changes state every 42.666 microseconds and the initial value is 0x15 and it increments until reaching 0xFF.

The value obtained from this hardware register is compared with the value at offset 0x70 and the DMA operation is performed only when the contents of the register are less than the indicated value. Keeping in mind that the HRP does not make use of the NMI (non maskable interrupt), we can determine, through instruction timings, if there is sufficient time remaining in the RTI to perform the move, restore the 8237, and clear the interrupt prior to the next LFDR sample.

Version 2.7 software has a default value of 0x??.

5.6 High Rate Processor Resources

The ***HRP*** has very limited hardware resources available for performing it's tasks. The operating system provides the application code with mechanisms to control access to and allocate limited resources. It is possible (likely) that a data acquisition schedule will be specified that will request operations that are physically impossible to perform and will be rescheduled. This rescheduling is evident in the data time tag and cannot be avoided although it should be possible to adjust the acquisition schedule to better meet the science objectives of an observation.

5.6.1 HRP Memory

The ***HRP*** memory consists of 96K bytes of main memory and 64K bytes of ***BULK memory***. The complement of main memory is statically allocated to be used as data buffers for the Langmuir Probe, WBR, WFR, and LFDR (Dust analysis is a derived mode of the WBR and shares buffers with WBR). In addition to the data acquisition buffers, the software requires various buffers and, of course, space for the operating code.

The BULK memory is not directly addressable by the processor and is not used in support of data acquisition. *BULK memory* typically is used to hold a backup copy of the operating software.

WFR and LFDR have separate buffers to accommodate the different data rates at which these logical instruments operate. This allows the WFR to continue to operate without being blocked due to a buffer of LFDR data that hasn't been completely processed.

WBR and DUST are not expected to operate at the same time so buffer contention is not considered an issue. In the event that DUST and WBR are scheduled concurrently, the impact of the shared buffers in minimal as there are minimal delays imposed by the delivery process (WFR and LFDR have significant delivery delays)

5.6.2 8237 controller

DMAThe 8237 controller is a 4 channel controller. The requirements imposed by the design require the use of 7 channels.

1. L/P DAC
2. L/P A/D
3. WBR (8 bit A/D)
4. WFR (12 bit A/D)
5. IPC (communications channel to LRP/DCP)
6. Memory to Memory
7. Memory to Memory (requires 2 channels of the 8237)

The ***Memory to Memory*** channel does not require any additional hardware support so it simply makes use of two channels used by WBR, and WFR. In addition, the WBR channel is shared with the L/P DAC and is used to perform sweep operations.

5.6.2.1 DMA Channel 0/1 (block move)

Using the ***8237*** to perform block moves requires the first 2 channels of the device. There are several schemes used to insure that the device operates correctly.

- Programmed

Programmed moves are enumerated here to make the block move list complete. The programmed move makes use of a programmed loop to move characters (i.e. it does not require the use of the 8237). This method limits the bit rate that HRP can format to about 100,000 bits/second.

- Simple

This is the simple case. The mutual exclusion flags for both DMA channels are requested and the process relinquishes control of the CPU until both channels are free. The 8237 is then programmed to perform the move. Note that the overhead associated with the *Mx* (mutual exclusion) activity is significant and limits the bit rate the instrument is capable of operating at.

- Fast

This case attempts to reduce the overhead associated with acquiring and releasing the lock on the 2 DMA channels. The move routine inspects the memory location used to keep track of the *Mx* associated with the channels and performs a programmed move when the *Mx* for either channel would block. If both *Mx* flags are free, interrupts are disabled and the 8237 is programmed to perform the move. This scheme avoids most of the overhead involved with managing access to the 8237. Note that this scheme may result in lower than expected data rates as the *Mx* flag for one of the 8237 channels will tend to be in use.

- Interrupt

This case attempts to use the 2 DMA channels while one channel is active servicing the LFDR at a sample rate of 100Hz. The hardware generates an interrupt when the LFDR sampling is complete (i.e. every 10mS). The interrupt service routine must then determine how much time has elapsed from the interrupt request until the service routine receives control of the CPU and proceed to setup a block move if sufficient time is available before the next LFDR sampling period.

The kernel provides a mechanism to attach an interrupt routine into the kernel interrupt handler in support of this hardware assist. The interrupt routine has access to the hardware timer used to generate the LFDR sample clock. This mode must be used with extreme care (i.e. both 8 bit and 12 bit configuration must be considered jointly, it is rather simple to misconfigure the hardware and cause the HRP to stall in an interrupt routine).

5.6.2.2 DMA Channel 0 (Langmuir Probe/WBR)

The Langmuir Probe DAC and the WBD ADC share this DMA channel. L/P software must have exclusive access to the DMA channel before setting the control bits that determine which of the instruments are connected to the DMA channel.

When performing density measurements, the Langmuir Probe requires a single DMA channel (i.e. channel 3). The density measurement does not typically present any resource conflicts as the sample rate is slow enough to allow a full WBR data set to be acquired between samples.

When a swept measurement is required, a DAC is supplied with data from a table in memory by channel 0 of the 8237. This is shared with the WBR and protected by an *Mx* flag that is acquired before the 8237 hardware is accessed.

5.6.3 Compression Hardware(ISFLIP)

The hardware compression hardware (the ***ISFLIP*** chip) may be connected to either the WBR converter of the WFR converter. Although it is impossible for both WFR and WBR to make use of the ISFLIP hardware simultaneously, access to the compression hardware is protected through the use of a mutual exclusion flag.

The compression hardware is primarily intended to be used with the WBR.

5.6.4 CPU(block move)

Performing data movement using the 8085 limits bit rates to approximately 100K bit/second.

5.6.5 CPU(WBR high band)

The WBR, when operating in high band, causes the 8237 to dominate the processor bus for the duration of a data acquisition. This causes all other activities on the HRP to be suspended. This potential interference is managed by having the WBR handler synchronize with the L/P through an mutual exclusion flag and with the WFR using one of two methods.

5.6.5.1 WFR MX

The WBR handler acquires the lock on the WFR DMA channel to prevent WFR from acquiring data when WBR is active. In the event that WFR is active when this request is made of the kernel, WBR will relinquish control of the CPU until the WFR has completed it's activities.

5.6.5.2 LFDR SYNC

The WBR handler ignores the lock on the WFR DMA channel, but sets a hardware control bit to cause the WBR acquisition to occur following the next WFR sample. When the WFR is sampling at 100Hz and the WBR data set size is at or below 2048 samples, the WBR will complete a data acquisition before the next WFR sample occurs.

5.6.5.3 WBR MX

The WFR acquisition process will ask for the WBR Mx when it is operating in high band mode and notices that either WBR or DUST has been commanded to operate in high band mode. This allows the LFDR to operate in low band with WBR

synchronizing with the LFDR sampling (WBR acquires 2048 samples or less to complete activity prior to the next LFDR sampling) while having occasional WFR acquisitions in high band.

The Langmuir Probe shares resources with the WBR and must, therefore, suspend operations when WBR is active (reverse situation also applies). The WFR, however, may be active when L/P is performing a sweep. Version 2.2 of the L/P driver does not attempt to lock out the WFR when performing a sweep, resulting in interference being visible on the electric antenna.

5.6.6 IPC and BIU direct

When operating in any HRS modes, the IPC channel must handle this traffic and there are potential interactions with other uses of the IPC channel. Carefree scheduling of outgoing traffic can result in unexpected depressions of the HRS bit rate.

Consider, for a moment, how BIU Direct traffic is scheduled for delivery by the IPC handler on HRP. We do not have a dedicated means of sending status from LRP to HRP to interlock delivery of HRS traffic. The IPC mechanism is too slow to allow an acknowledge for each buffer delivered so HRP must establish some means of avoiding an overrun condition. This is accomplished by delaying delivery of a BIU Direct buffer until the beginning of the RTI period (through the use of the DELAY system service). This effectively forces a period of time where outgoing traffic is suppressed. In most instances, other data sources on the HRP will not be generating IPC traffic at a sustained level (two or fewer IPC packets per RTI, on average). In most cases, BIU Direct traffic will be interspersed with other IPC traffic and the delay that occurs prior to the delivery of HRS traffic simply slows other traffic slightly without affecting delivery to any significant degree.

If, on the other hand, we start to accumulate any volume of normal IPC traffic, each succeeding BIU Direct packet may not be processed soon enough to cause the delay call to occur in the same RTI period as the preceding BIU Direct packet. When this occurs, more than one RTI period will have elapsed between BIU Direct packets, causing a corresponding drop in bit rates.

In particular, we have a WFR/LFDR setup that attempts to produce high bit rates (high for WFR, anyway). We configure WFR to collect on a 24 second schedule and pack the data (packing is faster than compressing as it only performs the 16 bit to 12 bit pack). The WFR delivery schedule is setup to deliver a minipacket segment every second (there are 20 segments). To allow the data to flow through IPC at this rate, the minipacket assembler has been flagged to deliver the WFR segment (consisting of 5 IPC buffers) without the normal 1 RTI delay. This causes a group of 5 IPC packets to be delivered to IPC handler in a group. When BIU Direct traffic is present, this group of WFR packets will depress the HRS bit rate as the BIU Direct traffic cannot be delivered every RTI period. The solution is simple, although there is the potential to cause buffer overruns on DCP when using this. Change the delivery schedule (WFRC offset 0x54) from 8 (1 segment per second) to 4 (1 segment every ½ second) and restore the minipacket assembler delay flag to cause the segment to be delivered 1 IPC packet per RTI. This prevents loading the IPC transmit queue with too many IPC packets while maintaining the WFR throughput. One side-effect of this is that DCP will now be holding the segment in F5 buffers for several RTI periods, with the potential to cause buffer overruns (DCP has about 10K-11K of buffer space with each LFDR segment requiring 768 bytes and each WFR segment requiring 1280 bytes).

5.6.7 CPU cycles on DCP

Although not part of the HRP resource pool, CPU cycles on the DCP will impact some of the high bit-rate WFR/LFDR modes. One area that is not as obvious is the loss of AGC function when insufficient cycles are available to perform an occasional analysis of the 12 bit waveform data in order to adjust the gain levels. Reducing bit rates or eliminating AGC control will solve this contention.

5.7 Science Software memory dump

In order to perform a ground analysis on the science software the following memory dump may be used. This dump assumes that science telemetry in excess of 4500 bits/second is available. The dump is performed in this order to leave the memory tweak process bank select byte pointing to bank zero on the LRP (in order to protect IEB memory from corruption).

00:00	**73MEM_TWEK, LRP, WORD, 0x60, 0x00, LOCK**
00:05	**73MEM_TWEAK, ALL, BYTE, 0X14, 0XC0, TWEK**
00:10	**73MRO, HRP, TLM, 8000, FFFF**
00:15	**73MRO, LRP, TLM, 8000, FFFF**
05:30	**73MEM_TWEAK, ALL, BYTE, 0X14, 0X00, TWEK**
05:35	**73MRO, DCP, TLM, 0000, 7FFF**
05:40	**73MRO, HRP, TLM, 0000, 7FFF**
05:45	**73MRO, LRP, TLM, 0000, 7FFF**
11:00	**73MEM_TWEAK, ALL, BYTE, 0X14, 0X40, TWEK**
11:05	**73MRO, DCP, TLM, 8000, FFFF**
11:10	**73MRO, HRP, TLM, 8000, FFFF**
11:20	**73MRO, LRP, TLM, 8000, FFFF**
17:00	**73MEM_TWEK, LRP, WORD, 0x60, 0xFF, LOCK**

5.8 Internal Interference

Several of the receivers perform active stimulation of the local environment resulting in interference the is coupled from one antenna to another (Langmuir Probe sweeps, HFR Sounder activity). Active stimulation is noted in the ancillary data that is delivered to the S/C each second. HFR Sounder activity would normally be coordinated with activity on the other receivers due to the use of a common antenna. The Langmuir Probe, on the other hand, has the spherical probe dedicated for it's exclusive use. It is possible to perform a L/P sweep while other parts of the instrument are active and the sweep will couple into the electric antennas.

Two sources of internal interference will also be evident in the data. The antenna switching matrix within the WFR, WBR, and MFR is susceptible to switching transients. In other words, when any of these three receivers is switched, the newly selected antenna will, in most cases, have a transient impressed on that antenna (this interaction is described earlier in this section). This interference is entirely within the WFR/WBR/MFR electronics.

A third source of interference in the HFR direction finding mode.

5.9 Muted operations

A mode of operation suggested for use during probe checkout is to leave the instrument powered and simply mute the 1553 transmitter. The muting can be achieved in at lest two different manners, first by CDS sending a 1553 bus command to mute individual RT's and second by CDS tripping the BIU watch-dog timer (this can be accomplished by command or through benign neglect).

The question then comes up as to how RPWS software handles this situation. In the nominal case, we can expect no adverse reactions to having the transmitter muted. If carelessly applied, however, the instrument can become constipated and require a transition through a SLEEP mode to recover.

5.9.1 Benign muting

If muting occurs when the S/C telemetry mode does not include RPWS data pickups, then the instrument should simply stop producing data and nothing unusual should occur.

This scenario causes high rate data to be suppressed by LRP.

In the event that CDS continues to request data from RPWS, the BIU may think data is being picked up and the LRP will continue to post data in the BIU for delivery (LRP would not be aware that the BIU is muted).

5.9.2 Impacted muting

If muting occurs when S/C telemetry mode indicates that RPWS should be sending data (either high rate or low rate), then several problems might occur.

CDS would not receive a response from RPWS and mark the instrument as off-line? Telemetry mode continues to be broadcast (indicating RPWS should be producing data) but CDS would not schedule RPWS pickups. LRP would then have to discard LRS data and the free space buffers would be over committed resulting in free space exhaustion. Commanding would be compromised as there would be no memory available to store incoming commands. This may trigger a timing interaction that may cause the HRP to remove power from the WBR/WFR A/D section and to power down the entire L/P.

Recovery from a HRP power-down would require transition to SLEEP, back to ACTIVE, then a 73IEB_TRIGGER, MASK, 0, 0.

This type of muting can be simulated (on the engineering model)by simply powering down the S/C simulator (i.e. Heurikon) for a period of time. This leave RPWS in a valid telemetry mode with no pickups. Strange things happen although it is important to note that the instrument does not crash (no software reload is required to recover).

5.9.3 Mute tests (5/2003)

In preparation for some of the probe relay activities, we have proposed that RPWS simply be muted, rather than being placed in sleep. This is simply to avoid a power cycle of the receivers that is caused by placing the instrument in sleep mode.

5.9.3.1 1st. Test

Instrument was loaded with FSW V2.6, powered up and started with trigger 26. After several minutes (about 10 science telemetry packets) the BIU was muted using the command: **73rt_wdterr_rpws, set** which causes the BIU transmitter to be disabled. The instrument was the left running for about an hour, during which no data was received by the PPCRTIU (due to the BIU being muted). This test indicates that the instrument operates with no ill-effects. When the BIU is un-muted at the end of the test, data flow resumes (i.e. Housekeeping and Science appear again). The only indication that the mute occurs is the loss of data and a corresponding gap in the packet sequence numbers.

TLM MODE S&ER-3
Load FSW V2.6
73IEB_TRIGGER, MASK, 0, 0
73IEB_TRIGGER, ID, 28, 0
73RT_WDTERR_RPWS, SET
(about 1 hour)
73RT_WDTERR_RPWS, CLEAR

5.9.3.2 2nd. Test

We will attempt to simulate the sequence of events a little better in this test. This time we are operating in trigger 28. These next test will attempt to verify that the instrument is rather insensitive to having data collection suspended.

With this sequence of events the packet sequence numbers will not indicate any lost data. This is because the telemetry mode has no pickups so BIU does not produce any HSK or SCI telemetry. Muting in this sequence is not even noticed in the telemetry data.

TLM MODE S&ER-3 (from previous test)
FSW V2.6 (from previous test)
73IEB_TRIGGER, ID, 28, 0 (from previous test)
TLM MODE PRLY-ONLINE (C303 0321)
73RT_WDTERR_RPWS, SET
(about 1 hour)
73RT_WDTERR_RPWS, CLEAR
TLM MODE S&ER-3

5.9.3.3 3nd. Test

We will attempt to simulate a slight change in the sequence of events. This time we are operating in trigger 28 (as before). This difference here being a change in to order that the mute and telemetry mode occur in.

As with the 1st test, we drop those packets that were asked for when the BIU is muted, so there is a gap in the packet sequence numbers. As expected, the instrument operates through the muted operations without problem.

TLM MODE S&ER-3 (from previous test)
FSW V2.6 (from previous test)
73IEB_TRIGGER, ID, 28, 0 (from previous test)
73RT_WDTERR_RPWS, SET
TLM MODE PRLY-ONLINE (C303 0321)
(about 1 hour)
TLM MODE S&ER-3
73RT_WDTERR_RPWS, CLEAR

6 IEB Internals

Instrument Expanded Blocks.

Memory within the instrument used to store sequenced commands.

Version 2.6 changes

Version 2.6 Flight Software introduces two minor changes in IEB handler behavior.

With this update the occurrence of the **73IEB_LOAD** command alters a timer that allows the IEB handler to run every RTI period. This timing change allows a load to occur at a 1 command per RTI rate (this is to support the LSF addition to CDS software).

The IEB handler also marks IEB memory as invalid when a **73IEB_LOAD** command is received. The only way to mark IEB memory as operational is to successfully perform a checksum validation. The checksum validation is triggered when a **73IEB_EOF** command is received by the instrument.

Version 2.4 changes

Version 2.4 Flight Software introduces a major change in the method used to load IEB tables into the instrument. Prior to this release, the IEB checksum was calculated within the IEB handler as part of the IEB loading process. A lost block would not be automatically detected when using the old load method. With the new method, the checksum table is calculated and built on the ground to be delivered with the IEB Load. Any lost blocks should result in a mis-match between the table delivered with the load and that table as calculated within the instrument. The result of the checksum calculation is now available in the housekeeping telemetry. In addition, an invalid checksum will inhibit execution of IEB triggers, preventing the instrument from executing invalid commands.

An additional change in the IEB handler allows about half of IEB memory to be loaded via the ALF mechanism, as part of the instrument software load. This capability is intended to allow the instrument to be loaded with a baseline set of IEB triggers without any additional effort on the part of CDS or ground control.

As mentioned above, the IEB checksum must be valid in order to execute ID triggers. Supplying the checksum table externally aids in detecting lost or defective records. As occasional lost records are to be expected, the loading mechanism does not abort the load process when a lost record is detected (although the missing record is noted in a status word in housekeeping). This also allows the load to be attempted several times to reduce the impact of random missing load records.

6.1 How to manage and understand IEB loads.

Some tips and comments about how this function is implemented within RPWS.

IEB loads may be delivered to the instrument at a maximum rate or 1 IEB record per second (8 RTI periods). Delivering the **73IEB_LOAD** records to RPWS at a faster rate may cause a resource exhaustion within the instrument resulting in rejection of some of the **73IEB_LOAD** commands. For short loads, this restriction is not particularly noticeable but the restriction should be observed for all loads.

Starting with Version V2.4 flight software, a new command **73IEB_HALT, CLEAR** has been added to allow a single command to clear all of IEB memory and invalidate the checksum table. This command should be always used prior to an IEB load. Keeping in mind that the command clears IEB memory, it must be issued prior to the load(s) (in other words, the order is *clear, load, load, load* **not** *clear, load, clear, load, clear, load*).

Another point to keep in mind when working with IEB is that there is little in the way of error checking to prevent the user from sending an invalid trigger. The current software load (through at least V2.4) will attempt to process any trigger address it receives. One indication of an invalid trigger command is having the *Invalid Command Count* increment.

The steps involved in creating an IEB load are as follows:

1) For each instrument mode, construct a command script, including 73MEM_TWEAK commands to define the receiver cycle times. Test the command script by sending from the ground.

2) When the command script is working satisfactorily, create an AVOCET assembler file by invoking the command parser on the HP like this

parser -avocet [*mode_name*] < *input_file_name* > *output_file_name*

Note whether the command script was done as a two-part script. It is necessary to do this when the commands total 256 bytes or more. The parser automatically does this, but you need to be aware of it when you are invoking the [*mode_name*] from within the IEB.

3) Place the assembler file on the PC and assemble it by invoking the Avocet assembler:

avmac85 *output_file_name*

4) Repeat this process for as many modes

6.2 IEB Memory (V2.4 and later)

This is an area of memory used to store the internal sequence. Starting with version 2.4 flight software some changes/enhancements are present to make handling IEB memory a little easier.

Following a processor reset, IEB memory is inactive (i.e. deselected by the bank select hardware) and inaccessible (i.e. processor does not have code in the ROM to select the memory bank containing the IEB memory). There is no provision to directly load IEB memory using the ALF loader. Also keep in mind that part of the initialization process is to clear memory from 0000 through 7BFF to zero.

Following an ALF load (science software V2.4 or later) the IEB handler looks in memory (at location 5C00) for an IEB memory image. If location 5C00 is non-zero the IEB handler assumes that there is an 8K IEB memory image at 5C00 and moves the image to the beginning of IEB memory. The upper 8K of IEB memory is then cleared and the checksum table, located at 5B00 is moved to the top of IEB memory.

If the reload is from BULK Memory (due to a software crash, for instance), the IEB memory image will not appear in memory (it is not saved in BULK Memory) and the IEB handler will skip the entire data movement operation described above and IEB memory will remain unchanged through the reset/reboot operation.

At this point the IEB handler will verify the integrity of the IEB checksum table. We can expect the checksum to be valid in the case of an ALF load (as we carefully tested the load prior to submission), and since IEB memory is typically inactive, we might reasonably expect IEB memory to survive a software crash. If the checksum validation indicates an invalid checksum, the entire IEB memory area is cleared to zero and marked as invalid (preventing any **73IEB_TRIGGER, ID** commands from executing).

A valid checksum, of course, marks the IEB memory as being valid and will allow **73IEB_TRIGGER, ID** commands to execute.

As mentioned at the beginning of this section, IEB memory is marked as invalid upon reception of any **73IEB_LOAD** command and may only be marked as useable by successfully completing checksum validation (by receiving the **73IEB_EOF** command) [V2.6].

6.3 IEB_LOAD (V2.4 and later)

IEB memory is loaded using the **73IEB_LOAD** command. In this version of the software the sequence number is no longer used, anticipating that command loss in not unusual. Rather, the checksum is calculated on the ground and delivered with the load. If a record is lost, the checksum table is expected to indicate the problem through housekeeping so that additional attempts may be made to load IEB Memory from the ground. And invalid checksum during this load process does **not** result in IEB memory being cleared.

A load may be presented to the instrument several times with no ill effects. Assuming that errors are random in nature, we expect that at least one good copy of each record arrives at the instrument after several load attempts.

The arrival of the **73IEB_EOF** command triggers the checksum validation and subsequent flagging of the IEB Memory as being valid (or invalid, as the case may be).

Prior to the first **73IEB_LOAD** command, IEB Memory may be cleared using the **73IEB_HALT, CLEAR** command. This command may be issued several times in order to accommodate lost commands. This pre-clear operation forces IEB Memory into an initial condition (of all zero) in order to prevent a false indication of a valid checksum. (Note that when memory is cleared, the checksum pattern is well known).

6.3.1 IEB_LOAD in Real-Time

IEB Load Test

Testing of the V2.4 Software includes some testing of the improved IEB load facility. This test is intended to verify that the loader is capable of dealing with damaged loads, typically in the form of lost commands to the spacecraft.

The **73IEB_HALT, CLEAR** command(s) must be grouped together at the beginning of the load sequence. When this command is encountered, IEB Memory is cleared to zero so it should be obvious that this command will eliminate any previously received load commands. Multiple occurrences of the command are legitimate when a minimum of ?? seconds is allowed after each command.

Following the clear commands, several occurrences of the IEB load image may appear. In the course of testing, several successive loads were delivered with a **73IEB_END** (**73IEB_LOAD, 0, sequence**) following each load. Each occurrence of the 73IEB_END command causes a check of the integrity of the load with the status bits in housekeeping being updated.

A typical real time load might occur as follows:

- Instrument Power on and Flight Software load.

  Since V2.4 Software has an internal IEB Load, expect the IEB GOOD status bit to be set. All other IEB status in the housekeeping page should be zero.

  It is also reasonable to reload IEB's when the instrument is on and running. This would be accomplished by sending **73IEB_TRIGGER, ID, 10, 0** to bring the instrument to an idle state (allowing at least 60 seconds for the command to complete) or, possibly, by putting the instrument to sleep.

- **73IEB_HALT,CLEAR**

  This command may occur several times, with the appropriate delays following each command. All IEB status in the housekeeping is cleared to zero in preparation for loading a new IEB.

- **73IEB_LOAD**

  The IEB load is presented to the instrument, followed by an **73IEB_END** to cause checksum calculation to check the integrity of the load. Command good counters reflects the number of records loaded. If any are lost, the BAD status bit will be set to indicate the checksum did not match. If all goes well, the good count will indicate the correct number of records and the good status bit will be set.

- **73IEB_LOAD**

  Keeping in mind that light time to Saturn is on the order of 90 minutes, we can choose to send several copies of the load to the instrument to allow for loss during transmission. The subsequent loads are placed in memory, overwriting the current contents. If all goes well the good count in housekeeping will indicate the number of records in this load that were received (although not necessarily a complete load) with the good status bit indicating if the checksum correlated.

  It is possible, at this point, to have both the GOOD and the BAD status bit set. The GOOD bit is the significant status, it indicates the state of the checksum table. The BAD bit indicates that, at some point, that the checksum verification was unsuccessful.

  There is also, no instrument imposed limit on the number of time that the IEB Load may be delivered to the instrument.

6.3.2 IEB_LOAD V2.6 enhancements

Version 2.6 adds a fast IEB Load capability to the instrument software. This is to support the *Library Storage Facility* added to CDS flight software around the time the RPWS V2.6 was uploaded (late 2002/early 2003).

The IEB handler introduces a 1 second delay into each group of commands it processes to reduce impact on the free space buffers on the LRP. Without this delay, free space resources on LRP can easily be consumed distributing commands to the various subsystems. This delay is in the path of the **73IEB_LOAD** processing and limits the rate with which we can deliver an IEB load to the instrument. Attempting to deliver the command any faster consumes free space (as the command decoder does not see the 1 second delay) and typically results in a lost command, but may cause the LRP to stop processing commands and delivering data for a period as the IEB handler slowly releases buffers.

The V2.6 RPWS flight software has an *improved I*EB handler that accommodates the timing requirements imposed by the *Library Storage Facility*. In addition, the changes to the IEB handler inhibit execution of IEB triggers when the IEB memory checksum has not been verified. Any **73IEB_LOAD** command that the instrument encounters will inhibit execution of the **73IEB_TRIGGER** command until the checksum table has been verified (using the **73IEB_LOAD,0** form of the command). Timing in the 73IEB_LOAD path is conditioned by the checksum flag maintained by the IEB handler. When the execute inhibit flag is set , indicating that IEB_TREIGGER commands cannot be processed, the IEB handler will accept IEB_LOAD commands that are only 1 RTI apart. Once the execute inhibit flag is clear, the 8 RTI rate limiting is re-established.

In addition to allowing full speed loading of IEB images when using the *Library Storage Facility* this change allows for full speed loading of IEB's at any time.

The IEB execution inhibit flag is set whenever a valid **73IEB_LOAD** command is encountered (V2.4 and V2.5 only set the execute inhibit flag upon reception of a **73IEB_HALT, CLEAR** command). This, in effect, means that triggers are disabled unless a valid checksum pass has occurred since any changes to IEB memory have occurred. This alleviates the need to send **73IEB_HALT, CLEAR** command as part of a library load. In most other cases, the **73IEB_HALT, CLEAR** command should be used to clear memory to insure that all remnants of the old load are removed.

Note that the IEB loader flags in housekeeping are still rather tied to the use of the **73IEB_HALT, CLEAR** command to clear the good/bad bits. Although the appearance of a **73IEB_LOAD** command with a sequence number of 0 will clear the 7 bit good counter, the good bit remains in its last state. The occurrence of the last **73IEB_LOAD** command, the one with a length field of zero, causes the checksum operation to run and set the good bit if the load checksums. It is possible to have an IEB load that will not execute with the good bit set with the V2.6 load (good bit is set by the internal load, perform a load without clearing IEB memory that fails, and the good bit remains set even though the load will not execute).

6.3.3 IEB_LOAD command patterns (notes about zero fill)

The **73IEB_LOAD** command is a special case command. Most other RPWS commands (with the exception of **73ALF** commands) make use of a 3 bit length field in the first command word to define the command length and allow multiple commands to be packed together for delivery in a single RTI period. The **73IEB_LOAD** command consists of a single 2 word command followed by up to 120 words of IEB image (for a total of 124 words in the command block). The 1st Word, like all commands, contains a 4 bit routing field, a 3 bit length field (that indicates this is a two word command), a 7 bit command field and 2 parity bits. The 2nd word holds a word count and sequence number (both byte fields without parity). The word count indicates the number of words that follow that will be loaded into IEB memory beginning with an address word and ending with a checksum word.

The command decoder recognizes this command buffer as being a **73IEB_LOAD** command, extracts the needed words and passes them on to the IEB handler to be loaded into IEB memory. If there is zero fill, the command decoder treats the zero fill as additional commands in the command buffer by extracting and validating the parity. Since we make use of odd byte parity, a command of all zero is rejected as invalid due to the invalid parity. The net result is that commands with zero fill at the end will cause the *invalid command count* to increment. Since this zero fill is recognized as improper, it is discarded, along with the remaining portion of the buffer (i.e. any additional zero-fill words). This checking, being serial in nature, does allow the **73IEB_LOAD** to be delivered, so everything works out.

Note that zero fill will be tolerated in commands although the invalid command count increases with each occurrence. The invalid command count will, effectively, count command buffers, not the number of zero words that are encountered.

6.4 IEB Memory Readout Results

The following commands may be used to verify that a ***73IEB_LOAD*** was correctly received.

The first command changes the bank byte in the *memory readout process* to the appropriate bank (i.e. the bank in which the IEB data is stored). If this step is not taken, records that make up the MRO may come from either bank of memory.

The second command dumps a single *micro-packet* into the housekeeping that will contain the version string of the loaded IEB. This may be used to verify that the correct IEB was uplinked.

The third line dumps the checksum area of the IEB into the housekeeping. This dumps an area of 128 bytes at a rate of 10 bits/seconds, so it will take several housekeeping records to view all of the information. The housekeeping data is collected in 10 byte records.

The fourth line is a repeat of the 3rd line but dumped to the telemetry stream. In order for this to be delivered to the ground the S/C must be in a data mode that collects data from RPWS and the instrument must be in a mode that generated science telemetry. If either of these conditions are not met, the data will not appear on the ground in a timely manner.

The fifth command returns the memory readout process to a state that does not select a memory bank when active. This eliminates a small window where IEB memory may be easily corrupted by a system crash.

Note that the MRO that specifies telemetry in the routing field assumes that some form of science telemetry is active (such as MFR) to flush the MRO through the system. If none of the subsystems within the instrument have been powered on the alternate commands should be used to guarantee that a full CDS record is formatted and delivered to the spacecraft system.

6.5 IEB Memory Readout Commands

The alternate commands dump sufficient memory to flush the dump record out of the BIU.

- **73mem_tweak, lrp, byte, 0x14, 0xc0, TWEK**
- **73mro, lrp, hsk, 8000, 800f**
- **73mro, lrp, hsk, bf00, bf7f**
- **73mro, lrp, tlm, bf00, 0**
- **73mem_tweak, lrp, byte, 0x14, 0x40, TWEK**

Alternate

- **73mem_tweak, lrp, byte, 0x14, 0xc0, TWEK**
- **73mro, lrp, hsk, 8000, 800f**
- **73mro, lrp, hsk, bf00, bf7f**
- **73mro, lrp, tlm, bf00, c2ff**
- **73mem_tweak, lrp, byte, 0x14, 0x40, TWEK**

6.6 Memory Readout Interpretation

Two items are required to verify the IEB load. First the checksum area from a successful load (this would be obtained from the bench model during testing of the load) and the number of ***73IEB_LOAD*** records used to create the load.

If the number of ***73IEB_LOAD*** records is even the comparison is trivial. Simply compare the same number of bytes as there are ***73IEB_LOAD*** records. Each 16 bit field in the checksum area covers 256 bytes of memory while each ***73IEB_LOAD*** covers 128 bytes of memory (this effectively matches up ***IEB_LOAD*** records with checksum bytes).

If the number of ***73IEB_LOAD*** records is not divisible by two (i.e. is an odd number) then the comparison becomes difficult as the checksum covering the last ***73IEB_LOAD*** record does not necessarily operate over identical memory blocks (although the beginning of the page may be identical, the back half of the page does not necessarily match and it makes no difference if it doesn't match). This can be remedied by clearing IEB memory to zero or by creating ***IEB_LOAD*** in groups of two.

Note that the tally of ***73IEB_LOAD*** records counts only those records with data. Any leading or trailing records that are short should not be tallied.

When loading a new ***IEB_LOAD*** on to of an old ***IEB_LOAD*** memory contents are not cleared. This can lead to unexpected results, particularly when an initial ***IEB_LOAD*** fails and a successive load is attempted. Please bear in mind that the IEB_LOAD must be in sequence and checksum fields must be valid. If any of the verification steps fail, the failed record and any following are discarded. This also makes repeated reloads difficult to debug as you will not be able to pinpoint where a load failure occurs.

6.7 73IEB_TRIGGER, MASK, 0, 0

This trigger is part of the IEB handler (it is not downloaded using the 73IEB_LOAD command) and may be used at any time the science software is loaded to switch the instrument to full power (i.e. applies power to all of the receivers).

Power is sequenced to prevent current spikes from exceeding the power allocated to the instrument. The subsystems that have large initial current requirements are switched on before the sections that do not have noticeable surge current are powered.

6.8 73IEB_TRIGGER, MASK, 0, 1

This trigger is part of the IEB handler (it is not downloaded using the 73IEB_LOAD command) and may be used at any time the science software is loaded and power has been applied to begin producing science data.

6.9 73IEB_TRIGGER, MASK, 0, 2

This trigger is part of the IEB Handler loaded with the special maintenance software. It is intended to apply power to the HFR digital electronics and perform a diagnostic on the HFR with the results being stored in memory of the LRP, DCP, and HRP.

V2.3 and later software loads map this trigger to **73IEB_TRIGGER, MASK, 0, 1**.

6.10 73IEB_TRIGGER, MASK, 0, 3 V2.3

This trigger is part of the IEB Handler loaded with the science software. It is used to place the L/P into maintenance mode where the sphere is biased to 32 volts.

This trigger internally issues **73POWER_CNTL, SLEEP, ACTIVE** command prior to sending power commands for the L/P. This can cause the instrument to apply power to all receivers if this trigger is issued following a power down caused by the use of the **73POWER_CNTL, SLEEP, SLEEP** command.

6.11 73IEB_TRIGGER, MASK, 0, 4 V2.3

This trigger is part of the IEB Handler loaded with the science software. It is used to bias the L/P sphere to zero volts following the use of the previous trigger.

6.12 LOAD Problems

There are provisions in the IEB handler and ground software to address real-time loading of IEB memory. Discussions of workarounds appear near the end of this user guide.

6.13 Default IEB load

Version 2.4 flight software changes the load methodology. This change is intended to make the instrument more tolerant of random loss of IEB blocks. The IEB memory is relatively small, so this change will load all IEB_LOAD blocks presented to the instrument (previous versions stopped loading when a bad or missing block was encountered). Also, the checksum table for the load must be calculated on the ground and delivered along with the IEB load.

When the IEB handler sees the IEB_EOF command, the checksums will be validated and the results saved. IEB triggers are only executed if the checksum is valid.

Also with version 2.4 flight software, the IEB handler looks for a small (8K bytes) IEB load in memory and moves it to the IEB area when present. This small load may be presented to the instrument as part of the ALF load although it need not be saved in bulk memory. By placing the IEB image ahead of the flight software (with the BULK bit cleared) in the ALF load, a default set of triggers are made available without the need to send the load from the ground or through onboard sequencing. The downside of this scheme being that altering the IEB load involves re-submitting the flight software.

Keeping in mind that the ROM clears memory from zero through address 0x7BFF, the IEB handler looks for a non-zero value in location 0x5C00. If this location is non-zero, the IEB handler moves the 8K block and the checksum table to the appropriate region of LRP memory and proceeds to validate the checksum table. If this location (0x5C00) is zero, the handler skips the move operation and proceeds with the checksum.

If the checksum is valid, the handler will accept and process triggers. In the event that the instrument suffers a restart (SEU?), the bulk memory will contain only code (i.e. no IEB information), and the IEB handler will attempt to make use of the existing IEB load.

The IEB load may be replaced at any time by simply sending a new load to the instrument. The most recent load is used following a restart (assuming the memory containing the load has not been corrupted).

6.13.1 Default IEB details

Value	Description
0x5C00	LRP memory location Default IEB table ALF address
8192	LRP memory allocation Size of default IEB table
0x5B00	LRP memory location Default IEB table checksum ALF address

6.14 IEB binary images.

This section describes the layout of the IEB Memory are in the RPWS instrument. This block of memory is located on the LRP in the 16K region of memory that is banked. Within this area there are 2 data structures defined by the IEB handler as well as the commands patterns required to configure the various receivers and handlers within the instrument.

6.14.1 1st. Level, Program Control.

The first level data structure is program control. At this level we implement simple looping and program control. There are currently 3 (out of 4) instructions implemented. One of the instruction causes a group of instructions to be delivered within the instrument and the other two are used for program control.

Two of the commands are 32 bits in length while the third is 16 bits.

First Word		2nd. Word	Command / Operation Code
D15 D14	D13 . . . D00	D15 . . . D00	
0 0	Address of command block	Delta-T expressed in seconds	MODE
0 1	Address of next control element	(not used)	GOTO
1 0	Address of next control element	Loop Count	LOOP
1 1			spare

Since IEB memory consists of only 16K bytes, a 14 bit address is sufficient to address the entire block of memory. The address field is the byte offset within IEB memory. Since IEB Memory is naturally aligned (i.e. on a 16K byte boundary), the address field may also be generated by truncating the actual memory address to 14 bits.

Note that GOTO and LOOP commands point to more MODE, GOTO, and LOOP commands. A MODE command with a Delta-T of ZERO will, effectively, idle the IEB handler until additional IEB commands arrive. The result being the instrument handlers continue to produce data in their last commanded state.

6.14.2 2nd. Level, Command Control

This is the data structure pointed to by the MODE command. This structure points to a block of commands and contains additional command following.

Field Width	Field Contents	Explanation
16 bits	Command Block Address	Common commands This block of commands may be reused by any number of *Command Control* Blocks.
8 bits	Command Block Length 16 bit words	This field indicates the number of words that the command block contains.
8 bits	Tweak Block Length 16 bit words	Number of words that follow in the *Tweak Block*.
multiple of 16 bits	Tweak Block	These commands are delivered following the commands in the command block.

The intent of this structure is to allow blocks of commands to be referenced by multiple *Command Control Blocks*. Multiple MODE commands may be strung together to execute a sequence of commands with minor changes being applied in the *Tweak Block*.

Multiple MODE commands require a 1 second delay to allow the IEB handler to execute (a zero delay in the MODE command indicates that no more processing by the IEB handler is required)

6.14.3 Command Block

The command block consists of any reasonable number of instrument commands. These commands are delivered to the command decoder much as though they were delivered by CDS (at a point past the incoming byte counter).

When building Command Blocks, you must take care to avoid exceeding command decode limits within the instrument handlers. In most cases, avoid placing groups of similar commands together that exceed 100 bytes or so. Similar commands are those that have the same deliver index in the 1st. 4 bits of the command.

The Memory Tweak handler and the HFR handler are well know cases of command decoders that use a signed byte during command processing. When a byte count exceeds 7 bits, the command decoding will fail. The other parts of the instrument don't require the volume of commanding that HFR does for some operations. In the case of Memory Tweaks, most of the handlers use this mechanism to complement the commands defined for the instruments.

6.14.4 Tweak Block

The *Tweak Block* is used as an aid to allow reuse of *Command Blocks* and as a timing aid.

Existing command block from other triggers may be referenced with minor changes applied using the tweak block. This is an attempt to allow the use of IEB memory to be reduced as it is such a limited resource.

As a timing aid, the *Tweak Block* is delivered following the *Command Block*. Commands to begin data collection typically end up in the Tweak Block. These would usually be the *Memory Tweaks* required to initiate scheduling of data acquisition activities for the receivers.

6.15 An IEB overlay scheme for the *LSF* (Library Storage Facility)

CDS flight software added this enhancement early in 2003. The *LSF* provides around 20K bytes of storage on the spacecraft for IEB images (in the form of 73IEB_LOAD commands). CDS provides a mechanism to deliver some or all of the images to the instrument when needed.

RPWS loads are rather small and can be uploaded from the ground with little trouble (keeping in mind that there were enhancements in an earlier RPWS FSW load to accommodate a lossy uplink).

Consider, then, providing a relatively static IEB load from the ground that occupies about 12K of IEB memory and them overlaying the remainder is several portions from the *LSF*. The static portion of the IEB would hold the unchanging triggers (triggers 10-BE, for example) while the dynamic portion would be hold a small number of triggers (in the range C0-FE in this example) that are reloaded on an ongoing basis. The static portion of the IEB would have the dynamic triggers mapped through a *worm hole table* back to the idle trigger. Loading one (of n available) dynamic load would overwrite the *worm hole table* and the upper 4K of IEB memory and the checksum table. Now, since the loading of the dynamic portion is all onboard the S/C (and, therfore, very fast) we can change the dynamic portion as needed within a sequence. This would, in effect, increase the size of IEB memory to as much as 32K (i.e. 12K static and up to 5 by 4K dynamic), effectively doubling the size of IEB memory.

Power cycle survivability of this scheme is poor, at best, as it requires a large enough commanding window from the ground to reload the static portion of the IEB in the event of a power cycle. Also, if the system suffers a software glitch, either due to a bug of SEU, we run the risk of losing the contents of IEB memory (keeping in mind that during startup a single bit error in IEB memory is sufficient to cause it to be scrubbed).

A preliminary test of this scheme is present in the V2.6 BASE IEB image. In this load triggers C0 through FE point to a worm hole table located at 0x9FC0 (i.e. Trigger C0 is a jump to location 0x9FC0 which in turn contains a jump to the Idle Trigger, Trigger C2 is a jump to location 0x9FC2 which has a separate jump to the Idle Trigger). Page 0x9F00 does not contain any other code (i.e. it is filled with zero's) so it should be trivial to recalculate a checksum when the instruction are changed. Pages 0xA000 through 0xBE00 are available with the checksum located in page 0xBF00 as always).

As of 7/2002 HEXBUILD V16.2 contains the beginning of support for building a partial checksum table. The variable ***IEB_CHECKSUM_TABLE*** is used to control which portions of the checksum table are emitted. This capability allows the checksum table to be partially updated with only the entries affected by the overlay being written (this assumes that the overlay is contiguous and being loaded on page boundaries).

HEXBUILD V17.1 has additional features to support building IEB overlays. This version attempts to list the triggers that are in use and those that are pointed to the IEB overlay area. In order to make use of this feature the IEB load must be built accoring to the following guidleines.

1. Trigger 10 must be constructed such that it looks like all the other unused triggers in the load. HEXBUILD will compare the trigger-10 pattern with all the other trigger patterns in the first page of IEB memory. Those that match are assumed to be unused triggers that point to the idle trigger.

   This can be accomplished by placing an IEB *GOTO* instruction at the trigger-10 location (this is how unused triggers are handled).

2. Trigger FE must be pointed to the overlay wormhole table and the wormhole table must contain an IEB *GOTO* instruction that points to the idle trigger.

3. The wormhole table must be the last memory page in the IEB load (excluding the checksum table) and must NOT contain and IEB instruction other than the *GOTO*'s that make up the wormhole table.

The wormhole table must not contain any resident code as it will be overwritten by the overlay and to make the checksum work correctly, we need to write the entire page. Note, however, that it can contain overlay code (i.e. the checksum can be handled properly if the wormhole table has IEB instruction prior to the wormhole table within the same page).

Also, to make bookkeeping manageable, all of the overlay segments that are built (to go along with a given resident load) should be the same length. This eliminates the need to clear portions of IEB memory (which the V2.6 code does not handle).

Although the resident portion could contain code that is loaded into the overlay area, this is probably not a wise choice. The only way to recover the code located in the overlay area is to reload the resident IEB (Although this may be something to keep in mind for running a one-time sequence). It seems like the appropriate method is to save 4 or 5 overlays in the LSF, locating the wormhole table as shown in the table below at 0xB100. This allows the 14 IEB_LOAD records that are available to start loading at 0xB1C0 or 0xB1E0 and continue to near the end of page 0xBE00, leaving the checksum table untouched.

ALF records	IEB Records	Checksum and EOF	Memory Bytes	Wormhole Table	IEB Image
128	16	14	3360	0xB100	0xB200

Structure of the resident IEB:

```
              Org    IEB_Table          ; 0x8000
                     Defb   "IEB NAME"
                     Defb   " VXX.X  "
Trigger_10:   GOTO   trig_10            ; 0x8010
Trigger_12:   GOTO   trig_12
Trigger_14:   GOTO   trig_14
        .
        .
        .
Trigger_30:   GOTO   trig_30
Trigger_32:   GOTO   trig_10
        .
        .
        .
Trigger_C0:   GOTO   trig_C0
        .
        .
        .
Trigger_FE:   GOTO trig_FE
              Org    IEB_Memory         ; 0x8100-0xAEFF
trig_10:
              Trigger-10 code
trig_12:
              Trigger-12 code
trig_14:
              Trigger-14 code
trig_30:
              Trigger-30 code

              Org    IEB_Overlay
trig_C0:      GOTO   trig_10
        .
        .
        .
trig_FE:      GOTO   trig_10
```

Structure of the IEB overlay:

```
trig_10       equ    0x8010
              Org    IEB_Overlay
wormhole_table:
trig_C0:      GOTO   trig_C0
        .
        .
        .
trig_FE:      GOTO   trig_10

trig_C0:
```

6.15.1 Our current understanding of LSF

The LSF (Library Storage Facility) is an area of the SSR that is used to store ALF images from CDS that are not needed (confusing, yes?. SSR has a fixed area to hold ALF records, all of which is not needed, the un-needed space is being re-allocated and will hold IEB images) The space has been reallocated for use by the science instruments with 640 ALF records assigned to RPWS. CDS still views this area of the SSR as ALF records, so the IEB images are held in the data area of the ALF records, 16 words of IEB image in the 22 word ALF record.

Some important numbers:

- Each ALF record is capable of holding 16 words (32 bytes) of data (out of 22 words).
- RPWS command structure allows somewhat less than 256 bytes of command per RTI interval. The buffers used to move commands within the instrument have several bytes of overhead.
- A **73IEB_LOAD** command that contains 120 words is a tested (and workable) size for the IEB images. A size of 121 is also workable but 122 words is large enough to cause problems within RPWS. We simply choose 120 to make the address management a little easier for people (even number).
- 128 ALF records is a magic number internally for CDS. This is the maximum chunk that is retrieved from SSR. The minimum number is 8 (a single IEB image).
- 8 ALF records is also a magic number for CDS. This is the number of ALF's that are gathered to build a single **IEB_LOAD** command (i.e. the minimum).

We therefore structure our overlays to make close to the maximum use of the group of 8 ALF records (i.e. making the **73IEB_LOAD** command close to 128 words long where a data length of 120 gives an ALF length of 124). The 128 ALF group is used to establish the size of the overlay area.

73IEB_LOAD usage:

- Records 1-14 are 73IEB_LOAD images for IEB memory, the first 64 bytes being the worm-hole table near the end of page 0xB100.
- Record 15 is a portion of the checksum table that covers pages 0xB100 through 0xBEFF. This will be a short record that will have a large level of zero-fill.
- Record 16 is the End-of-file indicator and is only 2 words long. This record is mostly zero-fill.

From this list, we calculate that each overlay segment will hold 3360 bytes of which 3296 may contain useful IEB commands (64 bytes are the worm-hole table that contains *GOTO* instructions). This works out to just shy of 13 pages (of 256 bytes each) of IEB memory image along with 64 bytes of jump vectors. Locating the jump vectors at 0xB1C0 (in IEB memory) with IEB instructions beginning at 0xB200 appears to make just about maximum use of available resources (reducing the jump table to 16 vectors, starting a 0xB1E0 achieves peak use).

This allows out allocation to be partitioned into 5 distinct overlays.

6.15.2 September 2002: More Telecon's

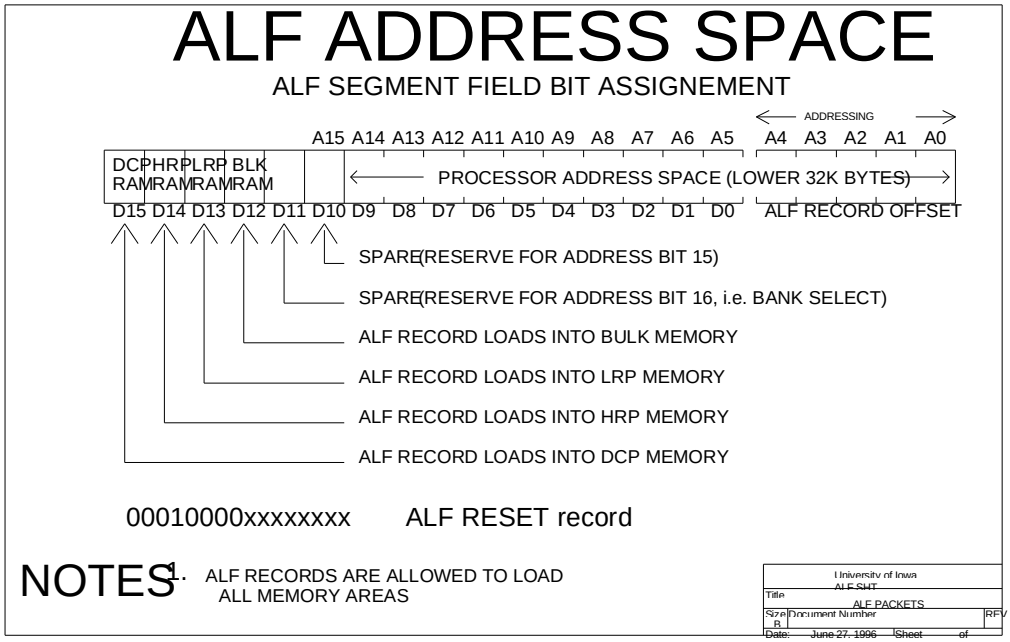
More information about LSF arrives. The LSF is accessed with a load command (6SSR_LOAD???) that specifies a start block and block count. The 8 record grouping is evident in this command (?) and we are not restricted to any reasonable number of IEB records that can be loaded with a singe instance of the command (RPWS only has 640 ALF records available in the LSF)

With this information in hand, it appears that we can manage complete loads as well as overlays using a single command to CDS (i.e. we won't require several commands to perform a full load). This also indicates that we can make use of the overlay facility with a finer granularity than 16 IEB records (128 ALF records). We can make use of virtually any size of overlay area.

7 Memory Download

Memory downloads make use of standard ***ALF*** formatted records as documented in ***CAS 3-291***. Memory downloads are accepted by the ***ROM*** software and ignored (treated as invalid commands) by any RAM software (*science* and *deploy*).

CAS 3-291 does not document the specific format of the segment field as this is instrument specific. The following chart describes the meaning of the bits in this field.



7.1 Error Recovery

Several simple problems may occur during the download process.

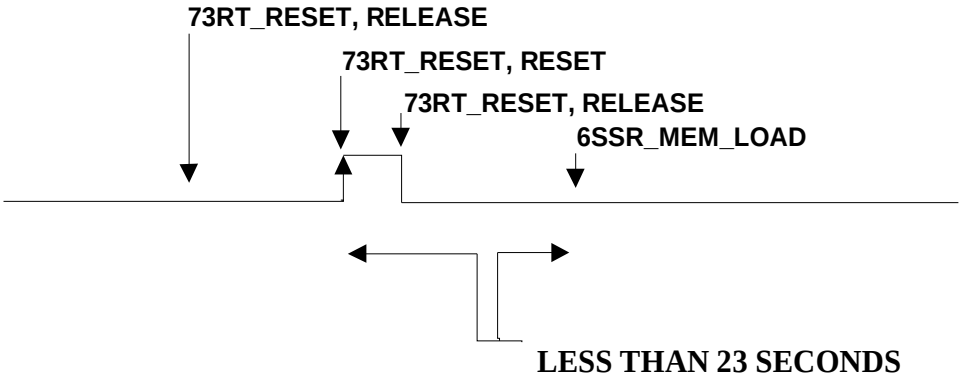
7.1.1 Internal download timeout

The instrument remains idle for only 23-24 seconds following the assertion of the reset signal (i.e. the reset occurs when the **73RT_RESET, RESET** command arrives, not when the reset line is returned to a released state as it is capacitor coupled). Following this idle period, the instrument will attempt to reload memory from an internally stored copy of the last load (a copy of the last flight software load is stored in bulk memory, where it remains until power is removed).

If the instrument is allowed to begin an internal reload, it may not correctly handle an external load as there are typically insufficient CPU cycles available to complete all the tasks that are required to correctly prepare for a software load (various counters are required to be initialized correctly so that we make sure no records have been lost or corrupted). This is not an issue following power-on as the bulk memory will not contain a valid load image. When reloading software this restriction/limitation must be kept in mind when specifying command timings.

Basically, the **73RT_RESET** commands should occur at 1 second intervals with the **6SSR_MEM_LOAD** command occurring 10 seconds later.

RESET and RELOAD TIMING



PREFERRED TIMING 10-15 SECONDS

7.1.2 SLEEP

All of the operating procedures mention that the instrument must not be in ***SLEEP*** when attempting a *memory download*. If the instrument is left in *SLEEP* when performing a download the housekeeping would typically indicate that a ***sequence error*** has occurred. This is caused by the reduced processor speed during sleep. The solution is to issue the ***73RT_SLEEP, ACTIVE*** command to exit sleep mode.

As a workaround for a stuck SLEEP bit, the download activity can still be performed if the reduced processor speed is taken into account. When operating in sleep the load will be accepted at a lowered rate. The suggested rate is 1 ALF group (of up to 5 records) per second. As an alternative, one ALF record every 4 RTI's is acceptable. These are **maximum** rates, slower rates will work although there is a lower bound imposed by the operation of the internal ALF reload mechanism (the minimum rate is one ALF transaction every twenty to thirty seconds).

7.1.3 ALF

The only word in the ALF record that is handled in a manner not documented in 3-291 is the segment field. This field, as documented in 3-291, contains upper address bits for the ALF record. This, obviously, is an instrument specific field with RPWS using the bottom 12 bits as upper address bits to form the 17 bit memory address in 8085 address space (the extra bit is bank select) and the upper bits used as processor select bits. Note that the segment field is defined differently for the ALF_EOF and ALF_SKIP record (i.e. this word becomes a counter). The BULK MEMORY select bit is propagated from the last occurring ALF record to allow internal downloads to be processed correctly.

The ROM performs integrity checks on the ALF record The ID word, sequence word, and checksum words must be correct in order for the download to proceed.

7.1.4 ALF_RESET

The ALF_RESET record is simply the first ALF record in the load with all three of the processor select bits cleared. The ROM recognizes the lack of processor select bits as an abnormal condition and keeps track of the sequence field to use for sequence verification of upcoming ALF records.

The ROM completely ignores the data words in this records so they are available to be used for storing version control information or anything else that is useful (see the *Version Control Information* section for details)

7.1.5 ALF_SKIP

During early ground testing, the ground system did not properly handle the ***73ALF_SKIP*** record. The purpose of this record type is to allow failed bits in the ***solid-state recorder*** to be bypassed. This failure is evidenced by a sequence error occurring at the ***73ALF_SKIP*** record.

The ROM performs integrity checks on the ALF_SKIP record, the ID word, sequence word, and checksum words must be correct. The skip count field is added to the internal sequence counter to allow for the bad blocks in the SSR that are about to be skipped.

73ALF_SKIP must **NOT** occur as the first record in the load!

The first record in any load must be the **ALF_RESET** record. As mentioned above, this is simply an atypical ALF record (with the processor bits cleared). It probably turns out that the occurrence of the ALF_SKIP as the first record may be inconsequential as long as the ALF_RESET record appears. This thoroughly confusing paragraph will be clarified following testing on the bench.

7.1.6 ALF_EOF(ALF_END)

This record marks the end of an ALF load and signals the ROM to proceed with next step in the boot process.

The ROM performs similar integrity checks on the ALF_EOF record, the ID word, sequence word, and checksum words must be correct. The record count field, however, will not abort the boot process if it contains an invalid value.

7.1.7 Wrong Instrument

If ALF records for an instrument other than RPWS are received by RPWS one might expect the peculiar RPWS reset record to be missing. If this is the case, the instrument will report a sequence error due to the sequence counter not being initialized correctly by the reset record.

It is more likely, however, that the segment field will contain zero in the upper bits resulting in something being processed and possibly loaded into RPWS memory. This should not cause a problem for the instrument, as first code fragment in the download that is executed contains a signature and checksum that are validated before the code is executed. If the signature/checksum are not validated, the processor will not pass control to this code fragment (this initial block of code is referred to as the ***boot block***).

7.1.8 Wrong SSR

This has occurred during ATLO testing and has resulted in no traffic being delivered to the instrument. No command counters were incremented when this has occurred. If CDS delivers improperly formed ALF records the normal ALF checking (command pattern, sequence, and checksum) is expected to reject bad data.

Rejected data should be reflected in the *invalid command count* field of the housekeeping display.

7.1.9 Status from DCP/HRP.

Normally the DCP will not return status information that may be viewed in the housekeeping. Due to the way the ROM formats housekeeping and the nominal sequence of events, the DCP is simply not in a position to provide status information.

The HRP is in a similar situation, although it is somewhat visible during internal reloads. The HRP will provide some status as the internal reload is in progress (status information is provided along with the internal download record).

It is possible, however, to trigger an error condition in which both the DCP and HRP will provide some status information. In the event that a software load (internal or external) fails, the LRP will begin to query HRP and DCP for status. As long as HRP provides internal LAF records, LRP will avoid reading status from the DCP. Only when the HRP has invalid ALF records will LRP poll both processors for status. If the instrument is provided with an incomplete download, the DCP will eventually be asked (by LRP) to provide a status report.

If the ALF_END record is suppressed or and invalid record appears in the ALF load, status from the DCP will eventually appear. This behavior may also be used as an indication that the *Bulk Memory* on the HRP has a bit error.

7.1.10 SSR Partition problems

During the early cruise phase of the mission, two separate software loads are maintained on the SSR. These two loads provide separate Science and Maintenance/Venus activities. Following Venus-1 encounter, these loads were consolidated into the single SSR that was powered (following launch, the separate loads were kept in separate SSR's).

With both loads being present in the same SSR, a problem can occur to cause the instrument to load incorrectly. Consider a command sent to the spacecraft to load the instrument from the first partition. Further consider that the first partition(s) contain errors that cause CDS to switch to the next partition. This scenario would have parts of memory loaded from one software set and parts of memory loaded from the other software set.

RPWS does NOT have a mechanism to uniquely identify a particular software load set. Although the 'spare' word in the ALF record could be used to contain version and/or function information, the ROM loader does not perform any verification/validation on this word. RPWS has no mechanism to determine if a load is being appropriately delivered (other than the checksum).

Considering the way the application software is collected for RPWS, it is conceivable that switching partitions would result in a successful load of the wrong software set. Examining the beginning of maintenance and science loads should reveal that the first 106 ALF records contain identical instructions (version information may differ, but the executable code should match).

If, however, the switch occurs later (in the code areas that make Science and Maintenance functions unique), completely unexpected results can occur. Since we have no internal power sequencing issues (i.e. the L/P is hardware interlocked to sequence power correctly), there would probably not be any internal problems that could not be resolved by cycling power, it is possible that the instrument would autonomously switch power in an order inappropriate for the S/C.

There are several possible solutions to this problem.

1. Eliminate the Maintenance load and consolidate the functionality into the existing Science load. A maintenance IEB is being built to address this solution. Ultimately the science software could be modified to make use of the maintenance bit (Science software ignores the maintenance bit currently).

2. Remap the Science and Maintenance loads such that they fit into a single partition. The first record of the loads in the 3rd. and 4th. Partitions could then be modified to cause the Maintenance load to be used.

7.2 Memory Download Procedure: *NO POWER*

Note that when we initially apply power, BULK MEMORY is, effectively, empty. The timing requirements are minimum times. There is no maximum time that mat elapse prior to step 4 below.

1. Apply Power to the instrument

    - **73PS_RPWS**

2. Allow the processors to complete housekeeping tasks, 15 seconds.

3. De-assert sleep using

    - **73RT_SLEEP, ACTIVE**

4. Send the ALF traffic

    - **6EXT_MEM_LOAD**

7.3 Memory Download Procedure: *ROM to RAM*

Note that when BULK MEMORY is empty there is no maximu time in effect prior to step 4 of the procedure.

1. Exit maintenance mode (if required) using

    - **73RT_MAINT, OFF**

2. Allow processors to remove L/P power if maintenance was active, 15 seconds.

3. De-assert sleep using

    - **73RT_SLEEP, ACTIVE**

4. Send the ALF traffic

    - **6EXT_MEM_LOAD**

7.4 Memory Download Procedure: *RAM to RAM*

When reloading software (i.e. instrument is already running software in RAM) a reset will cause the instrument to attempt to reload from internal ***BULK MEMORY***. If the bulk memory contains valid download records (from a previous load) the processor will complete the internal reload in several minutes and start operating from that (internal) load and ignore any subsequent download attempts.

In order to avoid the problem directly, the external download (from CDS) must begin prior to the internal download (from ***BULK MEMORY***). For most cases, 10 to 15 seconds should be sufficient to avoid problems. If the load present in BULK MEMORY is exceedingly short, the window may be very short and require that the instrument be power-cycled to successfully complete an external load.

The 10 to 15 second time period mentioned here allows sufficient tine for LRP, HRP, and DCP to complete initialization steps prior to the arrival of the ALF download. Waiting longer than about 20 seconds and the instrument will begin an internal download process. Avoiding the internal download process avoids any timing issues (and avoids the need to cycle instrument power when performing the ALF load).

1. De-assert sleep
    - **73RT_SLEEP, ACTIVE**
2. Reset the instrument
    - **73RT_RESET, RELEASE**
    - **73RT_RESET, RESET**
    - **73RT_RESET, RELEASE**
3. Allow the processors to complete housekeeping tasks, 5-10 seconds, sending the ALF traffic within 15 seconds,
    - **6EXT_MEM_LOAD**

Note also, that the critical timing period is from the rising edge of the reset pulse to the 1st. ALF record. The falling edge of the reset pulse is not particularly important (it must remain active for at least 1 RTI period, but may remain active for any length of time as the signal is AC coupled).

Once the reset occurs, the internal ALF load will begin after about 25 seconds.

7.5 Building the download image.

The ***HEXBUILD*** utility is used to gather the individual modules together and produce an image file (i.e. an **ALF** file) to be used to download the instrument. ***HEXBUILD*** takes, as input, a list of HEX files, a list of SSR bad blocks, a memory map from a previous run, and the ***Intel HEX files*** generated by the assembler/linker. The output file may be formatted for ATLO (in the form of hexadecimal-text encoded ALF records) or for SEQGEN (in the form of a 73ALF command string).

Note that ***HEXBUILD*** provides a mechanism to load the data words in the **ALF_RESET** record with up to 16 words of data. This is intended to provide a convenient method to implement a version control and checking mechanism. ***HEXBUILD*** has some features to allow the version control information to be easily inserted into the ALF files as they are constructed.

HEXBUILD also has the capability of processing bad block information for the SSR. When this capability is used, **ALF_SKIP** records will be generated as required and placed in the load file. The documentation for ***HEXBUILD*** (i.e. a users guide) is typically bound with this users guide.

7.6 Dump Analysis

In the event that it becomes necessary to perform a dump analysis on the contents of memory, some interesting information occurs here. As alluded to in the section titled *memory allocation scheme* performing a simple memory verification using a memory dump involves comparing static areas and ignoring dynamic areas.

As a starting point it is possible to load a matching image on the ground (using the engineering model) to use for analyzing a memory dump from the spacecraft. By filtering MRO packets and displaying them using the **DSP5** program a formatted memory dump may be obtained. Corresponding dumps from the E.M. and the spacecraft may then be passed through a compare utility on a workstation to find areas of memory that do not match.

Many areas will occur where the stack and data areas do not match up with each-other (it is nearly impossible to duplicate the spacecraft timings on the ground). These areas, that are not expected to match, can be identified using the load maps and by locating the process descriptors in the dump

Any code space that doesn't match should be indicative of a problem as we have made an effort to avoid self-modifying code. Although there are portions of the kernel and kernel utilities that are re-entrant, most of the instrument handlers are simple single threaded tasks.

The specific areas of memory that are static will change with each software version. It is beyond the scope of this manual to provide low-level details for a specific load. Of particular interest, however, are the load maps generated when the software is assembled and linked. The load map will reveal the locations that are static and can be expected to be identical on the bench and the spacecraft.

Another item that is worthy of close scrutiny in the memory dump is the *process descriptor* and stack area. The stack area should be located in memory immediately following the process descriptor. The software provided at launch follows this convention and there should be some area of zero-filled memory between the process descriptor and the bottom of the stack. When inspecting the stack, one must keep in mind that many of the modules keep scratch variables immediately following the process descriptor (in order to make them easy to access with a 73MEM_TWEAK/73MRO) with the stack being located a little higher in memory. Also, only the LRP makes use of the NMI interrupt, requiring an additional 10 bytes of stack be reserved on all processes to allow for the 2 level interrupt scheme (although in most cases the NMI will only occur when the idle process is current).

7.6.1 Dump Analysis Tools

Several tools may be used to assist in a dump analysis. It is expected that a *known good dump* may be obtained from the engineering model or the flight spare for use in a comparative analysis.

Memory dumps are available is several forms. MRO records may be routed to the science telemetry and these will probably be the most useful if the RAM software is loaded and functioning. The ROM delivers a similar size dump block for the LRP with each housekeeping record.

The tools on the GSE system may be used to format the memory dump into a human readable (i.e. text) form that can be compared with a reference dump using the UNIX ***diff*** utility. Although there is a large area expected to be the same, stack and variable areas are present throughout the code area. Areas of data memory would not be expected to match.

0Processor memory dump command sequence

```
00:00   73MEM_TWEK,  LRP, WORD, 0x60, 0x00, LOCK
00:05   73MEM_TWEAK, DCP, BYTE, 0X14, 0X40, TWEK
00:10   73MEM_TWEAK, HRP, BYTE, 0X14, 0X40, TWEK
00:15   73MEM_TWEAK, LRP, BYTE, 0X14, 0X40, TWEK
00:20   73MRO, DCP, TLM, 0000, 7FFF
00:25   73MRO, HRP, TLM, 0000, 7FFF
00:30   73MRO, LRP, TLM, 0000, 7FFF
07:10   73MRO, LRP, TLM, 8000, 83FF
07:30   73MEM_TWEK,  LRP, WORD, 0x60, 0xFF, LOCK
```

1. GSE Tools
    - hexbuild

      More recent versions of **HEXBUILD** produce a memory dump as part of the ALF build process.

    - dsphk (see -dump/+dump)
    - dsp5 (see +mro)
2. UNIX tools
    - diff

7.7 Doing the seemingly impossible

Some fun things to keep in mind when you discover it will not (?).

7.7.1 BULK MEMORY LIMITATIONS

In the event that the ***Bulk Memory*** becomes over subscribed, there are some work-around that may temporarily help. The scenario in mind for this discussion is the dust detection algorithm being added and overflowing the ***Bulk Memory***, but this is applicable to similar events.

Consider that the dust detection code will be the last major function added to the flight software load. The memory available for implementing the algorithm is very limited. In at least the initial implementation, we can expect to swell the size of the science load to a size in excess of what can be stored in bulk memory. This, of course, means that the instrument will no longer be capable of recovering from an error.

Although not optimal, consider a configuration where the dust detection code is loaded into processor memory but not into bulk memory. As long as no software problems occur (either code errors or cosmic ray events), dust detection proceeds without a problem. If a reset occurs, the dust detection code will be scrubbed from memory and the processors will reload from ***Bulk Memory*** and restart without the dust detection.

To accomplish this, the dust detection routine (and any other associated routines) must allocate resources in such a way that the dust detection process is not needed. Once this is completed, the download may be built in such a way that the first part is loaded into *processor memory* but not into ***Bulk Memory***.

It is acceptable to place more than one ***00ALF_RESET*** record in a download.

73ALF, nn, 0x0000, 0, 0, ?

73ALF, nn+1, 0x8080, ?

73ALF, nn+2, 0x8081, ?

73ALF, nn+3, 0x8082, ?

73ALF, nn+4, 0x1000, 0, 0, ?

73ALF, nn+5, 0xF000, 0, 0, ?

73ALF, nn+6, 0xF001, 0, 0, ?

73ALF, nn+7, 0xF002, 0, 0, ?

etc.

Note that the first section of the load does not load into bulk memory (i.e. only into DCP memory). In the middle of the load another reset record appears (sequence nn+4) that is loaded into ***Bulk Memory*** followed by some code that is loaded into all 3 processors as well as ***Bulk Memory***.

7.7.2 Stuck SLEEP discrete

This is discussed earlier in this chapter.

7.8 SSR Strategy

The SSR space allocated to hold the RPWS software load is divided into 8 areas. They are broken down in the following subsections.

7.8.1 SSR A / SSR B

There are 2 SSR devices on the spacecraft. For most of the cruise portion of the mission it is expected that only one of the two recorders will be powered. This, of course, eliminates half of the partitions as candidates for differing operating code.

All partitions of **SSR A** are expected to be identical to those in **SSR B**.

7.8.2 Primary / Secondary

The primary and secondary partitions are expected to contain identical loads to enable error recovery. The S/C may then switch between these partitions when an error is encountered.

As with the two SSR's, the **primary** and **secondary** partitions are expected to be identical to accommodate error recovery.

7.8.3 Default / Non Default

During the early portions of the mission, RPWS will make use of two separate software sets. The science software will be located in the default partition while the non-default partition will contain antenna deploy software during launch. The deploy software load will be replaced with a maintenance load following a successful deploy operation.

Note that antenna deploy and science operations are effectively performed concurrently (i.e. switching from deploy to science within minutes). This indicates that both software sets must be resident on the SSR in order to perform the antenna deploy and checkout activity. The equivalency bit, therefore, must be set to allow the default and non-default partition to contain different loads.

Following early cruise the requirement for the maintenance load to appear in the non-default partition will be eliminated (maintenance operations on the L/P are performed when the S/C is close to the Sun). At this point in the mission the non-default partition will be used to support upgrades to the science software load.

7.8.4 RPWS SSR Allocation

RPWS has sufficient allocation, assuming a limited number of SSR bad blocks, to store both science and deploy or science and maintenance within the instrument allocation on the SSR. For the case of *Science + Maintenance* there is a 10% buffer to accommodate SSR bad blocks.

Blocks V2.2	Blocks V2.3	Blocks V2.4	Blocks V2.5	Description
2458			2458	RPWS SSR Allocation
1489			1489	RPWS BULK Memory Capacity
1439	1441	1461	1464	Science Load
793	n/a	n/a		Maintenance Load
456	n/a	n/a		Deploy Load
2232	n/a	n/a		Science + Maintenance
1892	n/a	n/a		Science + Deploy
n/a	n/a	1716	1630	Science and Baseline IEB

7.8.5 An update strategy

Consider the problems that may arise from using new operating software in the instrument. There may be inadequate data available to test the software (particularly true of dust impact detection) or there may be important observations in the near term.

If a new software load is processed without the *BULK memory* bit set, it will not be loaded into BULK memory and will not be available should a reset occur. Also keeping in mind that the LRP allows a window following a reset before BULK memory is examined for ALF records:

```
00:00    73RT_RESET, RELEASE
00:05    73RT_RESET, RESET
00:10    73RT_RESET, RELEASE
00:15    6EXT_MEM_LOAD, CONNECTED, DEFAULT, RPWS
01:00    73RT_RESET, RELEASE
01:05    73RT_RESET, RESET
01:10    73RT_RESET, RELEASE
01:15    6EXT_MEM_LOAD, CONNECTED,
                              NON_DEFAULT, RPWS
```

The result of this operation being that the old code is placed into BULK memory in case of a watch dog timer trip, and the new code is running in processor memory.

7.8.6 A dual software strategy

Consider the following as a means to allow both RPWS software loads to exist in the SSR at the same time. This scheme assumes that altering the RPWS start address (i.e. the 7067 address) is difficult or impossible and that altering a single ALF record is possible. Note in these examples that the reset record is shown as a separate command for clarity (the **73ALF_RESET** is a special case of the **73ALF** command). The choice of the starting record number for the 2nd. load is arbitrary and simply makes the calculations simpler. This spacing also leaves room to reallocate blocks in the first load in the event that bad bits appear in the SSR (in both cases that follow, the starting record number for the second load has been selected to allow approximately 175 spare ALF records following the science load).

It is also important to keep in mind that the **73ALF_RESET** and **73ALF_SKIP** records that appear before the 2nd. load are vital . These records must appear in order for the block count in the **73ALF_END** record to be totaled correctly.

The required patch for these load addresses is shown in both the 3-291 format and in a hexadecimal format.

- **73ALF_RESET commands**

  The reference to a unique command, **73ALF_RESET**, may be somewhat confusing to everyone outside of the RPWS group. This is simply a simple way to describe the first ALF record the RPWS instrument must receive. Although this record is coded as a typical **73ALF** command, it is unique in that the 2nd. argument will be either 0 or 4096 (examining the Deploy Software will reveal the case where this word is zero). This indicates to the instrument that this record is the first record of a download and that the instrument should begin ALF verification procedures (i.e. this is the method used to tell the instrument about the starting sequence number). All other **73ALF** records will have additional bits set in this field.

Science

```
73ALF,7067,4096,1111,0,11,0,230,230,230,230,
    230,230,230,230,230,230,230,230

73ALF,7967,4096,1111,0,11,0,230,230,230,230,
    230,230,230,230,230,230,230,230
```

Maintenance

```
73ALF,7067,4096,2222,0,2200,0,230,230,230,230,
    230,230,230,230,230,230,230,230

73ALF,8667,4096,2222,0,2200,0,230,230,230,230,
    230,230,230,230,230,230,230,230
```

Build loads in one of the following manners

- **Science Load at the beginning of the RPWS allocation**

73ALF_RESET, 7067, xx, xx, xx

73ALF_SKIP, 7068, 1

73ALF, 7070, xx, xx, xx (science)

73ALF, 7071, xx, xx, xx

etc.

73ALF, 8506, xx, xx, xx

73ALF_END, 8507, 1440

73ALF_RESET, 8667, xx, xx, xx

73ALF_SKIP, 8668, 1

73ALF, 8670, xx, xx, xx (maintenance)

73ALF, 8671, xx, xx, xx

etc.

73ALF, 9460, xx, xx, xx

73ALF_END, 9461, 794

By patching record 7068 with the following information, the entire science load will be treated as if it were bad blocks by CDS and the maintenance software will be loaded. Changes indicated with bold text (i.e. only 2 words are different).

73ALF_SKIP, 7068, 1601

2500 1B9C **0641** 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 **80EB** 7146

- **Maintenance Load at the beginning of the RPWS allocation**

73ALF_RESET, 7067, xx, xx, xx
73ALF_SKIP, 7068, 1
73ALF, 7070, xx, xx, xx (maintenance)
73ALF, 7071, xx, xx, xx
etc.
73ALF, 7860, xx, xx, xx
73ALF_END, 7861, 794

73ALF_RESET, 7917, xx, xx, xx
73ALF_SKIP, 7918, 1
73ALF, 7920, xx, xx, xx (science)
73ALF, 7971, xx, xx, xx
etc.
73ALF, 9456, xx, xx, xx
73ALF_END, 9357, 1440

By patching record 7068 with the following information, the entire maintenance load will be treated as if it were bad blocks by CDS and the science software will be loaded. Changes indicated with bold text (i.e. only 2 words are different).

73ALF_SKIP, 7068, 851

2500 1B9C **0353** 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 **7DFD** 7146

7.8.7 Unworkable strategies

There are several methods that may be suggested that present problems within the instrument. This section will try to keep track of some suggestions that have come up that will not work satisfactorily and discuss the problems with them.

There are two crucial items to keep in mind about the RPWS instrument when working with downloads. As discussed in other sections of this document, the **BULK MEMORY** consists of 64K bytes of memory and limits the overall size of the ALF traffic that may be presented to RPWS when an internal download is required. The second item, a requirement that memory must contain zero in order for a download to work correctly as there are no download records that contain all zero.

7.8.7.1 Removal of the **ALF_END** record between loads.

This conflicts with both issues mentioned above.

The idea behind this method is to alter/remove the **73ALF_END** record that lies between two loads in order to allow the second load to overwrite the first. The **73ALF_END** would be changed to **73ALF** or **73ALF_SKIP**.

This would definitely exceed the capacity of **BULK MEMORY** eliminating the possibility of a warm restart if a software failure were to occur. Although loading one load on top of the other may work with one version of software, any updates could potentially alter areas of memory that are expected to contain zero causing the load to fail or run erratically.

7.8.8 Additional Thoughts

It may be possible to perform some additional compression on the combined science and maintenance loads. Both science and maintenance loads make use of common code in the kernel. It should be possible to provide a single copy of the code that is common and reduce SSR requirements by about 100 ALF records.

The common code used by science and maintenance differs in that the maintenance code has, in addition to all the common components used by the science load, a memory storage manager called *ramdisk*. As long as *ramdisk* is placed at the end of the common code, it should be possible to select a slightly different point to place the skip record.

7.9 Software Build Procedure

This section will attempt to describe the procedure and tools used to build the RPWS software loads. This applies, with minor changes, to both operating software and to IEB loads.

Keeping in mind that the operating software is very modular, in nature, and that the binding scheme is unique to the instrument, and is not supported by commercial software. Also of interest is the format of the download file required by the S/C and ground system, being specific to the CASSINI spacecraft, is not supported by commercial software.

Commercial software is used in the first step of building both the operating software and IEB loads.

Software change procedure.

1. Requirements evaluation
   - 0Scientist requests change or improvement
   - 1Software deficiency (i.e. bug) discovered
   - 2Planned upgrade
2. Implementation & Testing
   - 0Coding and bench level test
   - 1Engineering model test (stimulus)
3. ITL Testing
   - 0Flight spare with CDS
4. Delivery
   - 0PSL delivery
   - 1SRCR
   - 2Flight testing

7.9.1 Requirements Evaluation

RPWS team must, as a first step, evaluate the need to actually expend the effort to implement a change to the flight software.

When a bug or lack of function is deemed to have significant impact on the science effort, the team will proceed with a fix.

7.9.2 Implementation and Testing

Once the RPWS team has made the decision to implement a change or bugfix, the software team must implement the change. This portion of the effort will be detailed in this section.

As the RPWS instrument was designed and built, the software was built using an 8085 assembler from *Avocet Systems*. This assembler will be used for the foreseeable future although any assembler that is capable of producing an *Intel Hex Format* file is permissible.

The Avocet assembler/linker is used to produce individual modules (or handlers) that control the instrument. Each module consists of one or more tasks (process) that control a particular subsystem within the instrument. Memory allocation is performed manually, the effort to manage the limited address space of the 8085 is much less than that of building and debugging a utility for this purpose. Memory conflicts must be manually resolved (although the tools do help in detecting conflicts).

Due to the presence of multiple processors in the instrument, some means must be provided to generate a download to the instrument that loads the appropriate memory. This is handled by the utility *HEXBUILD*. This utility is used to perform all of the reformatting and marking in order to translate between the *Intel Hex* and the *ALF* format required by JPL (did that make any sense?)

The *HEXBUILD* step also performs several error checks. During this step several checks/operations are performed.

0 collect or gather together all of the modules required to load one/all of the processors. Part of the load (i.e. the kernel) is used by all processors with a single copy present in the load to conserve SSR/BULK space.

1 validate checksum information on incoming files

2 insure memory locations are allocated once (i.e. by a single module/task). This involves tallying the number of times all memory locations are referenced in the incoming files, the memory referenced in a previous build, and saving a memory map.

3 insure that only the correct memory is used (i.e. bottom 16K is dedicated to code use). This step is intended to prevent code from overflowing into uninitialized buffer space.

4 generate ALF_RESET records. This is a special form of the ALF record that is required at the beginning of a load in order to initialize the error flags and sequence counter within the instrument.

5 generate ALF style checksums.

6 generate ALF_SKIP records. This being possible only if there is knowledge of the bad blocks that must be ignored.

Some degree of manual memory management is performed to guard against improper memory allocation within the instrument. Although HEXBUILD performs several checks, the software can be architected in such a way that HEXBUILD has no way to detect a problem. As an example, the WBR/WFR handler and the L/P handler place portions of their code into the upper memory area of the HRP. This code movement is performed by the 8085 as part of process initialization. HEXBUILD has no mechanism to discover what are of memory WBR/WFR and L/P intend to use, so a small change in either application may result in a memory conflict that can only be detected by inspecting the load maps or by bench testing.

Once the HEXBUILD step has been performed, the aggregate load may be delivered to the instrument located at Iowa for testing (Iowa makes use of the ATLO and 3-291 format files). This process is iterative and may involve low level hardware testing (i.e. scope, logic analyzer, in-circuit emulators, etc.)

The steps used to produce an IEB load are similar with one extra step available to perform the translation from *command mnemonics* to *hexadecimal command patterns (*see the documentation/section on the command parser). The assembler is used to build the command and control tables that form an *IEB load*. The entire *IEB load* is usually produced as a single *Intel Hex File* and presented to *HEXBUILD* for translation to the required format (usually a file with several(many) 73IEB_LOAD commands). *HEXBUILD* has provisions to accommodate some shortcomings in the ground system (limitations in the way that the 73IEB_LOAD command is parsed).

7.9.3 ITL testing

If changes are significant, the team may request testing on the S/C simulator in the Integration and Test Lab at JPL. This step would be required when a change to telemetry formats or commanding is implemented. This testing would require delivery of software to JPL.

7.9.4 Delivery

When the team is satisfied with the changes, the software is delivered to the PSL and a SRCR is scheduled.

7.10 Download Internals

The manner in which a download is processed is discussed in this section.

One key mechanism in the downloading process is the **BOOT BLOCK**. This code fragment is used to make the transition from ROM to RAM based execution. Another key function performed by the boot block is to allow/prevent a warm restart action to be initiated from within the RAM based system.

Another key point is the memory architecture. The 8085 begins execution at location zero and requires memory starting at location zero to store interrupt vectors. If ROM is permanently addressed at location zero, the interrupt vectors require the use of and extra level of indirection if they need to be changed. Alternately, the ROM may be located in another area of memory and a means provided to provide a jump instruction to transfer control to the ROM. In the RPWS processors, the ROM may be disabled, exposing the underlying RAM. The RAM that is located underneath the ROM is disabled only during read access. The implication of this being that the contents of low RAM may be altered (i.e. loaded) when the ROM is enabled, simplifying the job of downloading the application software.

7.10.1 Memory architecture.

The memory architecture is similar on all processors. The bottom half of the address space (i.e. 32K bytes) contains RAM that is not banked (i.e. 32K bytes of RAM occupy 32K bytes of address space). As mentioned earlier, the lowest portion of memory contains ROM that is enabled following a processor reset. Sufficient ROM is present in the processors to support basic BIU activities, a minimal maintenance capability (i.e. bias the L/P sphere to 10 volts), and a memory download capability.

The upper half of the memory address space (32K bytes of address space) contains between 32K bytes and 64K bytes of memory in two banks. The "SOD" line from the 8085 is used as a bank select bit to enable one of two memory banks. The actual complement of upper memory differs on the three processors (LRP banks 16K bytes of address space , HRP banks 32K bytes of address space, and the DCP has no banked memory).

7.10.2 Rom activities

Following processor reset, the ROM is mapped into memory. All memory reads in the lower address range (i.e. the bottom 4K of memory) are taken from the ROM.

The hardware reset signal is also routed to most of the gates that deliver signals to points located off of the CPU boards. The reset signal forces most of the hardware into a quiet, low power state.

7.10.2.1 Hardware initialization.

The first activity following a reset is to program all of the hardware into a minimum power state. This involves writing to the control registers of all of the peripheral chips in order to bring all output lines to a zero voltage level. Clocks are suppressed to the MFR and the A/D converters on the HRP are powered down.

7.10.2.2 Memory scrub.

The next step is to clear memory from address 0x0000 through address 0x7FFF to zero. This is intended to accomplish two goals, first, the contents of memory may always be assumed to contain zero, resulting in some savings in the number of ALF records required to load memory. Second, it is hoped that any software problems that cause the processor to execute randomly through memory may be easier to debug when memory contains the same pattern.

The memory scrub operation specifically avoids changing the contents of upper memory so that upper memory can be used to save data through a reset.

7.10.2.3 Special Initialization

Setup patterns are written to the SCC chip that is used during bench level testing at this point. Although the flight hardware complement does not include the SCC chip, the initialization does not cause any ill effects on the flight hardware.

7.10.2.4 Waiting for commands.

The ROM software then enters a polling loop waiting for commands from the S/C to arrive. S/C commands may be in the form of ALF records (i.e. the memory download) or in the form of a BIU discrete command.

Two BIU discrete commands are recognized; a SLEEP/ACTIVE bit and a MAINTENANCE bit.

In addition to ALF commands that may arrive from the S/C, the **BULK MEMORY** on the HRP may contain a memory download image. The LRP also requests status records from the DCP and HRP. The HRP status record contains a record from **BULK MEMORY** that will be processed as though it were an ALF command from the S/C.

7.10.3 SLEEP recognition

BIU discrete bit 1 is used to trigger SLEEP mode where the instrument clocks on the HRP and DCP run at a reduced rate. The power up state of this bit, a zero, is used to indicate that the processors are expected to operate in a low power mode (i.e. at the reduced clock rate).

7.10.4 Maintenance recognition

Assuming that the sleep bit indicates operation in high power state is allowed (i.e. BIU discrete 1 is a one), asserting the maintenance bit will eventually cause the L/P power to be applied and the sphere will be biased to 10 volts.

This activity involves little intervention by the processors, so once the instrument is acknowledging entering maintenance, it will correctly process an ALF download.

7.10.5 ALF recognition

As each command block is received, either from S/C or **BULK MEMORY**, it is passed through several steps before being stored in memory. The command pattern, sequence field, and checksums must be valid before the record is loaded into memory.

Any non-ALF commands are, effectively, ignored. They will be treated as ALF records with invalid ALD ID field and ignored.

7.10.6 ALF_EOF recognition

The last record in the load is the ALF_EOF record, recognized by it's unique command pattern. This record triggers the transfer of control activities that pass control from the ROM code to the newly downloaded image. As with the ALF records, the command pattern, sequence field, and checksums must be valid for the record to be processed.

7.10.7 Reserved memory areas.

The area of memory from 0x7C00 through 0x7EFF is used by the ROM for workspace (stack, variables, command buffers, communications buffers, etc.). Access to the workspace by an ALF record is blocked by the ALF processing routines. Attempted writes to the workspace are **ignored without producing an error**.

It is expected that some support utility, used to produce the ALF load, contains checks to insure that this area of memory is not used.

7.10.8 Boot block pre-processing

Once the ALF_EOF record is accepted (acceptance indicating the memory was successfully and completely loaded), some final checks are performed prior to passing control to the boot block.

Since the general ALF formatting requirements are similar for all instruments on the spacecraft, the **BOOT BLOCK** is the last integrity check used to verify that the load is, indeed, intended for the RPWS instrument. The **BOOT BLOCK** which is part of the ALF load is located at address 0x7F00. It contains a pair of jump instructions followed by the string "BOOT BLOCK: ", and the instructions necessary to disable and re-enable the ROM.

Two integrity checks are performed on the **BOOT BLOCK**. The ROM checks that the string appears at location 0x7F06. If this check is successful, the ROM then proceeds to calculate a checksum on the memory from 0x7F00 through 0x7F3F. If the checksum is valid, the ROM will complement the checksum at location 0x07F3E and then passes control to location 0x7F00.

7.10.9 Boot block processing

The boot block contains two distinct/complementary functions.

The primary function is to disable the ROM and pass control to the newly loaded image. The starting address is typically zero, but the actual start address is entirely up to the **BOOT BLOCK**.

Currently the boot block initializes the 8155 such that Ports A and C are set for output and Port B is set for input with all output bits presenting a zero voltage level to the outside world. Port A bit 8 is then brought high to disable the ROM. At this point the ROM is no longer accessible ant the entire memory area consists of RAM. Finally the first 64 bytes of memory are copied to location 0x7F40 for bench level debugging.

The secondary function is to re-enable the ROM and return control to the ROM. Again the typical address used to return control to the ROM is zero, but this is entirely up to the **BOOT BLOCK**. In the case of the version 2.x code that was on the S/C at launch, there is no capability to perform a warm restart. This version of the software expects that **BULK MEMORY** will be used to handle a software crash.

7.10.10 Kernel initialization

Assuming all integrity checks have been successfully completed, the newly loaded kernel will receive control at location zero and finish any hardware initialization and proceed with the downloaded task.

7.11 Radiation and Hardware related issues

We may expect to see, at some time in the future, a problem with the processors (related to a part failure or radiation induced failure) that might be remedied by modifications to the software.

Failures in the receivers and their associated analog front-ends can be easily dealt with by simply reworking the IEB loads to avoid collecting data from a failed receiver.

Failed 8237 can be handled by changing the IPC driver to avoid use of this resource (i.e. the ROM does NOT depend on the 8237 to download the processors). WBR, WFR, and L/P require the use of the 8237 and would be impaired by loss of the device on the HRP.

CPU failures would probably be difficult (at best) to work around.

7.11.1 Resource requirements

In order for flight software to be downloaded, certain parts of the hardware must be functioning correctly. The 1553 interface and the LRP (CPU, ROM and a small area of RAM from 0x7C00 through 0x7FFF) must be operational. Neither DCP nor HRP are required for obtaining housekeeping or downloading flight software.

A failed DCP or HRP will impact the flight software. Reworking the IPC driver and a flight software rebuild should provide a means to disable traffic to a filed processor.

Failed memory one or more processors may be handled by mapping the ALF load in such a way as to avoid loading failed memory (this may impact resources, such as free space, on the affected processor)

8 RPWS Housekeeping

The housekeeping record contains a fixed area and a context dependent area. The fixed area is formatted the same by all software loads and may be decoded by common software. The context dependent area changes with each software load and requires context sensitive decoding.

The following tables describe the contents of the housekeeping packet in terms of 8085 address space.

This is, in most cases, backwards when viewed in a raw data file. Data on the RPWS GSE systems has been switched to a natural order for those accustomed to working with the 8085.

Note that the housekeeping module used on all of the software loads (i.e. all the operating software that is downloaded into the instrument) is identical (i.e. there is not a housekeeping process that is unique for the deploy software). Also the housekeeping software that operates out of ROM is very structured in the way it operates and this causes some unique signatures to appear in the ROM housekeeping.

The ROM initializes the housekeeping area of BIU memory using a scrubbing operation (clear to zero). The top of the housekeeping loop then waits for CDS to transfer a housekeeping packet before proceeding. This results in the first ROM housekeeping packet being predominantly zero.

The RAM housekeeping operates in a similar manner but waits for the housekeeping pickup to occur at the end of the loop. This results, for most cases, in housekeeping appearing to operate as one might expect. It is possible, however, for events to occur such that the housekeeping loop does not complete before the first housekeeping packet is collected. In other words, it is considered normal for the first packet of housekeeping following a software change to be predominantly zero.

Fields that appear as zero in this first housekeeping packet may be ignored.

8.1 RAM Housekeeping at reset

When the processor is reset, the housekeeping process requires some time following the first packet pickup (by the spacecraft) to cycle through all of the analog and digital values that appear in the housekeeping packet. The first houskeeping packet that is picked up by the spacecraft will contain predominantly zero data (this usually includes the time tag on the packet).

Also note that the instrument will tolerate elevate pickup rates on the housekeeping channel. In bench test we typically use a pickup rate of 192 bits/second which results in a housekeeping packet being picked up every 8 seconds. Pickup rates of 1536 bits/second (one packet per second) have been used on the bench. Note, however, that the primary housekeeping process requires roughly 6 seconds to cycle through all of the analog and digital samples (i.e. most of the housekeeping information updates at this rate, even if records are collected faster than this). Some of the information in the housekeeping record is updated as CDS picks up the packet (such as packet sequence and time tags). Micro packets are moved into the housekeeping packet as they arrive from various sources within the instrument (various micro-packet generators, within the instrument, limit the aggregate bit rates to well below 16 bits/second).

8.2 Channelized Housekeeping Tables

Housekeeping data is decommutated by the telemetry system and made available for display. The channels are assigned an arbitrary identification number.

8.2.1 Command/Power

Command and Power		
Channel ID	**Description**	**DMD Title**
S-1400	Antenna Bracket Temperature	
S-1401	*	
S-1402	Good Commands ??	
S-1403	Invalid Commands ??	
S-1404	Command Byte Count	CMD_Bytes
S-1405	BIU Discrete Command	Disc_Cmd
S-1406	BIU Discrete Status	Disc_Stat
S-1407	Power Status	SS_Power

* Channel S-1401 was originally documented with JPL as containing the lower 16 bits of time (i.e. 13 bits of seconds and 3 bits of RTI) but this information is redundant as the housekeeping record already contains a complete time field. None of the flight software ever loaded this 16 bit word of housekeeping with time information.

This field contains the IEB Status Word presented by the IEB handler (this is described shortly).

8.2.2 LRP Analog Mux

LRP Analog Multiplexer		
Channel ID	**Description**	**DMD Title**
S-1408	MFR2 Analog	MFR2
S-1409	MFR1 Analog	MFR1
S-1410	MFR3 Analog	MFR3
S-1411	HFR Analog 0	HFR_An0
S-1412	Antenna Motor Current	Ant_Mot_I
S-1413	+X Motor Temperature	+X_Temp
S-1414	Z Motor Temperature	-Z_Temp
S-1415	-X Motor Temperature	-X_Temp
S-1416	Search Coil Temperature	SC_Temp
S-1417	-X Position	-X_Pos
S-1418	+X Position	+X_Pos
S-1419	Z Position	-Z_Pos

8.2.3 HFR Analog Mux

HFR Analog Multiplexer		
Channel ID	**Description**	**DMD Title**
S-1420	ME2 Current	HFR_PS_ME2
S-1421	HFR Current	HFR_PS_HFR
S-1422	ME1 Current	HFR_PS_ME1
S-1423	L/P Current	HFR_PS_LP
S-1424	HFR +6 Volt Monitor	HFR_+6_V
S-1425	HFR +5 Volt Monitor	HFR_+5_V
S-1426	MFR +6 Volt Monitor	MFR_+6_V
S-1427	MFR +12 Volt Monitor	MFR_+12_V
S-1428	L/P +45 Volt Monitor	L/P_+45_V
S-1429	MFR +5 Volt Monitor	MFR_+5_V
S-1430	LRP +12 Volt Monitor	LRP_+12_V
S-1431	LRP +5 Volt Monitor	LRP_+5_V
S-1432	MFR -12 Volt Monitor	MFR_-12_V
S-1433	HFR -6 Volt Monitor	HFR_-6_V
S-1434	L/P -45 Volt Monitor	L/P_-45_V
S-1435	MFR -6 Volt Monitor	MFR_-6_V

8.2.4 Langmuir Probe

Langmuir Probe		
Channel ID	**Description**	**DMD Title**
S-1436	L/P Bias Voltage	LP_BIAS

8.2.5 S/C Systems

S/C Systems		
Channel ID	**Description**	**DMD Title**
E-2303	Instrument Temperature	BAY 04 T
E-0553	Instrument Load Current	RPWS_Elec_LC
F-0815	SSPS state	RPWS_ELEC_a
F-0816	SSPS state	RPWS_ELEC_b
F-0817	SSPS status	RPWS_ELEC_sw
E-1553	Instrument Load Current	RPWS_Elec_LC
F-1815	SSPS state	RPWS_ELEC_a
F-1816	SSPS state	RPWS_ELEC_b
F-1817	SSPS status	RPWS_ELEC_sw
E-0748	Power BUS voltage	30VBus_HF_V
E-0749	Power BUS Voltage	30VBus_LF_V
H-0064	Telemetry bit rate	Rate

8.2.6 Instrument Currents

Instrument Load Current E-0553 E-1053		
DN Data Number	**Instrument operating state**	**Current**
6	SLEEP / RAM	
7	ROM	
10	RAM Maintenance	148mA
36	Full Power low pickup rate	
	Full Power high pickup rate	

8.2.7 Instrument Temperature

Instrument Temperature Bay 4 (E-2303)		
DN Data Number	**Instrument operating state**	**Temperature**
	NO Power	
	RAM/SLEEP	
	ROM	
150	RAM Maintenance	27.6
	Full Power low pickup rate	
	Full Power high pickup rate	

8.2.8Antenna Temperature

Antenna Bracket Temperature Instrument Power Supply Bay 4 (S-1400)		
DN Data Number	**Instrument operating state**	**Temperature**
	NO Power	
	RAM/SLEEP	
	ROM	
	RAM Maintenance	
	Full Power low pickup rate	
	Full Power high pickup rate	

8.2.9 Derived Channels

Derived Channels		
Channel Designation	**Source Channels**	**Description**
R-1750		MFR Alarm
	S-1407.012	Internal Power Status
	S-1420	ME2 Current (Analog)
	S-1426	MFR +6 Volt Monitor
	S-1427	MFR +12 Volt Monitor
	S-1429	MFR +5 Volt Monitor
	S-1432	MFR -12 Volt Monitor
R-1751		HFR Alarm
	S-1407.012	Internal Power Status
	S-1421	HFR Current
	S-1424	HFR +6 Volt Monitor
	S-1425	HFR +5 Volt Monitor
	S-1433	HFR -6 Volt Monitor
R-1752		LRP Alarm
	S-1407.012	Internal Power Status
	S-1422	ME1 Current (Digital Logic)
	S-1430	LRP +12 Volt Monitor
	S-1431	LRP +5 Volt Monitor
R-1753		Langmuir Probe Alarm
	S-1407.012	Internal Power Status
	S-1423	L/P Current
	S-1428	L/P +45 Volt Monitor
	S-1434	L/P -45 Volt Monitor
R-1757	S-1405.3	Antenna Enable EX-
R-1758	S-1405.4	Antenna Enable EZ
R-1759	S-1405.5	Antenna Enable EX+
R-1760	S=1406.4	Antenna Motors Enabled

Derived channels are used to drive alarms on the SOPC (at Iowa and at JPL) allowing notification when housekeeping indicates there is a condition that needs attention within the instrument.

R-2757, R-1758, R-1759 and R-1760 should never occur and indicate significant hardware problems in the instrument as there is no software currently on board the spacecraft that manipulate the antenna deploy hardware.

8.3 Alarm Limits

The alarm limits originally supplied during instrument delivery and integration were, at best, hastily chosen. Following several instrument activities, the following tables may be used to select alarm limits.

Alarm Limits Selection, February 2000					
Channel ID	**Description**	**Red Lower**	**Red Upper**	**Yellow Lower**	**Yellow Upper**
S-1420	ME2 Current Monitor	90.0	110.0	94.0	106.0
S-1421	HFR Current Monitor	175.0	220.0	180.0	215.0
S-1422	ME1 Current Monitor	180.0	350.0	200.0	330.0
S-1423	L/P Current Monitor	38.0	47.0	40.0	45.0
S-1424	HFR +6 Volt Monitor	5.6	6.1	5.7	6.0
S-1425	HFR +5 Volt Monitor	4.9	5.6	5.1	5.4
S-1426	MFR +6 Volt Monitor	5.65	6.45	5.8	6.3
S-1427	MFR +12 Volt Monitor	11.2	12.6	11.4	12.4
S-1428	L/P +45 Volt Monitor	45.0	54.0	47.0	52.0
S-1429	MFR +5 Volt Monitor	4.95	5.65	5.1	5.5
S-1430	LRP +5 Volt Monitor	4.8	5.4	4.9	5.2
S-1431	LRP +12 Volt Monitor	11.3	12.5	11.5	12.3
S-1432	MFR -12 Volt Monitor	-11.3	-12.2	-11.5	-12.0
S-1433	HFR -6 Volt Monitor	-5.6	-6.25	-5.7	-6.15
S-1434	L/P -45 Volt Monitor	-45.0	-55.0	-47.0	-53.0
S-1435	MFR -6 Volt Monitor	-5.6	-6.5	-5.8	-6.3

This table is base on data collected on 1999-230 through 1999-233 and 2000-039 through 2000-042.

ME01 current occasionally spikes and causes a yellow alarm. Alarm limits were purposefully set tight so that we would receive a red alarm in the event that voltages or currents go beyond previously observed values.

Langmuir probe voltage occasionally spikes and causes a yellow alarm.

8.4 BIU Discrete Command & Status

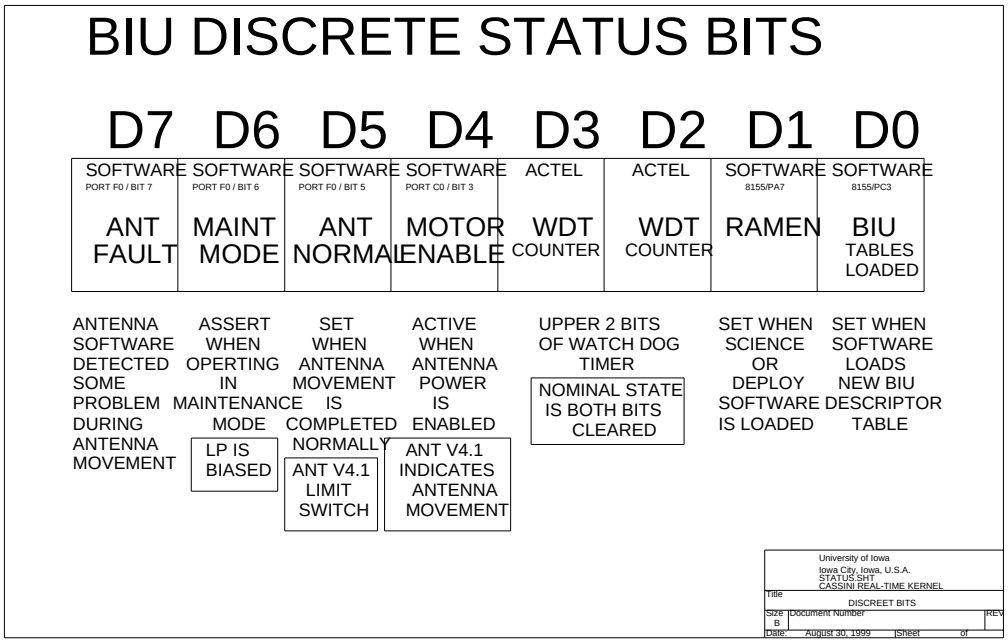
There are 8 command and 8 status bits managed by the BIU that do not appear in the housekeeping frame. In addition, the instrument does not have access to the internals of the BIU and cannot place any internal BIU status within the instrument housekeeping. These status bits are collected directly from the BIU by CDS and eventually delivered to the ground. This BIU ancillary data is typically available as *channeled data*.

The pattern **0x4B4B** usually indicates that RPWS has not powered (i.e. CDS has not been able to collect data from the instrument). Also note that when power is removed from the instrument, it seems that the last BIU discrete status words will be delivered to the ground without an indication that power has been removed from the instrument (this may be simply an issue with DMD, but DMD is the only visibility we currently have into the BIU status). The pattern **0x304** usually appears on a DMD page when the instrument is not powered. This is simply the last discrete data item retrieved from the instrument (with power applied). It is inappropriate to *believe* these values when the instrument is not powered.

Also, it is important that the C-0266 data item (and derived channels D-0280 through D-0294) be used to determine RPWS status. The items in C-0265 (and it associated derived channels D-0264 through D-0271) **not** be used to determine RPWS status. The C-0265 data reflects the last command key sent to subaddress 3 in the BIU and does **not** give an accurate indication of the state of the discrete command lines to RPWS.

RPWS Discrete Command and Status.

These two tables show the BIU discrete bits as seen by the LRP. These bytes are presented in housekeeping telemetry in bytes at offset 20 and 21.



BIU DISCRETE COMMAND BITS

D7	D6	D5	D4	D3	D2	D1	D0
MEMORY WRITE PROTECT DISABLE	SOFTWARE MAINT MODE	ANTENNA EX+	ANTENNA EZ	ANTENNA EX-	PROCESSOR (SLEEP/) RUN	PROCESSOR LRP WDT DISABLE	PROCESSOR LRP RESET
ASSERT TO DISABLE WRITE PROTECT CIRCUITS TO FUNCTION		ASSERT TO APPLY POWER TO EX+ ANTENNA LOGIC	ASSERT TO APPLY POWER TO EZ ANTENNA LOGIC	ASSERT TO APPLY POWER TO EX- ANTENNA LOGIC	ASSERT TO OPERATE IN NORMAL MODE (DEFAULT TO LOW POWER OPERATION)	ASSERT TO DISABLE THE WDT FUNCTION	RESET LINE ON 8085 CHIP

University of Iowa
Iowa City, Iowa, U.S.A.
DISCREET.SHT
CASSINI REAL-TIME KERNEL

Title: DISCRETE BITS

Size: B | Document Number: | REV:

Date: August 30, 1999 | Sheet of

RPWS Discrete Command and Status bits

These tables describe the BIU discrete status bits as shown in the DMD displays.

S Channel C-0266	Bit	Destination Status Source	Description Controlling Command
D-0280	D0 Command D-0	8085 Reset Port E0 / D0	Processor Reset Control 73RT_RESET
D-0281	D1 Command D-1	Actel Array Port E0 / D1	Processor Watch Dog Timer 73RT_WDT_CNTL
D-0282	D2 Command D-2	Software Port E0 / D2	Processor SLEEP command 73RT_SLEEP
D-0283	D3 Command D-3	Motor Control Port E0 / D3	Element EX- control 73RT_EX_M_CNTL
D-0284	D4 Command D-4	Motor Control Port E0 / D4	Element EZ control 73RT_EZ_P_CNTL
D-0285	D5 Command D-5	Motor Control Port E0 / D5	Element EX+ control 73RT_EX_P_CNTL
D-0286	D6 Command D-6	Software Port E0 / D6	Maintenance mode select 73RT_MAINT
D-0287	D7 Command D-7	Actel Array Port E0 / D7	Processor Write Protect 73MEM_WRT_PRT
D-0288	D8 Status D-0	8155/PC3	BIU Tables loaded
D-0289	D9 Status D-1	8155/PA7	RAM Enable
D-0290	D10, D11 Status D-2 Status D-3	ACTEL	WDT Counter Bit 0 WDT Counter bit 1
D-0291	D12 Status D-4	Port C0/D3	Motor Enable
D-0292	D13 Status D-5	Port F0/D5	Antenna Normal Venus Observation Normal (Success)
D-0293	D14 Status D-6	Port F0/D6	Maintenance Mode
D-0294	D15 Status D-7	Port F0/D7	Antenna Fault Venous Observation Fault (Failure)

The following status word reflects internal BIU status. These bits are used for internal BIU operations and are not affected by operations within the RPWS instrument (with the exception of the BCRTM Write Protection Violation Flag).

The lower 8 bits reflect the most recent command key delivered to the instrument *and should not be used to determine the current state of the BIU discrete command bits* (the previous table reflects command and status bits as seen and presented by the BIU).

Note that these field can cause some confusion as the 73RT commands manipulate a single bit at a time so the fields D-0264 through D-0271 will only reflect the last **bit** that was altered (if the bit was set to a zero, the lower 8 bits will all be zero).

S Channel C-0265	Bit	Description BIU Command Key
D-0264	D0	Processor Reset Control
D-0265	D1	Processor Watch Dog Timer disable
D-0266	D2	Processor SLEEP* command
D-0267	D3	Element EX- control enable
D-0268	D4	Element EZ control enable
D-0269	D5	Element EX+ control enable
D-0270	D6	Maintenance mode select
D-0271	D7	Processor Write Protect Disable
D-0272	D8	BCRTM Write Protect Disable
D-0273	D9	BIU Watch Dog Timer Disable
D-0274	D10	Set Write Protect Violation Flag
D-0275	D11	Set Watch Dog Timer Expiration Flag
D-0276	D12	BCRTM Write Protect Status
D-0277	D13	BCRTM Write Protect Violation Flag
D-0278	D14	BIU Watch Dog Timer Status

Bits D-0280 through D-0287 are controlled by CDS using the BIU discrete commands (i.e. **73RT**_).

8.4.1 D-0280, Processor Reset Control

AC coupled reset to LRP 8085 processor.

The rising edge of this signal causes the LRP to reset. Leaving the line asserted will NOT cause the LRP to hang.

Bit Status	Meaning
0	Reset line to LRP inactive
1	Reset line to LRP asserted (rising edge resets processor)

8.4.2 D-0281, Processor WDT Control

Asserting this line disabled the watch dog timer circuit on the LRP. In normal operations, the WDT may be enabled as the flight software correctly handles the timer.

Bit Status	Meaning
0	WDT enabled
1	WDT disabled

V2.5 flight software has a method to include DCP and HRP in the checking such that a failure of any one of the 3 processors will result in a trip.

8.4.3 D-0282, processor SLEEP command

This status bit reflects the state of the internal sleep control line in the instrument. This bit, when set to ***1*** indicates that LRP has commanded DCP/HRP to operate at a reduced clock rate. Note that this is opposite polarity of the BIU discrete command bit displayed in D-0266.

The reason D-0266 is negative-true (i.e. zero indicates sleep) is that the instrument can power up in a sleep state without any additional commanding by CDS. Internally, the logic requires that a high level be sent to DCP/HRP (no inversion is in the path to the BIU).

Bit Status	Meaning
0	SLEEP (Slow Clock)
1	ACTIVE

8.4.4 D-0283, Element EX- Control

Bit Status	Meaning
0	Disable motor drive electronics for EX- element
1	Enable motor driver, EX-

This control bit should remain cleared.

This bit is a hardware interlock that prevents the antenna deploy electronics from applying power to the indicated antenna element.

8.4.5 D-0285, Element EZ Control

Bit Status	Meaning
0	Disable motor drive electronics for EZ element
1	Enable motor driver, EZ

This control bit should remain cleared.

This bit is a hardware interlock that prevents the antenna deploy electronics from applying power to the indicated antenna element.

8.4.6 D-0285, Element EX Control

Bit Status	Meaning
0	Disable motor drive electronics for EX+ element
1	Enable motor driver, EX+

This control bit should remain cleared.

This bit is a hardware interlock that prevents the antenna deploy electronics from applying power to the indicated antenna element.

8.4.7 D-0286, Maintenance Model Select

Bit Status	Meaning
0	Normal Operations
1	L/P Biased to 10 Volts ROM ONLY

ROM ONLY

This bit indicates that the instrument should be biasing the L/P electronics to +10 volts. Note that none of the current downloaded software honors this bit.

This function is available in the form of an IEB trigger. A MASK variety applies power to the Langmuir probe electronics and an ID variety that assumes the L/P electronics are powered.

8.4.8 D-0287, Processor Write Protect Control

Bit Status	Meaning
0	Write Protect Enabled
1	Write Protect Disabled

The LRP has hardware write protect capability that inhibits writing to lower 16K bytes of memory on the LRP. The protected region is under control of the LRP and may be selected in 2K blocks.

Current flight software does not make use of this capability (Code and data areas for individual processes are not segregated in such a way that the memory protect is useful).

Processor reset clears the write protect select register, that is used by the 8085 to select 1K blocks to write protect, so the state of D-0287 is ineffective.

8.4.9 D-0288, BIU Tables Status

This status bit, when set, indicates that the software has created a working set of control tables for the BIU. This bit may be set to zero for a short period of time following a reset (other times indicates a software fault).

Bit Status	Meaning
0	Default BIU Table in use
1	BIU tables loaded by host processor BCRTM R2 not equal to 0

8.4.10D-0289, RAM Status

This bit is connected to the ***ROM Disable*** signal within the LRP. This bit, when set, disables the boot ROM, placing the ROM into a low power standby state. The instrument, therefore, is executing out of RAM (i.e. this must be using downloaded software).

Bit Status	Meaning
0	ROM enabled
1	ROM disabled Processor executing downloaded code

8.4.11 D-0290, Watch Dog Timer Status

The watchdog timer is implemented within an ACTEL gate array on the LRP. When the timer overflows the processor reset line is pulsed. The upper 2 bits of the watch dog timer are external to the gate array and are connected to the BIU status lines.

In most cases these status lines should be zero.

Version 2.4 flight software adds the capability to monitor both DCP and HRP with the watch dog timer. If this capability is enabled, one might expect to see the timer bits set on occasion.

Bit Status	Meaning
0 0	WDT normal, more than 45 seconds remain
0 1	WDT more than 30 seconds remain
1 0	WDT more than 15 seconds remain
1 1	WDT trip within 15 seconds

8.4.12 D-0291, Antenna Motor Status

This bit is connected to the antenna motor control line. This line is used by the LRP to route power to one of the antenna mechanisms. Assuming the deploy software is loaded and operating, this line is set when the antenna motor is powered. Note that it is possible for this line to be active when the antenna motor is *not* powered, but this is not the expected method of operation.

Bit Status	Meaning
0	Antenna Motor Disabled
1	Antenna Motor Enabled

8.4.13 D-0292, Antenna Status

The deploy software sets this bit to indicate that the antenna operation completed successfully (i.e. the mechanism tripped the appropriate limit switch).

This discrete bit is also used for indicating a successful observation when the special maintenance software is performing a Venus Observation. This observation is performed blind and minimal status (i.e. the discrete bits) is available following the observation.

Bit Status	Meaning
0	
1	Antenna Deploy Operation Successful

This bit may be available for other uses. It holds no interest for conveying deploy activity status since the antennas were successfully deployed.

8.4.14 D-0293, Maintenance Status

The ROM sets this bit to indicate that the Langmuir Probe is biased in response to the assertion of the maintenance mode bit. This action typically requires about 30 seconds to execute so the intermediate state may be observed where the maintenance command bit is set while the maintenance status bit is clear. This delay is used to sequence power to the L/P electronics in the correct order and to allow the voltage present on the L/P electronics to decay to zero before power is applied (in case an attempt is made to cycle power too rapidly).

Version V2.2 of the flight software contains a bug in the handling of this status bit. Although the maintenance state is handled correctly (the housekeeping packet may be used to verify that the L/P is being biased), the status bit is not set during the maintenance activity. Once the maintenance state is removed, the status bit will appear as the software enforces a 30 second idle period following maintenance activities (this prevents the L/P from being powered up immediately following a maintenance activity).

Version V2.3 of the flight software consists of a science load only. No additional maintenance load is provided with this release. A trigger is included in the science load to set the L/P 32 to volts. Due to limitations in the hardware and software it is not possible to present status information on the discrete status line when in maintenance mode when using the V2.3 software.

8.4.15 D-0294, Antenna Fault Status

The deploy software sets this bit to indicate that the antenna deploy operation failed for some reason. The housekeeping data will contain a more detailed status field that describes the error that occurred.

This bit may also be used to indicate a problem with the Venus Observation when the special maintenance software is loaded (although with version 2.2/2.3 software this status bit in ***NOT*** used). As with D-0292 this bit is used as an early status indication as the observation is performed when the S/C is out of contact with earth and minimal status is available when contact is re-established.

8.5 C-0266, Expected Status word Values

The following table shows some expected values for C-0266, the BIU discrete status word.

Instrument State	HEX Value	Discussion
ROM, SLEEP	0x0100	Following Power UP
ROM, Active	0x0104	Prior to loading Science Software
ROM, Maintenance	0x4144	LP biased to 10V
RAM, SLEEP	0x0300	By 73RT command
RAM, Active	0x0304	Normal Operations
RAM, Maintenance	0x0304	By IEB trigger, LP 32V

8.6 Common Area

The following fields do not change.

BYTE	DESCRIPTION		BYTE	DESCRIPTION
0-1	CCSDS Header		21	BIU Discrete Command
2-3	Source Sequence Count		22	Antenna Limit Switches
4-5	Packet Length		23	Power Status
6-11	Time & Error flags		24-39	LRP Analog MUX
12-13	IEB status word		40-55	HRP Analog MUX
14	Valid Command Count		56	BIU misc status
15	Invalid Command Count		57	BIU RTI status
16-17	Command Byte Counter		58	L/P Sphere bias
18	Loop Counter		59	L/P Cylinder bias
19	BIU soft reset count		60	L/P Multiplexer
20	BIU Discrete Status		61	L/P 8155 bits

8.6.1 CCSDS Header

This 16 bit field uniquely identifies this data as belonging to CASSINI/RPWS instrument and that this is housekeeping data.

Data Type	LSB	MSB	VALUE
ROM Based Housekeeping	0x90	0x0A	0x0A90
Antenna Deploy Housekeeping	0x93	0x0A	0x0A93
Special Maintenance Housekeeping	0x93	0x0A	0x0A93
Science Housekeeping	0x95	0x0A	0X0A95

Note that both ***deploy*** and ***special maintenance*** share a common CCSDS pattern. It is expected that deploy will be used only at the beginning of the mission so there will be no confusion over the contents of the housekeeping records.

8.6.2 Source Sequence Count

This field is a 14 bit counter that is incremented as each packet is delivered. The upper 2 bits are always set. This field is used to identify lost packets as well as a point in time when software is re-loaded (when ever the operating software changes, the ***source sequence count*** is reset to zero)

8.6.3 Packet Length

This is seven less than the overall packet length. For housekeeping the value should be 185 indicating an overall length of 192.

8.6.4 Time & Error Flags

40 bit time field and 1553 error control bits.

8.6.5 IEB Status Word

Indicates the last address in IEB memory used as a step of an internal IEB operation. Typically this would contain the IEB trigger number of the latest IEB_TRIGGER issued although many triggers have a level of indirection that may change the value. Since the IEB processor is present in the science code, none of the other handlers will change this location from a value of zero.

8.6.6 Valid Command Count

The number of commands processed by the command decoder. This includes both internal commands (such as those issued by IEB processing) and external commands (such as those issued by CDS).

8.6.7 Invalid Command Count

Number of command buffers that contained a parity error in the first word of a command (once an error is encountered the remaining portion of a command block is discarded).

8.6.8 Command Byte Count

Number of bytes (8 bits) moved from the BIU command buffer for decoding.

8.6.9 Loop Count

Number of commands processed by the "**TWEK**" process. This includes ***73MEM_TWEAK***, ***73IEB_TWEAK***, ***00MEM_TWEAK***, and ***00PORT_TWEAK*** commands. Note that internally, ***00MEM_TWEAK*** commands are generated to allow time tracking on DCP and HRP, so the ***loop count*** should be constantly incrementing.

8.6.10 BIU Soft Reset Count

This field is incremented whenever a soft reset is performed on the BIU. This involves pulsing one of the reset lines to the BIU (**not** the power on reset). This is triggered by a hardware timer on the LRP that monitors the length of time that the LRP waits for access to BIU memory. If the timer expires, the memory access is terminated and an error flag is set and the BIU handler pulses the reset line. In normal operation this should never occur. If the BIU is incorrectly programmed this can occur (but will it ever be seen on the ground?).

8.6.11 BIU Discrete Status

This item contains the most recent value delivered to port 0xF0 on the LRP. Note that bit 2 is similar in function to the BIU discrete bit D-0282, but of opposite polarity.

BIT	DESCRIPTION
7	BIU discrete status bit 7, antenna fault
6	BIU discrete status bit 6, maintenance mode
5	BIU discrete status bit 5, antenna normal
4	read as zero
3	LRP Sleep (XOR with bit 2 to make LRP sleep)
2	DCP/HRP Sleep (1=SLEEP)
1	HRP Reset
0	DCP Reset

8.6.12 Antenna Limit Switches

This item reports the state of the antenna limit switches.

BIT	DESCRIPTION
5	Z Retract
4	Z Extend
3	X minus Retract
2	X minus Extend
1	X plus Retract
0	X plus extend

Although all of the antenna elements have been successfully extended, do not expect the limits switches to remain closed. The antenna elements were driven only long enough to trip the switches and there is no locking mechanism to prevent the elements form moving. It is reasonable to expect the element to relax and release the extend limit switch over time.

Although not expected, one can make use of the element position reading to determine if a creep problem exists?

8.6.13 Power Status

This item reports the state of the power control register.

BIT	DESCRIPTION
7	HFR Command Enable
6	HFR Command Clock
5	HFR Command Data
4	HFR Cold Reset
3	HFR Warm Reset
2	Langmuir Probe Power
1	ME02 (MFR, WBR, WFR) Power
0	HFR Power

8.6.14 LRP Analog Mux

This group of 16 bytes are the conversions of the 16 analog multiplexer channels on the LRP. Several of the channels (not listed) are not used (connected to ground).

	Channel	DESCRIPTION
	0	MFR Band 0
	1	MFR Band 1
	2	MFR Band 2
	3	Power Supply Thermistor
	4	Motor Current
	5	(not recorded) HFR analog multiplexer
V2.4		IEB Error flag and counter
	6	(not recorded) Ground
V2.4		IEB Valid flag and counter
	7	Motor Temperature X plus
	8	Motor Temperature X minus
	9	Motor Temperature Z
	10	(not recorded) Ground
	11	Search Coil Temperature
	12	Antenna element position X plus
	13	Antenna element position X minus
	14	Antenna element position Z
	15	Halt integrator

Several channels of this multiplexer are grounded and always report a zero when read. In version 2.4 software, the housekeeping process does not record the analog channels that are grounded. These positions will contain other interesting housekeeping information as required (when we figure out what we want to see?). Also, the mux channel that connects to the HFR is ignored in this sweep and is read is a separate sweep of the 16 channel mux on the HFR (eliminates an additional redundant byte).

Also keep in mind that the ROM behavior is unchanged. When the instrument is operating from ROM, all 16 channels are presented as read.

5	IEB_LOAD bad record count (D6..D0)
5	IEB_LOAD checksum failed flag (D7)
6	IEB_LOAD good record count (D6..D0)
6	IEB Checksum OK flag (D7)
10	

8.6.14.1 IEB_LOAD bad record count

This counter records the number of IEB_LOAD records that arrive at the instrument that do not pass integrity checks (such as an invalid checksum). We should expect this field to remain zero on the spacecraft. A non-zero value probably indicates that a record has been poorly formed on the ground (verify with the EM).

8.6.14.2 IEB_LOAD good record count

This counter records the number of good IEB_LAOD records that arrive in the instrument in the lower 7 bits of the byte. This field is exactly large enough to handle a full IEB image (one record less than 16K; 63 pages plus the checksum table).

This field does not record movement of the internal IEB table that can be included with the ALF load, in other words, this field as always zero following an ALF load.

8.6.14.3 IEB Checksum error flag

The top bit in the byte is used to indicate that the checksum verification step failed. Starting with V2.4 flight software, the checksum table is included with the IEB load in order to make load verification easier. If a partial IEB load is placed in memory the checksum should fail and prevent the IEB from being used. In the presence of a poor uplink, the load may be sent several times assuming that errors/dropouts occur in random places.

This flag propagates through multiple attempts to load IEB memory and may remain set even when IEB memory is valid. This behavior is intended to give some visibility into the load process (keep in mind that housekeeping delivery rate is 24 bits/second resulting in the delivery of a housekeeping frame every 64 seconds).

This bit is reset using the 73IEB_HALT, CLEAR command that also clears IEB memory.

8.6.14.4 IEB Checksum OK flag

This status bit is set when IEB memory checksum calculation indicates that IEB memory is intact and ready for use. The occurrence of a checksum error will clear this bit.

The initial software load may contain an IEB memory image, and if this image is successfully moved to IEB memory and the checksum operation is successful, this bit is set to make it visible that the internal load was successful.

Another indication of an internal load is when this bit is set, but the lower bits of the byte are all zero, indicating that a successful checksum operation has occurred but that no IEB_LOAD commands were processed (i.e. the load didn't arrive after science software was loaded)

8.6.15 HFR Analog Mux

Channel 5 of the previous multiplexer is connected to an additional 16 channel multiplexer located in the power supply. These 16 channels are all used to monitor voltages and currents within the power supply.

Channel	DESCRIPTION
0	HFR Current
1	ME02 Analog Current
2	Langmuir Probe Current
3	Digital Electronics Current
4	HFR +5 Volt
5	HFR +6 Volt
6	ME02 Analog +12 Volt
7	ME02 Analog +6 Volt
8	ME02 Analog +5 Volt
9	Langmuir Prove +45 Volt
10	Digital Electronics +12 Volt
11	Digital Electronics +5 Volt
12	HFR -6 Volt
13	ME02 Analog -12 Volt
14	ME02 Analog -6 Volt
15	Langmuir Prove -45 Volt

This multiplexer is located within the HFR and contains low-pass filters to eliminate noise. The location of the LPF imposes a rather large settling time (housekeeping process allows several RTI periods for settling when reading these channels).

8.6.16 BIU Misc Status

This item contains the value read from port 0xE0 on the LRP, which is the current state of the 8 discrete command bits from the BIU. Note that these bits are commands from the BIU to the LRP being echoed to housekeeping.

BIT	DESCRIPTION
7	LRP Write Protect Disable
6	Maintenance Mode
5	Element EX plus enable
4	Element EX minus enable
3	Element EZ enable
2	Run (Sleep when zero)
1	LRP Watch Dog Timer Disable
0	LRP Reset

8.6.17 BIU RTI Status (RTI and Dead Time)

This item reports that an RTI interrupt was simulated. In the event that CDS fails to issue the RTI message, the hardware on the LRP will generate an interrupt at the appropriate time and log the occurrence.

Dead Time Start status is not currently logged in the housekeeping page. The software, as of version V2.6, does not store dead-time-start status.

Both RTI-status and DTS-status registers work in a similar manner. Occurrence of the RTI/DTS signal sets a flip-flop that is read by software. The flip-flop is then cleared by the software (writing to a *clear* port). Since this function is split between hardware and software there are some timing issues to be aware of when interpreting the status that is delivered in housekeeping.

8.6.17.1 RTI timing and sequence of events.

Software writes to the *clear* port, resetting the status flip-flop. At some later point, an RTI interrupt occurs causing the CPU to pass control to the RTI interrupt routine. The interrupt routine (some time after the RTI interrupt) reads the status port and examines the **RTI status bit**. The status bit, when set, indicates that S/C has delivered the RTI signal since the reset occurred. If the bit remains cleared, the software increments a counter in housekeeping, indicating loss of RTI.

The RTI signal from S/C (**S/C RTI**) also resets a counter that runs for slightly over the 125 mSec RTI period (see the instrument commands section for details of programming the exact value used to generate the RTI period, it may be changed if needed). When the counter overflows, the hardware generates an RTI interrupt, much the same as if it had come directly from the spacecraft. When RTI is coming from the spacecraft at 125Msec intervals, the **RTI counter** never quite overflows, and never generates a *fake* RTI.

If the **S/C RTI** comes a little late, the RTI interrupt may be generated by the hardware, causing the RTI interrupt service routine to begin execution. If the **S/C RTI** occurs prior to reading the RTI status port, the status bit will indicate that the **S/C RTI** has occurred. In other words, we can't really tell if **S/C RTI** comes a little late(but soon enough to be seen).

8.6.17.2 DTS timing and sequence of events

The *Dead Time Start* happens in much the same manner as RTI. There are some subtle differences, however. DTS is delivered from S/C as the last 1553 transaction during the bus activity of the RTI (the RTI is delivered as the first transaction). The DTS may occur at any time in the RTI prior to the last 5 mSec in the interval. The operating software in the instrument, however, does not want to see DTS interrupt until late in the RTI period. The instrument addresses this by generating the DTS interrupt approximately 120mSec after the RTI interrupt. As with the RTI timing, this is programmable from about 114 to 130 mSec with a nominal time of 120mSec.

The **S/C DTS** signal is only used to set the DTS status flip flop, it is not used to generate the DTS interrupt. The DTS interrupt is only generated by the DTS timer. Because the DTS interrupt is always generated by the timer, we satisfy the need for scheduling the DTS interrupt late in the RTI period.

As with the **S/C RTI** signal, the **S/C DTS** sets the DTS status flip flop when it occurs and the DTS interrupt routine then reads the DTS status flip flop to determine if the DTS signal arrived from S/C. As with the RTI signal, if DTS is slightly late, the latency of servicing the interrupt can mask a late DTS. Unlike the RTI signal, the DTS will usually occur many milliseconds prior to the interrupt generated by the DTS timer (there is no hardware provision to determine the delta-t between **S/C DTS** and DTS timer overflow).

There are bits in the RTI/DTS status register that indicate when the dead-time is active In addition, the status register is **not** clear-on-read, so it is possible to obtain an estimate of when the **S/C DTS** is delivered (although the current software has not provisions to do this).

8.6.18 L/P Sphere Bias

Value loaded into DAC-0 within the Langmuir Probe. MUX and Relay settings are required to determine the actual bias applied to the sphere.

8.6.19 L/P Cylinder Bias

Value loaded into DAC-1 within the Langmuir Probe. MUX and Relay settings are required to determine the actual bias applied to the electric antenna.

8.6.20 L/P Multiplexer

This item reports the state of the Langmuir Probe multiplexer control.

BIT	VALUE	DESCRIPTION
7	0	CYBMR, Cylinder bias mid range
	1	CYBFR, Cylinder bias full range
6 5 4 3	0 X 0 X	SPBMR, Sphere bias mid range
	0 X 1 0	SPBLR, Sphere bias low range
	0 X 1 1	SPBHR, Sphere bias high range
	1 X 1 0	SPBFR, Sphere bias full range
2	0	LPF, Low Pass Filter
	1	HPF, High Pass Filter
1 0	0 0	ADCP2, A/D Cylinder 2
	0 1	ADCP1, A/D Cylinder 1
	1 X	ADSP, A/D Sphere

8.6.21 L/P 8155 Bits

This item reports the state of the 8155 port C. These bits are used to control the routing of signals to the 8237 as well as controlling the power to the Langmuir Probe digital electronics.

BIT	DESCRIPTION
7,6	Undefined, may be 0 or 1
5-3	Should always read ZERO
2	LP DAC Enable
1	LP A/D Enable
0	LP Power enable

8.7 ROM Unique Housekeeping

When operating from ROM the housekeeping area typically contains a memory dump image. The memory address field increments by 128 bytes when the entire housekeeping record contains memory image.

BYTE	DESCRIPTION	LSB	MSB	VALUE
0-1	CCSDS Header	0x90	0x0A	0x0A90
62-63	Memory Address			

Contents of the variable area when no download has occurred.

BYTE	DESCRIPTION			
64-191	Memory Dump			

When downloading an ALF image, the memory dump is reduced to 32 bytes with the remaining area containing diagnostics.

BYTE	DESCRIPTION		BYTE	DESCRIPTION
64-95	Memory Dump		128-143	DCP Status Area
96-111	LRP Status Area		148-191	Most Recent ALF record
112-127	HRP Status Area			

8.7.1 Memory Dump

32 / 128 byte memory dump. The memory dump size is restricted to 32 bytes to make room for the ALF load status that follows. When the instrument is first powered on, it will only dump 32 byte records into the housekeeping frame in preparation for an ALF download. In the event that no ALF traffic is delivered to the instrument, it will begin dumping 128 byte records when the address counter reaches 0x100 (about 8 to 9 minutes).

Only the memory on the LRP is dumped in this manner. There was not sufficient code space in the ROMs to allow HRP and DCP to dump their memory contents. The DCP and HRP ROMs can only be checked by downloading a block of code to handle the IPC hardware (although this seems a pointless task, if it downloads the ROM is functioning).

There is a little visibility into the operation of the IPC during the download process in the form of a counter that increments with each successful IPC transfer to the DCP and HRP. This counter may be difficult to examine as it is updated by the ROM and will be cleared when any download clears the housekeeping buffer (a download takes less than half of the time between housekeeping updates).

It is expected that one may use the GSE housekeeping display software to obtain a formatted dump of the ROM on the LRP for comparison purposes. On the LRP the ROM occupies locations 0 through 0x0FFF (a total of 4K bytes). Note that the ROMs on all of the RPWS instruments are identical so any ground unit may be used as a reference (the external memory boards should also contain *as-flown* flight images).

8.7.2 LRP, HRP and DCP Status Area

LRP/HRP/DCP Status area. During CDS downloads only the LRP area is updated. During internal downloads, only the LRP and HRP areas are updated. When a failure occurs, the DCP area is updated. Keep in mind that there is a unique area reserved for each of the three processors.

BYTE	DESCRIPTION		BYTE	DESCRIPTION
+0-1	ALF Sequence		+8-9	Local ALF record count
+2-3	Local Load Address		10	Reason Code
+4-5	Local Skip Count		11	Count
+6-7	Local Load Count			

8.7.2.1 ALF Sequence

The sequence field from the last correct ALF record.

8.7.2.2 Local Load Address

Decoded local address. This is the memory location within the 8085 where the most recent ALF record was stored.

8.7.2.3 Local Skip Count

Number of ALF_SKIP records processed.

8.7.2.4 Local Load Count

Number of ALF records that were loaded into memory on the local processor. Keep in mind that parts of the load are processed (i.e. loaded into memory) on all processors.

8.7.2.5 Local ALF Record Count

Number of ALF records processed on the local processor.

8.7.2.6 Reason Code

Status of the ALF operation. This status remains zero until an unusual condition occurs. The status field is updated and processing stops.

	VALUE	MEANING		VALUE	MEANING
	0x00	Normal (no problem)		c	checksum word error
	B	Booting		I	invalid identifier
	C	Word Count not 22		p	packet count in ALF_EOF
				s	Sequence word error

8.7.2.7 Counter

Well, it counts, doesn't it ?!?

8.7.3 Most Recent ALF Record

This portion of the housekeeping record contains the most recent ALF record that has been processed by the LRP.

8.8 Deploy Unique Housekeeping

When operating with antenna deploy software loaded, the variable area contains a line of static status data (i.e. most recent values read from the hardware registers) while the remainder contains dynamic monitoring data.

BYTE	DESCRIPTION	LSB	MSB	VALUE
0-1	CCSDS Header	0x93	0x0A	0x0A93
62-63	Spare			

Contents of the variable area.

BYTE	DESCRIPTION			
64-79	Static Area			
80-191	Dynamic Area			

The Static area contains the following when executing a most commands.

BYTE	DESCRIPTION		BYTE	DESCRIPTION
64-65	Run Timer		73	Test Mask
66-67	Command RTI		74	Motor Current
68-69	Command pattern		75	Motor Temperature
70	Command Index		76	Element Position
71	Antenna Index		77	Limit Switch Register
72	Status		78-79	Reason Code

8.8.1 Run Timer

This field echoes the number of RTI periods remaining before the current operation will be terminated. If the timer expires a error status (***reason code***) will indicate that the timer expired before the antenna element reached a limit switch.

8.8.2 Command RTI

This field indicates the time the most recent command was received.

8.8.3 Command Pattern

This is the 1st. 16 bits of the most recent command. The internal monitoring software may generate an *Antenna Hold* command to stop element movement.

Command Pattern	Source	Function
0x1080	External	Antenna Hold
0x1090	Internal	Antenna Hold (dynamic monitoring HOLD)

8.8.4 Command Index

This is a decoding of the most recent command.

1. Extend Element
2. Retract Element
3. Status Report

8.8.5 Antenna Index

This is a decoding of the antenna selection in the most recent command.

1. Element EX Plus
2. Element EX Minus
3. Element EZ

8.8.6 Status

This is an internal status flag.

8.8.7 Test Mask

This is an internal mask used to examine the limit switches.

8.8.8Motor Current

This is the value read from the A/D channel associated with motor current.

8.8.9Motor Temperature

This is the value read from the A/D channel associated with the currently selected antenna mechanism.

8.8.10 Element Position

This is the value read from the A/D channel associated with the currently selected antenna mechanism.

8.8.11 Limit Switch Register

This is the most recent value read from the antenna element limit switches. This register is address at 0xC0 on the LRP.

8.8.12 Reason Code

This is a 16 bit field used to return the status of the antenna control software. The table indicates the BIU discrete status bit that is set when the status code is presented in housekeeping. Several of the status codes are not expected to appear and indicate a problem if encountered (indicated with ***pfr*** in the DISC column).

Hex Status Value	Reason Code	DISC
0x56, 0x**nn**	Flight Antenna Software version **nn**	n/c
0x54, 0x**nn**	I&T Antenna Software version **nn**	n/c
0x89, 0x01	Bad Antenna Command	D7
0x00, 0x03	Drop during GO	***pfr***
0x81, 0x04	Attempted to clear relay	D7
0x00,0x05	Drop before GO	***pfr***
0x81, 0x06	ON Time limit reached	D7
0x21, 0x07	Extend limit switch	D5
0x21, 0x08	Retract limit switch	D5
0x21, 0x09	Limit Switch reached	D5
0x81, 0x10	Over Current	D7
0x81,0x11	Position potentiometer limit	D7
0x81, 0x12	Element stopped	D7
0x81, 0x13	High temperature limit	D7
0x81, 0x14	Low temperature limit	D7
0x81,0x15	BIU discrete bit cleared	D7
0x81, 0x16	BIU SLEEP asserted	D7
0x02, 0xFF	attempting element retract	D4
0x02, 0xFE	attempting element extension	D4
0x02, 0xFD	attempting initial retract before extend	D4
0x03, 0xF2	ANT_STATUS command received	n/c
0x03, 0xF1	ANT_SETUP command received	n/c
0x02, 0xF0	ANT_HOLD command received	n/c

The 1st. byte is a *status class* indicator. Status class of 2 indicates an antenna movement command was received. Movement commands may be extend, retract or hold commands. The extend command may enable or suppress handling of the caging pin but the status indicates the type of command used to generate movement (in other words, 02FD is the status expected for a normal extend operation).

The Static area contains the following after a setup or status command. Fields with the same name as the previous table are identical.

BYTE	DESCRIPTION		BYTE	DESCRIPTION
64-65	Run Timer		73	Minimum Current
66-67	Command RTI		74	Maximum Temperature
68-69	Command pattern		75	Minimum Temperature
70	Command Index		76	Maximum Position
71	Antenna Index		77	Minimum Position
72	Maximum Current		78-79	Reason Code

8.8.13 Maximum/Minimum Current

These are the static limits checking values for the selected antenna element. A value of zero in both fields indicates that limits checking is disabled.

8.8.14 Maximum/Minimum Temperature

These are the static limits checking values for the selected antenna element. A value of zero in both fields indicates that limits checking is disabled.

8.8.15 Maximum/Minimum Position

These are the static limits checking values for the selected antenna element. A value of zero in both fields indicates that limits checking is disabled.

The Dynamic area contains the following record repeated 5 times.

BYTE	DESCRIPTION		BYTE	DESCRIPTION
+0-1	RTI		+12,14,16,18	Element Position (2 sec)
+2	Motor Temperature		+19	Average Current
+3,5,7,9	Motor Current (2 sec)		+20	Minimum Current
+11,13,15,17	Motor Current (2 sec)		+21	Maximum Current
+4,6,8,10	Element Position (2 sec)			

8.8.16 RTI

The RTI at the beginning of the 16 second collection period.

8.8.17 Motor Temperature

Average motor temperature for the 16 second period.

8.8.18 Motor Current, 2 second average

8 sets of 2 second averages.

8.8.19 Element Position, 2 second average

8 sets of 2 second averages.

8.8.20 Average Current

Average motor current for the 16 second period.

8.8.21 Minimum/Maximum Current

Minimum and Maximum motor current for the 16 second period.

8.9 Science Unique Housekeeping

When operating with science software loaded, the variable area contains micro-packets that may originate from any subsystem within the instrument. The special maintenance software is formatted in the same manner.

BYTE	DESCRIPTION	LSB	MSB	VALUE
0-1	CCSDS Header	0x95	0x0A	0x0A95
62-63	Micro Packet Counter			
64-191	Micro Packet area			

8.9.1 Micro Packer Counter

This micro-packet counter is incremented as each micro packet is moved into the housekeeping frame.

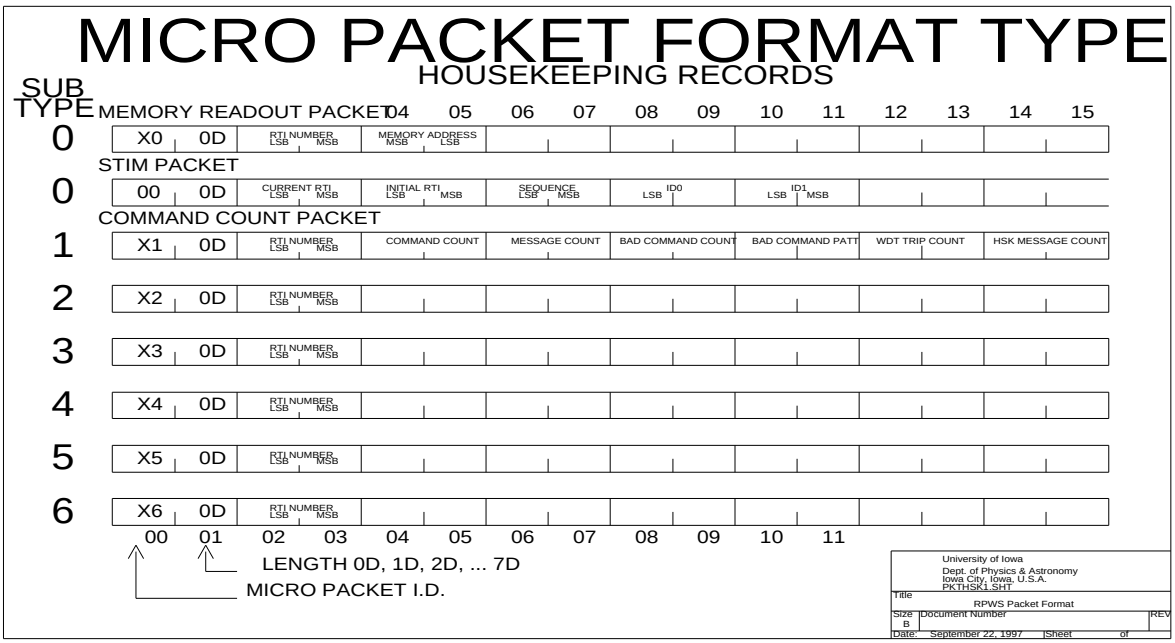
8.9.2 Micro Packet Area

BYTE	DESCRIPTION		BYTE	DESCRIPTION
+0	Source ID		+2-3	RTI
+1	Length		+4..n-1	status data

The micro packet appears much the same as a data mini packet. The length must be a multiple of 16 with a maximum size of 128 bytes. The length is encoded in the same manner as the data mini packet, a 16 byte packet will have a length of 13 (i.e. the length field is one less than the number of bytes that follow).

Micro Packets are stored in the ***micro packet area*** in the form of a stack (Last-in First-out). The micro packet located at byte offset 64 is the most recent. Using the length field the remaining micro packets can be extracted from the buffer.

There is no flow control mechanism so the source of the micro packets must make arrangements to avoid overflowing the ***micro packet area***.



8.10 Micro Packet: MRO

Memory Readout packets that appear in housekeeping present 10 bytes of dump data with 6 bytes of overhead (for a total of 16 bytes) . No status bits are available to indicate the processor or memory bank associated with this record.

BYTE	DESCRIPTION		BYTE	DESCRIPTION
0-1	ID/Length 0xD00D		4-5	Memory Address
2-3	RTI		6-15	Memory Dump

8.10.1 Memory Address

16 bit address

8.10.2 Memory Dump

10 bytes of memory image

8.11 Micro Packet: BFDL

BFDL status packets are generated when WBR, WFR, Dust or LFDR commands are received by the instrument. When a command is received a timer is started and when the timer expires an attempt is made to deliver a status micro packet to the housekeeping process. Several situations may occur that prevent the delivery of the BFDL status micro packet, first being continuous commanding may keep the timer from expiring.

In addition, the status delivery activity is performed in the context of the IPC Watch Dog Timer so no system service calls are allowed that may result in the process being blocked. The queue read to obtain a buffer to hold and deliver the status micro packet is non-blocking. This may result in failure to obtain a buffer with expected results. No attempt is made to recover from this failure (i.e. the status record will be lost).

BYTE	DESCRIPTION		BYTE	DESCRIPTION
0-1	ID/Length 0xC10D		8-9	Bad Count
2-3	RTI		10-11	Bad Command
4-5	Command Count		12-13	WDT Count
6-7	Message Count		14-15	WDT Packets

8.11.1 Command Count

Command counter. Each command extracted from the incoming buffer is registered with this counter.

8.11.2 Message Count

Incoming command buffer counter. This counter is incremented as each IPC buffer is received.

8.11.3 Bad Count

Count of the number of bad commands due to:

0Destination Invalid. Not a WBR/WFR/DUST/LFDR command

1Command Index Bad. The WFR/WFR/DUST/LDFR commands make use of a 3 bit command *index*. Not all of these sub-commands are valid. Commands that specify an invalid index are rejected and counted.

8.11.4 Bad Command

16 bit command pattern that was invalid. This field is only overwritten by bad commands so the most recent bad command received is always visible.

8.11.5 WDT Count

When either of the data acquisition processes begin data collection, a software timer is started to catch a failure of the DMA hardware. Any time this software timer expires the event is counted with this counter. This particular event is probably indicative of a hardware problem. (May occur with V2.2 during single channel LFDR activities that are using large buffers).

8.11.6 WDT Packets

Count of the number of micro packets delivered from the WBR/WFR command process on the HRP. If data is lost, there will be missing values from this field as it should increase monotonically.

8.12 Micro Packet: DUST

The DUST micro packets make use of the TWEK process to deliver an MRO record to housekeeping. Since the micro packet is formatted by the TWEK process, the header and format are that of an MRO packet.

BYTE	DESCRIPTION		BYTE	DESCRIPTION
0-1	ID/Length 0xD00D		10	# dust hits at 20 dB gain
2-3	RTI		11	# dust hits at 30 dB gain
4-5	Address 0x27E0		12	# dust hits at 40 dB gain
6-7	# WBR packets examined		13	# dust hits at 50 dB gain
8	# dust hits at 0 dB gain		14	# dust hits at 60 dB gain
9	# dust hits at 10 dB gain		15	# dust hits at 70 dB gain

8.13 Micro Packet: IPC

The IPC micro packets require an MRO command be issued (i.e. these are not automatic). This seems to be a convenient place to document the contents of the memory within the IPC driver we are interested in.

The command required to dump the status block within the IPC driver may be issued to any (or all) of the processors as follows:

73MRO, processor, HSK, 1170, 0

BYTE	DESCRIPTION		BYTE	DESCRIPTION
0-1	ID/Length 0xD00D		8-9	TX Retry
2-3	RTI		10-11	TX Abort
4-5	Address 0x1170		12-13	RX Lost Data
6-7	TX Lost Data		14-15	RX No F5 available

An expanded dump, comprised of two MRO records may be obtained using this command (causing a second MRO record to be generated):

73MRO, processor, HSK, 1170, 117F

BYTE	DESCRIPTION		BYTE	DESCRIPTION
0-1	ID/Length 0xD00D		6-7	RX Abort
2-3	RTI		8-9	Trigger (always 0x00)
4-5	Address 0x117A		10-15	"PROCES"

8.13.1 Address

This data area is mixed in with the process descriptor and is effectively located in a static data area. This address should not change and it is identical on all 3 processors.

8.13.2 TX Lost Data

Transmit failure: receiving side did not assert *Clear to Send.* This is indicative of high traffic loads or possibly a failed processor (although the loop count in housekeeping is probably a better indication that DCP/HRP are dead or alive).

8.13.3 TX Retry

This counter is incremented each time a transmit retry is attempted. This field will be incremented occasionally during periods of high traffic load (such as when high rate science is active). This field incrementing alone (i.e. *TX Lost Data* and *TX Abort* remain zero) is not an indication of a problem.

8.13.4 TX Abort

This field is incremented when the DMA activity for the transmit channel time out. This occurs only when both processors involved in a transaction have agreed to a data transfer so this counter is probably indicative of a problem.

8.13.5 RX Lost Data

This counter increments when no data buffer is available to receive an incoming transaction. When this occurs, the incoming data is accepted into a reserved buffer and immediately discarded. There is no indication that a problem has occurred on the transmit end.

8.13.6 RX No F5 Available

This counter increments when no data buffer is **immediately** available to the receive process. This counter is incremented before the receive process waits for activity on the

incoming *Request to Send* lines. This does **not** mean that data was lost, rather this counter is an indication that **F5** buffers are scarce.

8.13.7 RX Abort

This counter is the counterpart to the **TX Abort** counter above. It does not appear when a single MRO record is requested

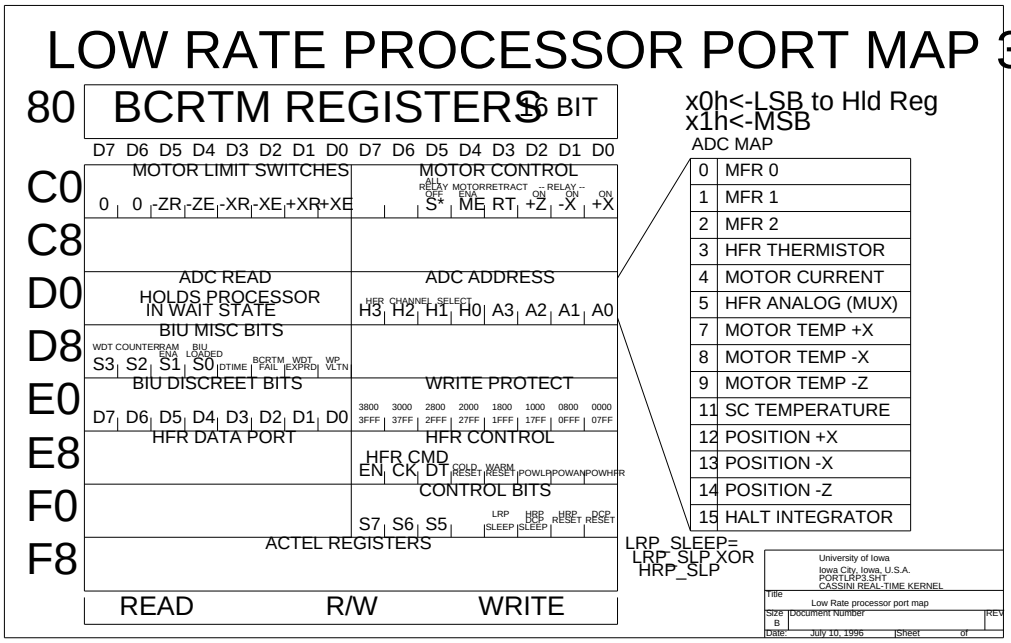
8.14 Ancillary Data

The instrument continuously maintains an ancillary data packet for collection by CDS. The instrument keeps the contents of the table constantly updated. When running maintenance software, the Langmuir probe bias voltage is valid and flagged as valid.

Word	Bit	Description	Value
0	15-8	Message Identification	0000 0010
	5	Producer error flag	1 indicates an error
	4	CDS collection error flag	1 indicates an error
	3-0	Data type ID	RPWS=0010
1	15-0	RTI number (13 bits of seconds)	
		Status for this second	
2	15	HFR status valid	1 = valid status
	14	HFR sounder active	1 = sounder running
	13	HFR sounder enabled	
	11	Langmuir Probe status valid	1 = valid status
	10	Langmuir Probe electronics on	
	9	Langmuir Probe sweep	1 = sweep in progress
	8	L/P Sphere Bias valid	1 = indicates voltage is valid
	7-0	LP Bias Voltage	0 indicates -45 volts FF indicates +45 volts linear scaling
		Status for upcoming second	
3	15-0	Duplication of word 2	

8.15 Low Rate Processor Port Map

Some of the processor ports are presented, essentially unchanged, in the housekeeping data stream.



8.16 Housekeeping Display: DSPHK/HKROM

The following display is typical for the RPWS instrument. This display indicates that the instrument is operating with the downloaded science software.

```
 1    CASSINI RPWS ROM housekeeping dump (HKROM5) V6.1 In: 46
 2  15 Housekeeping, Science    (.record_type)2048   RTIU length 42086
 3  Seq    68 Len 185 epoch:E96EF2EB Time:4ACB66A4.6 Dsp: 1 45 0
 4  0A95 C044 00B9 004A CB66 A4C0  1997 279 15:47:27 Out: 0 45 0
 5
 6 0000: 95 0A 44 C0 B9 00 4A 00 66 CB C0 A4 00 FF 9D 00  ..D...J.f.......
 7 0010: EE 01 37 00 00 04 15 06 4E 08 1A 79 00 03 00 8D  ..7.....N..y....
 8 0020: 8D 8C 00 A0 34 36 26 2D 04 A8 90 6A 00 00 D5 D1  ....46&-...j....
 9 0030: DC E5 BC D2 00 71 73 DB 30 00 80 80 56 C3 21 00  .....qs.0...V.!.
10 0040: C1 0D 87 1C 71 00 71 00 00 00 00 00 00 00 20 00  ....q.q....... .
11 0050: C1 0D DF 16 61 00 61 00 00 00 00 00 00 00 1F 00  ....a.a.........
12 0060: C1 0D FB 14 60 00 60 00 00 00 00 00 00 00 1E 00  ....`.`.........
13 0070: C1 0D 17 13 5F 00 5F 00 00 00 00 00 00 00 1D 00  ...._._.........
14 0080: C1 0D 33 11 5E 00 5E 00 00 00 00 00 00 00 1C 00  ..3.^.^.........
15 0090: C1 0D 4F 0F 5D 00 5D 00 00 00 00 00 00 00 1B 00  ..O.].].........
16 00A0: C1 0D 6B 0D 5C 00 5C 00 00 00 00 00 00 00 1A 00  ..k.\.\.........
17 00B0: C1 0D 87 0B 5B 00 5B 00 00 00 00 00 00 00 19 00  ....[.[.........
18  ------------------------- Analog Monitors VOLTS -------------------
19  MFR 1 [1.529]  MFR 2 [0.157]  MFR 2 [0.510]
20  HFR T 23.9C Ant mtr I [0.000]  HFR Ana 0 [0.059]
21  Motor Temp +X 20.4C  -X 20.4C  +Z 20.7C
22  Search Coil Temp 26.3C  HALT 83%
23  Motor Position +X [1.020]089  -X [1.059]095  +Z [0.745]059
24   HFR   4.2mA  +5:0.000V  +6:0.000V  -6:0.000V
25   LRP 213.2mA  +5:5.024V +12:11.833V
26   MFR 108.4mA  +5:5.306V  +6:6.065V +12:11.903V -12:-11.840V  -6:-6.103V
27   LP   42.1mA           +45:49.841V -45:-49.984V
28  ---------------- Digital Monitors ------------------
29  Ant sts (in C0) [15] -ZR -XR +XR  Pwr_Sts (out E8) [06] LP ME2
30  BIU_Disc_Cmmd (in E0) [04]   BIU_rst 0 BIU_RTI 0
31  Sts (out F0) [00] ACTIVE  Misc (in D8) [30] RAME BIUL
32  ---------------- Command Monitors ------------------
33  Cmd Count [494]  Valid [157]  Invalid[0] Loop[55]
34  IEB ID   [52 00] Bad-0[00] Good-0[0] IEB Mem Empty
34  ---- HRP status --------------------
35   LP P8155 port C [C3]       ADC PWR
36   LP  MUX [56]  DAC-0 [80]  DAC-1 [80]
```

8.16.1 Line 1

Housekeeping display software version is shown toward the end of the line along with a count of the number of records processed by the housekeeping display program.

8.16.2 Line 2

This line contains an indication of the software that is operating in the instrument. There are three potential sets of downloaded software along with the ROM based set. Two of the loads (maintenance and deploy) make use of identical CCSDS identification words (it will take the display program several records to correctly differentiate between deploy and maintenance).

There is some additional information that is useful for debugging problems with the S/C simulator used at Iowa that are beyond the scope of this document (i.e. the record_type and RTIU length fields)

8.16.3 Line 3

Displays the sequence number and length field that occurs in the CCSDS header, the epoch of the data as well as the raw time contained in the CCSDS header. The epoch information is not presented in the housekeeping telemetry although it may appear in the CHDO data delivered from the SOPC.

The **Dsp** fields indicate the number of various packet types processed by the display software. The first value being the number of housekeeping records processed and the second value being the number of science packets processed. This second field is useful to determine if science telemetry is being received by the housekeeping program.

8.16.4 Line 4

This line contains a hex dump of the CCSDS header along with the time field converted to a human readable form.

The Out fields indicates the number of housekeeping and science packets that the display program have written to the output file.

The CCSDS header is composed of six words:

5. CCSDS ID word

   Assigned by JPL.

6. Sequence

   14 bit sequence, monotonically increasing. Upper two bit always set.

7. Length

Length of data following the primary header, minus one. In other words add seven to this field to arrive at the packet length. Should always be 0x00B9.

8. Error Flags / Time

   8 bits of error flags followed by bits 31-25 of SCLK.

9. Time

   Bits 24-8 of SCLK

10. Time

Bits 7-0 of SCLK and 8 bits of millisecond. Only the upper 3 bits are of milliseconds field are meaningful.

8.16.5 Lines 6-9

Hex dump of the fixed portion of the housekeeping record. This is the area of housekeeping that does not change with any of the software loads. This is essentially the *global* housekeeping data.

The hex dump include an ASCII display that is probably of little use.

8.16.6 Lines 10-17

Hex dump of the *private* housekeeping data. This portion of the record changes with each software download. The detailed format for each load is described earlier in this section.

This section also includes an ASCII display of the data. When dumping process descriptors or other text areas, the ASCII dump becomes useful.

8.16.7 Lines 19-27

This area contains the 32 analog channels converted to some convenient form. The display program has several selectable conversion methods but the nominal conversion method is to convert to the most appropriate units.

8.16.7.1 Line 19

MFR channels. These are the voltages present on the 3 MFR integrators. Since there is no useful conversion these channels are simply converted to the voltage presented to the A/D.

8.16.7.2 Line 20

HFR temperature, converted to degrees Celsius.

Antenna motor current. No useful conversion may be done so this is simply the voltage presented to the A/D by the current monitor within the antenna drive electronics.

P/S multiplexer. This is the voltage obtained from channel 0 of the HFR mux. This should be the voltage presented to the A/D by the HFR current monitor in the power supply.

8.16.7.3 Line 21

Motor temperature. This is the temperature of the drive motors in the three antenna mechanisms, converted to degrees Celsius.

8.16.7.4 Line 22

Search coil temperature, converted to degrees Celsius.

Halt Integrator. The processor typically executes a HALT instruction when no work remains to be performed. The processor provides a status signal to indicate the HALT instruction is executing and this status signal is passed through a simple filter to yield a voltage level that is an indication of how busy the processor is. This voltage is converted to a percentage indication of how much idle time the processor has. Since the ROM does not make use of the HALT instruction, the value is 0 following a reset (i.e. no idle time).

Also note that the EM does not contain the circuitry to implement the halt integrator (the reading will always be at the extreme end of the scale).

8.16.7.5 Line 23

Motor position, converted to volts. Calibration of the position pot requires a conversion table unique to each of seven elements. The reading is simply the voltage at the A/D.

Later versions of *hkrom/dsphk* will also display an approximation of the value that appears on the antenna mechanism GSE. Tables exist that correlate the physical length of the antenna with the readout on the GSE box.

8.16.7.6 Lines 24-27

Power Supply numbers. These are the 16 channels of data from the power supply. All of the data numbers have been converted to current and voltage as appropriate.

8.16.8 Lines 29-31

Digital Monitors
Title (port) [value] BITS

The digital monitors are status values extracted from the global housekeeping data.

All of the lines follow the same general format starting with a short descriptive title for the status word. In parenthesis, the source of the data is indicated, usually a hardware port in the 8085 processor. Next the hexadecimal value of the status word is shown in brackets. Finally, short mnemonics (in uppercase) are listed for the active bits.

8.16.8.1 Ant sts - Antenna Status

The following tags are displayed when the antenna element is at the extend limit switch or the retract limit switch

	Retract Limit Switch	**Extend Limit Switch**
X Dipole +	+XR	+XE
X Dipole -	-XR	-XE
Z monopole	-ZR	-ZE

8.16.8.2 Pwr_Sts - Power status

These tags indicate that power is applied to the indicated receiver.

Affected Section	**tag**
Langmuir Probe Analog Electronics	LP
Main Electronics MFR, WBR, WFR	ME2
High Frequency Receiver Digital and Analog	HFR

8.16.8.3 BIU_Disc_Cmmd - BIU discrete command

This the BIU discrete command bits presented to the LRP. No mnemonic decode is currently attempted on this data.

8.16.8.4 BIU-rst - BIU reset count

Indicates the number of times that the software, running in the LRP, has attempted to reset the BIU. There are some rare situations that can trigger an interaction between the BIU and the LRP. This counter indicates that the BIU has failed to grant LRP access to BIU memory for a period exceeding approximately 10 microseconds. When the LRP detects this condition it reacts by resetting and reconfiguring the BIU in an attempt to continue.

This is usually an indication that the LRP is accessing the BIU in an inappropriate manner.

An early version of flight software seems to have addressed the problem this item addresses. Any time this counter has a non-zero value it is probably an indication of a problem.

8.16.8.5 BIU_RTI - Simulated RTI count

Indicates the number of time that the RTI signal was simulated by the hardware. The hardware that simulated the RTI notifies the software of this occurrence so that it may be logged.

As part of the startup sequence, the hardware will mark exactly one loss of the RTI. Any value other than 1 indicates that the RTI signal was lost. Occasional events are nothing to worry about, but seeing several occurrences may indicate a 1553 problem(?).

Complete loss of RTI will probably not be visible in this counter as the number of RTI periods between housekeeping pickups is 512 while this counter, being 8 bits, rolls at 256.

8.16.8.6 Sts - Status

This reflects the BIU discrete status information available within the processor.

Load	Status	tag
Deploy Maintenance	Antenna movement Fault Venus observation Fault	AFLT
ROM Maintenance	L/P Biased to 10 Volts L/P biased to 32 volts	MAIN
Deploy Maintenance	Antenna movement successfully complete Venus Observation complete	ANOR
Deploy	Antenna motor power enable (does not indicate movement)	AMOT
	LRP is operating at ½ Clock Speed (750 Khz)	LSLP
	DCP and HRP are operating at 1/16 clock speed (around 100Khz)	SLEEP
	Processors operating at full clock speed (1.5 Mhz)	ACTIVE
	High Rate Reset	HRST
	Data Compression Reset	DRST

8.16.8.7 Misc - Miscellaneous status

Other status indicators from within RPWS

Load	Status	tag
	Watch Dog Timer Status	WDT0 WDT1
	RAM Enable LRP ROM is disabled	RAME
	BIU Tables Loaded by LRP with non-default values BIU should be compatible with CDS	BIUL
	Dead Time	DTIM
		BCRTM
	Watch Dog Timer Enable	WDTE
	Write Protect Violation	WPV

- Watch Dog Timer Status

Watch dog timer status is normally sitting at a value of 0. LRP should never become busy to the point that the watch dog timer cannot be serviced. Any time either of these bits is non-zero this is an indication of a problem.

- RAME

This bit is set when downloaded software disables the ROM and starts to execute the downloaded software. This bit would indicate a problem if it is set when the instrument first powers up or remains cleared following a download.

- BIUL

This bit indicates that the BIU tables have been loaded with the values used by RPWS. This bit should always be set and showing the bit as a zero indicates a problem.

8.16.9 Line 33

Command statistics.

Cmd Count is the number of command bytes correctly processed by the 1st level command decoder. This counter indicates that the initial command word has the correct parity. This is a BYTE counter in spite of all commands being multiples of 16 bits.

Valid is the number of valid commands processed by the 1st level command decoder. Commands are variable length, ranging from one to eight words. Only ALF and IEB_LOAD commands exceed eight words in length. ALF and IEB_LOAD commands are counted individually (i.e. the command decoder recognizes these special cases).

Invalid indicates the number of command buffers that failed the parity test by the first level command decoder. When the parity is suspect, no attempt is made to process subsequent commands (i.e. any subsequent commands are discarded).

As noted in earlier sections, ALF commands have opposite parity with respect to all other commands and will be rejected by any downloaded software.

8.16.10 Line 34

This line shows the IEB handler status. This line is meaningful only with software versions V2.4 and newer. The IEB ID field from the IEB handler indicates the current step that the IEB handler is executing. The Bad/Good counters and flags give an indication of the number of records processed by the IEB handler. Bad records are those with an invalid checksum field. The Good field is the number of IEB records loaded into memory. The top bit of each of these fields is a status flag. The Bad flag being set when a memory checksum operation indicates that IEB memory is corrupt (and IEB Triggers will not be executed). The Good Flag is set when IEB memory has been loaded and the checksum operation is successful.

The Good Flag will be set when an internal IEB is moved to IEB memory during the ALF loading process.

The text at the right end of the line indicates the current status of IEB memory as determined by the status flags. The flags indicate one of four possible conditions that IEB memory may be in.

Checksum Error Status	Checksum Valid Status	IEB Memory condition
0	0	IEB memory not loaded (memory load may be in progress)
0	1	Successful load (checksum valid)
1	0	Unsuccessful Load (checksum invalid, IEB_TRIGGER commands suppressed)
1	1	Successful load (checksum valid) following an unsuccessful load

8.16.11 Lines 36-37

L/P status area. These lines display the available Langmuir Probe control bits (Power and DMA enables) and the current voltage applied to the L/P sphere and the X antenna (multiplexer bits determine if the sensors are actually being biased).

8.17 Housekeeping process notes

A description of how housekeeping telemetry is gathered and time-tagged.

The housekeeping process is used for all telemetry modes, it internally determines the instrument operating configuration form the *Data_LRP_Antenna_Flag* located in the system data page. Based on this flag, the CCSDS ID word is loaded with the appropriate value.

As part of the initialization activity, the housekeeping buffer is cleared to zero. If CDS happens to collect housekeeping shortly after this, the housekeeping packet will, of course, show these zeros (setting off most of the Red Alarms for RPWS).

Housekeeping priority if rather low, with only the HFR Sounder process at a lower priority. In normal operation the Housekeeping process completes in 33 RTI periods. If, for some reason, LRP becomes CPU bound, the Housekeeping process may be delayed for about 477 additional RTI periods without loss of any housekeeping data.

Since the main housekeeping loop completes in 33 RTI periods, the peak pickup rate for housekeeping is just over 4 seconds.

8.17.1 A/D Channel

Note that MFR and Housekeeping share use of the LRP analog MUX (i.e. The MFR channels are connected to to 3 of the A/D channels). The hardware is shared using a 2-level scheme.

At one level, the Housekeeping process has exclusive access to the HFR bits in the MUX control register. MFR must collect the current HFR Mux setting and combine it with it's LRP MUX setting in order to allow the analog signal from the HFR MUX to the LRP MUX to stabilize. This is accomplished by simply having MFR and Housekeeping disable interrupts when writing to the MUX control register and when accessing the System Data Page location that holds these bits. The Disable-Interrupt / Enable-Interrupt instructions that bracket these accesses prevent the other process from obtaining the CPU at a critical time. The group of protected instructions is small, so this method has less CPU impact than using an Mx Flag to protect access to the A/D hardware.

The second level operates much as the first, but does not require a location in the System Data Page for MFR and Housekeeping to communicate. Again, we simply make use of Disable-Interrupt / Enable-Interrupt instructions to prevent a process dispatch from occurring when accessing the A/D hardware. Again this is less impact on the CPU than making use of the Mx Flag.

8.17.2 Housekeeping Loop

8.17.2.1 Flag Set: F_LRP_HSK2

This flag was used to notify another process that housekeeping cycle has finished. It seems to have been removed at some point, but the flag operation has been retained in the housekeeping process.

8.17.2.2 Flag Wait: F_LRP_HSK

BIU handler (BIUK6.ASM) sets this flag following a housekeeping packet pickup, allowing housekeeping process to release CPU until the next housekeeping record needs to be collected.

Note that the timing of the housekeeping cycle is limited by the frequency with which this flag is set and how many steps are in the housekeeping loop (each step requires 1 RTI to complete)

8.17.2.3 Place CDS Timetag

Immediately following the flag wait, the CDS time tag is placed into the housekeeping buffer. The time tag, therefore, marks the beginning of the housekeeping collection cycle. In normal operation, this timetag will be almost 64 seconds old when the housekeeping packet is collected by the spacecraft.

8.17.2.4 Collect HFR Analog Channels

16 channels of HFR analog data are collected, one channel per RTI period. MUX channel is set prior to the delay and the A/D conversion is performed following the delay.

The delay period is nominally 1 RTI period, but the housekeeping process simply schedules a wait that expires at the next RTI period. This means the delay period can be short if there is significant activity on the LRP.

8.17.2.5 Collect LRP Analog Channels

Similar to the HFR collection activity, each channel of the LRP MUX is collected with an intervening 1 RTI delay. The delay serves to spread the CPU requirements over a longer period of time (settling time of the MUX is not an issue with the LRP analog channels).

8.17.2.6 Collect Bit Status

Several status bytes are moved from the system data page to the housekeeping page. This activity is not spread out, it all occurs in a single RTI period.

8.17.3 Housekeeping MicroPackets

Micro packets may be generated by any process in the instrument. The Housekeeping process hooks the watch dog timer (which is part of the IPC process) to provide a mechanism to poll for the arrival of micro packets every 4 seconds. During each cycle, all micropackets that are available on the LRP are moved to the micropacket buffer are of the housekeeping packet (in BIU memory).

The polling interval of 4 seconds is do reduce the impact of micropacket processing on the CPU. Currently there are only a few sources of micropackets: WBR/WFR handler generates command statistics on a occasional basis (typically less than 1 bit per second); DUST generates status packets on a regular basis (about 2.25 bits per second); and TWEAK/MRO can generate about 10 bits/second(through external command).

There is no provision to throttle delivery of micropackets, if the presented load exceeds the housekeeping pickup rate, micro packets will be overwritten in the housekeeping buffer (the micropacket delivery process does not have any notion of housekeeping pickup rate, it simply delivers micropacket traffic whenever it arrives_.

8.18 Stupid Housekeeping Tricks (as seen on Letterman)

Some notes on doing the seeming impossible.

8.18.1 Housekeeping presented in the Science Telemetry Stream

Version 2.2 software has no direct method to place housekeeping records directly into the Science Telemetry stream. It is straightforward, however, to use the 73MRO command to route a memory dump record containing the housekeeping buffer in BIU memory to the science telemetry stream.

73MRO, LRP, TLM, CC40, 0

or

73MRO, LRP, TLM, CC40, CCF0

The first variation will present the fixed area of the housekeeping record along with half of the variable area of the housekeeping record.

The second variation will dump the entire housekeeping record.

Flight software version V2.4 changes the length of an MRO record to 192 bytes allowing the entire housekeeping buffer to be dumped using a single MRO record. This means the either of the above commands will result in the entire housekeeping buffer appearing in the science telemetry stream.

8.18.2 Housekeeping MRO decoding

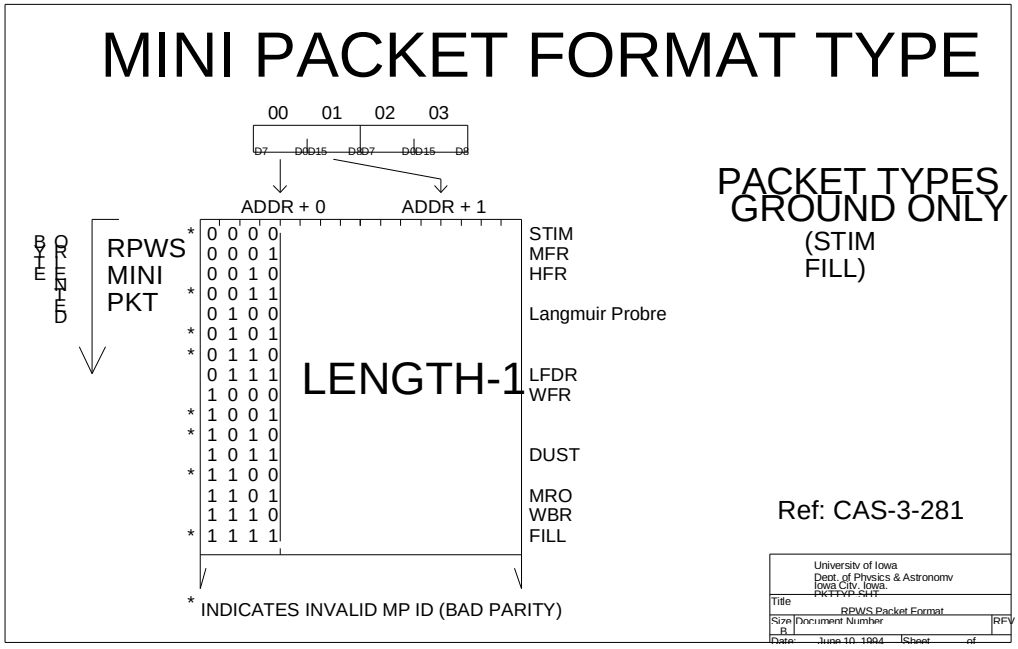
HKROM/**DSPHK** understand how to decode housekeeping buffers that are embedded within the science telemetry stream. All that is necessary is to redirect a minipacket file (either segmented or unsegmented) into **HKROM**/**DSPHK**. The housekeeping display program will search for MRO records from the LRP containing the correct memory address.

Note that the housekeeping buffer is 192 bytes in length and that MRO records are 128 bytes long (at least as of version 2.3 of the flight software). Also, the MRO record may begin on a 32 byte boundary. The decoding of the MRO records keeps an internal image of the 192 byte housekeeping buffer and updates the portion delivered by the MRO record. Each and every MRO record results in updating the housekeeping display (i.e. the display update is not very sophisticated). Although it may be expected that most values in the housekeeping display update slowly, the display method may cause unexpected updates in the housekeeping display.

Version V2.4 flight software changes the default MRO record size to 192 bytes to match the housekeeping buffer size. This keeps MRO packets within the 256 byte F5 buffer and eliminates some redundant bytes, reducing the telemetry bandwidth requirements when routing housekeeping through science telemetry (by sending MRO commands at regular intervals).

9 RPWS Science Telemetry

This section is intended to provide a brief discussion of the CCSDS ID's and mini packet format used to transport the instrument telemetry through the spacecraft to the ground.



9.1 RPWS Mini packets

The visible internal transport packaging entity is referred to as a mini-packet. The mini-packet is a stand-alone data set containing a segment of data from a particular receiver within the instrument. The basic format of the mini-packet does not handle data segmentation although several of the receivers has segmentation information in the body of their mini-packet to allow data sets to be delivered to the ground that exceed the limits of the 12 bit length field of a mini-packet.

The mini packet consists of a consistent header that contains a 4 bit identification field, a 12 bit length field and a 16 bit time field. Following these non changing header bits are the status and data generated by the receiver. The status field tends to be optimized for the data delivered by each receiver, containing only enough bytes of status to fully describe the hardware setup of the associated receiver.

Note the order of data items carefully as they are laid out to make the software within the instrument manageable. Also keep in mind that the instrument is a little-endian system connected to a word system using the 1553 bus and that byte order may not be as expected.

Mini Packet:

	BYTE:BIT	DESCRIPTION
	0 : 7..4	Data source
	0 : 3..0	Length MSB
	1 : 7..0	Length LSB (3 less than the total packet length)
	2 : 7..0	RTI of data, LSB
	3 : 7..0	RTI of data, MSB
	4..end	Receiver Status and Data

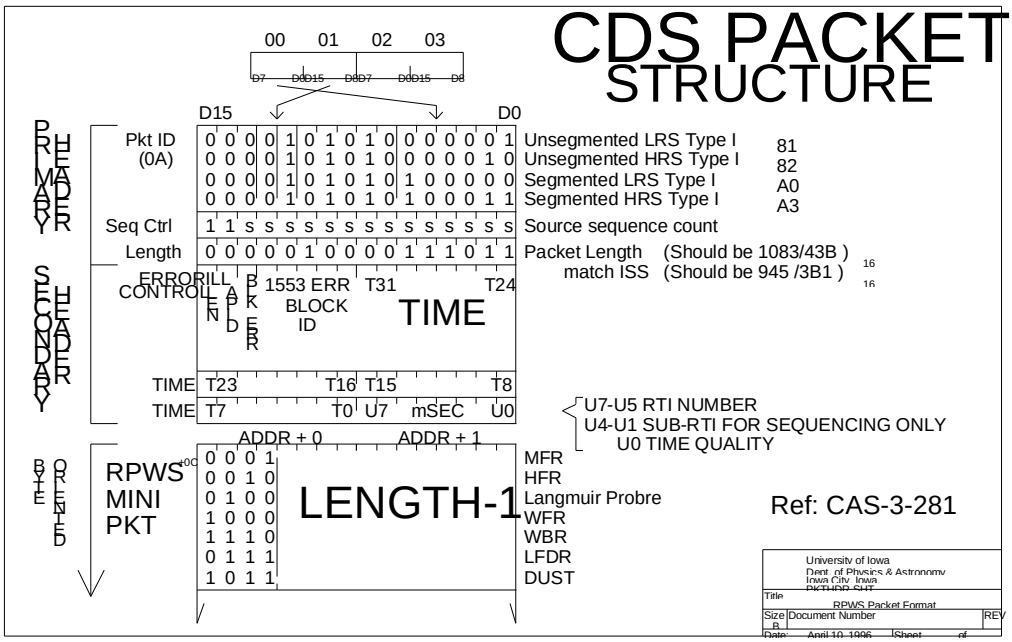
9.2 RPWS CCSDS packets

Having described the mini-packet, now take a step back and let us examine the transport mechanism used to ferry data from RPWS through the spacecraft to the ground. The fields will be described in essentially the same terms as in the JPL document describing spacecraft intercommunications.

The issue of byte order is the same throughout these discussions. The CCSDS headers are arranged in 8085 memory as 16 bit words. The time field becomes somewhat confusing as it is larger than a word and therefore appears in a rather peculiar order to the 8085.

CCSDS field in the science telemetry.

WORD	DESCRIPTION		VALUE	COMMENTS
0	CCSDS Header		0x0A81 0x0AA0 0x0A82 0x0AA3	Unsegmented Low Rate Segmented Low Rate unsegmented High Rate Segmented High Rate
1	Source Sequence Count		0xC000 to 0xFFFF	14 bit counter
2	Packet Length		0x3B1	always 952-7
3-5	Time & Error flags			big-endian across words. 40 bits of time 8 bit error cntl
6-475	Instrument data			mini-packets



RPWS operates 2 logical data channels, *Low Rate Science* and *High Rate Science*. Each of these channels may contain segmented and unsegmented data. An unsegmented record has a mini-packet that begins in word 6 that contains instrument status. A segmented record has a mini-packet that is continued from a previous CCSDS record. ***HRS*** and ***LRS*** streams are sequenced separately (keeping in mind that *Segmented* and *Unsegmented* are **not** considered unique telemetry streams)

Due to some of the streamlining within its formatting routine, the HRS stream tends to have a significant number of *unsegmented* CCSDS records. This is simply a result of a slightly different approach used to improve the rate at which data is formatted.

In some cases, the last several bytes of a CCSDS record may be zero filled. This occurs particularly in high-rate telemetry where the cost of fragmenting a the mini-packet header and status, in terms of CPU use, is excessive.

The RPWS minipacket is a standalone data set. Each minipacket begins with a status area that describes the configuration of the receiver at the time the data set was acquired. Note that a minipacket is limited, in size, to less then 4K data bytes so the handlers for those receivers that generate data sets in excess of this size decommutate and segment the data set where necessary. Segmented data sets will contain duplicate status information in each minipacket. Decommutated data sets may not contain complete information on the state of the receiver, but they do contain complete status for the data set (i.e. WFR in multi channel mode, each data set describes only the channel of data in the data set, ignoring the other channels).

9.2.1 CCSDS Packet Sequence

This section is effectively a review of the previous sections' discussion of the scheme used to sequence the CCSDS science telemetry. The S/C telemetry document lists 6 unique identification patterns for science telemetry. These 6 telemetry *types* may be further reduced to a 3 by 2 matrix using the following table:

Segmented Low Rate Science	Unsegmented Low Rate Science	HRS
Segmented High Rate Science	Unsegmented High Rate Science	LRS
Segmented Memory Read Out	Unsegmented Memory Read Out	MRO

There are three basic types of data defined; **Low Rate Science**, **High Rate Science**, and **Memory Read Out.**

Each of the these three *types* may be either **Segmented** records or **Unsegmented** records.

Each of the three basic types are independently sequenced because each of these three data types are logically independent data. Also note that with the initial software release, the **Memory Read Out** packet type is not used (and should not appear in any telemetry produced by the instrument).

The Segmented and Unsegmented data for each of the three basic types simply indicate that the minipacket data contained in the CCSDS packet is the start of a new minipacket (in the case of **Unsegmented**) of a continuation of a minipacket (in the case of **Segmented**).

Another way to make this a little more obvious is to go into a little more detail on how LRS and HRS are produced within the instrument:

LRS data is handled by one process on the LRP (i.e. the BIU handler) and makes use of a single buffer within the BIU. This process maintains a private sequence counter which is placed in the LRS packets. (You may also notice that the single buffer in BIU memory limits the peak LRS data rate to a single packet every other RTI period, or approximately 30Kbps).

HRS data is handled by one process on the HRP (i.e. BIU Direct) and makes use of two multi-packet buffers within the BIU. This process also maintains a private sequence counter which is placed in each HRS packet. In order to support the high data rate, each buffer provides space for six CCSDS packets.

MRO data is produced by one of three copies of the MEM_TWEAK handler (one copy on each of three processors). In the initial software release, the MRO data is treated almost like science instrument by the data formatting routines so the MRO data appears in the LRS telemetry stream as a unique type of mini-packet. Since the MRO data is embedded in the LRS data, no MRO packet type is produced (in other words no special MRO-CCSDS packet is produced).

The **segmented** / **unsegmented** indication within the CCSDS packet is simply an indication of where the RPWS mini-packet begins within the CCSDS packet. When a mini-packet begins at the beginning of a CCSDS packet, the packet is marked as an **unsegmented** packet . In most cases, however, a mini-packet will span the boundary of a CCSDS packet and the CCSDS packet will be marked as a **segmented** packet. **Segmented** / **unsegmented** does not affect sequencing, in other words, RPWS has defined three types of data **LRS**, **HRS**, and **MRO**. Current software produces two data types; **HRS** and **LRS**.

RPWS produces some LRS traffic when the receivers are powered, the minimum data rate being on the order of 30 to 50 bits/sec. LRS traffic may be produced at any rate up to the architectural limit of 30Kbps (although a more typical rate would be a few thousand bits / second). HRS traffic, when active, is produced at rates from 30Kpbs to 350Kpbs and will be mixed with LRS data.

9.2.1.1 LRS Sequencing

LRS Sequencing is performed on the LRP as all LRS traffic is directly handled by the LRP (i.e. It is impossible to sequence LRS anywhere else in the system).

9.2.1.2 HRS Sequencing

HRS sequencing may be performed on either LRP or HRP. All HRS data originates and is formatted on the HRP. Formatting on HRP includes generating and placing sequence numbers in the CCSDS records. In addition, the LRP has code that can sequence CCSDS records.

Up through Version 2.4 sequencing was performed on the LRP. This presented an uninterrupted sequence to CDS.

Version 2.5 disabled sequencing on the LRP to address an ISA filed early in 2001. Sequencing on LRP strips some information vital to reassembling mini-packets as well. With sequencing moved to HRP, data gaps will become more evident. When we sequenced on LRP. Data loss, within the instrument, during mode transitions, would be masked by the LRP sequencing. With the V2.5 load, the sequencing task is moved to HRP so CCSDS records that are discarded within the instrument will be evident.

9.3 CCSDS Timestamp

The timestamp placed in the CCSDS record is accurate to 125mS. No additional timing information can be extracted from the CCSDS timestamp as far as RPWS is concerned. What this means is that the lower 5 bits of the 40 bit time field are implicitly zero for all intents and purposes. Ground processing, however, associates the full 40 bit time (SCLK) with an actual event time (SCET).

Seconds	Sub-Seconds (i.e. approximately milliseconds * 4)							
	RTI number			HRS Packet Counter				Time Quality
D31..D00	R2	R1	R0	C3	C2	C1	C0	Q0

9.3.1 Seconds

This clock is a monotonically increasing clock with a nominal interval of 1 second. The reference epoch is nominally the 1958 astronomical epoch. Since the spacecraft does not have a perfect clock, corrections to the spacecraft clock are calculated and applied on the ground.

The seconds field, in its entirety is supplied on a regular basis by the spacecraft systems to the instrument. Time delivery is scheduled to occur during RTI-6 allowing the instrument to accurately establish time to within a single RTI interval of 125 mSec.

9.3.2 RTI

The RPWS processors and operating software keep track of time to an accuracy of 125 mSec (also referred to as the RTI interval). The time, accurate to 125mSec is placed in all of the CCSDS data products produced by the instrument (i.e. science and housekeeping telemetry packets).

Note that the processors do not have a mechanism, with the exception of the WBR receiver, to keep track of time with any more resolution than 125 mSec.

9.3.3 HRS Packet Count

As the instrument only keeps track of time to the granularity of the RTI interval, less significant bits in the time field are used for other purposes within the instrument.

Housekeeping telemetry has this field set to zero.

LRS science has this field set to zero.

HRS science places a formatting count in this field. As each CCSDS record is formatted, this field is incremented (starting from 1). **The net effect being that all science telemetry records have a unique time label**. One side effect of this is also that ground processing will calculate the SCET for all HRS records that does not correspond to the beginning of the RTI period.

9.3.4 DCP/HRP SCLK maintenance

DCP and HRP do not have a direct connection to the BIU and, therefore, do not have direct access to the regular time updates. We accommodate this limitation by sending two hardware signals to the DCP and HRP to allow the lower bits of the SCLK to be aligned without the need to have reliable updates using the IPC delivery mechanism.

This is accomplished primarily through the use of the RTI-0 signal that is generated on the LRP. RTI-0 is triggered on LRP by the BIU handler during RTI-6 or RTI-7. The hardware then asserts RTI-0 during RTI 0 by counting RTI edges (LRP does not need to have CPU cycles available during a critical period, only at some point in a preceding RTI period. HRP and DCP check the state of the RTI-0 signal in each RTI period and clear the lower bits of the SCLK to zero when RTI-0 is asserted. The upper bits may then be updated at any point prior to the next RTI-0 signal. Once an update occurs, the time is automatically advanced during each RTI period. As long as no RTI interrupts are missed, HRP and DCP should update their SCLK in a manner identical to LRP.

Bad time on DCP is of little consequence as there are no science data products that require a time-tag (STIM records are tagged, but a bad time is not particularly fatal).

Bad time of HRP, however, affects timetags for all of the dataproducts gathered on HRP. Unfortunately, we have seen occurrence of this on the spacecraft. The initial impression is that a RTI interrupt is lost (at the time this paragraph was added, we don't know if CDS, LRP, or HRP are to blame).

If CDS skips an RTI period, this behavior would be observed.

It is not clear if LRP can cause a problem as the RTI signal is not under software control and the RTI-0 signal is trigger by software but synchronized by hardware.

If HRP keeps interrupts disabled too long, thereby missing the SCLK update, this behavior would be observed.

What appears to happen is that as we are advancing time, the RTI-0 signal appears at the beginning of HRP's RTI 7. This causes the lower 16 bits of the 40 bit SCLK to be zeroed. Normally, the RTI-0 signal would be presented in RTI 0 and HRP would increment the current SCLK (which would match LRP) prior to testing the RTI-0 line and clearing the lower SCLK bits. If HRP is an RTI period behind LRP, the SCLK update is not performed (as it is not yet time), but the low order bits of SCLK are cleared. This results in HRP being 256 seconds late (in other words, the SCLK is turned back 256 seconds).

Normally this would be a short term problem as the LRP sends a time update message to correct the upper 24 bits of SCLK 9 RTI periods following RTI-0. If the IPC traffic is not lost, this update will bring the upper bits of SCLK forward to the correct time. When there is HRS traffic, however, we occasionally drop the SCLK update message. In most cases this isn't a problem as HRP and LRP are marching time in lockstep. If, however, HRP is behind and the update is lost we will spend 257.125 seconds with HRP tagging data with bad time information.

Rectification

We can, potentially, mitigate the effects by changing the meaning of the signal from RTI-0 to RTI-1. The ACTEL hardware that generates the RTI-0 signal can be triggered in either RTI-6 or RTI-7. Current software triggers the RTI-0 signal during RTI 7. The hardware is triggered by a port write and asserts the RTI-0 signal at the next RTI pulse from BIU. The hardware is also capable of asserting the RTI-0 signal on the 2nd RTI signal following the port write (i.e. During RTI 6). If we change the meaning to RTI-1 and trigger the RTI-0 on the 2nd. RTI (but during RTI 7) we will see the RTI-0 signal during RTI 1.

In addition, the handler on DCP and HRP will need to write a 0x20 to the low order SCLK field (i.e. This is the bit pattern that indicates RTI 1, the RTI number is the upper 3 bits of this field) when the RTI-0 signal is active (rather than the 0x00 that is currently written).

This does not completely clear the problem, but does allow for a 1 RTI slip without causing corruption of SCLK.

Cause of the Problem

It looks like (prior to tests) that biuint_5 (or some other process) is keeping interrupts disabled for long periods of time. Haven't yet looked to see if this is peculiar to a particular data movement mode.

9.3.5 Time Quality Flag

The time quality flag is used to indicate that a time update has been delivered from the spacecraft. In the event that S/C fails to deliver time, the instrument software will increment the RTI field during the RTI-7/RTI-0 interrupt and set the *time quality flag*. A zero indicating that S/C updated the time (as expected) and a 1 indicating the time update did not occur.

Note that on the DCP and HRP no time updates from the S/C occur so this bit is, as expected, always set to a one.

9.3.6 Time calculations using CHDO information

When using the SCLK/SCET pairs supplied in the *CHDO type 94* record keep in mind that the SCET may not correspond with the start of the CCSDS record. In other words, one must use all 40 bits of the SCLK found in the SCLK/SCET data in the *CHDO type 94* record but only the top 35 bits of the SCLK (with the lower 5 bits set to zero) that is derived from the CCSDS packet or mini-packet.

The following calculations represent the steps take to derive an spacecraft *event time of interest*, such as the start time of an MFR observation. Assume for the purpose of this discussion that we have derived the 40 bit SCLK of the event from the 40 bit CCSDS time tag and the 16 bit MFR minipacket timetag.

- OFFSET-TIME40 = CHDO-SCLK40 - (MP-SCLK40 & 0xFFFFFFE0)
- MP_SCETmS = CHDO-SCETmS - (OFFSET-TIME40 * 3.906)

In the first calculation, we determine the offset, expressed in 3.906 mSec units (1000/256 or 1 second expressed in 8 bits) from the known time (CHDO) to the time of interest (MP). Note that the instrument can only resolve to the nearest RTI, so the lower 5 bits are suppressed.

Now we know the time difference, in 4mS steps, between a known time and the desired time. The offset may now be subtracted from the known time (CHDO) to obtain the desired time (MP). SCET in the CHDO record is expressed in days/milliseconds so it becomes a simple matter to come up with the desired time and adjust for any rollover in the milliseconds field.

This all assumes, of course, that the ground system at JPL is including correct SCLK/SCET pairs with the data. We also operate under the assumption that *the CHDO type 94 record* is relatively recent. The spacecraft makes use of a stable oscillator, rather than a precision standard, so there is some clock drift. As long as the CHDO tag appears in the data within several minutes of the time of interest, any errors introduced by this simple method are below the uncertainty within the instrument.

Also, bear in mind that we are accounting for a difference in the way RPWS treats the sub-second portion of the time field and the way JPL treats it. For the most part, all RPWS scheduling is based on (or begins at) the beginning of the RTI. Only WBR provides a mechanism to schedule at some point other than the beginning of the RTI (and this is only used at very high data rates, those in excess of 150,000 bits/second).

9.4 Time relationships

The point at which the CCSDS time stamp is applied is subtlety different between the LRS and the HRS data. This difference arises due to the impact HRS has on HRP resources (i.e. CPU).

The minipacket time field consists of 13 bits of second and 3 bits of RTI. This allows accurate timing of minipackets that linger within the instrument for periods over 2 hours. Any data production rate above a few bits per seconds should be able to accurately attach time tags.

Note, however, that starting the instrument may flush stale mini packets through the system. The BIU handler does not attempt to discard data or deliver partial CCSDS packets (i.e. the BIU handler does not zero fill and deliver data left in BIU memory).

9.4.1 LRS timestamp

The LRS timestamp (i.e. the 32 bit seconds field in the CCSDS header) is applied when the CCSDS record is marked for delivery to the spacecraft.

As a result, the CCSDS time field is always recorded following the timestamp in the minipacket. In some cases the time stamps may be the same, but the CCSDS time never occurs after the minipacket time stamp.

Since the minipacket time is always before the CCSDS time, one may simply decrement the CCSDS seconds field until the 13 overlapping seconds bits match to arrive at the time a minipacket was acquired.

9.4.2 HRS timestamp

The HRS timestamp is applied sequentially as the formatting process places data into the CCSDS packet buffer.

In other words, the timestamp in the CCSDS record is recorded when the first data bytes are moved into the CCSDS buffer. This difference should result in a CCSDS timestamp that is, at most, a few seconds newer than the second minipacket in the CCSDS buffer as HRS is expected to be used when data production rates are in excess of what the LRS architecture is capable of delivering.

The approach to resolve this potential problem within the GSE software is to allow up to 15 seconds of overlap in the time stamp.

As with the LRS, stale data may be expected when the instrument is started following a period of inactivity.

This method of applying a timestamp to the data allows the formatting software to proceed in a serial fashion, placing the timestamp into the CDS buffer at a convenient time.

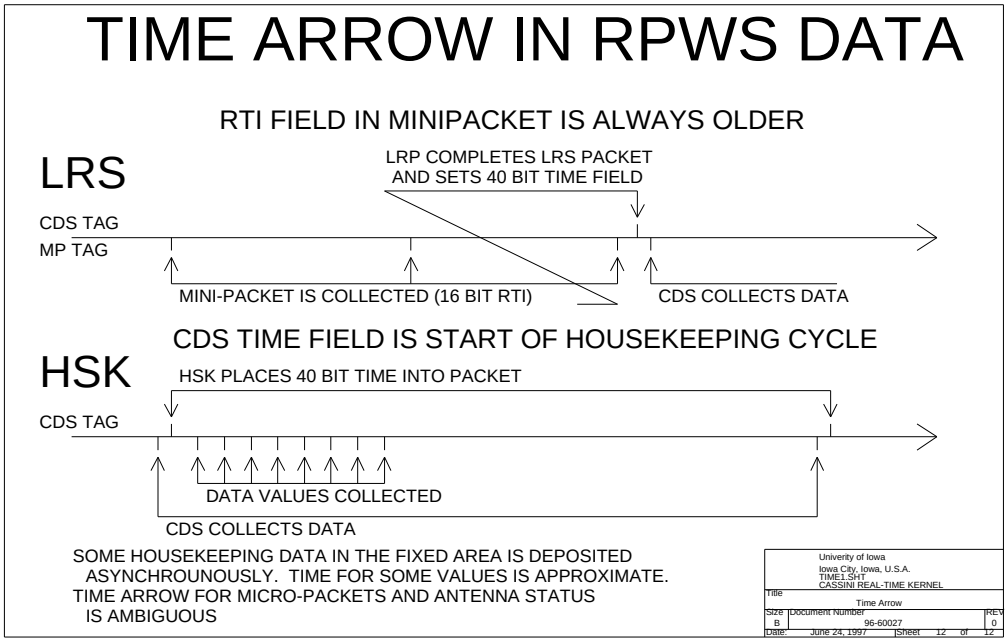
This is not expected to be a problem as the high rate telemetry is not required when the instrument is operating at lower bit rates, no packets are expected to remain on the HRP for more than a few seconds.

9.4.3 Timestamp Strategy

Given the small ambiguity in the HRS timestamp it appears that one should not operate the instrument using the HRS delivery mechanism at low data rates. LRS is architecturally limited to approximately 30 Kb/sec and that is the minimum pickup rate at which the BIU handler will allow HRS to operate. A reasonable breakover point is probably in the 8Kb/sec range. (One must differentiate between the pickup rate that the BIU is operating at and the data delivery rate that the instrument handlers are operating at).

The time delta between data sets is effectively the most time that can normally elapse between data production and application of the CCSDS time stamp.

The GSE software that calculates the full event time assumes that the scheduling delta never exceeds 15 seconds.

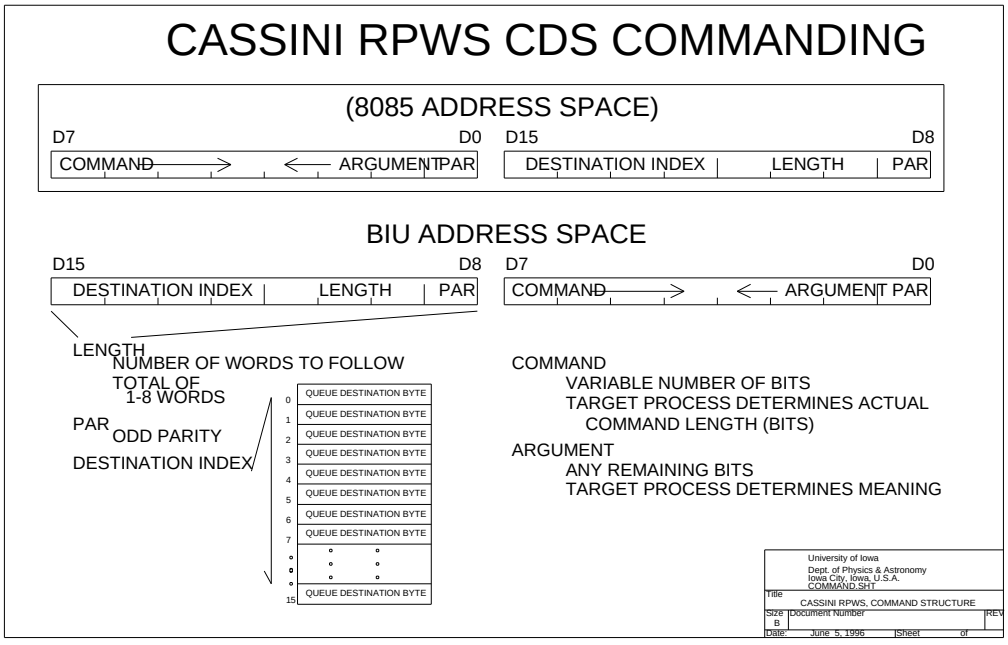


10 RPWS Commands

These are the commands the RPWS instrument is capable of recognizing. Commands listed in CAS 3-291 start with *73*, the identification for RPWS. Commands starting with *00* are undocumented and intended to be issued only at the bench level or internal to the instrument.

Most of the commands are available only when science software is loaded (unless otherwise indicated).

All of the commands contain parity of one type or another. ALF commands are odd parity across the 1st. 16 bits of the command while all other commands are even parity across the 1st. word (more specifically odd parity across the 1st. 2 bytes of the command). The ROM code can process only the 3 types of ALF commands, so there are only three 16 bit patterns that the ROM considers acceptable. The command decoder used in the downloaded software checks parity on the first command word and will reject any ALF records that arrive (incorrect parity).



CASSINI RPWS CDS COMMANDING
COMMAND DECODE BY SOFTWARE LOAD

		ROM	ANTENNA	SCIENCE	MAINT
STIM/DISC	0	IGNORE	IGNORE	ACCEPT	IGNORE
DEPLOY	1	IGNORE	ACCEPT	IGNORE	IGNORE
ALF	2	ACCEPT	IGNORE/I	IGNORE/I	IGNORE/I
TWEAK/MRO	3	IGNORE	ACCEPT	ACCEPT	ACCEPT
IEB	4	IGNORE	IGNORE	ACCEPT	IGNORE
POWER	5	IGNORE	IGNORE	ACCEPT	IGNORE
BIU HANDLER	6	IGNORE	ACCEPT	ACCEPT	IGNORE
DCP	7	IGNORE	IGNORE	ACCEPT	IGNORE
MFR	8	IGNORE	IGNORE	ACCEPT	IGNORE
HFR	9	IGNORE	IGNORE	ACCEPT	ACCEPT
LP	10	IGNORE	IGNORE	ACCEPT	ACCEPT
WFR	11	IGNORE	IGNORE	ACCEPT	IGNORE
WBR	12	IGNORE	IGNORE	ACCEPT	IGNORE
LFDR	13	IGNORE	IGNORE	ACCEPT	IGNORE
LFDR (DCP)	14	IGNORE	IGNORE	ACCEPT	IGNORE
DUST	15	IGNORE	IGNORE	ACCEPT	IGNORE
MAINT		HONOR	UNIMPL	UNIMPL	HONOR

ACCEPT
COMMAND IS ACCEPTED AND PROCESSED

IGNORE
COMMAND IS IGNORED (NO PLAN TO IMPLEMENT) WILL NOT CAUSE PROBLEMS IF ISSUED

IGNORE/I: Invalid

UNIMPL
FUNCTION IS CURRENTLY UNIMPLEMENTED (FUTURE RELEASE) WILL NOT CAUSE PROBLEMS IF ISSUED

NO
COMMAND WILL CAUSE SOFTWARE TO CRASH

HONOR
SETTING THIS BIT WILL RESULT IN DOCUMENTED ACTION

University of Iowa
Dept. of Physics & Astronomy
Iowa City, Iowa, U.S.A.
COMMAND2.SHT

Title: CASSINI RPWS, COMMAND STRUCTURE

Size: B | Document Number | REV

Date: June 20, 1997 | Sheet of

10.1 ALF

ROM ONLY

Assisted load format. These are the 3 commands used to define the format of memory downloads presented to the RPWS instrument. The ALF format provides several levels of error checking to allow a valid download to be verified before attempting to execute code.

Note that ALF records have invalid parity and will be ignored by the ***DEPLOY LOAD*** and the ***SCIENCE LOAD.*** If ALF records are delivered when the instrument is operating from RAM they should be ignored and flagged as invalid commands. The ALF mechanism is not useful for performing patches to memory.

The ROM code, in turn, is capable of processing only ALF records and will ignore any other commands that arrive (treating them as invalid ALF records).

All of the fields are documented in JPL documents with the exception of the ***segment field.*** This field is used to place the 16 word memory image into the appropriate memory location. The lower bits of the field are used as the upper address bits to locate the memory image. The upper 4 bits select any combination of processor and bulk memory allowing common code to be loaded into all processors using a single ALF record.

The segment field is defined as follows:

BIT	FUNCITON		BIT	FUNCTION
D15	DCP Select		D12	BULK Select
D14	HRP Select		D11	BANK Select
D13	LRP Select		D10-D0	Address A15..A5

10.1.0.1 DCP Select

This bit, when set causes the 16 word memory image to be loaded into the Data Compression Processor.

10.1.0.2 HRP Select

This bit, when set causes the 16 word memory image to be loaded into the High Rate Processor.

10.1.0.3 LRP Select

This bit, when set causes the 16 word memory image to be loaded into the Low Rate Processor.

10.1.0.4 Bulk Select

This bit, when set causes the 16 word memory image to be loaded into the Bulk Memory on the High Rate Processor.

10.1.0.5 Bank Select

This bit may be used to control the bank select hardware on the LRP and HRP. The flight ROM's delivered at the time this document was written do NOT allow code to be downloaded into banked memory.

10.1.0.6 A15

The upper bit of the address. The flight ROM's delivered at the time this document was written do NOT allow code to be downloaded into upper half of memory.

10.1.0.7 A14..A5

The upper address bits used to locate the ALF record in memory.

10.1.1 73ALF, sequence, segment, 16 x data, 2 x checksum ROM ONLY

32 byte memory line. Naturally aligned (32 byte boundary). Sequence is monotonically increasing. Segment contains explicit addressing (data words have implicit addressing by position). Checksum is a dual 16 bit checksum (over every other word).

Note that the description of the *segment* field in the 3-291 document is misleading/inaccurate. Although the segment field is used to generate the address within processor memory, it is not a simple calculation (in other words, note that the segment field is not simply composed of upper address bits).

10.1.2 73ALF_END, sequence, count, 2 x checksum ROM ONLY

Indicates that the memory load is complete and execution of the downloaded image may begin. The sequence field and checksum are verified and must be correct for the ALF END to be recognized and processed.

The count field is not used. The assumption here being that if the sequence field is correct throughout the load, there are no problems.

10.1.3 73ALF_SKIP, sequence, count, 2 x checksum ROM ONLY

Indicates that a break in the sequence will occur to skip over a bad storage area in the SSR.

This must **NOT** occur as the first record in a download!

The previous segment field encountered is used to determine if this record is to be placed in bulk memory. The SKIP record must be placed in bulk memory in order for the internal download to occur so skip records must be counted when calculating bulk memory use.

10.1.4 00ALF_RESET ROM ONLY

This command is simply a special case of 73ALF that is used to initialize the sequence checking when performing a download. This record must be the first record of a download and the segment field should have the upper 3 bits of the ***segment field*** cleared.

The 16 data words of this record may contain any desired data pattern (such as version control information). The ROM will discard the contents of this type of data record.

10.1.4.1 73ALF, sequence, 4096, 16 x data, 2 x checksum

This indicates that the record is not to be loaded into any processor memory; being treated as a special case that saves the sequence field.

The ***bulk memory*** select bit must be set to match the rest of the load. In other words, this record must be present in ***bulk memory*** in order for an internal reload to occur.

10.1.4.2 73ALF, sequence, 0, 16 x data, 2 x checksum

This, again, is a special case of the 73ALF command. Note that the segment field is set to zero. This is how the reset appears at the beginning of a load that will not be placed into ***Bulk Memory***. This would be the case at the beginning of the ***Deploy Load***.

10.1.5 6EXT_MEM_LOAD, ssr, load type, device

CDS command used to download RPWS instrument memory. This command is listed here as a convenience to the instrument user.

ssr	Description
SSR_A	Use SSR A
SSR_B	Use SSR B
CONNECTED	Use currently active SSR

In most cases, it seems like **CONNECTED** is probably the optimal choice here. CDS may have either A or B active (the inactive SSR may be off-line and therefore inaccessible) and using **CONNECTED** will allow CDS to make the correct choice.

load type	Description
DEFAULT	Select the default SSR partition this should contain the SCIENCE LOAD
NON_DEFAULT	Select the **non**-default SSR partition would be used to hold updated flight software prior to final acceptance.

device	Description
RPWS	Selects RPWS instrument load
CAPS, CDA, CIRS, INMS, NAC, WAC, MAG, MIMI, RADAR, UVIS, VIMS	Selects the other guy, not of interest to RPWS

10.2 Antenna Control

RESTRICTED COMMANDS

Antenna Movement Commands. These commands are now listed as RED FLAG items and will not be processed by ground software following successful deploy on 25 October 1997. These commands were processed only when the deploy software is loaded into the processor. Additional hints and information may be found in the *Release Description Document* for the deploy software.

Although the antenna control commands are processed when the instrument is in SLEEP mode, it is suggested that commands be issued with care (i.e. verified before additional antenna commands are issued).

These commands are harmless when operating from ROM or when science software is loaded into the instrument. At some time prior to Venus-1 the deploy software will be overwritten when the maintenance software is loaded on to the spacecraft, rendering these antenna commands ineffective (i.e. no chance of an antenna command causing movement).

10.2.1 73ANT_HALT Restricted Command

10.2.2 00ANT_HALT Restricted Command

These commands cause power to be removed from the antenna mechanism. The commands are identical in function with the internal form being reserved for use by the monitoring software. This makes the source of the command simple to determine if the antenna element deploy operation terminates prematurely.

command	hexadecimal pattern reflected in housekeeping
73ANT_HALT	0x1080
00ANT_HALT	0x1090

10.2.3 73ANT_CNTL, direction, antenna Restricted Command

Basic antenna movement command. Selects one of the elements and the direction to move. In order for the element to move, the deploy electronics must be powered, the appropriate BIU discrete bit must be enabled, and the element must be at the end of its travel (i.e. against the corresponding limit switch).

direction	Description
out	extend the antenna element
in	retract the antenna element

antenna	Description
ex_plus	dipole element EX+
ex_minus	dipole element EX-
ez_plus	monopole EZ

10.2.4 00ANT_CONT, direction, antenna Restricted Command

Allows a deploy operation that terminated early to be resumed without performing the initial 1 second retract. As this command is not documented in the ***3-291*** command document, it must be ***wrapped.*** This command is intended to allow recovery from a deploy operation that terminated early.

10.2.5 00ANT_CNTL, direction, antenna, 2 x I lim, 2 x pot lim, deploy tmr Restricted Command

Adds alteration of the static limits checking to the ***73ANT_CNTL*** command. The preferred method to change limit values is through the ***00ANT_SETUP*** command.

direction	Description
out	extend the antenna element
in	retract the antenna element

antenna	Description
ex_plus	dipole element EX+
ex_minus	dipole element EX-
ez_plus	monopole EZ

I lim	Description
nn	current limit

pot lim	Description
nn	position pot limit

timer	Description
nn	deploy timer expressed in RTI periods

10.2.6 00ANT_SETUP, antenna, 2 x I lim, 2 x pot lim, deploy timer Restricted Command

Alters the static limits checking fields for the selected antenna. This command has been used during ground testing to alter the deploy timer (requires ***73WRAP***).

In the following examples *iiII* represent the min/max current levels and *ppPP* represent the min/max position values. Setting them to zero effectively disables the static check during the antenna operation.

EX+ 73WRAP, (1643, iiII, ppPP, tttt)
EX- 73WRAP, (1645, iiII, ppPP, tttt)
EZ 73WRAP, (1649, iiII, ppPP, tttt)

10.2.7 00ANT_STATUS, antenna DEPLOY ONLY

Causes the static limits table for the selected antenna to appear in the housekeeping data.

antenna	Description
ex_plus	dipole element EX+
ex_minus	dipole element EX-
ez_plus	monopole EZ

10.2.8 73mem_tweak, LRP, BYTE, 0x60, 0xNN, ANT_ Restricted Command

Limit switch mask. This tweak may be used to alter the limit switch exclusive-or mask from it's default value of 0x3F.

10.3 BIU Control

1553 Bus Interface Unit control.

10.3.1 73BIU_TLM_CNTL, command, parameter 1, parameter 2

command	Description

parameter 1	Description

parameter 2	Description

10.3.2 73MEM_TWEAK, LRP, BYTE, 0x32, nn, BIUH

High Rate Science enable.

nn	Description
0x00	Disable HRS
0xff	Enable HRS

10.3.3 73MEM_TWEAK, LRP, BYTE, 0x33, nn, BIUH

High Rate Science Sequencing control. CCSDS records are sequenced with a monotonically increasing 14 bit sequence number. When sequenced on LRP, visibility into the loss of records from HRP to LRP is lost. HRP always sequences CCSDS traffic and the LRS sequencing may be suppressed or enabled.

Loss of sequence from HRP causes mini-packet reassembly to fail in odd way, so starting with Version 2.5 the sequencing on LRP was suppressed by default.

Version 2.4 and prior.

nn	Description
0x00	LRP sequences HRS (default)
0xff	HRP sequences HRS

Version 2.5 and later.

nn	Description
0x00	HRP sequences HRS (default)
0xff	LRP sequences HRS

10.3.4 00MEM_TWEAK, LRP, Word, 0x10C6, nn

Sets the target BIU DIRECT buffer size. Values for various pickup rates are listed below (we are interested only in the size of the buffer, not if the pickup is every RTI or every other RTI).

This number is located in the system data page rather than in any specific process. Since this value is not stored in a process, the absolute memory tweak command is required.

nn	**Description**
0x03B8	1 Packet per RTI
0x0770	2 Packets per RTI
0x0B28	3 Packets per RTI
0x0EE0	4 Packets per RTI
0x1298	5 Packets per RTI
0x1650	6 Packets per RTI

10.4 DCP data compression

Data compression processor control.

This command is used to control the data compression routines running on the DCP.

Please keep in mind that the DCP makes use of a 1.5Mhz 8085 for processing *power*. L/P and WFR are both capable of overloading the DCP. In most setups, the DCP is not heavily taxed (WFR bit rate is nominally 360 bits/second and the L/P is less than 100 b/s). When building modes that approach several thousand bits/second it is not unusual to experience transient overloads on the DCP. If the rate is simply too high data will be lost, as one might expect. If the overload is minimal, you may expect to see occasional lost data sets. Bench testing may provide a sufficient indication that the flight model will handle a given load.

If the bench model RTI timing is off by just a few milliseconds, the resulting data production rate is lower (as it is based on the RTI timing) and the corresponding CPU cycles available on the DCP are increased).

Note that the flight software revision 2.3 added a capability to the IPC driver in the handling of free space. Prior to the 2.3 release, the DCP could deadlock when using the IPC handler/ mini-packet assembler to deliver a completed (i.e. compressed) data set to LRP. This could occur when incoming data exhausted the F5 queue. The 2.3 release changes the minipacket assembler to allow it to conditionally request free space and abort delivery in the event that the F5 is exhausted. The abort results in a partial minipacket being delivered to LRP (where it is discarded), and the calling process being un-blocked so that it can process the next data set. Although this prevents the deadlock (which, effectively kills the DCP), a minipacket segment is lost (this being preferable to a non-functioning DCP).

Earth Encounter Notes

During Earth encounter, a trigger to exercise the L/P was tested on the bench successfully. The trigger performed a set of 10 sweep/density cycles at 1 minute intervals. The first 5 (or so) sweeps were successfully processed on the DCP (16 to 12 bit packing only, to achieve peak throughput). The remaining data sets were missing a minipacket segment, indicating that DCP had become overloaded and was dropping data (missing a complete segment indicates either DCP or LRP were getting behind). The DCP is indicated as the CPU usage meter on the LRP didn't indicate an overload and the combined L/P and WFR bit rates were well within the capability of the LRP.

The theory being that the bench model runs slightly slower than the flight unit (Heurikon 68000 based RTIU without RTI interrupt hardware), allowing the DCP slightly more time to process a slightly slower data rate.

10.4.1 73DCP_CNTL, source, method, size, backup, parameter 1, parameter 2

source	Description
wfr	setup compression scheme for the wfr
lp_sweep	setup compression scheme for Langmuir Probe sweeps
lp_density	setup compression scheme for Langmuir Probe density measurements
lp_lock	setup compression scheme for Langmuir Probe lock

method	Description
walsh	
walsh-rice	
packed	
none	suspend compression for this source

size	Description
0..12	walsh algorithm data word size

backup	Description
compress	algorithm selects for maximum compression
speed	algorithm selects for maximum speed

parameter 1	Description
0..1023	

parameter 2	Description
0..32767	

10.4.2 73MEM_TWEAK, DCP, WORD, 0x60, nn, CMPX cycle delay

This delay sets the number of RTI periods that the compression process will idle between minipackets. The minimum effective value is 1 (i.e. loading a zero in this field will result in a delay until the next RTI interrupt.

10.4.3 73MEM_TWEAK, DCP, WORD, 0x62, nn, CMPX minipacket control flags

This field controls some features of the mini-packet assembler for data delivered from the DCP to the LRP.

01 Checksum enable

Setting this bit causes a checksum of the IPC packet to be generated before delivery to the IX queue.

12 Non Blocking

Setting this bit will cause MiniPKT processing to abort in the event that no free space is available. This bit is required to prevent the compression process from deadlocking when free space is overrun.

24 Fast Delivery

Setting this bit eliminates the 1 RTI delay between successive IPC packets.

10.5 Dust Detection Control

Dust detection data acquisition and analysis commands. Note that the compression switch is present in two separate commands and this can present an inconsistency in building the commands. Please refer to the *suggested order list* when building **DUST** commands to avoid placing the instrument into an invalid state.

Commands may be sent to the DUST control process when the instrument is in *SLEEP* mode although they will not be immediately processed. It should not be possible to bring the instrument out of SLEEP mode inadvertently (i.e. modifications to offset 0x56).

DUST is a derivative instrument that makes use of the WBR to obtain sample data that is analyzed within the instrument. There are some potential problems when operating both the WBR and DUST concurrently. The WBR section contains additional notes on interactions between the two logical 8 bit instruments.

Timing control of the dust acquisition may be setup to force synchronous acquisition in spite of the fact that data acquisition is controlled by the number of spare CPU cycles available on DCP. This is accomplished by performing a synchronous poll for data acquisition commands. The timing control words at offset 0x50 and 0x52 are used to schedule polling for DUST commands from the DCP.

Note that dust operates in a manner different than that of WBR/WFR/LFDR in that the analysis is performed on DCP. The analysis process on DCP operates at a low priority so the data acquisition and analysis are closely coupled (a new data set is explicitly requested as each analysis is completed). Pay particular attention to the memory tweaks that control timing on the data acquisition process for insight on meeting scheduling needs.

Version V2.6 adds a burst scheduling mode to WBR that can affect DUST data acquisition. This burst mode allows a programmable number of WBR data sets to be acquired with WBR automatically idling. To allow concurrent DUST activities with AGC active, the WBR scheduling mode can force DUST to suppress data collection during the WBR burst.

Suggested order of commands:

73DUST_AUTO_CNTL

00DUST_DATA_CNTL

73DUST_GAIN_CNTL

73DUST_MODE_CNTL, HOLD

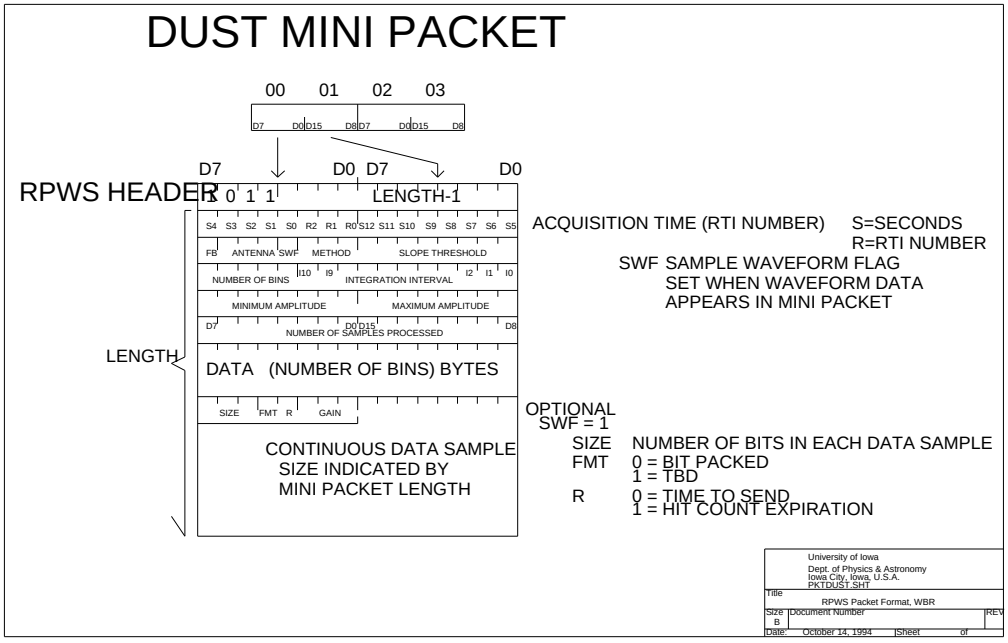
73DUST_CMPRS_CNTL

73MEM_TWEAK offset 0x50

73MEM_TWEAK offset 0x52

73MEM_TWEAK offset 0x54

73MEM_TWEAK offset 0x56



10.5.1 DUST Timing Control Process: Setup Storage Area

OFFSET	Label	Default	Description
0x2590	CMDp_Divisor	0x0008	MOD timing divisor
0x2592	CMDp_Remainder	0x0004	MOD timing remainder
0x2594	CMDp_Packet_Delay	0x0008	LRS inter packet delay (Not used for HRS)
0x2596	CMDp_Run	0x01	Run Flag
0x2597	CMDp_AGC	0x00	
0x2598	CMDp_Offset	0x0990	8254 timer, offset from RTI to start of data capture
0x259A	CMDp_Mode	0x0000	
0x259C	CMDp_Band	0x0000	Band Select
0x259E	CMDp_Length	0x0400	Dataset byte count (8237 word count)
0x25A0	CMDp_Destination	0x49 0x05	Data Routing Destination
0x25A2		0x0000	
0x25A4	CMDp_AGC_H_L	0x1C 0x32	AGC set points
0x25A6	CMDp_Antenna	0x02	Antenna Selection
0x25A7	CMDp_Gain	0x38	Gain Setting (D5..D3)
0x25A8	CMDp_AGC_Flag	0xFF	AGC Enable
0x25A9		0x00	
0x25AA		0x0000	
0x25AC	CMDp_DCC	0x4E	ACTEL register 43 Pattern
0x25AE			

10.5.2 DCP resource requirements

Discussion of the CPU resource requirements to perform the dust analysis.

DCP CPU requirements	Description
N/A	RAW (bypass DCP)
N/A	PACK
N/A	RICE
N/A	WALSH
N/A	WALSH/RICE
.125 seconds ??? 512 sample packet	DUST analysis

10.5.3 73DUST_AUTO_CNTL, low set, high set, average interval, time constant

This command is delivered to the acquisition process. Automatic Gain Control settings. The preferred high set point for dust is lower than that for WBR data. While the rule of thumb for the WBR is to allow a factor of three (or about 10 dB) of headroom, the DUST detection algorithm prefers about 20 dB of headroom. This helps eliminate some of the spurious events (e.g., antenna switching transients) which can be mistaken for dust hits. This means for the high set point a value of 36 decimal would be optimum, although this would usually prevent the 70-dB gain setting from being used. If more sensitivity is desired at the risk of picking up false hits, then the value of 77 decimal should be used. Here the low-band is assumed to be the DUST detection setting. For the low set point, one should remember that it must result in a window which is wider than the 10-dB step size of the gain amp. To give a window which is about 14 dB wide, the low set point should be 30 for the 10-dB headroom case, and 27 for the 20-dB headroom case.

Low Set		**High Set**		**Description**
EM	**Flight**	**EM**	**Flight**	
	27		77	Dust, 10dB Headroom
	28		50	C35 value
	30		36	Dust, 20 dB headroom

10.5.4 73DUST_CMPRS_CNTL, OFF, enable, word count

This command is delivered to the acquisition process. Controls size of data set delivered to dust analysis code. This command mimics the ***73WBR_CMPRS_CNTL*** command although the compression should never be enabled. This is the command used to specify data set size. The route field may be specified with either legal value as it has no effect when compression is not used.

Enable	Description
off	hardware compression not used
on	hardware compression enabled but you won't be able to perform the dust analysis (WBR toggle mode!)

route	Description
wc_out	EOP generated by WCR in 8237 only
wc_in	EOP generated by WCR in Actel or 8237

word count	Description
nn	number of samples in data set

10.5.5 00DUST_DATA_CNTL, destination

This command is delivered to the acquisition process. Controls routing of the raw 8 bit samples. During operations there is no reason to alter data routing. During testing it is useful, at time, to route the data directly to the Low Rate Science telemetry stream to bypass the analysis code.

destination	Description
lrs	WBR data delivered to Low Rate Science stream on Low Rate Processor (appears as WBR data)
hrs	DUST Data delivered to High Rate Science stream on High Rate Processor (appears as WBR data)
dcc_lrs	DUST data delivered to DCP for impact analysis

10.5.6 73DUST_DET_ANAL, method, parameter 1, parameter select, parameter 2

This command is delivered to the analysis process. Control the dust analysis algorithm. Currently this command does nothing; to command the DUST detection process, use **73MEM_TWEAK** as detailed in section 10.5.7.

method	Description
gallagher	
null	

parameter 1	Description
0..1023	

parameter select	Description
threshld	
bins	
amp_rnge	
intg_intvl	
waveform	
samples	
bitsize	
spr12	spare
spr13	spare
spr14	spare
spr15	spare

parameter 2	Description
0..2047	

10.5.7 73DUST_GAIN_CNTL, enable, gain select

This command is delivered to the acquisition process. Sets the digital gain control.

enable	Description
man	manual gain control
auto	automatic gain control

gain select	Description
0, 10, 20, 30, 40, 50,60, 70	gain level

10.5.8 73DUST_MODE_CNTL, HOLD, band, antenna select

This command is delivered to the acquisition process. Band and antenna select. The format of this command was defined early in the development process with subsequent software architecture rendering the ***trigger field*** useless and it should always contain the keyword **HOLD**.

band	Description
lband	36 uSec Sample period
hband	4.5 uSec sample period

antenna select	Description
ex	electric dipole, X
ez	electric monopole, Z
bx	search coil, X axis
hf	HFR downconvertor
lp	langmuir probe

The antenna select field is also stored in the W08I process when this command is issued. The selection may be overridden with either a memory tweak or a **73WBR_MODE_CNTL** command. This setting(i.e. Within the W08I process) impacts the way AGC behaves. The antenna selection is saved for both WBR and DUST as we seldom use both of these at the same time.

73MEM_TWEAK, HRP, BYTE, 0x52, 0x00, W08I

The setting of this field is critical as it affects other receivers when improperly set. In particular, the WFR will show switching transients when operating in low band due to the length of time required to acquire a dataset (i.e. several seconds will usually cover several dust acquisitions). This isn't usually evident in the timing plots as DUST delivers data to the ground at such a low data rate (i.e. DUST acquires many datasets prior to sending a data packet to the ground).

10.5.9 73DUST_MODE_CNTL, TRIGGER, na, na

Trigger a dust data capture. This form of the **DUST_MODE_CNTL** command would not be issued from the ground. It is used internally by the Dust Analysis Process on the DCP to obtain additional data for analysis. The band and antenna select fields are, effectively, ignored and would be set to zero.

This form of the command does not alter either the band or antenna selection.

The acquisition control byte, located at offset 0x56 from the DSTC process, must be set to *Idle* (a value of 1) for this command to be effective.

The following items control the dust data acquisition activity on HRP.

Dust normally operates with the acquisition process waiting in an idle state and the analysis routine requesting data sets as CPU cycles on the DCP become available. The dust acquisition idle state makes use of a MOD function during the delay time to allow acquisitions to be synchronized, if needed. Set locations 0x50 and 0x52 to the desired schedule, keeping in mind that the actual data capture occurs at the start of the next RTI period. Although this does **not** imply that dust captures occur every cycle, it will cause captures that are requested by the analysis process on the DCP to occur on the schedule.

This scheduling is always followed, so some reasonable value needs to appear in location 50/52.

10.5.10 73MEM_TWEAK, HRP, WORD, 0x50, *nnnn*, DSTC

Scheduling period expressed in RTI ticks.

10.5.11 73MEM_TWEAK, HRP, WORD, 0x52, *nnnn*, DSTC

Scheduling offset expressed in RTI periods.

Dust is not scheduled directly by the acquisition control process, rather the analysis activity on DCP interacts with the acquisition process on HRP to limit the minimum period between datasets. The data acquisition control process, DSTC, normally is programmed to be idle (using location 0x56). The analysis process the sends a trigger command causing a single dataset to be acquired, when it is ready for data.

Dust acquisition uses 0x50/0x52 values as a MOD timing to check for acquisition requests from DCP. When a request arrives from DCP, it will be processed when the DSTC process next awakes. This allows dust acquisitions to be synchronized with other activities within the instrument (as well as with activities on the spacecraft that are scheduled in a similar manner).

Another way to think of this is that this causes data acquisitions to **slow down** to meet the desired synchronization goals.

Dust is the only process on HRP to use 0x50/0x52 during idle (WBR, WFR, and LFDR idle in an unsynchronized 1 second polling loop).

10.5.12 73MEM_TWEAK, HRP, WORD, 0x54, *nnnn*, DSTC

Delay (expressed in RTI's) between minipackets. Used to throttle the data delivered to the DCP.

10.5.13 73MEM_TWEAK, HRP, BYTE, 0x56, n, DSTC

Scheduling mode.

0 Stop

Data acquisition is stopped. ***73WFR_MODE_CNTL, TRIGGER*** is not processed. Internal triggers are not processed. This state is entered whenever sleep is asserted.

1 Idle

Data acquisition is stopped but the process will accept a trigger to perform a single acquisition. After the acquisition the process will return to idle to await further triggers.

When idle, the handler releases the CPU by using the DELAY_ system service. This DELAY_ function is MOD based and may be used to force data acquisition to occur, in effect, on a MOD schedule.

2 Run

Data acquisition is continuous and based on the schedule specified at offset 0x50 and 0x52.

3 Trigger

Single data acquisition.

10.5.14 73MEM_TWEAK, HRP, WORD, 0x58, *nnnn*, DSTC

8254 Offset from RTI. Used to control when the WBR sample occurs with respect to the RTI pulse from the S/C. This field is expressed in WBR samples so it is dependent on the current WBR sample rate. In addition there are 2 special cases.

nnnn	Description
0x0000	Immediate scheduling
0x0001..0xFFFE	Offset from next RTI
0xFFFF (-1)	Synchronize with WFR sampling

Offset table to place WBR sampling at the end of the RTI period.

	sample/RTI	512 samples	1024 samples	2048 samples
36uS	3,472	2,960 (0B90)	2,448 (0990)	1,424 (0590)
4.5uS	27,777	27,265 (6A81)	26,753 (6881)	25,729 (6481)

10.5.15 73MEM_TWEAK, HRP, WORD, 0x60, *value*, DSTC (00DUST_DATA_CNTL)

IPC destination queue. This location is used to route the raw mini packets for DUST. Normally the DUST traffic is delivered to the DCP for analysis, but the data may be routed directly to the LRP if necessary to implement a WBR toggle mode.

value	Description
0xC6	WBR Low Rate Science
0x47	DUST Analysis
0x01	WBR High Rate Science

10.5.16 73MEM_TWEAK, HRP, BYTE, 0x68, *nn*, DSTC

AGC Enable Flag. This is the location that stores the AGC enable bit. It may be directly altered to enable/disable gain without changing the gain level. Trigger 10 alters this memory location in lieu of using the **73WBR_GAIN_CNTL** command to avoid changing the current gain selection. This avoids having WBR step gain levels whenever we execute a trigger that uses trigger 10 code to perform preliminary WBR configuration/setup.

10.5.17 73MEM_TWEAK, DCP, WORD, 0x6E, *nnnn*, DUST

Target bit rate, expressed in bits per second, allocated for dust packets. Default is 10. This essentially limits how often a dust sample waveform minipacket will be produced.

10.5.18 73MEM_TWEAK, DCP, BYTE, 0x70, *n*, DUST

Three point slope criterion value. For three consecutive 8-bit samples in a WBR data set, referred to as W(I), W(I+1), and W(I+2), the DUST detection algorithm requires that abs(W(I)-W(I+2)) >= n. Default is 40 hexadecimal. Also note that all 8-bit data has its lower 4 bits zeroed by the algorithm, so in fact only the upper 4 bits are significant. If this value is sufficiently high, it eliminates most low frequency sinusoids.

10.5.19 73MEM_TWEAK, DCP, BYTE, 0x71, *n*, DUST

Zero-crossing criterion value. After the 3-point slope criterion has been met, a recovery time is looked for in the waveform. The zero-crossing value is taken to be either 70 hexadecimal or 80 hexadecimal. After the initial upward or downward steep change required by the 3-point slope criterion, a minimum number of points are required to be above (for upward) or below (for downward) the zero crossing. Default value is 12. The purpose of this criterion is to eliminate high frequency features.

10.5.20 73MEM_TWEAK, DCP, BYTE, 0x72, *n*, DUST

Initial flatness criterion value. Before the 3-point slope criterion is checked, an initial flatness is required, such that abs(W(I)-W(I-1) <= n. Default value is 10 hexadecimal.

10.5.21 73MEM_TWEAK, DCP, WORD, 0x7B, *nnnn*, DUST

Time between DUST housekeeping micro packets (see ***RPWS Housekeeping*** chapter ***Micro Packet: DUST*** section), given in RTIs. Default is 512, which corresponds to once every 64 seconds. Note that these micro packets are only produced as long as the DUST detection algorithm continues to receive data from the DSTC process on the HRP. If the DSTC process is shut down, the requests for data which come from the DUST detection process are ignored, and no DUST micro packets are produced. If the DUST detection process finds it is being ignored, it waits for about a minute before requesting more data from the DSTC data acquisition process. Between requests, it simply goes back to its usual game of DOOM, although it will accept any invitation for a game of NETDOOM also.

10.6 HFR control

French HFR experiment.

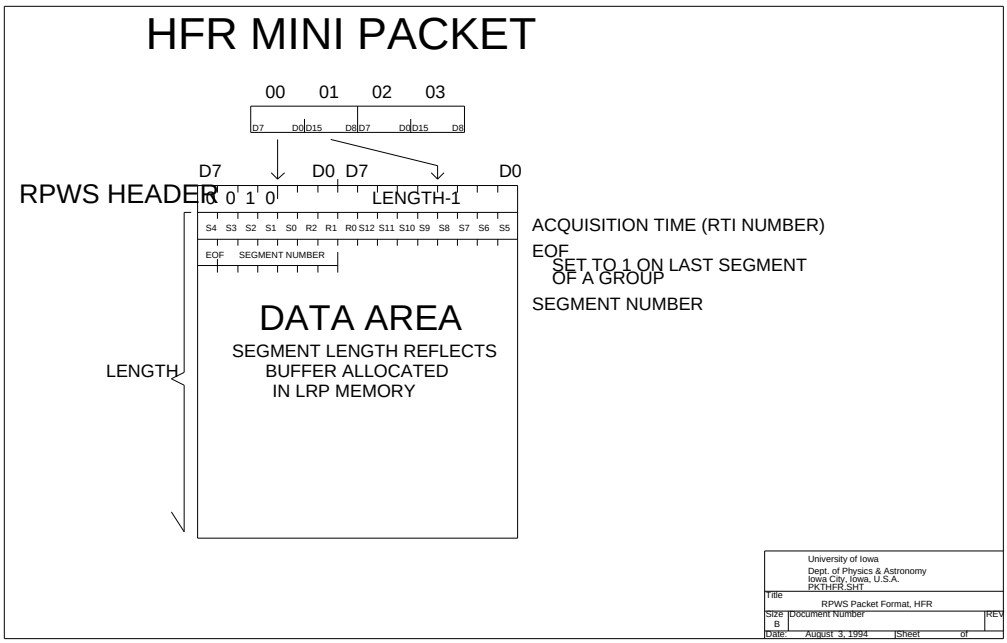
Commands should **NOT** be issued to the HFR when the instrument is in *SLEEP* mode or whenever the HFR is powered off. Commands cannot be processed, resulting in Free Queue elements being tied up. This can, eventually, result in LRP being uncommandable and eventually resetting itself.

Command blocks to the HFR must be limited to no more than 31 commands in order to limit the command buffer size. The command decoder in HFR and Sounder use a signed 8 bit counter to extract commands and supplying a buffer with more than 127 bytes will cause the command buffer to be ignored. This restriction is most noticeable when using the IEB handler to send commands to the HFR/Sounder.

HFR error recovery is modified in the 2.1 release to address a problem encountered when using the 2.0 software. When this problem occurs the 2.1 software performs a cold reset. If the reset fails several times, any outstanding commands will be discarded to avoid a deadlock where incoming commands consume all available free space and block the entire LRP.

The HFR breaks commands down using the upper three bits (of the 16 bit command word issued to the HFR) as a major function code. The command structure here mimics the internal decoding of the commands. Some of the descriptions are taken from the HFR command document and refer to the upper three bits of the HFR command word when indicating *Command Type*.

Some of the HFR commands detailed here are decoded using a special HFR command table located in the file: cmd_hfr.txt (ftp://rpwshp2.physics.uiowa.edu/usr/cassini/cfg/cmd_hfr.txt). These commands do not start with 73 or 00 as all other RPWS commands do.



10.6.1 73HFR_ANALYSIS, analysis

Command type 111 is the command which commands the HFR to perform an analysis, using the measurement mode setup earlier with the Parameter Load #0 and Parameter Load #1 commands. The table below lists the currently defined sub-types for this command.

analysis	Description
0xE000..0XE0FF	Analysis
0xE100..0xE1FF	Internal Calibration

10.6.1.1 14ANAL, command, ms, h2, h1, c, b, a

command	Description
ANALYSE	Analysis
CALIBRATE	Internal Calibration

Use the receiver name to enable and the NA to suppress data collection on the indicated channel.

Channel	MS	H2	H1	C	B	A
Enable	MS	H2	H1	C	B	A
Disable	NA	NA	NA	NA	NA	NA

10.6.1.2 Analysis

This command tells the HFR to do an analysis, using the mode settings previously defined. The command parameter selects the bands to analyze, as follows

Bit Number	Contents
5	MS
4	Band HF2
3	Band HF1
2	Band C
1	Band B
0	Band A

When the bit 5 is 0 a normal analysis will be performed. The bits set in the command parameter determine which bands will be analyzed. Thus if bit 4 is set, high frequency band #2 will be analyzed, if bit 0 is set low frequency band A will be analyzed, and so on. Any combination of set bits is permitted. When the bit 5 is set a special millisecond (MS or Fast) sequence is initiated where only one AGC is sampled at rapid intervals. The parameters for this special mode are determined by commands **73HFR_INITIALIZE, AExx** and **73HFR_INITIALIZE, AFxx**. Only the bits 3 and 4 are used in this mode: whichever of these bits are set determines whether the HF1 or the HF2 band is used. The selection defaults to HF1 when both bits are set to 0 or 1.

10.6.1.3 Internal Calibration

This command pattern tells the HFR to perform an Internal Calibration sequence, using the calibration settings stored in the PM RAM table. Normally these calibration settings would never be changed The command parameter selects the bands to calibrate, as follows

Bit Number	Contents
4	Band HF2
3	Band HF1
2	Band C
1	Band B
0	Band A

The bits set in the command parameter determine which bands will be calibrated. Thus if bit 4 is set, high frequency band #2 will be calibrated, if bit 0 is set low frequency band A will be calibrated, and so on. Any combination of set bits is permitted. Normally all five bits would be set, since we will usually want to calibrate the entire instrument.

10.6.2 00HFR_DEVELOPMENT, command, value

Commands used for instrument development and debugging. New commands are frequently added and removed from this list as dictated by instrument debugging needs. Of the command sub-types described below, only sub-types 0 (a NOP command) and sub-type 1 (firmware revision command) are likely to be useful in flight. The table below lists the subtypes for this command.

command	binary	Description
NOOP	0x0000..0x00FF	No Operation
VERS	0x0100..0x01FF	Return HFR firmware version
STEP or DELAY	0x0200..0x02FF	Set frequency step of sounder delay
SWITCH	0x0300..0x03FF	Switch between two frequencies
START	0x0400..0x04FF	Start sounder test
BAND	0x0500..0x05FF	Force sounder band for test
TEST	0x0600..0x06FF	Enable/Disable test mode
UNDEF	0x0700..0x07FF	Undefine subtype

10.6.3 73HFR_HK, value

value	Description
0xE000..0xFFFF	

10.6.4 73HFR_INITIALIZE, value

Command type 011 groups together a sub-set of the commands used to control HFR operation. The table below lists the currently defined subtypes for this command

value		Description	
0x6000..0x67FF		Set FPGA Words 0..7	
0x6800..0x68FF		Re Initialize FPGA 2	
0x6900..0x69FF		Re Initialize measurement modes	
0x6A00..0x6AFF		Recalculate Filter Coefficients	
0x6B00..0x6BFF		Force Synthesizer selection	
0x6C00..0x6CFF		Set Data Compression Mode	
	0x6C00	No Compression	
	0x6C20	Meander	
	0x6C40	Rice	
0x6D00..0x6DFF		Set MFR antenna / calibration	
	D6	Cal signal Enable	
	D5	MFR Cal #2 enable	
	D4..D3	00	MFR Cal #1 OFF
		01	MFR Cal #1 +
		10	MFR Cal #1 -
		11	MFR Cal #1 +/-
	D2	Antenna MFR #2 EZ enable	
	D1..D0	00	Antenna MFR #1 OFF
		01	Antenna MFR #1 Ex +
		10	Antenna MFR #1 Ex -
		11	Antenna MFR #1 Ex +/-
0x6E00..0x6EFF		Set antenna limit relay state	
0x6F00..0x6FFF		Set attenuator state	

10.6.4.1 6FPGA2-0, synth-hi, synth-lo, band, A-counter

10.6.4.2 6FPGA2-1, M-counter

10.6.4.3 6FPGA2-2, sounder-enable, sounder-band, sounder-pulse

10.6.4.4 6FPGA2-3, internal-cal, mfr-cal, fast-slow, level

10.6.4.5 6FPGA2-4, mfr-ez-cal, mfr-ex-cal, mfr-ex+cal, mfr-ex-ant, mfr-ex-ant, mfr-ex+ant

10.6.4.6 6FPGA2-5, mfr-ez-cal, mfr-ex-cal, mfr-ex+cal, mfr-ex-ant, mfr-ex-ant, mfr-ex+ant

10.6.4.7 6FPGA2-6, clamp-relay, sounder-relay

10.6.4.8 6FPGA2-7, attenuator, band

10.6.4.9 6CALC-COEF, high, sounder, low

10.6.4.10 6SYNTH-SEL, frequency

frequency	
auto	synthesizer selection is automatic
lf	low frequency synthesizer selected
hf	high frequency synthesizer selected

10.6.4.11 6COMPRESS, method

method	
off	no compression selected
meander	meander code
rice	rice code

10.6.4.12 6MFR-ANT, cal, cal-ez, cal-ex, ant-ez, ant-ex

Command type 100 groups together a sub-set of commands used to control HFR operation. The various sub-types define parameters for HFR Analysis modes. The table below lists the currently defined sub-types for this command.

setup	Description
0x8000..0x81FF	Set measurement mode ABC
0x8200..0x83FF	Set extended measurement mode ABC
0x8400..0x85FF	Set ABC repeat count
0x8600..0x87FF	Set First Frequency Channel HF1
0x8800..0x89FF	Set frequency step HF1
0x8A00..0x8BFF	Set number of frequency steps HF1
0x8C00..0x8DFF	Set measurement mode HF1
0x8E00..0x8FFF	Set extended measurement mode HF1
0x9000..0x91FF	Set HF1 repeat count
0x9200..0x93FF	Set First Frequency Channel HF2
0x9400..0x95FF	Set frequency step HF2
0x9600..0x97FF	Set number of frequency steps HF2
0x9800..0x99FF	Set measurement mode HF2
0x9A00..0x9BFF	Set extended measurement mode HF2
0x9C00..0x9DFF	Set HF2 repeat count

0x9E00..0x9FFF	Set cycle count

10.6.4.13 8ABC-MODE, period, df, ant-2, ant-1, filters

period
1s
1/2s
1/4s
1/8s

df
df-on
df-off

ant-2	
a2-off	
a2-on	Ez

ant-1	
a1-off	
a1-plus	Ex +
a1-ex+	
a1-minus	Ex -
a1-ex-	
a1-both	Ex +/
a1-on	

filters	
8 filters-8	8 filters per band
16 filters-16	16 filters per band
32 filters-32	32 filters per band

10.6.4.14 8ABC-EXT, auto, cross

auto
auto-off
auto-on

cross
cross-off
cross-on

10.6.4.15 8ABC-REP, count

The number of times to repeat the low frequency measurement per frequency sweep. A frequency sweep is one pass through each of the bands A, B, C, HF1, and HF2.

10.6.4.16 8HF1-MODE, period, df, ant-2, ant-1, filters

period
160mS
80mS
40mS
20mS

df
df-on
df-off

ant-2	
a2-off	
a2-on	Ez

ant-1	
a1-off	
a1-plus	Ex +
a1-ex+	
a1-minus	Ex -
a1-ex-	
a1-both	Ex +/
a1-on	

filters	
1 filters-1	1 filters per band
2 filters-2	2 filters per band
4 filters-4	4 filters per band
8 filters-8	8 filters per band

10.6.4.17 8HF1-EXT, auto, cross

auto
auto-off
auto-on

cross
cross-off
cross-on

10.6.4.188HF1-REP, repeat count

The number of times to repeat the low frequency measurement per frequency sweep. A frequency sweep is one pass through each of the bands A, B, C, HF1, and HF2.

10.6.4.19 8HF1-START, start frequency

This command set the first (lowest) frequency for HF1. Expressed in 25Khz units.

10.6.4.20 8HF1-STEP, step size

Frequency step size expressed in units of 25Khz.

10.6.4.21 8HF1-COUNT, step count

This command set the number of frequencies to scan. Range is determined by start frequency, step size, and number of steps.

10.6.4.22 9HF2-MODE, period, df, ant-2, ant-1, filters

period
80mS
40mS
20mS
10mS

df
df-on
df-off

ant-2	
a2-off	
a2-on	Ez

ant-1	
a1-off	
a1-plus	Ex +
a1-ex+	
a1-minus	Ex -
a1-ex-	
a1-both	Ex +/
a1-on	

filters	
1 filters-1	1 filters per band
2 filters-2	2 filters per band
4 filters-4	4 filters per band
8 filters-8	8 filters per band

10.6.4.23 9HF2-EXT, auto, cross

auto
auto-off
auto-on

cross
cross-off
cross-on

10.6.4.24 9HF2-REP, repeat count

The number of times to repeat the low frequency measurement per frequency sweep. A frequency sweep is one pass through each of the bands A, B, C, HF1, and HF2.

10.6.4.25 9HF2-START, start frequency

This command set the first (lowest) frequency for HF2. Expressed in 50Khz units.

10.6.4.26 9HF2-STEP, step size

Frequency step size expressed in units of 50Khz.

10.6.4.27 9HF2-COUNT, step count

This command set the number of frequencies to scan. Range is determined by start frequency, step size, and number of steps.

Command type 101 groups together a sub-set of the commands used to control HFR operation. The various sub-types define parameters for HFR Sounder modes, and the relay and attenuator settings. The table below lists the currently defined sub-types for this command.

setup	Description
0xA000..0xA0FF	Set Sounder Mode
0xA1..0xA1	Set first sounder frequency
0xA2..0xA2	Set last sounder frequency
0xA3..0xA3	Set Sounder T1 Delay
0xA4..0xA4	Set sounder T2 delay
0xA5..0xA5	Set sounder T3 delay
0xA6..0xA6	Set passive sweeps / cycle
0xA7..0xA7	Set active sweeps / cycle
0xA8..0xA8	Set number of cycles
0xA9..0xA9	Set sounder relay state
0xAA..0xBA	Set sounder output A
0xAC..0xDC	Set sounder output B
0xAE..0xAE	Set MS parameters A
0xAF..0xAF	Set MS parameters B

10.6.4.28 10SND-MODE, antenna, mode

10.6.4.29 10MS-ANT, antenna, sample rate, sample count

Note that a sample count of 32768 is invalid as the HFR has insufficient memory for this many samples.

Sample rate patch is available and may be downloaded using the appropriate IEB LOAD along with the appropriate IEB TRIGGER.

antenna	
EZ	
EX+	Ex +
EX-PLUS	
EX-	Ex -
EX-MINUS	
EX	Ex +/
BOTH	

sample rate		
keyword	un-patched	patched
500uS	160uS	500uS
1mS	160uS	1mS
2mS	160uS	2mS
4mS	160uS	4mS
8mS	160uS	8mS
16mS	160uS	16mS
32mS	160uS	32mS
64mS	160uS	64mS

sample count
256
512
1024
2048
4096
8192
16384

10.6.4.30 10MS-FREQ, frequency

Center frequency for the mixer. When using HF1 the frequency is expressed 25Khz steps. When using HF2 the frequency is expressed in 100Khz steps with a 25Khz offset.

Somehow we manage to select H1/H2 with this command, but the command decode table is probably set up inappropriately at this time.

If the WBR is in use, note that **the 8HF1-START** command must be issued to send the appropriate frequency to HRP for inclusion in the WBR minipacket.

10.6.5 73HFR_LOAD_MEM, value

Command type 010 groups together commands used for uploading new data to either Program or Data memory. Data is only written to memory when an Upper byte load command (subtype 101) is received. For this command sub-types, bit 12 of the command is used to choose either DM (P = 0) or PM (P = 1). The table below lists the subtypes for this command.

The command column is used for the extended commands that follow.

command	***value***	Description
LOWER	0x4000..0x40FF	Lower byte of load address
UPPER	0x4100..0x41FF	Upper byte of load address
RAM	0x4200..0x42FF	RAM page to load
LSB	0x4300..0x43FF	Lower byte of load
MSB	0x4400..0x44FF	Middle byte of load
USB	0x4500..0x45FF	Upper byte of load, PM
PC	0x4600..0x46FF	Load PC with upload address
ENAB	0x4700..0x47FF	Load Enable/Disable
USBPM	0x5500..0x55FF	Upper byte of load, PM

10.6.5.1 4LOAD_MEM, command, byte-value

The command mnemonic is taken from the table and the byte-value as required.

10.6.6 73HFR_MEM_DUMP, value

Command type 001 groups together commands used for dumping the contents of Program or Data memory. Some of the subtypes below are applicable to either program memory (PM) or data memory (DM). For these sub-types, bit 12 of the command is used to choose either DM (P = 0) or PM (P = 1). The table below lists the subtypes for this command.

command	***value***	Description
LOWER	0x2000..0x20FF	Lower byte of dump address
UPPER	0X2100..0X21FF	Upper byte of dump address
RAM	0X2200..0X22FF	RAM page to dump from
BYTE	0X2300..0X23FF	Dump bytes, DM
WORD	0X2400..0X24FF	Dump words, DM
BYTEPM	0X3300..0X33FF	Dump bytes, PM
WORDPM	0X3400..0X34FF	Dump words, PM
INST	0X3500..0X35FF	Dump Instructions

The procedure for dumping data from memory is as follows. Before dumping memory, the first (that is, lowest) address to dump from must be loaded. This is done using two commands, Low byte and High byte, Subtypes 000 and 001 below. The order in which low and high address bytes are sent is not important. Then the RAM page to dump from is loaded using Subtype 010. Finally, the data is dumped using the appropriate instruction. The subtypes are described below.

10.6.6.1 2DUMP_MEM, command, byte-value

The command mnemonic is taken from the table and the byte-value as required.

10.6.7 73HFR_RELAY_ACC

Relay Actuate.

10.6.8 73HFR_RELAY_ENA, position

Controls the clamp relay.

position	Description
open	Antenna potential may exceed ±12V
closed	Clamped

10.6.9 73HFR_RESET, reset

Used to perform a ***warm*** reset (i.e. software reset) or ***cold*** reset (i.e. hardware reset) on the HFR processor.

The HFR COLD RESET requires approximately 3 RTI periods to complete. The handler allows a period of 4 to 5 RTI for the reset to occur. In the event that the cold reset fails to occur the handler will eventually detect this condition and attempt to reissue the reset. After several unsuccessful attempts, the handler will flush a single HFR command and repeat these steps.

Reset	Description
cold	cold reset, reloads memory
warm	warm reset, **no** memory reload

For either reset method, note that the HFR is susceptible to a reset failure that may require power cycling the HFR to recover. Also note that all other internal power switches must be off prior to applying power to the HFR.

The symptom of a reset failure is no LRP idle time as indicated in the housekeeping page. In addition, loss of MFR data may occur due to a blocked CPU (i.e. HFR recovery activity may be at a higher priority than the MFR processes).

10.6.10 73HFR_SOUND, sounder

sounder	Description
0xC000..0xDFFF	Perform sounder analysis

Command type 110 is the command which commands the HFR to perform a sounder analysis. There are no sub-types for this command, and the command parameter is ignored. Command 110 simply performs a sounder analysis using the previously defined sounder mode settings.

10.6.10.1 12SOUND

No parameters required.

10.6.11 73MEM_TWEAK, LRP, WORD, 0x0104, *mod-count*, HFRI

This location controls the timing of the HFR analysis activity specified in location 0x010A. Expressed in RTI periods. The next analysis command will be synchronized with the time based on this RTI cycle.

10.6.12 73MEM_TWEAK, LRP, WORD, 0x0106, *mod-remainder*, HFRI

This location contains the remainder used to time the next HFR analysis activity. The remainder of (SCLK ÷ mod-count) will be equal to this value when the next activity is scheduled.

10.6.13 73MEM_TWEAK, LRP, WORD, 0x010A, *hfr-command*, HFRI

This command loads the ***73HFR_ANALYSIS*** command pattern that is used during normal **HFR** operations. This is the command that is sent to the HFR repeatedly.

This table simply lists some sample command patterns that may appear in commands issued to the instrument (i.e. these are taken from existing IEB's and commands sets).

HFR-command	Description
0xE01F	Do Analysis all bands A, B, C, HF1, HF2
0xE018	Do Analysis bands HF1, HF2
0xE01C	Do Analysis bands HF1, HF2, C
0xE007	Do Analysis bands A, B, C
0x0000	Idle HFR

10.6.14 73MEM_TWEAK, LRP, WORD, 0x60, *enable-flag*, SOND

This location to enables the sounder. Set to 0x0000 to disable sounder. Set to 0xFFFF to enable sounder.

10.6.15 73MEM_TWEAK, LRP, WORD, 0x62, *delay*, SOND

This location to sets the delay, in RTIs, between sounder commands sent to the HFRI command queue. If set to 0, then no commands are sent to the HFRI command queue.

10.6.16 73MEM_TWEAK, LRP, WORD, 0x64,
10.6.17*sounder-command*, SOND

This location contains a sounder command. Normally this should be the pattern 0xC000, which is the HFR sounder command. But it could be any HFR software command which needs to be sent repetitively.

10.6.18 00PORT_TWEAK, LRP, 0x60, delay HFR DMA timing control register

This tweak may be used to alter the DMA timing for the HFR data interface to the LRP. This port control the clock used to strobe data from the HFR during data transfers. The default timing value is 32 uSec when the LRP is reset.

As of version V2.3 the HFR handler does not alter this default timing. Any changes to the timing will remain in effect until the LRP is powered down (or additional changes are made)

delay	timing
0xFE	4 µSec
0xFD	8 µSec
0xFB	16 µSec
0xF7	**32 µSec, default**
0xEF	64 µSec
0xDF	128 µSec
0xBF	256 µSec
0x7F	512 µSec
0x00	1020 µSec

The HFR data collection is performed open-loop. There is no signal from the HFR to indicate that it is ready to deliver the next data word so the timing must be sufficiently relaxed to allow HFR to perform bookkeeping tasks between data blocks (in most cases the HFR handler will **not** be in synchronization with the HFR so we must allow for worst case conditions here).

The value in this register is the ones-complement of the desired transfer timing expressed in 4 µSec units. The table above is provided only as a guide, and is not intended to limit the choice of timing values.

10.6.19 Down-convert Mode for the WBR

This is a discussion of the method used to tag the WBR data that is obtained through the HFR with appropriate status information with respect to the down-convert frequency.

The WBR, when commanded into HF mode (i.e. using the HF to down-convert signals) either HF1 or HF2 may be used to perform the mixing task. We make use of H1 for frequencies below 4Mhz and H2 for frequencies above that point. **The HFR must be commanded into a quiescent mode, such as Millisecond mode, to eliminate interference in the signal delivered to the WBR** (Note that millisecond data is required to obtain AGC information to calibrate the WBR data). The HFR handler passes the frequency value in the **8HF1-START** command to the HRP so the WBR handler can mark the WBR data with an indication of the frequency band of the data.

The spacecraft exhibits interference line at multiples of 50 Khz. Due to this interference the H1 down-convert mode is only commanded with odd numbers in the **8HF1-START** command (this being an operational constraint, not a limitation within the HFR). This has less effect on the HF2 as it is spaced at 100Khz intervals when in the millisecond mode, but with only 8 bits available to mark the data, we need a scheme to uniquely encode the frequency in a compatible and consistent manner.

Keep in mind that when running in a down-convert mode, the WBR data is useful only when HFR is quiet (i.e. idle or millisecond mode). This means, in effect, that we need only interest ourselves in millisecond mode where the H2 frequency is specified in 100Khz increments.

The following table shows the value that should be used as the HF1 start frequency for the various down-convert frequencies for both H1 and H2. Note that it is necessary to send the H1 command in order to deliver the correct value to HRP/WBR for the status byte in the WBR minipacket (it has no effect within the HFR)

8HF1 START frequency	H1 frequenc y Khz	H2 frequenc y Mhz
00		4.025
01		
02		4.125
03		
04		4.225
05	125	
06		4.225
07	175	
08		4.425
09	225	
0A		4.525
0B	275	
0C		4.625
0D	325	
0E		4.725
0F	375	
10		4.825
11	425	
12		4.925
13	475	
14		5.025
15	525	
16		5.125
17	575	
18		5.225
19	625	
1A		5.325
1B	675	
1C		5.425
1D	725	
1E		5.525
1F	775	

8HF1 START frequency	H1 frequenc y Khz	H2 frequenc y Mhz
20		5.625
21	825	
22		5.725
23	875	
24		5.825
25	925	
26		5.925
27	975	
28		6.025
29	1025	
2A		6.125
2B	1075	
2C		6.225
2D	1125	
2E		6.325
2F	1175	
30		6.425
31	1225	
32		6.525
33	1275	
34		6.625
35	1325	
36		6.725
37	1375	
38		6.825
39	1425	
3A		6.925
3B	1475	
3C		7.025
3D	1525	
3E		7.125
3F	1575	

8HF1 START frequency	H1 frequenc y Khz	H2 frequenc y Mhz
40		7..225
41	1625	
42		7.325
43	1675	
44		7.425
45	1725	
46		7.525
47	1775	
48		7.625
49	1825	
4A		7.725
4B	1875	
4C		7.825
4D	1925	
4E		7.925
4F	1975	
50		8.025
51	2025	
52		8.125
53	2075	
54		8.225
55	2125	
56		8.325
57	2175	
58		8.425
59	2225	
5A		8.525
5B	2275	
5C		8.625
5D	2325	
5E		8.725
5F	2375	

8HF1 START frequency	H1 frequenc y Khz	H2 frequenc y Mhz
60		8.825
61	2425	
62		8.925
63	2475	
64		9.025
65	2525	
66		9.125
67	2575	
68		9.225
69	2625	
6A		9.325
6B	2675	
6C		9.425
6D	2725	
6E		9.525
6F	2775	
70		9.625
71	2825	
72		9.725
73	2875	
74		9.825
75	2925	
76		9.925
77	2975	
78		10.025
79	3025	
7A		10.125
7B	3075	
7C		10.225
7D	3125	
7E		10.325
7F	3175	

8HF1 START frequency	H1 frequenc y Khz	H2 frequenc y Mhz
E0		15.225
E1		
E2		15.325
E3		
E4		15.425
E5		
E6		15.525
E7		
E8		15.625
E9		
EA		15.725
EB		
EC		15.825
ED		
EE		15.925
EF		
F0		16.025
F1		
F2		
F3		
F4		
F5		
F6		
F7		
F8		
F9		
FA		
FB		
FC		
FD		
FE		
FF		

10.7 IEB Commands

Instrument Expanded Block commands are used to load and execute macro level commands within the instrument. The IEB memory may be loaded with up to 16K bytes of command and control tables.

The commands consist of instrument commands as documented in this section of the Users Guide. These patterns are no different than the commands delivered by the spacecraft and are inserted into the command processor as if they arrived from the spacecraft.

The control consists of timing and loop control directives. Due to lack of decision making capability, the programmability is limited and depends on some assistance from the spacecraft in the form of additional **73IEB_TRIGGER, ID** commands.

10.7.1 73IEB_HALT, modifier

Stops the currently executing IEB sequence.

modifier	Description
idle	suspend data acquisition
run	data acquisition continues in last mode step executed in the IEB
clear	IEB execution is halted and IEB memory is cleared to ZERO

10.7.1.1 73IEB_HALT, CLEAR

This command is intended to be used when an IEB load is provided from the ground. Version 2.4 software has a default IEB load as part of the ALF load. This internal IEB load should be cleared prior to loading an IEB from the ground in order to force all checksums and unused memory to a known state. This *primes* the integrity checking mechanism; checksums must be calculated on the ground and included with the IEB load; the loader simply recalculates and verifies the checksum information provided by the ground.

10.7.1.2 73IEB_HALT, IDLE

This command will cause MFR to halt (along with all the other receivers). This command issues a **73POWER_CNTL, PAUSE** command which has the potential to hang HRP if WBR in operating in high band mode.

NOTE that this command suspends all data collection activities.

10.7.2 73IEB_LOAD, size, sequence, data?

Loads IEB memory. This command is a special case and is expected to arrive as the only command during any given RTI period. Command length is variable although the nominal command is used to deliver 64 words (128 bytes) if IEB image. The minimum number of data words is, of course, one word.

Although the IEB_LOAD command may specify an arbitrary command size, the desired size is 64 words. This results in a packet that contains 68 words (136 bytes) and limits the command delivery system on the spacecraft to delivering no more than one command per RTI period. Use of a size field below 60 can result in multiple commands being delivered in a single RTI period (causing the command decode within the instrument to fail). The maximum size is around 120 words in order to limit the total command size to a maximum of 125/128 words (imposed by the command decoder and spacecraft). 64 is the maximum natural size (i.e. power-of-two) that can be achieved while enforcing the *single command per RTI* limitation of the instrument software. A size of 120 is useful when storing loads using the Library Storage Facility.

The field ***size*** is used to specify the number of data words in the command.

The sequence field must start with zero at the beginning of the load and increase monotonically. A break in the sequence field will cause all succeeding ***73IEB _LOAD*** records to be ignored.

size	Description
1	Minimum word count
66	64 data words per record This value seems to work everywhere
34	32 data words per record (in case it's needed!) Observe 1 command/RTI limit

sequence	Description
0..n-1	monotonically increasing from zero

data[0]	Description
0xC000..0xFFFF	IEB data address
0x8000..0xBFFF	HFR Scientific Program address

data[1..n-2]	Description
0x0000..0xFFFF	IEB data
0x0000..0xFFFF	HFR program data

data[n-1]	Description
0x0000..0xFFFF	IEB data checksum
0x0000..0xFFFF	HFR program checksum

10.7.3 73IEB_LOAD, 0, sequence
10.7.4 00IEB_END, sequence
10.7.5 00IEB_EOF

This is the cue to validate the checksums to the IEB handler. The length

The 1st form of the command is documented in JPL 3-281. Note the length field is zero and that no data words follow the sequence number.

The 2nd and 3rd Forms of the command are present in the local command decoder only, they are not in the JPL 3-281 document. It serves simply as a shorthand form of the last load command that causes the checksum to be performed. The 3rd form skips the sequence number, by simply inserting a one in the sequence field.

size	Description
0	End of load indicator

sequence	Description
n	Next sequence (i.e. 1 greater than the last **73IEB_LOAD** command)

10.7.5.1 Some notes about EOF handling.

The length field in the 2nd word of the **73IEB_LOAD** command must be zero to indicate to the IEB handler that the checksum table is to be updated. The IEB handler is tolerant of additional words in the command, so a **73IEB_LOAD, 0, sequence, 0** command will cause the checksum to be calculated (try **73IEB_LOAD, 0, sequence, 2001** to eliminate incrementing the invalid command count). The length field being zero is vital!

10.7.6 73IEB_TRIGGER, type, identifier 1, identifier 2

Starts running a step in the IEB sequence. To trigger internal command sequences, the ***type*** field contains the keyword **ID** and the appropriate command sequence must be loaded using 73IEB_LOAD commands.

type	Description
mask	internal IEB control does **not** require IEB_LOAD data
id	IEB step number requires an IEB_LOAD to operate

identifier 1	Description
10, 12, 14,?	IEB trigger address this specifies an address in memory where an IEB trigger point begins

identifier 2	Description
0	parameter should be zero

10.7.6.1 73IEB_TRIGGER, MASK, 0, 0

Full **Power ON** command handled internally. This is the preferred method to bring the instrument to a full power state as it is timed and ordered to present the minimum peak current load to the S/C power bus.

10.7.6.2 73IEB_TRIGGER, MASK, 0, 1

Basic science collection mode. This trigger, effective only after the instrument is fully powered, begins collecting data from all of the receivers. Indicated bit rates are estimates **before** any data compression is applied.

Receiver	Bit Rate	Function
MFR	56	Toggle Ex/Bx
HFR	509	Composite mode
Sounder	0	OFF
L/P	0	OFF
WBR	0	OFF
WFR	360	5 Channel (Ex, Ex, Bx, By, Bz) Auto gain 2K samples
LFDR	32	Ex/Bx
DUST	10	Ez

10.7.6.3 73IEB_TRIGGER, MASK, 0, 2

Science Load

Same as previous trigger.

Special Maintenance Load

Triggers the HFR Venus observation. This requires that the special maintenance software is loaded in the instrument. Given the internal storage capacity of 128K bytes, this observation will take approximately 2 ¼ hours.

Receiver	Bit Rate	Function
MFR	0	OFF, No Power
HFR	130	Venus Observation Mode
Sounder	0	OFF
L/P	0	OFF, No Power
WBR	0	OFF, No Power
WFR	0	OFF, No Power
LFDR	0	OFF, No Power
DUST	0	OFF, No Power

10.7.6.4 73IEB_TRIGGER, MASK, 0, 3

Powers the Langmuir Probe electronics and set the sphere bias to 32 volts. This trigger is part of the V2.3 science software and is intended to allow maintenance activities to be performed without the special maintenance software (i.e. this allows maintenance software to be deleted from the SSR).

Receiver	Bit Rate	Function
MFR	0	OFF, No Power
HFR	0	OFF, No Power
Sounder	0	OFF, No Power
L/P	0	ON, Sphere bias to 32 Volts
WBR	0	OFF, No Power
WFR	0	OFF, No Power
LFDR	0	OFF, No Power
DUST	0	OFF, No Power

Exercise caution with the use of this trigger as it can apply power to all receivers. If all receivers are powered then this is not an issue, but if the instrument has been placed into sleep using the **73POWER_CNTL, SLEEP, SLEEP** command, then this trigger will cause the previous power state to be restored.

In normal use, this trigger would be issued in isolation, following power-on and a fresh load of the science software.

10.7.6.5 73IEB_TRIGGER, MASK, 0, 4

Exit Langmuir probe maintenance mode. The sphere is biased to 0 volts but the power is NOT removed from the L/P electronics. Use the SLEEP command to remove power.

Power is not commanded in this trigger to allow triggers 3 and 4 to be used when operating the instrument in a science mode.

10.7.7 73IEB_TWEAK

Special case of the ***73MEM_TWEAK*** command. This is one of those really nifty well-thought out commands that was defined too early to be really useful. Appears as ***73MEM_TWEAK*** in most command blocks. Refer to the ***73MEM_TWEAK*** command for details of the operation and use of this command.

This form of the command has some address range limits that are verified on the ground. The command will be rejected if the address specified is out of the allowed range.

10.7.8 73WRAP, (4310, *xx*00)

This is the hexadecimal representation of the **73IEB_LOAD** command used to terminate an *IEB_LOAD* where the ***xx*** is the sequence number. The command translation tool at JPL did not correctly process this form of the 73IEB_LOAD command and it typically needs to be wrapped in order to translate correctly. Note that ***xx*** must be expressed in hexadecimal.

10.8 IPC Handler

Although the IPC process is not directly commandable, there are some interesting areas within this handler that are of interest. This command generates housekeeping data that is described in the ***RPWS Housekeeping*** chapter under the ***Micro Packet: IPC*** section.

10.8.1 73MRO, *processor*, HSK, 1170, 0

Dumps the status area of the IPC handler.

10.8.2 73MRO, *processor*, HSK, 1170, 117F

Dumps all of the status area of the IPC handler.

10.9 LFDR Control

Low Frequency Digital receiver commands. Note that the compression switch is present in two separate commands and this can present an inconsistency in building the commands. Please refer to the *suggested order list* when building **LFDR** commands to avoid placing the instrument into an invalid state.

Commands may be sent to the LFDR when the instrument is in *SLEEP* mode although they will not be immediately processed. It should not be possible to bring the instrument out of SLEEP mode inadvertently (i.e. modifications to offset 0x56).

Version V2.6 add a band toggle capability to the LFDR. This allows for alternating between high band and low band for succeeding acquisitions. The only controls that are managed separately for the toggle mode are the gain settings. Since gain control resides external to the LFDR control process, there is support in the acquisition process to allow independent automatic gain control for the high band and low band acquisition.

Note on synchronous idle: we considered switching to a synchronous idle on the LFDR to accommodate some of the HFR/WBR modes in a little cleaner manner. Consider the effect if LFDR is operating in a peculiar mode with a scheduling interval that is unreasonably long (5 minute WFR period comes to mind). This will have noticeable impact when transitioning to a normal mode in that it will, in many cases, take take this long period to transition into a new mode (the command process would be stuck polling for commands at the slow rate).

The toggle mode is enabled and disabled using the **73LFDR_TOGGLE_CNTL** command (as documented a bit later in this section).

Suggested order of commands:

73LFDR_ANT_SEL

73LFDR_AUTO_CNTL

73LFDR_CHAN_CNTL

73LFDR_TOGGLE_CNTL

73LFDR_GAIN_CNTL

73LFDR_MODE_CNTL, HOLD, , NOCOMPRESS

73LFDR_CMPRS_CNTL, OFF

73LFDR_TOGGLE_CNTL

73LFDR_CNTL (DCP)

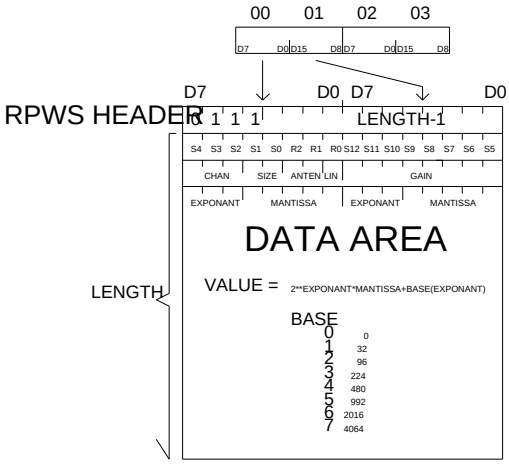
73MEM_TWEAK offset 0x50

73MEM_TWEAK offset 0x52

73MEM_TWEAK offset 0x54

73MEM_TWEAK offset 0x56

LFDR MINI PACKET



LIN
0=LOG
1=LINEAR

ANTEN
FROM WFR PACKET
(CHANNEL OF INTEREST)

SIZE 2**N*256
0=256
1=512
2=1024
3=2048

CHANNEL
0..4

GAIN
FROM WFR PACKET

DATA
8 BIT FLOATING POINT

University of Iowa
Dept. of Physics & Astronomy
Iowa City, Iowa, U.S.A.
PKTLFDR.SHT

Title: RPWS Packet Format, WFR

Size: B | Document Number | REV

Date: November 14, 1995 | Sheet of

10.9.1 LFDR Timing Control Process: Setup Storage Area

OFFSET	Label	Default	Description
0x24A8	CMDp_Divisor		MOD timing divisor
0x24AA	CMDp_Remainder		MOD timing remainder
0x24AC	CMDp_Packet_Delay		Inter packet delay
0x24AE	CMDp_Run		Run Flag
0x24AF	CMDp_AGC		
0x24B0	CMDp_Offset		8254 timer, offset from RTI to start of data capture
0x24B2	CMDp_Mode		Channel Mode: 1ch,3ch,5ch
0x24B4	CMDp_Band		Band Select
0x24B6	CMDp_Length		Dataset byte count (8237 word count)
0x24B8	CMDp_Destination		Data Routing Destination
0x24BA			
0x24BC	CMDp_AGC_H_L		AGC set points
0x24BE	CMDp_Antenna		Antenna Selection
0x24BF	CMDp_Gain_0		Gain Setting: Channel 0
0x24C0	CMDp_Gain_1		Gain Setting: Channel 1
0x24C1	CMDp_Gain_234		Gain Setting: Channels 2, 3, 4
0x24C2	CMDp_Walsh		Walsh number
0x24C4	CMDp_DCC		ACTEL register 43 Pattern
0x24C6			

Field	*Field Location*	*Description*
Minipacket ID	Byte 0 / D7..D4	1000
Minipacket Length	Byte 0 / D3..D0 Byte 1 / D7..D0	12 bit length
RTI	Byte 2 S4..S0, R2..R0 Byte 3 S12..S5	RTI time of data acquisition Seconds & RTI
Lin	Byte 4 / D0	
Anten	Byte 4 / D1..D2	
Size	Byte 4 / D3..D4	
Chan	Byte 4 / D5..D7	
Gain	Byte 5	

10.9.2 DCP Resource Requirements

Each FFT analysis requires about 5 seconds of CPU time on the DCP.

DCP CPU requirements	Description
N/A	RAW (bypass DCP)
N/A	PACK
N/A	RICE
N/A	WALSH
N/A	WALSH/RICE
5 seconds 512 sample packet	FFT analysis

10.9.3 73LFDR_ANT_SEL, sensor 0, sensor 1, sensor 2

Selects the antenna element that will be connected to the first 3 channels of the 12 bit analog converter. The 1st 3 channels have a 2 to 1 signal multiplexer.

Sensor 0	Description
exlo	electric dipole antenna, X
lmr	langmuir probe cylinder

sensor 1	Description
ezlo	electric monopole antenna, Z
lmr	langmuir probe cylinder

sensor 2	Description
bx	magnetic search coil
lp	langmuir probe sphere

10.9.4 73LFDR_AUTO_CNTL, low set, high set, average interval, time constant

This command is re-routed to the DCP by the first level command decoder (i.e. This is one of 2 HRP commands that are handled as special cases).

10.9.5 73LFDR_CHAN_CNTL, channel select

Sets the number of channels sampled by the 12 bit system.

Channel select	Description
CH0, CH1, CH2, CH3, CH4	single channel (using selected channel)
CH01	dual channel mode
CH012 PROBE	three channel mode using first three channels
CH234	three channel magnetic
CHALL	all channels

10.9.673LFDR_CMPRS_CNTL, OFF, route, word count

Used to define the length of an LFDR data capture expressed in samples. Data compression should not be enabled for the LFDR. As with Dust Analysis, the LFDR involves post processing so compression must never be enabled and the ***route*** field may contain any legal keyword.

The LFDR has a 6K byte buffer that is capable of holding 3072 samples. As the LFDR is expected to run with a low duty-cycle, a single buffer is available on the HRP for acquiring the raw data.

enable	Description
off	hardware compression not used
on	hardware compression enabled

route	Description
wc_out	EOP generated by WCR in 8237 only
wc_in	EOP generated by WCR in Actel or 8237

word count	Description
512, 1024	single channel sample count

1536, 3072	two channel word count (requires a 3-channel mode)

10.9.7 73LFDR_CNTL, select, mask, channels

Analysis control command delivered to the analysis process resident on the DCP.

select	Description
log	
linear	

The following is a hexadecimal bit mask of the channels to suppress.

mask	Description
Bit 0 ON	Suppress channel 0 analysis
Bit 1 ON	Suppress channel 1 analysis
Bit 2 ON	Suppress channel 2 analysis
Bit 3 ON	Suppress channel 3 analysis
Bit 4 ON	Suppress channel 4 analysis
0x1A	Default value Ex, Bx

channels	Description
0..255	

10.9.8 73LFDR_GAIN_CNTL, control, gain 0, gain 1, gain 2-3-4

Selects the gain setting for the 12 bit system. There are 3 separate controls with channel 0 and 1 having independent control and channels 2, 3, and 4 making use of a common gain control.

Version V2.6 software adds an enhancement to the AGC function by adding a control field selection that skips setting the gain when the AGC is set to OFF.

control	Description
man	manual gain control
auto	automatic gain control (data must be routed to DCP for AGC to function)

gain0, gain1, gain234	Description
0 10 20 40	gain level, in dB

10.9.9 00LFDR_AUTO_SET, agc, channel, gain 0, gain 1, gain 2-3-4

This internal command is used to select a gain settings independently for the primary and secondary bands (this is how we manage gain when operating in a toggle mode). The *agc* field selects manual(MAN) or automatic(AUTO) gain. The channel field selects the primary(PRI) or secondary(SEC) band, and the three gain fields are similar to the gain selects in the **73LFDR_GAIN_CNTL** command.

10.9.10 73LFDR_TOGGLE_CNTL, control, gain 0, gain 1, gain 2-3-4 (V2.6)

Selects the gain setting for the *alternate band* in toggle mode.

A toggle mode was added in the V2.6 software to allow LFDR data, to be collected in an alternate (or secondary) band, to replace a failing MFR band 2, at the same time as data is being collected in a primary band. This is accomplished by having LFDR flip the band selection with each data acquisition (we are still limited by the single 5 channel 12 bit data acquisition system on the HRP as well as the CPU cycles available on the DCP to perform data analysis) . To avoid gain problems, a separate gain state is maintained for the high band activity and this command is used to set this gain state.

It is important to keep the configuration of the LFDR in mind when discussing primary and alternate bands. Normally LFDR would be setup to operate in the low band (at least that's the way we ran it prior to the introduction of the TOGGLE mode) with the TOGGLE enable allowing collection of high band data. The correct image to use, however, is that of setting up the LFDR primary mode (channel selection, band, gains settings, timing, delivery, etc.) and then providing a starting gain point and enabling the toggle mode with this command. On odd acquisitions, the band select bit is flipped and the secondary gain (supplied in this command, updated as needed when AGC is on) is used to collect data. On even acquisitions, the primary gain setting along with the selected band is used. All the other settings are identical (no timing, channel, or antenna changes are allowed).

If you happen to setup the LFDR in high band (**73LFDR_MODE_CNTL, HBAND,...**) the gain selection in the **73LFDR_TOGGLE_CNTL** command would then specify the low band gain setting.

If LFDR is being operated exclusively in one band, the this command would not be used.

control	Description
OFF	Disable toggle mode
MAN	Enable TOGGLE mode without AGC control
AUTO	Enable TOGGLE mode ***with*** AGC control

gain0, gain1, gain234	Description
0 10 20 40	gain level, in dB

10.9.10.1Compatibility note

This command is implemented as an extension to the LFDR gain command (**73LFDR_GAIN_CNTL**) in a manner that is reasonably compatible with earlier version of the flight software. This trigger can be used with earlier versions of software as long as the trigger is immediately followed by a gain control command. The reason for this restriction is that older versions of the flight software will decode the **73LFDR_TOGGLE_CNTL** command as a **73LFDR_GAIN_CNTL** command and make use of the indicated gain levels. By following the toggle command with a normal gain command, the effect in prior versions is nullified while newer version will act accordingly.

In particular, trigger 10 will need to shut down toggle mode in a manner that is as universal as possible, and this is simply accomplished by adding the gain control command immediately prior to the gain control command (this was done around the time of the C32 submission).

10.9.10.2 Effects of AGC when operating in Toggle mode.

When we are operating in toggle mode, the AGC activity (that executes on DCP) has no means to pick the dataset it will perform gain analysis on. If there are few CPU cycles available to the gain process, it is possible for the AGC to become erratic. Operating in toggle mode increases the CPU load on the DCP (cycles needed to perform sufficient gain analysis to effect change on both channels on both bands).

AGC analysis is a low priority task on DCP

10.9.11 73LFDR_MODE_CNTL, HOLD, band, NOCOMPRESS

Selects sampling rate (140uSec or 10mSec) and anti-aliasing filter.

band	Description
lband	10 mSec Sample rate
hband	140 uSec sample rate

compression	Description
nocompress	no hardware compression
compress	hardware compression (ISFLIP chip) LFDR can **not** function with compressed data

10.9.12 73MEM_TWEAK, HRP, WORD, 0x18, 1, LFDC

This tweak may be used to bring the LFDR timing control process out of a delayed state when changing instrument modes. Many of the LFDR bit rates require rather long cycle periods and changing this period takes effect following the next scheduled data acquisition. If LFDR timing control is not currently delayed this tweak will have no effect. See the WFR command section for a discussion of the placement of this command.

10.9.1373MEM_TWEAK, HRP, WORD, 0x50, *nnnn*, LFDC

Scheduling period expressed in RTI ticks.

10.9.14 73MEM_TWEAK, HRP, WORD, 0x52, *nnnn*, LFDC

Scheduling offset expressed in RTI periods.

10.9.15 73MEM_TWEAK, HRP, WORD, 0x54, *nnnn*, LFDC

Delay (expressed in RTI's) between minipackets. Used to throttle the data delivered to the DCP.

10.9.16 73MEM_TWEAK, HRP, BYTE, 0x56, *n*, LFDC

Scheduling mode.

00 Stop

Data acquisition is stopped. ***73LFDR_MODE_CNTL, TRIGGER*** is not processed. Internal triggers are not processed. This state is entered whenever sleep is asserted.

01 Idle

Data acquisition is stopped but the process will accept a trigger to perform a single acquisition. After the acquisition the process will return to idle to await further triggers.

12 Run

Data acquisition is continuous and based on the schedule specified at offset 0x50 and 0x52.

23 Trigger

- Single data acquisition.

10.9.17 73MEM_TWEAK, HRP, WORD, 0x60, *nnnn*, LFDC

LFDR data routing control. It was not expected that this capability would be required, so the command decoder for this function is not implemented in the LFDR command decoder. Since the LFDR makes use of the same routing method as the WFR, it is possible to reroute data when needed.

Nnnn	Description
0x0343	Normal, data routed to DCP for analysis
0x0245	Route data to DCP for WFR compression
0x03C4	Route to WBR LRS queue (dual WFR activities)

Note the second route appears on the ground as WFR data.

10.9.18 73MEM_TWEAK, HRP, BYTE, 0x50, *nn*, W12J (affects WFR)

RST-5 Clock Enable flag. This field is normally set to a -1 and should not be altered. When cleared to zero, it causes the sample clock to the 12 bit A/D system to be stopped when data is not being actively acquired. This clock does not appear to interfere with anything within the instrument (i.e. the WBR).

Clearing this flag prevents WBR data acquisition when LBAND data is being acquired.

10.9.19 73MEM_TWEAK, HRP, WORD, 0x52, *nn*, W12J (affects WFR)

Gain change settling time. This field is normally a 2 to allow at least 125 mSec of settling time prior to data acquisition in the event that a gain change has occurred. If both LFDR and WFR are operating, setting this field to zero will probably cause problems. Gain level changes (i.e. auto gain) and antenna selection changes require this field to be set to 2 or greater.

10.9.20 73MEM_TWEAK, HRP, BYTE, 0x00, *nn*, LFDX

LFDR dual data routing control. The copy of the waveform analyzed by the LFDR process may be routed to the ground using this memory tweak. This feature is intended for debugging and checkout activities.

Nnnn	Description
0x00	Normal, data routed to DCP for analysis
0x45	Dual routing Route data to DCP for WFR compression
0xC3	Dual routing Route data to LRP (unpacked raw data)
0x01	Dual Routing to HRS **ERROR - NOT FUNCTIONAL**

When using the dual routing capability with WFR active, the WFR data **must** be routed to the HRS stream. Failure to observer this restriction will result in unrecoverable data sets. There are several reasons for this behavior. First or all, the LFDR raw data is simply WFR data that would normally be routed to the DCP for the FFT analysis. The result of this analysis is packaged as an LFDR minipacket that is forwarded for delivery to the ground. There is no way to distinguish the raw data from WFR data. If both the LFDR raw data and the WFR data are delivered to the *C3* queue on the LRP, the BIU handler will mix the data from the two sources together deleting some data in an attempt to keep the segmentation information sequential.

If one were to attempt something deviant, such as routing the raw LFDR data to and idle queue (such as the LP queue into the BIU handler, assuming that LP is inactive), the data might be recoverable on the ground but the task of sorting data in the presence of dropouts becomes a daunting task.

The only workable solution is to make use of the HRS data path to deliver one of the receiver streams to the ground and the LRS stream to deliver the other. Dual routing the LFDR data through the HRS stream seems to cause the HRP to hang, so this leaves HRS available only to the WFR so that LFDR raw data can be routed to LRS (i.e. queue C3 or queue 45).

10.9.21 73MEM_TWEAK, HRP, BYTE, 0x6A, *n*, LFDX

Minipacket delivery flags.

These values are added to obtain combined results. In particular, this location must be tweaked when high speed LFDR data is being delivered to the DCP for analysis.

0 Nominal behavior

Do not perform checksumming and delay 1 RTI between IPC packet delivery. This is appropriate for the limited data rates of the LFDR.

1 Checksum enable (not required by the analysis routines on the DCP)

Setting this bit causes a checksum to be calculated on each IPC packet sent to the IX queue for delivery.

2 Non blocking

Setting this bit will cause minipacket processing to abort whenever the free queue is empty (i.e. a conditional queue read is used to obtain delivery buffers and a read failure causes processing to be aborted).

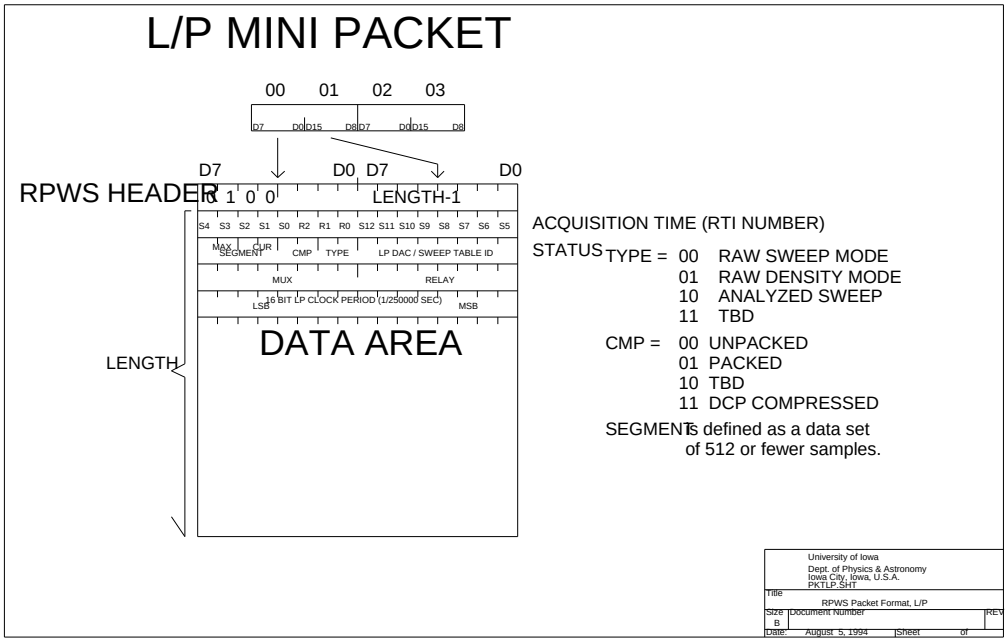
4 Fast delivery

Setting this bit suppresses the 1 RTI delay between each IPC packet normally imposed by the minipacket assembler.

Note that in most LFDR setups the default value for this field is used. IEB loads prior to C37 did not alter this field and the idle trigger did not restore this field to a proper defualt value. When changing this field, it must be changed back to the correct default setting.

10.10 L/P Control

Langmuir Probe.



Field	***Field Location***	***Description***
Minipacket ID	Byte 0 / D7..D4	1000
Minipacket Length	Byte 0 / D3..D0 Byte 1 / D7..D0	12 bit length
RTI	Byte 2 S4..S0, R2..R0 Byte 3 S12..S5	RTI time of data acquisition Seconds & RTI
Type	Byte 4 / D0..D1	Packet Type
Cmp	Byte 4 / D2..D3	Compression Type
Segment Cur	Byte 4 / D4..D5	Segmentation, Current Index
Segment Max	Byte 4 / D6..D7	Segmentation, Maximum Index
LP DAC Sweep Table ID	Byte 5	DAC value (density) Sweep Table (sweep)
Mux	Byte 6	Mux setting (bit pattern)
Relay	Byte 7	Relay setting (bit pattern)

10.10.0.1 Type

Packet type. Used to distinguish between Sweep and Density packets.

10.10.0.2 Compression

Compression status, indicating the compression applied to the data.

10.10.0.3 Segmentation

Minipacket segmentation control. Used to allow data packets in excess of 4K bytes to occur (there is only 12 bit length field in the minipacket).

10.10.0.4 DAC setting

DENSITY MODE: Value that appears in the DAC for this data set when it was captured.

10.10.0.5 Sweep Table

SWEEP MODE: Indicates the waveform applied as the sweep occurs.

10.10.0.6 Mux

Multiplexer control bits.

10.10.0.7 Relay

Relay control bits.

10.10.1 L/P Timing issues (sweep warning)

Langmuir Probe stimulates the plasma by biasing one of the sensors during a density sweep. This has the potential to interfere with other instrument sensors as well as other instruments. Other instruments on the spacecraft are warned using the sweep warning flag that is broadcast in the ancillary data (L/P handler sends notification through the LRP).

The sweep warning flag is sent one second before the sweep occurs. At this time the sensor bias is changed to the 1st. value in the sweep table to allow adequate settling time (for the L/P sweep that is about to occur). This bias change is evident in most of the other RPWS receivers

Therefore, when scheduling L/P sweeps, allow for the bias change that occurs 1 second prior to the beginning of the sweep.

10.10.2 DCP Resource Requirements

Langmuir Probe makes use of the DCP to pack or compress the collected data. Since the L/P makes use of a 12 bit A/D convertor (and we place each 12 bit sample into a 16 bit memory location), the data may be passed to the ground in raw form, bypassing the DCP altogether. For minimum impact, the DCP may be commanded to pack the data, requiring about 1 second of CPU time for each 512 sample minipacket.

DCP CPU requirements	Description
0	RAW (bypass DCP)
1 sec / packet	PACK
	RICE
	WALSH
	WALSH/RICE

10.10.3 73LP_MUX0_CNTL, function, state

Multiplexer control. The parameter ***function*** selects one of the bits within the mux to changes to the specified ***state***. This command does not apply to Langmuir Probe sweep mode; when a sweep is done, the mux is always set to value 56 hex, meaning A/D connected to Sphere, High-Pass Filter, and Sphere bias full range (-30.75 volts to +32.30 volts nominally for a ΔV of 63.05 volts).

function	Description
bit0..bit7	multiplexer bit number

state	Description
off	clear the selected bit
on	set the selected bit

Multiplexer Function	D7	D6	D5	D4	D3	D2	D1	D0
ADCP1 A/D Cylinder Probe Ex-			X				0	1
ADCP2 A/D Cylinder probe Ex+			X				0	0
ADSP A/D Spherical Probe			X				1	X
HPF High Pass Filter			X			1		
LPF Low Pass Filter			X			0		
SPBFR Sphere bias full range -30.75 to +32.30 volts		1	X	1	0			
SPBLR Sphere bias low range -9.91 to -2.36 volts		0	X	1	0			
SPBMR Sphere bias medium range -3.98 to +3.60 volts		0	X	0	X			
SPBHR Sphere bias high range +1.98 to 9.60 volts		0	X	1	1			
CYBFR Cylinder bias full range -12.26 to +12.79 volts	1		X					
CYBMR cylinder bias medium range -4.69 to +4.65 volts	0		X					

Low pass filter characteristics

Gain	Frequency
0.98 dB	DC
-1 dB	3.0 Hz
-3 dB	5.4 Hz
-6 dB	6.3 Hz
delay 218 mS in passband	

10.10.4 73LP_RELAY_CNTL, function, state

Latching relay control. The parameter ***function*** selects one of five relays to switch into one of two states as specified by ***state***.

function	Description
relay1..relay6	Relay to change (or set)

state	Description
coil_a	switch latching relay to coil A side
coil_b	switch latching relay to coil B side

Relay 1 through 4 control connection of the dipole (the electric dipole referred to as X+ and X- with the other receivers).

Relay 5 controls the Preamplifier gain setting.

	Coil A	Coil B	Function
Relay 1	disconnect	connect	LMR X- I+
Relay 2	disconnect	connect	LMR X- I-
Relay 3	disconnect	connect	LMR X+ I+
Relay 4	disconnect	connect	LMR X+ I-
Relay 5	High	Low	Pre amp gain
Relay 6	Not implemented		

10.10.5 73LP_SET_PARAM, function, value

Miscellaneous parameter control.

function	Description
tsamp	set sample rate
dmblok	density mode block size
swipid	select sweep table
func7	density mode sample rate

73LP_SET_PARAM, TSAMP, nnnn sets the sample rate for sweep mode. Nnnn is a 14-bit number which specifies how much the 125 kHz clock should be divided by to determine the sweep mode sample rate. The default value is 125, so that 1000 Hz is the sample frequency.

73LP_SET_PARAM, DMBLOK, nnnn sets the density mode block size, i.e., how many samples will be taken for each density mode measurement. The default value is 256. The maximum allowable size is 2048, since the Langmuir Probe dedicated buffer is 4096 bytes long, and each 12-bit sample requires 2 bytes of storage.

73LP_SET_PARAM, SWPID, nnnn specifies which internally defined sweep table should be used for the sweeps. The following are the possibilities:

> 0: simple linear up-going ramp, 512 samples, DAC0 starting at 0, going to FF hex, with a dwell time of two samples at each DAC0 setting. Sweep takes 0.512 seconds at 1000 Hz sample rate.
>
> 1: linear down-up ramp, 1024 samples, DAC0 starting at FF hex, going to 0 hex, then back up to FF hex with a dwell time of two samples at each DAC0 setting, except at 0, where there is a dwell time of 4 samples. Sweep takes 1.024 seconds at 1000 Hz sample rate.
>
> 2: logarithmic up-going ramp, 256 samples, DAC0 set to hex values 00, 40, 60, 70, 78, 7C, 7E, 7F, 80, 81, 83, 87, 8F, 9F, BF, FF with each setting at a dwell time of 16 samples. Sweep takes 0.256 seconds at 1000 Hz sample rate.
>
> 3: logarithmic up-going ramp, 256 samples, DAC0 set to hex values 00, 04, 08, 0C, 10, 14, 18, 1C, 20, 24, 28, 2C, 30, 34, 38, 3C, 40, 42, 44, 46, 48, 4A, 4C, 4E, 50, 52, 54, 56, 58, 5A, 5C, 5E, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 6A, 6B, 6C, 6D, 6E, 6F with each setting at a dwell time of 1 sample. Then DAC0 set to hex values 70, 71, 72, 73, 74, 75, 76, 77 with each setting at a dwell time of 2 samples. Then DAC0 set to hex values 78, 79, 7A, 7B with each setting at a dwell time of 4 samples. Then DAC0 set to hex values 7C, 7D with each setting at a dwell time of 8 samples. Then DAC0 set to hex values 7E, 7F, 80, 81 with each setting at a dwell time of 16 samples. Then DAC0 set to hex values 82, 83 with each setting at a dwell time of 8 samples. Then DAC0 set to hex values 84, 85, 86, 87 with each setting at a dwell time of 4 samples. Then DAC0 set to hex values 88, 89, 8A, 8B, 8C, 8D, 8E, 8F with each setting at a dwell time of 2 samples. Finally, DAC0 set to hex values 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 9A, 9B, 9C, 9D, 9E, 9F, A1, A3, A5, A7, A9, AB, AD, AF, B1, B3, B5, B7, B9, BB, BD, BF, C3, C7, CB, CF, D3, D7, DB, DF, E3, E7, EB, EF, F3, F7, FB, FF with each setting at a dwell time of 1 sample. Sweep takes 0.256 seconds at 1000 Hz sample rate.

73LP_SET_PARAM, FUNC7, nnnn sets the sample rate for density mode. Nnnn is a 14-bit number which specifies how much the 125 kHz clock should be divided by to determine the sweep mode sample rate. The default value is 6250, so that 20 Hz is the sample frequency. The maximum value which can be programmed into the 14-bit counter is 16383, which would limit the density mode sample frequency to 7.63 Hz. If lower sample rates are desired, there is a "Slow Mode" possibility, but the timing is under software control instead of the hardware clock. If nnnn is greater than 16384, then the lower 14 bits of nnnn are interpreted as a delay count between samples, where the count is given in RTI's, i.e., units of 125 milliseconds. For example, if nnnn is specified as 16385, then the sample frequency is 8 Hz. But there can be sample jitter in this mode.

10.10.6 73LP_TRIGGER, type, destination

This command does nothing.

10.10.7 73LP_VOLT_CNTL, lp dac, voltage 0, voltage 1

Selects the voltage to be applied to the Langmuir probe sphere and the electric antenna elements when operating in current mode (i.e. as part of the Langmuir probe). The selected bias requires an appropriate relay and mux setting before any voltage is applied to the sensors or antenna element.

If the Langmuir Probe is idle it is necessary to specify ***BOTH*** for the ***lp dac*** parameter in order to immediately effect a change on DAC0 (the sphere). If ***lp dac*** is DAC1 or BOTH, then the command is executed immediately. If ***lp dac*** is DAC0, then the setting takes effect at the next cycle time. The default cycle time is 16 seconds, so the DAC0 setting may be delayed by as much as 16 seconds. The reason for this delay is that it is assumed that normally sweeps and density mode measurements are alternated, and a fixed DAC0 setting can not take effect until the next density mode measurement.

lp dac	Description
dac0	sphere bias voltage
dac1	cylinder bias voltage
both	set both

voltage 0	Description
0..255	value applied to sphere (DAC 0) (if selected)

voltage 1	Description
0..255	value applied to cylinder (DAC 1) (if selected)

10.10.8 73MEM_TWEAK, HRP, BYTE, 0x30, *n*, LP_I

The byte at offset 30 controls operation of the 2 portions of the Langmuir Probe receiver. The upper nibble, when non-zero enables A/D converter only operations (referred to as density mode) while the lower nibble enables the D/A and A/D together (referred to as sweeps). If the value of *n* is FE hex, then a "smart sweep" is done. This means that after the sweep, an analysis of the sweep data is done, and the DAC0 setting for the density mode measurement is determined automatically. The algorithm used is to scan the 12-bit sweep data for the closest value to 800 hex, which represents zero current. The DAC value corresponding to this measurement is then added to an offset to determine the DAC0 value to use for the density mode measurement. The default for the offset is 40 hex, but this can be modified via a mem_tweak command. In summary,

n	Description
0xF0	enables density mode measurements only
0x0F	enables Langmuir Probe sweeps only
0xFF	*enables alternating sweeps and density mode measurements*
0xFE	*enables alternating sweeps and density mode measurements, with automatic setting of the DAC0 value for the density mode measurement*

10.10.9 73MEM_TWEAK, HRP, BYTE, 0x31, *n*, LP_I

The byte at offset 31 specifies the offset value to be used when doing a "smart sweep". The default value is 40 hex, which is a 16-volt offset at Full Range.

10.10.10 73MEM_TWEAK, HRP, WORD, 0x82, *n*, LP_I

The word at offset 82 specifies the cycle period in RTIs, i.e., the delta-T between sweeps. The default value is 128 (16 seconds).

10.10.11 73MEM_TWEAK, HRP, WORD, 0x84, *n*, LP_I

The word at offset 84 specifies the RTI # when the 1-second warning is given just prior to a sweep. The default value is 6.

Note that operating the Langmuir Probe in sweep mode requires 8237 resources that are shared. The other users of this resource are the WBR data acquisition and the High Rate Science formatting routine.

10.10.12 73Mem_Tweak, HRP, Byte, 0x98, n LP_I

The byte at offset 98 hex specifies the Langmuir Probe data destination queue. The default value for this is 46 hex, which corresponds to the DCP LP compression queue. Another useful value for this is CE hex, which corresponds to the LRP LP data queue. Specifying this value bypasses the DCP compression process, allowing faster output of the LP data; however the data would be unpacked (12-bit values in 16-bit words). If the queue destination is changed for some operation, then it should be reset to the default destination later when the operation is finished.

10.11 Misc. Utility, TWEAK

Miscellaneous utility commands. These commands are processed by the memory tweak process (not directly by any subsystem handler). Although these commands are rather generic in nature, they are used to control instrument operation in ways that the documented command structure is inadequate.

Note that the command decoder in the **TWEK** process is limited to 127 bytes of commands in any buffer. This affects IEB loads by limiting the number of consecutive 73MEM_TWEAK commands to 12 (each command requires 10 bytes).

These miscellaneous commands may be send when the instrument is in *SLEEP* mode for immediate execution as they are not dependent on any power setting. Keep in mind that both DCP and HRP are operating at reduced clock speed to conserve power when the instrument is in SLEEP mode exacting a performance penalty.

It is also interesting to take note of the internal process structure of the handler for these commands as it affects some operations (particularly MRO activities). The modules that handles this group of commands is internally named "TWEK" and you will notice that modifications are made to the process descriptor to control the bank select mechanism. The TWEK module is identical on all processors to reduce the number of ALF records required to load the instrument. The code, being identical on all processors, makes use of the destination field within the IPC packet used to deliver commands to determine the processor the command is being handled on. Following local processing the command is passed on to the next processor in the chain for processing. The command is routed from the 1st. level command decoder to the LRP followed by the HRP, the DCP, and finally back to the LRP where the loop count is incremented.

Each command buffer is completely processed before being passed on the next processor. In the case of MEM_TWEAK and PORT_TWEAK commands the processing delay is negligible and may be ignored. In the case of the MRO command, however, the processing time can be significant due to bit rate limiting discussed in the previous paragraph. Although there is no practical way to overlap MRO commands destined for the housekeeping, the science data rate is typically sufficient to handle concurrent dumps from all three processors. Generating a long MRO can cause significant problems as many internal activities require the use of the MEM_TWEAK. S/C clock is updated on DCP and HRP using the MEM_TWEAK mechanism, L/P activities are communicated from HRP to LRP using the MEM_TWEAK mechanism, and HFR activities are communicated from LRP to HRP using the MEM_TWEK mechanism.

Assuming that internal activities can be temporarily suppressed (there are mechanisms in place to accomplish this), concurrent MRO activity can occur when properly coordinated. First there is a specific order that the 3 MRO commands must be issued in, 1st. to the DCP, 2nd. to the HRP and 3rd. to the LRP. Altering this order will cause one of the MRO commands to become blocked and the MRO will not occur concurrently. Second, the size of the MRO blocks must be equal (or the time allowed for the dump to occur must be equal) to allow all commands to clear through all of the TWEK processes.

10.11.1 73MEM_TWEAK, processor, size, offset, value, process name

Memory Tweak command. Used to alter the contents of memory in any of the processors. This command alters memory within a selected process so the address is expressed relative to the process descriptor.

NO MORE THAN 12 MEM_TWEAK commands (120 total bytes) may be grouped together for delivery to the TWEAK process. Part of the error checking imposes this limit. In particular, IEB sequences are susceptible to violating this constraint.

processor	Description
lrp	execute on LRP only
hrp	execute on HRP only
dcp	execute on DCP only
all	execute on all processors

The processor select field is a bit field. The implementation of the command looks at each individual processor bit so it is possible to build a bit pattern that selects 2 of the 3 processors. We have chosen no to document this capability in the JPL command document, but the capability exists in the software.

size	Description
byte	alter 8 bit value
word	alter 16 bit value (little-endian)

offset	Description
0..0xFFFF	Bench (Iowa) new word value in hexadecimal notation
0..FFFF	Flight (JPL) new word value in hexadecimal notation

When building commands for execution at JPL, this field is a hexadecimal number, it may ONLY be expressed in hexadecimal. The Command parser used on the bench (in Iowa) is slightly more flexible in that it will accept both hexadecimal and decimal numbers. When supplying a value on the bench, use the c-convention of 0x preceding hexadecimal and supplying decimal directly.

value	Description
0..0xFF 0..0xFFFF	Bench (Iowa) new byte value in hexadecimal notation new word value in hexadecimal notation
0..256 0..65535	Bench (Iowa) new byte value in decimal notation new word value in decimal notation
0..FF 0..FFFF	Flight (JPL) new byte value in hexadecimal notation new word value in hexadecimal notation

As with the offset, this number is a hexadecimal only field when building commands for the spacecraft and either when building commands on the bench model.

process name	Description
A..Z, 0..9, _	process name

Process name is a 4 character name. Valid characters are A-Z and 0-9 as well as a space character. Lowercase is not used (and there better not be anyone generating a Process Descriptor within the instrument that violates this rule).

By using a *process-relative* addressing scheme, we can re-assign memory addresses within the processor complex without the need to alter any of the **73MEM_TWEAK** commands. This **73MEM_TWEAK** is used to augment the command structure by altering flags and settings that occur a fixed locations within the instrument handlers.

Code patches would be built using absolute memory addresses using the **00MEM_TWEAK** command that follows. Code patches, of course, would be used on a case by case basis (i.e. we stop using them when the next software release is uploaded).

Finally, the **00MEM_TWEAK** command is used to patch values in the system data page. The system data page occurs at a fixed location in memory so software updates do not alter the allocations.

10.11.2 00MEM_TWEAK, processor, size, address, value

Memory Tweak Command. Used to alter the contents of an absolute address in memory. This command omits the *process name* found in ***73MEM_TWEAK*** command so the address is not *process relative* but an *absolute address*.

NO MORE THAN 20 MEM_TWEAK commands may be grouped together for delivery to the TWEAK process (these commands are only 3 words long, hence the 20 command limit, but we are still restricted to 60 words in a command packet). Part of the error checking imposes this limit. In particular, IEB sequences are susceptible to violating this constraint.

processor	Description
lrp	execute on LRP only
hrp	execute on HRP only
dcp	execute on DCP only
all	execute on all processors

See 73**MEM_TWEAK** for additional comments about the processor selection field.

size	Description
byte	alter 8 bit value
word	alter 16 bit value (little-endian)

address	Description
0x0..0xFFFF	absolute memory address

value	Description
0..0xFF 0..0xFFFF	new byte value new word value

10.11.3 00PROCESS_CREATE, processor, address UNIMPLEMENTED

Unimplemented in the current version(s) of software due to memory constraints.

This command was intended to allow creation of a process that had, for example, been placed in memory by a series of ***00MEM_TWEAK*** commands. In the current design, it is actually simpler to build the additional function and resubmit the entire software load.

processor	Description
lrp	execute on LRP only
hrp	execute on HRP only
dcp	execute on DCP only

address	Description
0x0..0xFFFF	absolute memory address

10.11.4 00PORT_TWEAK, processor, port address, value

Port Tweak Command. Writes the specified pattern to the specified port.

NO MORE THAN 20 PORT_TWEAK commands may be grouped together for delivery to the TWEAK process (again, 3 word command here, still limited to 60 words per command block). Part of the error checking imposes this limit. In particular, IEB sequences are susceptible to violating this constraint.

Processor	Description
lrp	execute on LRP only
hrp	execute on HRP only
dcp	execute on DCP only
all	execute on all processors

See 73**MEM_TWEAK** for additional comments about the processor selection field.

port address	Description
0x00..0xFF	port address to be modified

value	Description
0x00..0xFF	new value, always 8 bit

Port Tweaks for RTI/Dead Time Control

10.11.5 73PORT_TWEAK, LRP, 0x65, RTI_period
10.11.6 73PORT_TWEAK, LRP, 0x66, DTS_period

The first tweak alters the timing values used to simulate RTI interrupts when S/C does not deliver RTI pulses. The second tweak controls when the Dead Time Start interrupt occurs.

The RTI transaction should be available at all times but in the event that the RTI transaction is not delivered by the S/C the hardware will generate an RTI interrupt signal to the processor 125.056 mS after the last RTI received from the S/C. Timing of this interrupt is not terribly critical so if S/C is a little late delivering the interrupt it has little impact on operations. The software keeps a count of the number of lost RTI interrupts and presents the lost interrupt count in housekeeping.

Selecting a value for the RTI register that is close to the actual period will allow the instrument to freewheel through periods when the RTI does not appear. Having the period a little short would allow RTI interrupt processing to commence a little earlier but may cause the lost RTI counter to increment, although this would have no other deleterious effects as long as S/C continues to supply the RTI signal (loss of more than one RTI begins to present problems if the programmed period is short).

The DTS transaction is generated by the S/C as the last transaction during the RTI period. The DTS transaction should occur no later than 120mS into the RTI period, allowing the instrument a minimum of 5mS to perform housekeeping tasks between each group of transactions. Since the DTS transaction is tied to the level of activity on the S/C bus, it can occur almost anywhere in the RTI period (while keeping out of the last 5mS). Keeping in mind that the processor performs part of the BIU processing in a dead-time interrupt handler and part of the processing following the RTI interrupt (i.e. the handler is a process and it schedules using the DELAY system service), it is important the DTS interrupt not occur too soon. The solution is two-pronged.

First, the DTS interrupt is simulated by the Actel gate array on the LRP. Unlike the RTI interrupt, the simulated DTS is the primary source of the processor (non-maskable) interrupt. The timer is reset when the RTI occurs (real or simulated) and when it expires the DTS interrupt is asserted.

The Second part of the DTS interrupt architecture is that the interrupt is masked within the interrupt and un-masked by the BIU handler when it is permissible to process the next interrupt. This prevents interrupts from overrunning the software.

Finally, the hardware on the LRP that generates RTI/DTS also logs the occurrence of the actual signal from the BIU. The processor can determine, in the RTI and DTS interrupt routines, if BIU supplied a RTI/DTS signal (this works because the interrupt processing will always occur after the actual interrupt should have occurred, the DTS from S/C appears a little to soon for our purposes).

The tables on the following pages may be used to select values to load into the timing control registers for RTI and DTS interrupts. Values are identical for both registers.

Value	Time	Value	Time	Value	Time
-00	131.008	31	127.872	62	124.736
01	130.944	32	127.808	63	124.672
02	130.880	33	127.744	64	124.608
03	130.816	34	127.680	65	124.544
04	130.752	35	127.616	66	124.480
05	130.688	36	127.552	67	124.416
06	130.624	37	127.488	68	124.352
07	130.560	38	127.424	69	124.288
08	130.496	39	127.360	6A	124.224
09	130.432	3A	127.296	6B	124.160
0A	130.368	3B	127.232	6C	124.096
0B	130.304	3C	127.168	-6D	124.032
0C	130.240	3D	127.104	6E	123.968
0D	130.176	-3E	127.040	6F	123.904
0E	130.112	3F	126.976	70	123.840
-0F	130.048	40	126.912	71	123.776
10	129.984	41	126.848	72	123.712
11	129.920	42	126.784	73	123.648
12	129.856	43	126.720	74	123.584
13	129.792	44	126.656	75	123.520
14	129.728	45	126.592	76	123.456
15	129.664	46	126.528	77	123.392
16	129.600	47	126.464	78	123.328
17	129.536	48	126.400	79	123.264
18	129.472	49	126.336	7A	123.200
19	129.408	4A	126.272	7B	123.136
1A	129.344	4B	126.208	7C	123.072
1B	129.280	4C	126.144	-7D	123.008
1C	129.216	4D	126.080	7E	122.944
1D	129.152	-4E	126.016	7F	122.880
1E	129.088	4F	125.952	80	122.816
-1F	129.024	50	125.888	81	122.752
20	128.960	51	125.824	82	122.688
21	128.896	52	125.760	83	122.624
22	128.832	53	125.696	84	122.560
23	128.768	54	125.632	85	122.496
24	128.704	55	125.568	86	122.432
25	128.640	56	125.504	87	122.368
26	128.576	57	125.440	88	122.304
27	128.512	58	125.376	89	122.240
28	128.448	59	125.312	8A	122.176
29	128.384	5A	125.248	8B	122.112
2A	128.320	5B	125.184	-8C	122.048
2B	128.256	5C	125.120	8D	121.984
2C	128.192	RT-5D	125.056	8E	121.920
2D	128.128	5E	124.992	8F	121.856
2E	128.064	5F	124.928	90	121.792
-2F	128.000	60	124.864	91	121.728
30	127.936	61	124.800	92	121.664

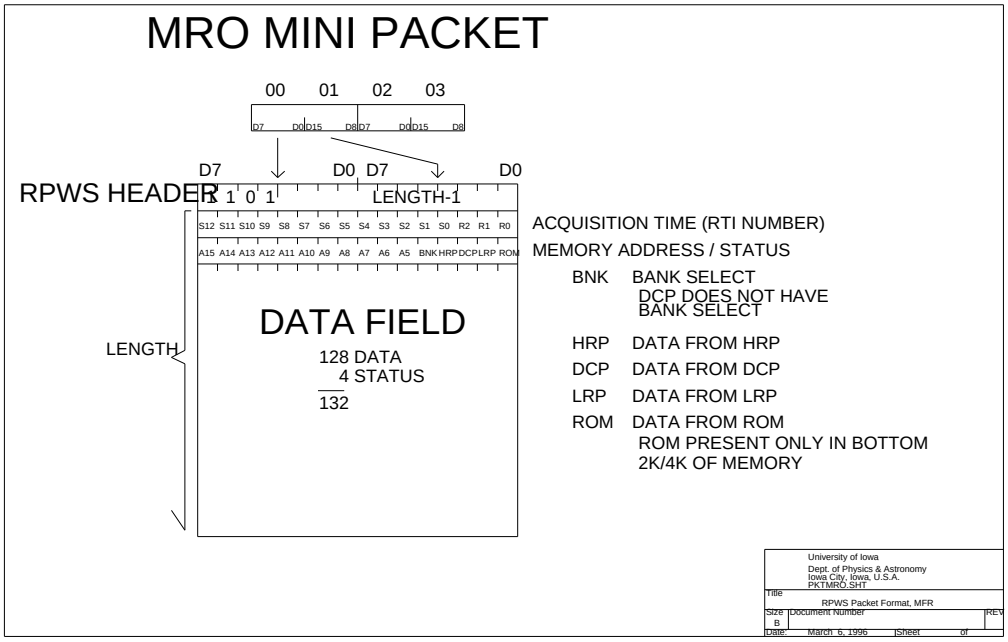
Value	Time
93	121.600
94	121.536
95	121.472
96	121.408
97	121.344
98	121.280
99	121.216
9A	121.152
9B	121.088
-9C	121.024
9D	120.960
9E	120.896
9F	120.832
A0	120.768
A1	120.704
A2	120.640
A3	120.576
A4	120.512
A5	120.448
A6	120.384
A7	120.320
A8	120.256
A9	120.192
AA	120.128
AB	120.064
AC	120.000
DT-AD	119.936
AE	119.872
AF	119.808
B0	119.744
B1	119.680
B2	119.616
B3	119.552
B4	119.488
B5	119.424
B6	119.360

Value	Time
B7	119.296
B8	119.232
B9	119.168
BA	119.104
-BB	119.040
BC	118.976
BD	118.912
BE	118.848
BF	118.784
C0	118.720
C1	118.656
C2	118.592
C3	118.528
C4	118.464
C5	118.400
C6	118.336
C7	118.272
C8	118.208
C9	118.144
CA	118.080
-CB	118.016
CC	117.952
CD	117.888
CE	117.824
CF	117.760
D0	117.696
D1	117.632
D2	117.568
D3	117.504
D4	117.440
D5	117.376
D6	117.312
D7	117.248
D8	117.184
D9	117.120
-DA	117.056

Value	Time
DB	116.992
DC	116.928
DD	116.864
DE	116.800
DF	116.736
E0	116.672
E1	116.608
E2	116.544
E3	116.480
E4	116.416
E5	116.352
E6	116.288
E7	116.224
E8	116.160
E9	116.096
-EA	116.032
EB	115.968
EC	115.904
ED	115.840
EE	115.776
EF	115.712
F0	115.648
F1	115.584
F2	115.520
F3	115.456
F4	115.392
F5	115.328
F6	115.264
F7	115.200
F8	115.136
F9	115.072
-FA	115.008
FB	114.944
FC	114.880
FD	114.816
FE	114.752
FF	114.688

10.11.7 73MRO, processor, stream, start, stop

Memory Read Out Command. Used to dump memory. Note that this command is capable of delivering the readout to either the housekeeping stream of the science telemetry stream. The size of the data packet is dependent on the delivery stream. Housekeeping packets are a total of 16 byte, 10 of which are memory data. The telemetry stream, being higher bandwidth, allows for 192 bye memory blocks to be dumped. Also, the *stop* address may be specified as zero, forcing a single packet to be delivered.



The MRO command is bit rate limited with the data rate dictated by the destination of the read out. Housekeeping data is limited to approximately 10 bits per second (a 10 byte dump record is generated every 64 RTI periods) while Science telemetry is limited to 1000 bits per second (actually a 128 byte dump is generated every 9 RTI periods).

Processor	Description
lrp	execute on LRP only
hrp	execute on HRP only
dcp	execute on DCP only
all	execute on all processors

stream	Description
hsk	dump delivered in the housekeeping data data rate is approximately 10 bits/sec
tlm	dump delivered in the science data data rate is approximately 1000 bits/sec

The ***start*** address is masked to allow room for status bit to be embedded in the minipacket. All currently know version use a granularity of 32 bytes when dumping data into the telemetry stream (The lower 5 bits indicate the source of the dump). In the case of performing a dump to housekeeping, this is assumed to be a real-time activity where the exact source of the data is know beforehand or that addresses are unique across processors.

start	Description for ***tlm stream***
0..0xFFE0	absolute memory address granularity of 32 bytes (i.e. 0x0020)

The ***start*** address is used, as is, without being masked.

start	Description for ***hsk stream***
0..0xFFFF	absolute memory address granularity of 1 byte

The ***stop*** address requires a little care in specifying. When requesting a dump that covers more than a single mini-packet (128 bytes) or micro packet (16 bytes), it works best to specify the stop address that is greater than desired. Specifying the address of the last location to appear in the last MRO packet should always work.

V2.4 changes the default packet size to 192 bytes plus overhead.

Stop	Description
0	dump a single block (size depends on stream)
1..0xFFFF	absolute memory address

MRO Packet Format TLM stream

Field	*Field Location*	*Description*
Minipacket ID	Byte 0 / D7..D4	1101
Minipacket Length	Byte 0 / D3..D0 Byte 1 / D7..D0	12 bit length
RTI	Byte 2 S4..S0, R2..R0 Byte 3 S12..S5	RTI time of memory dump Seconds & RTI
Address MSB	Byte 4 / D0..D7	Address Bits A15..A8
Address LSB	Byte 5 / D5..D7	Address bits A7..A5
Bank	Byte 5 / D4	Bank bit
HRP	Byte 5 / D3	Data from HRP
DCP	Byte 5 / D2	Data from DCP
LRP	Byte 5 / D1	Data from LRP
ROM	Byte 5 / D0	Data from ROM

MRO micro-Packet Format HSK stream

Field	*Field Location*	*Description*
MicroPacket ID	Byte 0 / D7..D4	1101
MicroPacket Length	Byte 0 / D3..D0 Byte 1 / D7..D0	12 bit length
RTI	Byte 2 S4..S0, R2..R0 Byte 3 S12..S5	RTI time of memory dump Seconds & RTI
Address MSB	Byte 4 / D0..D7	Address Bits A15..A8
Address LSB	Byte 5 / D0..D7	Address bits A7..A0
Memory data	Byte 5..Byte 15	Memory data

MRO Data Rate Control

10.11.8 73MEM_TWEAK, processor, BYTE, 0x60, 64, TWEK
10.11.9 73MEM_TWEAK, processor, BYTE, 0x62, 10, TWEK

These tweaks alter the data rate for 73MRO activities destined for the housekeeping stream. The value at offset 0x60 is set to 10 bits/second and probably should never be altered (i.e. this is the number of RTI periods between each micro packet). The value at offset 0x62 is the size of the micro-packet delivered to the housekeeping routine. The value of 10 results in a single 16 byte line of the variable area of the housekeeping buffer being used to deliver the MRO data.

MRO Data Packet Size Control

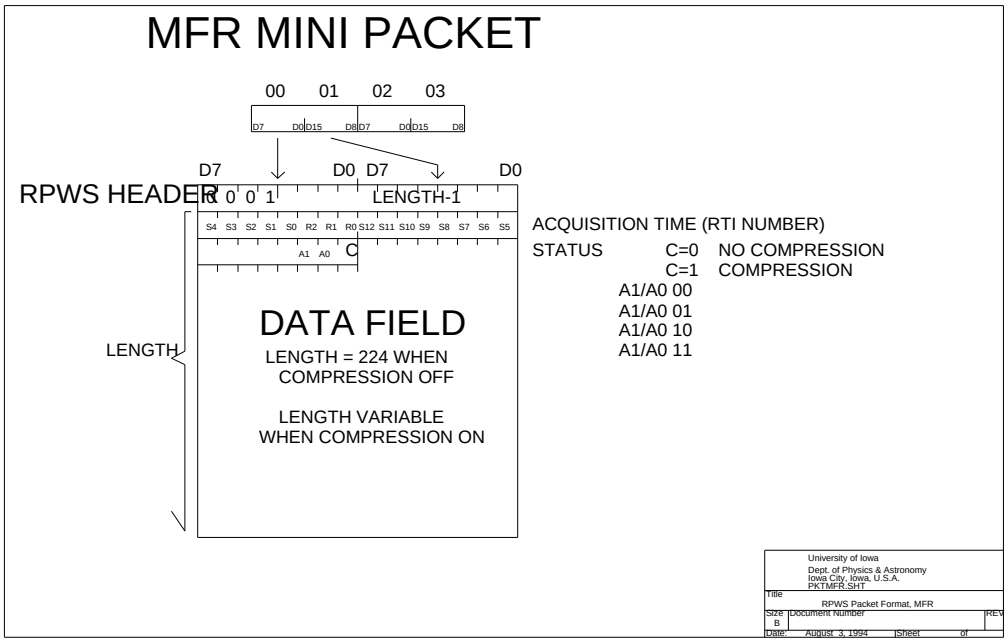
10.11.10 73MEM_TWEAK, processor, BYTE, 0x64, 9, TWEK
10.11.11 73MEM_TWEAK, processor, BYTE, 0x66, 128, TWEK (V2.4)

These tweaks alter the data rate for 73MRO activities destined for the science telemetry. The value at offset 0x64 is set to 1000 bits/second. This delay period can be raised without too much difficulty (i.e. this is the number of RTI periods between each mini packet). The value at offset 0x66 is the size of the micro-packet delivered to the BIU handler. The value of 128 results in a convenient size minipacket.

The value at offset 0x66 must not be set above 240 or the resulting data will not fit within a single free space buffer that is used to deliver the packet to the BIU handler. The nominal value is 128 with a change to 192 being used to accommodate the contents of the housekeeping buffer when routing housekeeping to the science telemetry stream.

10.12 MFR Control

Medium Frequency Receiver is a stepped spectrum analyzer operating from 20Hz to 12Khz.



Field	***Field Location***	***Description***
Minipacket ID	Byte 0 / D7..D4	1000
Minipacket Length	Byte 0 / D3..D0 Byte 1 / D7..D0	12 bit length
RTI	Byte 2 S4..S0, R2..R0 Byte 3 S12..S5	RTI time of data acquisition Seconds & RTI
Antenna	Byte 4 / D1..D0	Current Antenna Selection
Compression	Byte 4 / D2	Compression enabled
Fast Switch	Byte 4 / D3	Fast switching (16 second cycle)
Antenna-1	Byte 4 / D5..D4	
Antenna-2	Byte 4 / D7..D6	

10.12.0.1 D0..D1 Antenna

Antenna selection.

When Fast Switch is Asserted, indicates the antenna selection for the 1st. Half of the dataset with the 2nd. selection in (Antenna-1/Antenna-2???).

10.12.0.2 D2 Compression

Set to indicate that the data has been compressed using a ?? compression scheme.

10.12.0.3 D3 Fast Switch

This bit, prior to version V2.4, was a toggle indicator. This information was replaced , in version V2.4, with a *fast switch* status. When set, indicates that an antenna switch occurs in the middle of the dataset, effectively altering the data collection timing to switch antennas every 16 seconds.

10.12.0.4 D4..D5 Antenna 1

10.12.0.5 D6..D7 Antenna 2

10.12.1 73MFR_CNTL, command, compression, toggle, antenna 1, antenna 2

command	Description
trig	
mode	

compression	Description
compress	enables compression
nocompress	disables compression

toggle	Description
cont	antenna selection fixed on one antenna
single	
toggle	antenna selection toggles with each dataset

antenna 1	Description
ex	electric dipole antenna, X
ez	electric monopole antenna, Z
bx	magnetic search coil X
bz	magnetic search coil Z

antenna 2	Description
ex	electric dipole antenna, X
ez	electric monopole antenna, Z
bx	magnetic search coil X
bz	magnetic search coil Z
na	not applicable

10.12.2 73MEM_TWEAK, LRP, BYTE, 0X2A, mode, MFRI (V2.4)

mode	Description

0x00	Default antenna change every 32 seconds data packet covers 32 second period
0xFF	High temporal resolution antenna change every 16 seconds data packet covers 32 second period

Alter the MFR data collection cycle. Originally (and by default) the MFR collects data with a cycle time of 64 seconds. During the 64 second cycle, two 32 second data sets are collected from two antennas. This gives the basic instrument cycle time of 64 seconds.

Each 32 second cycle of the instrument gathers two complete cycles of the lowest frequency. If higher temporal resolution is required, this tweak may be applied to collect data from both antennas in a single minipacket.

Running the MFR at the higher temporal resolution can cause antenna switching interference in other instruments as well as making the MFR sensitive to antenna switching that is synchronized with the 64 second MFR cycle.

Commanding MFR into fixed mode (one antenna selected) with high temporal resolution has no unusual effect. Results are no different than running in low temporal resolution mode. Although it would seem peculiar to set this command state, it should be benign.

10.12.3 73MEM_TWEAK, LRP, BYTE, 0X2B, idle, MFRI (V2.4)

idle	Description
0x00	Default MFR collects data
0xFF	Idle MFR stops data collection

Prior to V2.4 software, the MFR would run any time that ME02 power was applied and the spacecraft collects data. The method to bring MFR to a quiescent state was to simply lower the priority of the MFR data collection process ("MFRI") to a priority below the idle process, effectively blocking its use of CPU. The downside of this being that data collection stopped as soon as the tweak was applied which would typically be in the middle of a data gathering activity.

Version 2.4 software adds a flag that may be set that causes the data gathering activity to cease following the completion of the current data gathering cycle (within 32 seconds). This has the benefit of always producing MFR minipackets that are correctly and accurately time tagged.

10.12.4 73MEM_TWEAK, LRP, BYTE, 0X15, priority, MFRI
Obsolete command

Alter the MFR control process priority. The MFR does not have any commands that may be used to suspend the data gathering and delivery activity prior to V2.4. The only way to accomplish this was to lower the process priority to a value below the *IDLE Process.*

To allow execution
priority - 0x64

To suspend execution
priority - 0x7F

10.13 Internal Power Control

Control of power switches internal to the instrument.

Note that indiscriminate use of power commands may result in exceeding the power allocation of the instrument. Flight rule 73FRC5 discusses the impact and implementation issues.

Also note that the Langmuir Probe handler, in loads prior to version 2.2, may be affected by duplicate power on commands that are issued to the HRP (i.e. **00PORT_TWEAK, HRP, 0x03, 0x01**).

10.13.1 73POWER_CNTL, assembly, state

Controls the 3 power switches located in the power supply. In addition to these power switches there are 2 additional switches located on the HRP that are typically controlled using internal ***00PORT_TWEAK*** commands.

This command is effectively ignored when the instrument is in *SLEEP* mode.

When powering up the instrument manually (i.e. not using the **73IEB_TRIGGER, MASK, 0, 0** command) the following command order should be used to prevent RPWS from exceeding power allocation. Also note that the instrument should be brought to a sleep-like state (i.e. **all power switches off**) before switching any power on to prevent the inrush current from causing RPWS to exceed power allocation.

```
73POWER_CNTL, HFR, ON
73POWER_CNTL, ANALOG, ON
00PORT_TWEAK, HRP, 0x03, 0x01
73POWER_CNTL, LPROBE, ON
00PORT_TWEAK, HRP, 0x40, 0x01
```

The commands may be wrapped as follows:

```
73WRAP, (0x5126)
73WRAP, (0x512A)
73WRAP, (0x32C8, 0x0103)
73WRAP, (0x5132)
```

assembly	Description
hfr	HFR digital/analog electronics
analog	ME02 (MFR/WBR?WFR) electronics
lprobe	Langmuir Probe analog
sleep	secondary SLEEP control
pause	stop acquiring data on (hrp) **DO NOT ISSUE THIS COMMAND**

state	Description
off	remove power (*hfr*, *analog*, *lprobe*)
on	apply power (*hfr*, *analog*, *lprobe*)
sleep	**sleep modifier**, enter sleep state (*sleep*: low power)
active	**sleep modifier**, exit sleep state (*sleep*: no power change)
hrp	**pause modifier**, affects HRP data activity (*pause*: idle hrp)

The **PAUSE** operates in a manner similar to the **SLEEP** modifier but was not intended to affect any of the power switches. As things always seem to turn out, this is not actuially the case due to the implementation of the L/P power switching electronics. L/P digital power is conditioned with the sleep status line on HRP (i.e. the very signal that we use to implement the pause command), so pulsing the sleep status line also pulses the digital electronics switch on the HRP. As a further consequence, the L/P analog power is conditioned by the L/P 5 volt digital power, so it also is removed briefly when pulsing the power line. This is NOT desirable as it risks placing the L/P in an nasty power state (although we haven't seen any evidence of a problem in flight or on the bench). We hope to remedy the situation in a future FSW release (i.e. V2.,7). In the mean time, we have removed all traces of the **73POWER_CNTL, PAUSE** command from the IEB's loaded on the S/C (the V2.6 base load and the C35 sequence do not make use of the pause, but the C35 load used for LSF testing continues to use the **73POWER_CNTL, PAUSE** command in several triggers.

This form of the command was intended to provide an out-of-band signaling mechanism that was used to stop data acquisition activities on the HRP. When traffic levels on the IPC bus are high, the LRP seems to have a relatively high rate of packet loss. This command is used to stop data acquisition activities on the HRP using the SLEEP line (it is brought active for 1 to 2 RTI periods). HRP reacts to activation of the SLEEP line by immediately stopping scheduling activities. If the SLEEP line is released in less than 4 RTI periods, the power switches will be left unaltered (i.e. power will remain on)

Version 2.3 & 2.4 have an error in the implementation that causes the LRP to operate at a reduced clock rate (i.e. ½ or normal speed) for the 2 RTI periods the sleep signal is sent to DCP/HRP. This seems to cause problems with the BIU (WPV bit is set) when running cyclic triggers that use the **73POWER_CNTL, PAUSE** command as part of a mode switch (see C20/Trigger-58). The error may be corrected by sending the following patches.

73MEM_TWEAK, LRP, BYTE, 0x65C, 0xF3, MFRI

73MEM_TWEAK, LRP, BYTE, 0x64C, 0x0C, MFRI

The patches may be verified by dumping memory using either of the following MRO commands

73MRO, LRP, HSK, 0x2F4C, 0

73MRO, LRP, HSK, 0x2F4C, 0

or

73MRO, LRP, TLM, 0x2F00, 0

This patch causes both sleep control bits to be set and cleared when the command issued.

Port F0 sleep control bits

D3	D2	LRP	HRP/DCP	Comment
0	0	1.5Mhz	1.5Mhz	Full Speed Operations i.e. Science operations
0	1	750Khz	287Khz	ROM Sleep
1	0	750Khz	1.5Mhz	ERROR Condition
1	1	1.5Mhz	287Khz	Science Sleep 2RTI Pause command

10.13.2 00PORT_TWEAK, HRP, 0x40, state

Power control for the WBR and WFR analog converters on the HRP.

state	Description
0x00	remove power from WBR & WFR A/D
0x01	apply power to WBR & WFR A/D

10.13.3 00PORT_TWEAK, HRP, 0x03, state

Power control for the L/P digital electronics. Note that L/P handler prior to version 2.2 does not contain a time-out handler to allow recovery if this port tweak is sent when the L/P is acquiring data (i.e. will permanently hang the L/P process).

state	Description
0x00	remove power from L/P A/D
0x01	apply power to L/P A/D

10.14 Misc Process Control, TIME LOCK

Misc tweaks that need to be documented, but don't belong to any receiver groups.

10.14.1 73MEM_TWEAK, LRP, BYTE, 0x60, active_flag, LOCK

Default value is 0xFF.

Setting this flag to zero suppresses time updates to DCP and HRP. A side-effect of this is that regular *mem tweak* traffic is not supplied to the other processors and, most likely, will result in a watch dog timer trip if the watch dog timer is active.

10.14.2 73MEM_TWEAK, LRP, WORD, 0x62, wdt_k, LOCK

Default value is 60 * 10 * 8, 60 seconds x 10 minutes x 8 RTI. 4800 RTI.

Time lapse from loss of looper traffic until watch dog timer is no longer reset.

Time updates occur every 256 seconds (2048 RTI), so the default value allows about 11 minutes to elapse (without *mem tweak* traffic circulating) prior to resetting the 8085.

10.14.373MEM_TWEAK,LRP,BYTE, 0x6E, behavior, LOCK

Watchdog timer control flag. This flag is used to enable the more restrictive watch dog timer control.

V2.4 code adds the capability to monitor the DCP/HRP by looking for arrival of MEM_TWEAK/MRO commands. These class of commands are routed through all processors and terminate on the LRP. The terminating packet causes the WDT update so loss of either DCP or HRP will cause loss of these updates and eventually a WDT trip.

Behavior	Description
0x00 default setting V2.4	V2.3 operation WDT hardware is *tickled* every RTI interrupt.
0xFF	V2.4 operation WDT hardware is *tickled* as long as looper packets are flowing.

The initial implementation of the Watch Dog Timer was not intended to be well behaved and functional. It was intended to simply keep the timer from tripping. This was accomplished by tickling the WDT in an interrupt service routine (it would keep running through all but the most catastrophic failures).

The V2.4 implementation requires that all 3 processors be operating and have sufficient resources available to deliver IPC traffic (i.e. F5 buffers and CPU cycles). This method is still susceptible to situations where handlers are blocked and cannot acquire and deliver science data, but it does provide better protection than the original implementation.

10.15 External Power Control

These are the commands used to cycle power to RPWS instrument and antenna deploy electronics.

These commands are PPS commands that are listed with the RPWS identifier.

10.15.1 73PS_ANT_MOTOR, *on/off* Restricted Command

Command to switch the SSPS that controls power to the antenna deploy mechanism

10.15.273PS_RPWS, *on/off*

Command to switch the SSPS that powers the RPWS instrument.

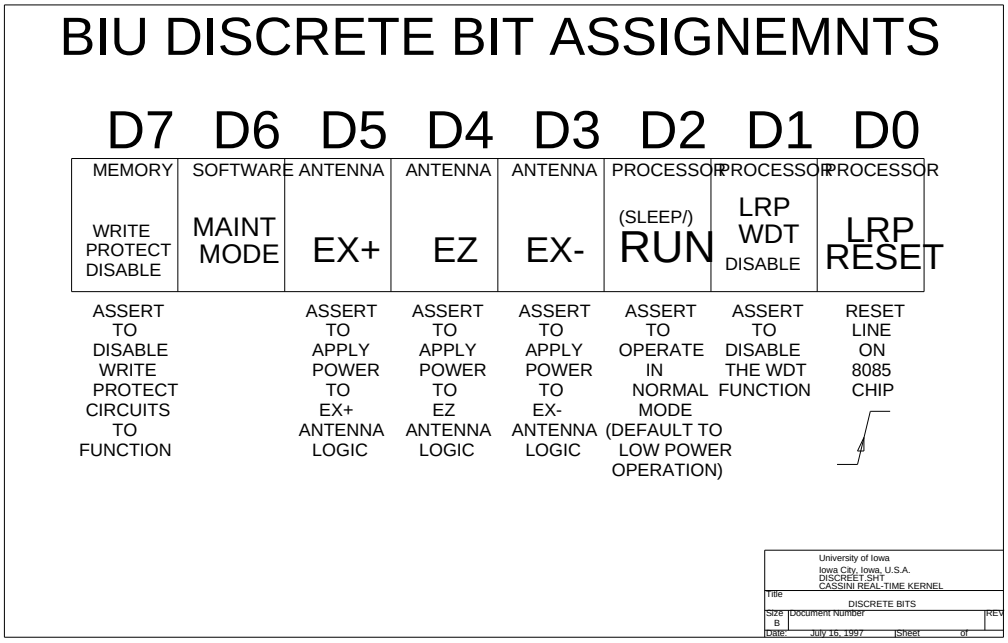
This power is applied only when the antennas are deployed at the beginning of the mission following launch.

10.16 BIU Control

BIU control commands are managed within the 1553 interface and may be issued at any time. The BIU control commands are grouped into two distinct groups, one group of commands that are defined by the RPWS instrument and a second group that are defined within the BIU.

Nominal command states are indicated in bold in the ***73RT*** commands.

RPWS Control Bits



10.16.1 73RT_EX_N_CNTL, *enable/**disable*** **Restricted Command**

BIU Discrete bit that controls the antenna control mechanism lockout for the X+ antenna element. The antenna element deploy circuit requires this bit to be set before the power routing relay is switched to allow power to reach the antenna deploy mechanism. In other words, this bit must be set to start a deploy but it does **not** remove power once it has been applied (although the software does detect the loss of the bit and reacts by removing power).

10.16.2 73RT_EX_P_CNTL, *enable/**disable*** **Restricted Command**

BIU Discrete bit that controls the antenna control mechanism lockout for the X- antenna element. The antenna element deploy circuit requires this bit to be set before the power routing relay is switched to allow power to reach the antenna deploy mechanism. In other words, this bit must be set to start a deploy but it does **not** remove power once it has been applied (although the software does detect the loss of the bit and reacts by removing power).

10.16.3 73RT_EZ_P_CNTL, *enable/**disable*** **Restricted Command**

BIU Discrete bit that controls the antenna control mechanism lockout for the Z antenna element. The antenna element deploy circuit requires this bit to be set before the power routing relay is switched to allow power to reach the antenna deploy mechanism. In other words, this bit must be set to start a deploy but it does **not** remove power once it has been applied (although the software does detect the loss of the bit and reacts by removing power).

10.16.4 73RT_MAINT, on/**off** AFFECTS ROM / MAINTENANCE ONLY

BIU Discrete bit that triggers maintenance mode when operating out of ROM or the Special Maintenance download

10.16.5 73RT_MEM_WRT_PRT, *ena/dis* DOES NOT AFFECT CURRENT SOFTWARE LOAD

BIU Discrete bit that disables the write protect on the lower area of 8085 memory on the LRP. Current software architecture does not make use of this capability(i.e. the 8085 never configures the memory protection hardware).

10.16.6 73RT_RESET, *reset/**release***

BIU Discrete bit used to reset the 8085 on the LRP. Typically the reset sequence desserts, asserts, then desserts the discrete bit. The initial deassertion helps eliminate problems with a stuck reset line (RPWS is edge sensitive so the line being stuck active will not prevent RPWS from operating).

Note that a flight rule dictates that 5 seconds elapse following any ***73RT_RESET*** command.

10.16.7 73RT_SLEEP, *active*/*sleep*

BIU Discrete bit to allow RPWS to operate in high power modes. When this bit is clear, RPWS operates in a low power mode with all internal power switches off.

The instrument will not honor the maintenance bit when the sleep discrete is cleared to 0 (i.e. in sleep state).

The instrument will not process a normal speed ALF load when the sleep discrete is cleared.

Note that the housekeeping telemetry may indicate that the instrument is still in ***SLEEP*** following the reception of a **RT_SLEEP, ACTIVE** command. This is simply an artifact of the way the BIU handler processes the sleep discrete. The BIU handler will unconditionally enter the sleep state when the discrete bit is asserted but it does not exit the sleep state. The power control code is responsible for releasing the sleep state so a power command is required before housekeeping telemetry indicates the instrument has exited sleep. The sleep indication in the housekeeping telemetry is simply an indication that the instrument has received no power commands and is currently in a low power state.

10.16.8 73RT_WDT_CNTL, *enable*/*disable*

BIU Discrete bit to disable the 8085 watchdog timer. Both the ROM and any downloaded software take care of operating the watchdog timer so this should never be necessary.

BIU Internal Control Bits

The BIU Internal commands control internal operation the BIU. Specifically the BIU implements a write-protected area of memory that contains the initial transaction table. This initial transaction table is write protected with any attempts to write from the host generating an error status.

The BIU also implements a watch dog timer that disables the local transmitter if not updated periodically. CDS is responsible for updating the watch dog timer. If CDS is not able to update the watch dog timer the BIU will disable it's transmitter until the **RT_WDTERR** command is received. In the event that a BIU begins to babble, these updates cannot occur and all of the BIU's will shut down, allowing CDS to systematically isolate the offender.

10.16.9 73RT_WDTERR_RPWS, *set/clear*

BIU Internal Control of the BIU watch dog timer. This command can mute the transmitters (set) or unmute the transmitters(clear). It can *not* be used to suppress the watch dog time function (i.e. if the time expires, the transmitter will be muted again).

This command is also used to clear a WDT failure (i.e. the WDT issued by CDS over the 1553 bus).

10.16.10 00RT_WDTFNC_RPWS, *enable*/*disable*

BIU Internal Control of the BIU watch dog timer. This command can be used, during internal testing, to disable the BIU internal watch dog timer. When disabled, the S/C simulator does not need to send regular watch dog timer triggers.

When the BIU fails to receive watch dog timer updates (i.e. when this flag is in the enable state), it will disable it's 1553 bus transmitter.

This command does not seem to appear in the JPL command document?

10.16.11 73RT_WPERR_RPWS, *set*/*clear*

BIU Internal Control of the BIU write protect violation flag.

This flag indicates either CDS or the instrument have attempted to write to the lower portion of memory.

CDS may also assert this flag for testing purposes.

10.16.12 73RT_WPFNC_RPWS, *enable*/*disable*

BIU Internal Control of the BIU write protect enable. Normally the internal write protect is enabled such that attempts to access to lower portion of BIU memory, containing the default transaction tables, is suppressed.

CDS may disable the write protect to perform diagnostics.

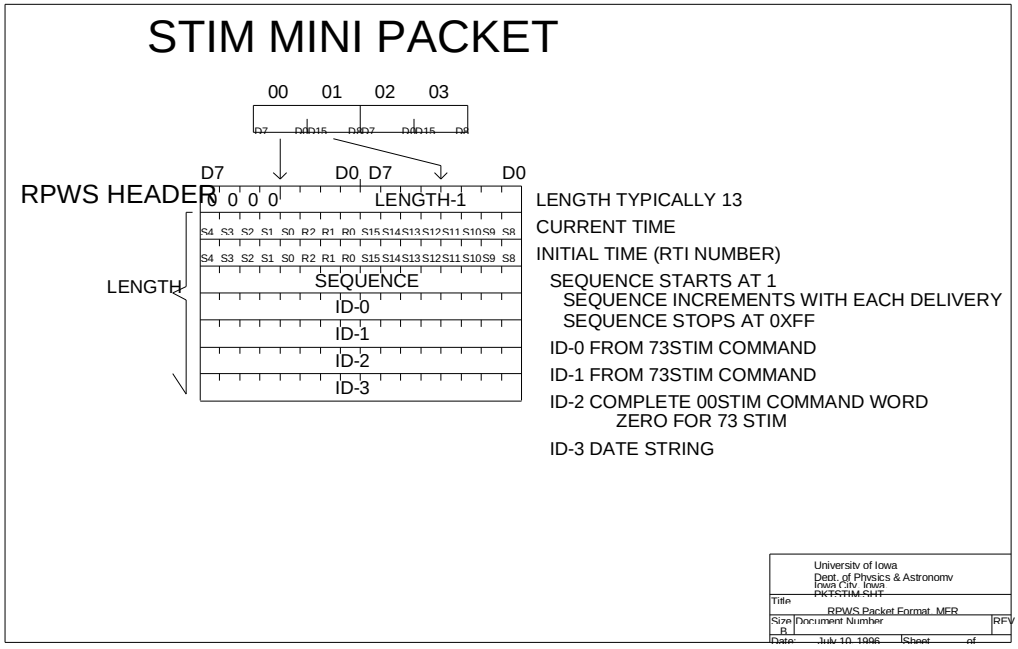
The instrument does not have access to this area and will cause a write protect violation to be flagged when attempting to write to this area of memory.

10.17 Stimulus Echo

Commands to echo a small packet into the science telemetry stream. The resulting packet, which is 16 bytes long, contains an echo of the ID fields, time information and the day of year that the stim process was assembled. The day of year is intended to be used for version verification of the loaded software.

These commands are not dependent on any power state and may be issued when the instrument is in *SLEEP* mode.

The STIM process does, however, meter the outgoing stim packets. This may result in packets being discarded if too many commands are delivered to the STIM process. This limit is approximately 1 command every 10 seconds.



Field	*Field Location*	*Description*
Minipacket ID	Byte 0 / D7..D4	1000
Minipacket Length	Byte 0 / D3..D0 Byte 1 / D7..D0	12 bit length

Field	*Field Location*	*Description*
Current RTI	Byte 2 S4..S0, R2..R0 Byte 3 S12..S5	RTI time of data acquisition Seconds & RTI
Initial RTI	Byte 4 S4..S0, R2..R0 Byte 5 S12..S5	RTI time of data acquisition Seconds & RTI
Sequence	Byte 6 / Byte 7	Sequence Number
ID-0	Byte 8 / Byte 9	ID-0 field from 73STIM command
ID-1	Byte 10 / Byte11	ID-1 field from 73STIM command
ID-2	Byte 12 / Byte 13	73STIM command pattern
ID-3	Byte 14 / Byte 15	Date String (version control)

10.17.0.1 Current RTI

RTI from when the STIM packet was generated.

10.17.0.2 Initial RTI

RTI from when the 73STIM command was first processed.

10.17.0.3 Sequence

Sequence number. This field is initially set from the 73STIM command and decrements with each STIM packet that is generated until the sequence number reaches zero.

In most cases, where we are using STIM as an indication of which IEB trigger is executing, the sequence count is set to zero, causing a single STIM packet to be generated from each 73STIM command.

10.17.0.4 ID-0

Bit field from the 73STIM command.

10.17.0.5 ID-1

Bit field from the 73STIM command.

10.17.0.6 ID-2

16 pattern of the 73STIM command (taken directly from the 73STIM command).

10.17.0.7 ID-3

Version control information. Se the version control section of the UG/SOM (this document) to see the expected value of this field. This field changes with each software update.

Software version	ID-3 Location	Offset
2.6	0x3E99	0x99

10.17.1 73STIM, id 0, id 1

ID-0 is a 7 bit unsigned integer. ID-2 is a 16 bit unsigned integer. If ID-0 is non-zero, 256 STIM packets will be delivered into the LRS stream at regular intervals.

id 0	Description
0	generate a single stim record
1..127	generate multiple stim records

id 1	Description
0..65535	

10.17.2 00STIM, id0

Internal form of ***73STIM*** that delivers a single record to the LRS stream.

10.18 WBR Control

Wide Band Receiver commands. Note that the compression switch is present in two separate commands and this can present an inconsistency in building the commands. Please refer to the *suggested order list* when building **WBR** commands to avoid placing the instrument into an invalid state.

Commands may be sent to the WFR when the instrument is in *SLEEP* mode although they will not be immediately processed. It should not be possible to bring the instrument out of SLEEP mode inadvertently (i.e. modifications to offset 0x56).

Although not evident, it is possible to combine the DUST and WBR logical instruments to implement a ***toggle mode***. This is accomplished by commanding the DUST instrument with a setup for the 2nd. antenna (i.e. identical sample size and timing parameters). Although it is possible to set the schedule up to request both sensors at the same time, the hardware lacks this capability and the software is structured so the acquisitions will occur sequentially. In order to obtain consistently time data sets, it is necessary to use the MOD timing to delay the data capture on one of the sensors (would work best to schedule them evenly when considering the impact on LRP). The command lists for DUST contains the MEM_TWEAK required to alter the data routing to deliver to the ground.

Version V2.6 implements a burst scheduling mode for the WBR. This allows a burst of WBR acquisitions to be collected with, effectively, an automatic idling of the WBR. An additional capability allows DUST collection to be suspended during the WBR burst to avoid AGC interactions.

Note that the LFDR sync mode is incompatible with the DCC hardware. Do not specify hardware compression when using LFDR synch mode!

Note that the **73WBR_MODE_CNTL** command alters the contents of offset 0x52 in the W08I process, forcing antenna selection between datasets to match that selected in the **73WBR_MODE_CNTL** command. Changes to this selection must be commanded after the **73WBR_MODE_CNTL** command as well as after the **73DUST_MODE_CNTL** command.

Suggested order of commands:

73WBR_AUTO_CNTL

73WBR_DATA_CNTL

73WBR_GAIN_CNTL

73WBR_MODE_CNTL, HOLD

73WBR_CMPRS_CNTL

73MEM_TWEAK DIR_ offset 0x60

73MEM_TWEAK W08I offset 0x52

73MEM_TWEAK WBRC offset 0x50

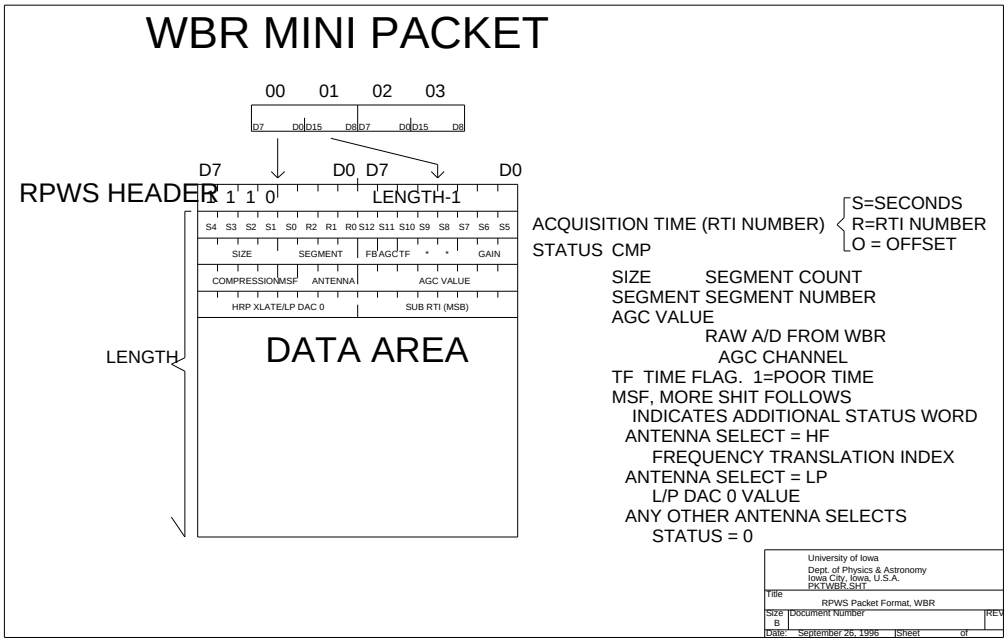
73MEM_TWEAK WBRC offset 0x52

73MEM_TWEAK WBRC offset 0x54

73MEM_TWEAK WBRC offset 0x58

73MEM_TWEAK WBRC offset 0x5A

73MEM_TWEAK WBRC offset 0x56


WBR MINI PACKET
00 01 02 03
D7 D0 D15 D8 D7 D0 D15 D8
D7 D0 D7 D0
RPWS HEADER
R 1 1 0
LENGTH-1
S4 S3 S2 S1 S0 R2 R1 R0 S12 S11 S10 S9 S8 S7 S6 S5
SIZE
SEGMENT
FB AGC TF * * GAIN
COMPRESSION MSF ANTENNA
AGC VALUE
HRP XLATE/LP DAC 0
SUB RTI (MSB)
LENGTH
DATA AREA
ACQUISITION TIME (RTI NUMBER)
S=SECONDS
R=RTI NUMBER
O = OFFSET
STATUS CMP
SIZE SEGMENT COUNT
SEGMENT SEGMENT NUMBER
AGC VALUE
RAW A/D FROM WBR
AGC CHANNEL
TF TIME FLAG. 1=POOR TIME
MSF, MORE SHIT FOLLOWS
INDICATES ADDITIONAL STATUS WORD
ANTENNA SELECT = HF
FREQUENCY TRANSLATION INDEX
ANTENNA SELECT = LP
L/P DAC 0 VALUE
ANY OTHER ANTENNA SELECTS
STATUS = 0
University of Iowa
Dept. of Physics & Astronomy
Iowa City, Iowa, U.S.A.
PKTWBR.SHT
Title
RPWS Packet Format, WBR
Size B
Document Number
REV
Date: September 26, 1996
Sheet of

10.18.1 WBR Timing Control Process: Setup Storage Area

OFFSET	Label	Default	Description
0x2520	CMDp_Divisor	64	MOD timing divisor
0x2522	CMDp_Remainder	0	MOD timing remainder
0x2524	CMDp_Packet_Delay	0	LRS inter packet delay (Not used for HRS)
0x2526	CMDp_Run	1	Run Flag
0x2527	CMDp_AGC		
0x2528	CMDp_Offset		8254 timer, offset from RTI to start of data capture
0x252A	CMDp_Mode		
0x252C	CMDp_Band		Band Select
0x252E	CMDp_Length		Dataset byte count (8237 word count)
0x2530	CMDp_Destination		Data Routing Destination
0x2532			
0x2534	CMDp_AGC_H_L		AGC set points
0x2536	CMDp_Antenna		Antenna Selection
0x2537	CMDp_Gain		Gain Setting (D5..D3)
0x2538	CMDp_AGC_Flag		AGC Enable
0x2539			
0x253A			
0x253C	CMDp_DCC		ACTEL register 43 Pattern
0x253E			

10.18.2 WBR Minipacket Format

Field	*Field Location*	*Description*
Minipacket ID	Byte 0 / D7..D4	1110
Minipacket Length	Byte 0 / D3..D0 Byte 1 / D7..D0	12 bit length (n-3)
RTI	Byte 2 S4..S0, R2..R0 Byte 3 S12..S5	RTI time of data acquisition Seconds & RTI
Size	Byte 4 / D7..D4	Number of segments in data set (n - 1)
Segment	Byte 4 / D3..D0	Segment number (n - 1)
FB	Byte 5 / D7	Frequency Band 1 = 4.5uSec (80Khz) 0 = 36uSec (10Khz)
AGC	Byte 5 / D6	AGC 1 indicates AGC enabled
TF	Byte 5 / D5	Time Quality Flag 1 indicates LFDR synch in use
TMO	Byte 5 / D4	Timeout 1 indicates dataset invalid
spare	Byte 5 / D3	
Gain	Byte 5 / D2..D0	Gain level 0dB to 70dB in 10 dB steps
Compression	Byte 6 / D7..D4	Compression status
MSF	Byte 6 / D3	More Status Follows 1 indicates bytes 6 and 7 present
Antenna	Byte 6 / D2..D0	Antenna Selection
AGC value	Byte 7 / D7..D0	AGC value Value read from AGC port following acquisition
HFR Xlate	Byte 8 / D7..D0	HFR translation frequency when MSF set and antenna select is 3
LP DAC 0	Byte 8 / D7..D0	Langmuir Probe DAC voltage when MSF set and antenna select in 4
Sub RTI	Byte 9 / D7..D0	Time stamp with sub-RTI resolution valid when non-zero (see discussion)

10.18.3 Status

Status field definitions for WBR data.

10.18.3.1 Size/Segment, Byte 4

Size and segment information is used to allow for a dataset that exceeds the size available in the length filed of the minipacket header. Although 2 segments are all that is required to accommodate hardware restrictions on the WBR, the field is provisioned with 4 bits for each field such that it matches the WFR packet format.

10.18.3.2 FB bit, Byte 5, bit 7

Frequency Band bit. Set to 1 to indicate high band data acquisition, 4.5uSec sample interval. Set to 0 to indicate low band data acquisition, 36 uSec sample interval.

10.18.3.3 AGC bit, Byte 5, bit 6

AGC active bit. Set to indicate that AGC control is enabled. Clear to indicate the AGC control is not in use.

This bit is added mostly as a verification that the intended command state has been achieved. WFR has a similar status bit (although it is replaced with other status information when the data is compressed).

Prior to V2.6 this bit is not used and will be set to zero.

10.18.3.4 TF bit, Byte 5, bit 5

Time quality Flag bit. Set to indicate the WBR is operating in synch with the LFDR sample clock. This indicates that the time status byte may be off by up to 10 milliseconds (we latch the high resolution time bits and the wait for the next LFDR sample to occur)..

10.18.3.5 TMO bit, Byte 5, bit 4

Time Out bit. Set to indicate a timeout has occurred while waiting for WFR data acquisition. **This bit, when set, probably indicates a corrupt data set that should be discarded**. This will most likely occur when operating WBR in synch with the LFDR when a WFR data acquisition occurs. This problem is obvious when viewed on a timing plot, where WBR acquisitions should occur frequently (every other RTI or faster) and a WFR data set is acquired. WBR data will occur once per second, which is the timeout interval for the WBR. When the timeout timer expires, the W08I process is released from a wait for data acquisition to complete and processes the data normally, even though a completion interrupt did not occur.

By setting this bit, the instrument informs the ground system that a potential problem with a specific data set has occurred.

10.18.3.6 Gain level, Byte 5, Bits 2-0

Hardware gain level used for acquisition of this dataset. 0 to 7 represent 0 to 70dB in 10dB steps.

10.18.3.7 Compression status, Byte 6, bits 7-4

We have both DCP and DCC compression available for use with the WBR data although it is very unusual to use DCP to perform this task.

Value	***Compression***
0000	RAW (no compression)

10.18.3.8 MSF Bit, Byte 6, bit 3, "More Status Follows"

The WBR status may be augmented with two additional status bytes when required. The presence of these status bytes is flagged using the ***MSF bit*** in a fixed location in the status area. When the ***MSF bit*** is set the minipacket length increases by two to accommodate the additional status bytes.

The additional status bytes will be included in the WBR minipacket under the following conditions.

- WBR antenna selection is set to the HFR input (and the HFR is operating in a frequency translation mode). The 1st. extra status byte indicates the HFR frequency selection. Information about the current HFR frequency selection is updated in the system data page by the HFR process on the LRP. WBR process can pick up the translation setting and place it in the minipacket.

- WBR antenna selection is set to the L/P input. The 1st. extra status byte indicates the value loaded into DAC-0 (i.e. the sphere bias). L/P process keeps the system data page updated with the current value that is loaded in to DAC-0 so the WBR may collect the information and place it in the minipacket.

- WBR is not operating synchronized with the RTI pulse (indicated when WBRC offset 0x58 is zero). Normally WBR data collection is synchronized with the RTI pulse using one channel of an 8254. This allows the data collection to begin a fixed period of time after the RTI pulse. In order to achieve very high bit rates, the WBR data collection is allowed to occur as soon as data buffers become available. The offset from the RTI pulse is measured using another channel of an 8254 and this value is shifted appropriately and placed into the 2nd extra status byte.

- WBR is operating synchronized with the LFDR sampling (indicated when WBRC offset 0x58 is 0xFFFF). This would be used to operate WBR in high-band mode at the same time that LFDR is running. In this case the Sub-RTI field will have reduced temporal accuracy as the WBR data acquisition may occur up to 10 mSec after the Sub-RTI counter has been read.

NOTE that scheduling data acquisition at a fixed offset from the RTI pulse is not noted in the status field of the minipacket anywhere.

Dust detection routines telemeter data using the WBR minipacket. The compression method is used to indicate that the data is a special 4bit format from the dust detection algorithm. This data may be collected late in to RTI period to keep S/C noise from triggering the detection algorithm (the L/P sphere seems to be sensitive to BIU transactions). No indication is provided in the minipacket status to indicate that the data collection was scheduled late in the RTI period.

Finally, keep in mind that the 1st. Extra status byte is not particularly interesting when the antenna select in not set for HFR or L/P antennas. The MSF bit indicates both bytes are present even if only on of them is required.

10.18.3.9 Antenna Select, Byte 6, bits 2-0

10.18.3.10 AGC Value, Byte 7

This field, an unsigned 8 bit value, contains the value read from the AGC integrator. This corresponds with the value supplied to the **73WBR_GAIN_???** command.

10.18.3.11 HFR Xlate frequency, Byte 8

This field contains an indication of the HFR downconvert frequency when the Antenna Select is set to the HFR. The HFR command section contains the frequency select table. See **73xxx** command in the HFR section.

10.18.3.12 LP DAC 0, Byte 8

This field contains an indication of the voltage on the Langmuir Probe when the Antenna select field is set to LP. We don't send down the MUX/Relay settings (We keep track of them through benign intuition).

10.18.3.13 Sub-RTI, Byte 9

This field ranges in value from 244 down to 9. Each count represents 512 micro seconds. The count starts at 244 at the beginning of the RTI period and counts down until the ***Dead Time*** begins, 120 milliseconds into the RTI period (i.e. 5 mSec prior to the next RTI pulse).

A value of zero indicates that the WBR is operating with a known offset from the beginning of the RTI period (i.e. the Sub-RTI is not valid when it contains a zero). This can occur when the antenna mux is set to select either the HFR or the L/P (i.e. when these antenna selections are in use, an additional 2 status bytes are delivered to the ground, you will see the Sub-RTI field as part of the additional status, even if it is not needed).

The ***RTI*** and ***Dead Time*** signals are combined and delivered as a single signal from LRP, the signal level transitions from low to high when the ***RTI*** begins and transitions from high to low when *Dead Time* begins (signal is high for 120 mSec and low for 5 mSec). This signal is connected to the 8254 gate and stops the counter when low and reloads the counter at the rising edge (we didn't notice the behavior in time to have it any other way).

Dead Time Start is synthesized in the Actel gate array on LRP. The starting point of this signal, by convention, is 120mSec into the RTI period. The starting point is programmable (on the LRP), however, and affects the point at which the 8254 counter is stopped. The Dead Time must be long enough to allow the processors sufficient time to accomplish dead time activities (in other words, it can't get shorter to accommodate HRP goals without interfering with BIU communications functions).

10.18.4 Data Area

The data area consists of 8 bit unsigned samples. The dataset can be any length up to 5120 bytes with a typical length of 2K or 4K bytes. Operating in synch with the LFDR imposes a hardware restriction of 2048 on the dataset size (if the hardware is programmed for a longer dataset size, the LFDR data may be corrupted).

10.18.5 73WBR_AUTO_CNTL, low set, high set, average interval, time constant

Sets trip points for the automatic gain control software.

The auto gain implemented within the data acquisition process is rather primitive. It reads the AGC level following each data set and adjusts the gain level based on the reading. This scheme depends on the AGC circuit in the receiver to supply a meaningful integrated level for the incoming signal. Several problems can make obtaining meaningful AGC readings difficult.

The first problem being that the 8 bit analog system operates in two distinct frequency bands which makes optimizing the AGC circuit difficult.

The next difficulty that may arise is antenna/gain switching. The AGC circuit requires some relatively long period of time to adjust it's output level when the incoming signal changes radically. This can occur when the antenna multiplexer is changed or when the gain setting is changed just prior to acquiring a data set. When WBR is operating in conjunction with the DUST, the 2 logical instruments are handled independently. Antenna selection and gain selection are made just before performing the data acquisition (possibly upsetting the ACG circuit). Problems associated with gain changes are reduced somewhat by immediately loading new gain selections into the hardware when the AGC decision is made following the acquisition of a data set.

And finally, there can be an oscillation induced into the system due to the double buffering used to improve the rate of data production. The current gain level is maintained in a data structure common to the entire thread of activity that is involved in producing and delivering data. What can occur when running at high speeds is a gain update can be made in the common data structure after the other data buffer has already been loaded with the current gain setting and queued for processing. When the 2nd. Buffer arrives, the original gain is used and the AGC processing performs a 2nd. gain change. This causes the receiver to oscillate by 2 gain steps. This problem is avoided by suppressing the gain update on every other acquisition. This solves the potential oscillation when running at high data rates at the expense of a lower AGC update rate (particularly evident when running DUST at a low rate).

Low Set		High Set		Description
EM	**Flight**	**EM**	**Flight**	
11	31	58	77	WBR, 10dB headroom
08	28	31	50	WBR, 15dB Headroom
07	27	42	60	WBR IEB (C18)

Version V2.5 software alters the AGC timing scheme in an attempt to improve the speed with which it reacts to changes in the environment. Previously the AGC was used every ***n*** data sets to change the gain, if required. This basically ignored the gain information is all other data sets. V2.5 handles AGC by suppressing updates for ***n*** cycles. The update rate numbers work, essentially, the same as before, but the AGC will react to changes almost immediately (AGC is sampled **after** the data set is acquired). As before, updates will not occur more frequently than the defined schedule, but the updates will not necessarily appear to be on a MOD schedule.

V2.5 software has a bug the can be patched. V2.6 has the AGC patch. The patch to V2.5 is as follows (this code runs on HRP):

```
3C30: 21 2211   Lxi H, AGC_Throttle
3C33: AF        XrA A
3C34: C9        Ret

31C4: CD 3C30   Call Patch
```

This code fragment inserts an instruction to clear the accumulator prior to a compare sequence. We simply overwrite a 3 byte instruction (the Lxi H) with a call and insert the clear accumulator (Xra A). The flaw in the V2.5 shows up as a long (i.e. up to 255 data sets) period of time where the gain will not change.

10.18.6 00WBR_AUTO_ADJ, direction

Used by internal gain control routine to adjust gain.

10.18.7 00WBR_RATE_CNTL, direction

Used by data throttling routine to adjust the sample period for the WBR. The change is accomplished by shifting the scheduling period word left or right by 1 bit.

10.18.8 73WBR_BURST, enable, target, data-set count V2.6
10.18.9 73WBR_BURST_TRIGGER, enable, target V2.6
10.18.10 73WBR_BURST_SIZE, data-set count V2.6 (no data acquisition)

The logical instrument must have the scheduling mode set to IDLE in order for this command to execute. If the scheduling mode is set to any other mode, the command will be ignored (see **73MEM_TWEAK, HRP, BYTE, 0x56, n, WBRC**).

These commands control the WBR burst data acquisition. The specified number of data acquisitions will be performed. An enable bit is used to suppress dust acquisition if needed (**i.e. 73WBR_BURST_SIZE only sets the number of data sets to acquire, it doesn't acquire any data**).

This is a one or two word command with two words being required to set the size of a data burst. A single word form may be used to trigger a burst using a previously stored burst count. An enable bit appears in the command word that is clear to suppress performing the data captures (i.e. This allows a method to initialize the burst count without actually triggering data collection, particularly useful in trigger 10).

Finally, keep in mind that the Mux Control Byte has a very noticeable impact on this AGC behavior. With the V2.6 release (i.e. This is where this command first appears) the IEB setups are not well configured to allow AGC to operate correctly (most of our triggers have Ez selected between datasets which artificially reduces the gain level).

Enable	Description
nodust	DUST data acquisitions are suppressed while the burst is active.
dust	DUST data acquisitions are allowed to continue while the burst is active.

Target	Description
MP	Counting data set acquisitions. (Counting Mini Packets)
CDS	Counting CDS records.

Data-set count	Description
nn	Number of data sets to acquire

The CDS packet count driven burst mode is very decoupled. Once the sepcified CDS packet count has been delivered, the data acquisition process will stop acquiring data. Any data that has not been delivered, up to this point, will be formatted and delivered, resulting in what appears to be an overflow. As WBR is double buffered, an additional pair of acquisition buffers may be outstanding, so an additional 80,000 bits or so may be delivered to the spacecraft after the burst has ended (maximal packet size is 5120 bytes, double buffered).

10.18.1173WBR_CMPRS_CNTL, enable, route, word count

This command turns the ISFLIP chip on/off, selects the word count used to count the data (i.e. uncompressed samples or compressed bytes), and sets the number of samples.

The ***route*** field is used to enable an external word count register that allows variable size minipackets to be produced (i.e. with a fixed number of samples). The ***8237*** always counts bytes transferred to memory.

The WBR is double buffered with a maximum sample count of 5120.

Enable	Description
off	hardware compression not used
on	hardware compression enabled

Route	Description
wc_out	EOP generated by WCR in 8237 only
wc_in	EOP generated by WCR in Actel or 8237

Hardware Limitation when using the DCC chip with **WBR** in **High Band**.

When WBR is operating at the 4.5µSec sample rate (i.e. high band) with hardware compression (i.e. using the DCC chip) while keeping the number of samples constant (i.e. wc_in, using the word count register in the Actel), it is susceptible to a deadlock condition that may be triggered any time the processor clock is slowed down (i.e. by asserting the SLEEP signal from the LRP). In order to avoid the potential condition with V2.3 or older software it is essential that **wc_in** not be used in conjunction with **hband**.

V2.5 software crashed once due to this hardware limitation.

V2.6 and later software removes (from internal IEB and subsequent IEB loads) a command that may trigger this.

V2.7 software disables the offending command. There is no longer code on LDRP that can trigger this.

Note that the instrument is, however, susceptible to having the assertion of the sleep discreet bit cause the HRP to hang, although it will reboot itself internally. The instrument would be rebooted before we take any recovery action.

word count	Description
nn	number of samples in data set 5120 samples maximum

Hardware Limitation: Do not use compression and LFDR sync at the same time.

The hardware will not start correctly when WBR high band is programmed when using the LFDR sample to trigger WBR data acquisition. This results in a complete hang of the HRP (8237 take over the processor bus and never releases, RESET is the only way out).

Version V2.7 software avoids the potential deadlock by forcing compression off whenever LFDR SYNC is specified (i.e. even if you command compression, it will not be used).

Hardware Limitation: Do not use word count in excess of 2048 when WBR is in high band and LFDR sync is specified.

2048 4.5 uSec samples will fit between 10mSec WFR/LFDR samples. Using a larger data set size will cause both WFR/LFDR and WBR data to be corrupted as the logic is not built to accommodate concurrent activities.

Version V2.7 software avoids the potential data corruption by limiting the dataset size to 2048 samples when LFDR Sync is enabled and WBR is set to high band mode.

The V2.7 software does allow for the simplification of some sequences involving the WBR hardware limitations. In particular, consider a set of triggers that would normally operate WBR with a data set size of 4096 samples using high band. With the WBR configured, we can change WBR to use the LFDR Sync mode without the need to alter compression and data set size (i.e. the driver will limit dataset size and compression setting as needed). Also note that this change in behavior affects WBR only when needed, it does not alter the commanded state of WBR, so switching WBR out of LFDR Sync mode will allow the compression setting and data set size to return to its previous setting.

10.18.12 73WBR_DATA_CNTL, destination

Selects the data route used to deliver the data to CDS. Allows selecting the HRS, LRS and possibly the DCP as a destination for the data (although selecting DCP is pointless as there is no software in place to accept WBR data).

destination	Description
lrs	WBR data delivered to Low Rate Science stream on Low Rate Processor
hrs	WBR Data delivered to High Rate Science stream on High Rate Processor

10.18.13 73WBR_GAIN_CNTL, enable, gain select

Select the current gain level and enables/disables the automatic gain control. See *73WBR_AUTO_CNTL* for additional notes.

enable	Description
man	manual gain control
auto	automatic gain control

gain select	Description
0, 10, 20, 30, 40, 50,60, 70	gain level

10.18.14 73WBR_MODE_CNTL, HOLD, band, antenna select

Antenna and filter selection.

band	Description
lband	36 uSec Sample period
hband	4.5 uSec sample period

See the hardware limitation note for the 73WBR_CMPRS_CNTL command when making use of hband with this command.

antenna select	Description	MSF
ex	electric dipole antenna, X	
ez	electric monopole antenna, Z	
bx	search coil, X axis	
hf	HFR downconvertor	set to 1
lp	langmuir probe	set to 1

MSF column indicates that when selecting either of these two antennas, the MSF bit will be set.

The antenna select field is also stored in the W08I process when this command is issued. The selection may be overridden with either a memory tweak or a **73DUST_MODE_CNTL** command. This setting(i.e. Within the W08I process) impacts the way AGC behaves.

10.18.15 73MEM_TWEAK, HRP, BYTE, 0x60, *nn*, DIR_

BIU Direct processing mode. Details of this tweak are discussed in the section labeled *High Rate Science*

nn	Description
0	Programmed move (data rate limit of approx 100Kb/sec)
1	Simple DMA (8237 acquired as needed)
2	DMA/Programmed Mx dependent
3	Restart-5 (8237 acquired between LFDR samples)
4	Burst DMA (8237 acquired for multiple move operations)

10.18.16 73MEM_TWEAK, HRP, WORD, 0x50, *nnnn*, WBRC

Scheduling period expressed in RTI ticks.

A value of zero may be used to indicate immediate scheduling. This becomes necessary when attempting to operate at elevated bit rates (i.e. in excess of 131,000 typically).

10.18.17 73MEM_TWEAK, HRP, WORD, 0x52, *nnnn*, WBRC

Scheduling offset expressed in RTI periods.

A data acquisition is scheduled, typically for the next RTI period, when the *remainder of the division of the system time by location 0x52* matches the offset in location 0x52.

10.18.18 73MEM_TWEAK, HRP, WORD, 0x54, *nnnn*, WBRC

Delay (expressed in RTI's) between minipackets. Used to throttle the data delivered to either the DCP or LRP.

10.18.19 73MEM_TWEAK, HRP, BYTE, 0x56, *n*, WBRC

Scheduling mode.

0 Stop

Data acquisition is stopped. ***73WBR_MODE_CNTL, TRIGGER*** is not processed. Internal triggers are not processed. This state is entered whenever sleep is asserted.

1 Idle

Data acquisition is stopped but the process will accept a trigger to perform a single acquisition. After the acquisition the process will return to idle to await further triggers.

2 Run

Data acquisition is continuous and based on the schedule specified at offset 0x50 and 0x52.

3 Trigger

Single data acquisition.

8 Burst

Collect a *BURST* of WBR data, scheduled as specified in locations 0x50 and 0x52, collecting the number of samples specified in location 0x5A. Following the burst of data, the scheduling mode reverts to 1 (Idle).

This scheduling mode must be selected with a MEM_TWEAK each time it is required (i.e. it does not reschedule automatically).

Location 0x5A must be loaded with a reasonable count prior to releasing the control process to acquire the burst of data.

9 Burst, no DUST

Same as scheduling mode 8, but DUST will suspend acquisition until the WBR burst completes.

This scheduling mode is intended to allow both DUST and WBR to operate with AGC active. Dust collection is suspended to prevent discontinuities in the AGC level selected by the data acquisition process.

Location 0x5A must be loaded with a reasonable count prior to releasing the control process to acquire the burst of data.

10&11 CDS counting burst

Similar to 8 and 9, but counts CDS packets. Note that the control and counting process are very decoupled, so there will be extra data that trickles out of the system.

12&13 Internal Burst state

Do NOT set this value, it will cause a very long WBR burst (of 65535 records) if the internal counter has completed a burst (and, therefore, contains a value of zero). The WBR control process uses this state internally after updating the data set counter. DUST process suspends acquisition while 13 is active.

10.18.19.1 Operational notes: scheduling modes 8 & 9

The burst of activity may be aborted early by altering the scheduling mode to a value of 0 or 1. Aborting the burst leaves the internal counter untouched, so it is possible to continue acquiring the burst by setting the scheduling mode to 12 or 13. Using a scheduling mode of 8 or 9 reloads the internal counter (from location 0x5A) and starts a new burst.

Consider a data throttling mode that provides for external initiation of a burst scheduling mode (i.e. 8 or 9) and where we achieve data throttling by shutting the burst down early by changing the scheduling mode to 0 or 1. Each time the burst is started (8 or 9) a new count value (from location 0x5A) is loaded and we start from the beginning part way through some bursts, data-throttling causes the burst to be terminated early, leaving the count non-zero. This will work fine as each time the burst is started the count is reloaded.

10.18.20 73MEM_TWEAK, HRP, WORD, 0x58, *nnnn*, WBRC

8254 Offset from RTI. Used to control when the WBR sample occurs with respect to the RTI pulse from the S/C. This field is expressed in WBR samples so it is dependent on the current WBR sample rate. In addition there are 2 special cases.

nnnn	Description	MSF
0x0000	Immediate scheduling	set to 1
0x0001..0xFFFE	Offset from next RTI	
0xFFFF (-1)	Synchronize with WFR sampling	set to 1

The MSF bit is set when the indicated scheduling offsets are selected. This will add 2 status bytes to the WBR minipacket.

These values will cause data sampling to occur near the end of the RTI period. It may be desirable to move the sampling up in the RTI period slightly to avoid hitting the 1553 traffic that causes the RTI interrupt. The 1553 transaction that causes the RTI interrupt (and the synchronization pulse within RPWS) occurs about 2uSec prior to the RTI as seen within RPWS.

	sample/RTI	512 samples	1024 samples	2048 samples
36uS	3,472	2,960 (0B90)	2,448 (0990)	1,424 (0590)
4.5uS	27,777	27,265 (6A81)	26,753 (6881)	25,729 (6481)

V2.7 note: When 0xFFFF is specified in this command, the compression command that is passed from the timing control process to the data acquisition process is forced to ZERO (i.e. Compression is disabled). This avoids triggering a hardware deadlock condition that will hang the HRP and eventually cause the processor complex to reset and reload from bulk memory.

V2.7 note: When 0xFFFF is specified in this command, the data set size is limited to 2048 samples. This avoids causing data corruption in both WBR and WFR/LFDR data.

10.18.2173MEM_TWEAK, HRP, BYTE, 0x5A, *nn*, WBRC (V2.5 only)

Version V2.5 flight software implements a software interlock to prevent interference with the WFR when WBR is operating in high band mode. This is accomplished by having W12I ask for the WBR Mx when both WFR and WBR are in high band (FSW assumes that when WFR is using low-band, that the WBR will be operating in synch with the WFR sampling pulse).

Unfortunately, the W12I process looks at the wrong control word (there is a simple error in the source that will be corrected in V2.6 software). This tweak is to the control word that W12I inspects for making the decision. Setting the variable to non-zero will cause W12I to ask for WBR Mx when W12I is performing a high-band acquisition.

nn	Description
0x00	Cause W12I to ignore WBR Mx
0xFF	W12I will seize WBR Mx

Version V2.6 flight software corrects the hiband lockout so that it is not necessary to explicitly manage the flag and reassigns the meaning of this field. In V2.6 this field holds the burst count. Altering this field in a V2.5 fashion is incompatible with using the new WBR burst mode.

10.18.22 73MEM_TWEAK, HRP, BYTE, 0x5A, *nn*, WBRC (V2.6)

Version V2.6 flight software uses this field to hold the WBR BURST mode count. This is a count of the number of WBR data sets to acquire in burst mode.

This field is treated in the same manner as all control fields, that is to say, they are read-only as far as the control process is concerned. The data-set count must be (re)loaded only when it changes.

The time required to acquire the data sets is determined by the timing control words at offsets 0x50 and 0x52. Also note that offset 0x54 can affect timing by preventing data buffers from becoming available for re-use when delivering data to the LRS handler (it is ineffective when delivering data through the BIU-direct path).

10.18.22.1 V2.5 / V2.6 compatibility

Upgrading to V2.6 should not present significant impacts as the use of offset 0x5A is somewhat mutually exclusive.

V2.5 compatible IEB's treat this location as a hi-band flag while V2.6 software correctly handles the interlock function correctly (looking at the hi-band location). Setting the location, with the intent of causing the lockout to occur will continue to cause the lockout.

V2.5 IEB's that make use of the burst collection scheduling mode are created knowing that the hi-band flag is handled correctly.

10.18.23 73MEM_TWEAK, HRP, BYTE, 0x68, *nn*, WBRC

AGC Enable Flag. This is the location that stores the AGC enable bit. It may be directly altered to enable/disable gain without changing the gain level. Trigger 10 alters this memory location in lieu of using the **73WBR_GAIN_CNTL** command to avoid changing the current gain selection. This avoids having WBR step gain levels whenever we execute a trigger that uses trigger 10 code to perform preliminary WBR configuration/setup.

10.18.24 73MEM_TWEAK, HRP, BYTE, 0x50, *n*, W08I

AGC update rate control. This field controls the rate with which the gain level is updated when operating the WBR. When operating the WBR at elevated data rates, the double buffering will cause erratic gain updates to the WBR when attempting to update with each data set.

N	Description
1	Updates ACG following each data set Use only when acquiring at rates below 1 data set per RTI
2	Updates AGC every other data set Suitable for elevated acquisition rates.
n	Updates AGC every n data sets Suitable for all acquisition rates.

V2.5 software alters the behavior of the AGC slightly. Rather than performing updates every n^{th} data set, AGC updates are suppressed for *n*-1 data sets. When conditions are changing, this allows updates at the desired rate but when conditions are static, it allows a change to occur quickly.

As a result of this new software, gain changes do not necessarily occur at a predictable time, they simply do not occur any closer than the prescribed number of data acquisitions.

V2.6 software actually

10.18.25 Byte offset 0x51 PROCESS W08I

Working counter for gain control activity.

Do not alter this location when using the new WBR Burst command/capability. It may be possible to load a value of 1 into this counter for early termination, but setting offset 0x56 in WBRC to a 0 or 1 would be easier to understand.

10.18.2673MEM_TWEAK, HRP, BYTE, 0x52, *n*, W08I

MUX Control. The default value (ZERO) for this field leaves the antenna MUX with it's current setting. Deselecting the MUX should allow the AGC integrator to settle out prior to acquiring a data set (this may improve AGC performance when operating the WBR in a toggle mode or when operating with DUST active).

Note that this tweak may also be used to select a specific channel of the MUX. This may be useful in a toggle mode where the 2nd. channel is not using AGC control.

n	Description
0	Leave MUX selected to current channel **73WBR_BURST**
0x80	MUX reverts to Ex antenna
0x81	MUX reverts to Bx antenna
0x82	MUX reverts to Ez antenna
0x83	MUX reverts to HFR
0x84	MUX reverts to Langmuir Probe
0x07	Deselect MUX between data sets

A non-zero value enables MUX reversion with the upper 5 bits being stripped prior to use in the antenna select port.

MUX control and AGC

MUX setting has a noticeable impact on what the AGC integrator delivers to the AGC A/D convertor. Switching the MUX can present two signals to the AGC integrator; one being the signal from the *other* antenna (i.e. When set to a different antenna than what the WBR is looking at), and the second being a transient caused by a DC offset between the two antennas. The transient can be relatively large (and, therefore, the impact on AGC can be quite noticeable).

To operate in BURST mode (i. e. using the 73WBR_BURST command) a value of 00 might be best in the W08I MUX control location.

It appears that gain level changes may intrude into WBR/MFR data when the MUX byte is set to zero ???

10.18.27 Byte offset 0x53 PROCESS W08I

Spare, to maintain word and page alignment. This location simply keeps the process descriptor of the next process aligned on a 16 byte boundary.

10.18.28 73MEM_TWEAK, HRP, BYTE, 0x60, *nn*, DIR_

High Rate Science data movement control. This field controls the HRS data formatting and is, effectively, shared for WBR and WFR delivery activities. Both data sources may not use HRS at the same time. A discussion of the appropriate values are listed in a previous section on HRS data.

10.18.29 Byte offset 0x61 PROCESS DIR_

Spare, to maintain word and page alignment.

10.18.30 Byte offset 0x62/0x63 PROCESS DIR_

Working location for 0x60 to allow for mode changes.

10.18.31 73MEM_TWEAK, HRP, WORD, 0x64, nnnn, DIR_

Minipacket size for WBR data. This field is used to control the size of the minipacket that is formatted for delivery through the HRS stream. The default value of 1024 is appropriate for all WBR modes.

Some WFR modes require a different value to be specified in this field to accommodate a large dataset size.

10.18.32 73MEM_TWEAK, HRP, WORD, 0x66, nnnn, DIR_

Minipacket size for WFR data. This field is used to control the size of the minipacket that is formatted for delivery through the HRS stream.

10.19 WFR Control

Wave Form Receiver commands. Note that the compression switch is present in two separate commands and this can present an inconsistency in building the commands. Please refer to the *suggested order list* when building **WFR** commands to avoid placing the instrument into an invalid state.

Commands may be sent to the WFR when the instrument is in *SLEEP* mode although they will not be immediately processed. It should not be possible to bring the instrument out of SLEEP mode inadvertently (i.e. modifications to offset 0x56).

Suggested order of commands:

73WFR_ANT_SEL

73WFR_AUTO_CNTL

73WFR_CHAN_CNTL

73WFR_DATA_CNTL

73WFR_GAIN_CNTL

73WFR_MODE_CNTL, HOLD

73WFR_CMPRS_CNTL

73WFR_TOGGLE_CNTL

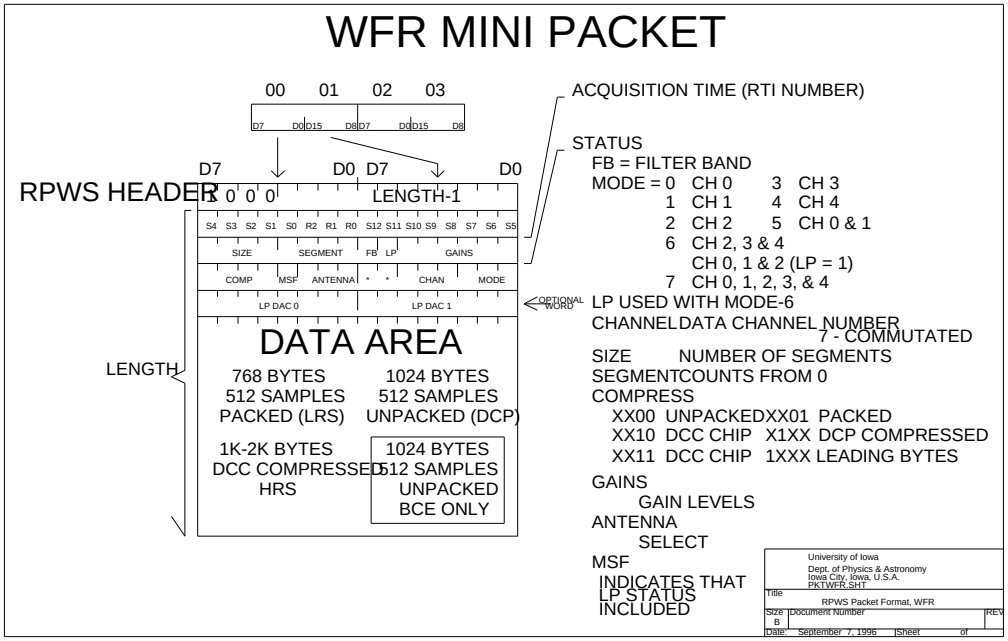
73MEM_TWEAK offset 0x50

73MEM_TWEAK offset 0x52

73MEM_TWEAK offset 0x54

73MEM_TWEAK offset 0x56

Version V2.3 software has a mechanism in place to prevent 12 bit data acquisition from being corrupted by 8 bit data acquisition. When either WBR or DUST are in high band mode, the 12 bit system will ask for the WBR Mx flag whenever performing a high band data acquisition. The Mx is not requested during low band activities under the assumption that 8 bit system will be using set to synchronize with the sampling on the 12 bit system and acquire small data sets (i.e. 2048 samples or less). This mechanism allows WBR high band acquisition to occur during ongoing LFDR acquisitions while allowing for the occasional WFR capture.


WFR MINI PACKET
00 01 02 03
D7 D0 D15 D8 D7 D0 D15 D8
D7 D0 D7 D0
RPWS HEADER
1 0 0 0
LENGTH-1
S4 S3 S2 S1 S0 R2 R1 R0 S12 S11 S10 S9 S8 S7 S6 S5
SIZE
SEGMENT
FB
LP
GAINS
COMP
MSF
ANTENNA
CHAN
MODE
LP DAC 0
LP DAC 1
OPTIONAL WORD
DATA AREA
LENGTH
768 BYTES
512 SAMPLES
PACKED (LRS)
1024 BYTES
512 SAMPLES
UNPACKED (DCP)
1K-2K BYTES
DCC COMPRESSED
HRS
1024 BYTES
512 SAMPLES
UNPACKED
BCE ONLY
ACQUISITION TIME (RTI NUMBER)
STATUS
FB = FILTER BAND
MODE = 0 CH 0 3 CH 3
1 CH 1 4 CH 4
2 CH 2 5 CH 0 & 1
6 CH 2, 3 & 4
CH 0, 1 & 2 (LP = 1)
7 CH 0, 1, 2, 3, & 4
LP USED WITH MODE-6
CHANNEL DATA CHANNEL NUMBER
7 - COMMUTATED
SIZE NUMBER OF SEGMENTS
SEGMENT COUNTS FROM 0
COMPRESS
XX00 UNPACKED XX01 PACKED
XX10 DCC CHIP X1XX DCP COMPRESSED
XX11 DCC CHIP 1XXX LEADING BYTES
GAINS
GAIN LEVELS
ANTENNA
SELECT
MSF
INDICATES THAT LP STATUS INCLUDED
University of Iowa
Dept. of Physics & Astronomy
Iowa City, Iowa, U.S.A.
PKTWFR.SHT
Title
RPWS Packet Format, WFR
Size B
Document Number
REV
Date: September 7, 1996
Sheet of

10.19.1 WFR Timing Control Process: Setup Storage Area

OFFSET	Label	Default	Description
0x2430	CMDp_Divisor		MOD timing divisor
0x2432	CMDp_Remainder		MOD timing remainder
0x2434	CMDp_Packet_Delay		Inter packet delay
0x2436	CMDp_Run		Run Flag
0x2437	CMDp_AGC		
0x2438	CMDp_Offset		8254 timer, offset from RTI to start of data capture
0x243A	CMDp_Mode		Channel Mode: 1ch,3ch,5ch
0x243C	CMDp_Band		Band Select
0x243E	CMDp_Length		Dataset byte count (8237 word count)
0x60	CMDp_Destination		Data Routing Destination
0x2442			
0x2444	CMDp_AGC_H_L		AGC set points
0x2446	CMDp_Antenna		Antenna Selection
0x2447	CMDp_Gain_0		Gain Setting: Channel 0
0x2448	CMDp_Gain_1		Gain Setting: Channel 1
0x2449	CMDp_Gain_234		Gain Setting: Channels 2, 3, 4
0x244A	CMDp_Walsh		Walsh number
0x244C	CMDp_DCC		ACTEL register 43 Pattern
0x244E			

Field	*Field Location*	*Description*
Minipacket ID	Byte 0 / D7..D4	1000
Minipacket Length	Byte 0 / D3..D0 Byte 1 / D7..D0	12 bit length
RTI	Byte 2 S4..S0, R2..R0 Byte 3 S12..S5	RTI time of data acquisition Seconds & RTI
Size	Byte 4 / D7..D4	Number of segments in data set (n - 1)
Segment	Byte 4 / D3..D0	Segment number (0 to n-1)
FB	Byte 5 / D7	Frequency Band 1 = 140 uSec (2.5Khz) 0 = 10 mSec (40hz)
LP	Byte 5 / D6	LP mode 0: mode-6 is channels 2,3,4 1: mode-6 is channels 0,1,2
Gain 2 3 4	Byte 5 / D5..D4	Gain level, channels 2, 3, 4 0dB to 30dB in 10 dB steps
Gain 1	Byte 5 / D3..D2	Gain level, channel 1 0dB to 30dB in 10 dB steps
Gain 0	Byte 5 / D1..D0	Gain level, channel 0 0dB to 30dB in 10 dB steps
Compression	Byte 6 / D7..D4	0
MSF	Byte 6 / D3	More Status Follows 1 indicates bytes 6 and 7 present
Antenna	Byte 6 / D2..D0	Antenna Selection
Walsh DGF	Byte 7 / D7..D6	Walsh Digital Gain Factor (valid only for DCP compression)
Toggle	Byte 7 / D7	Toggle Enable bit (when data is passed directly to the LRP)
AGC	Byte 7 / D6	AGC enable bit (when data is passed directly to the LRP)
Chan	Byte 7 / D5..3	Channel
Mode	Byte 7 / D2..D0	Channel Mode
LP DAC 0	Byte 8 / D7..D0	Langmuir Probe DAC 0 voltage when MSF set
LP DAC 1	Byte 9 / D7..D0	Langmuir Probe DAC 1 voltage when MSF set

10.19.2 Status Field

10.19.2.1 Size/Segment, Byte 4

WFR data may be up to 10,240 samples (20,480 bytes) which requires a minimum of 3 bits (each) for the size/segment field. By allocating a full 4 bits for each field, it becomes possible to acquire large datasets while still making use of the DCP to perform compression tasks (DCP will correctly process only WFR datasets of 512 or 1024 ?? samples).

The range of these fields is 0-15, so the number of segments in the datasets is expressed as (n-1).

10.19.2.2 Frequency Band, Byte 5, D7

As indicated in the table above, this bit indicates which sample rate was used to acquire this dataset.

0 indicates low band, 10mSec sample period, 26Hz Filter.

1 indicates high band, 140uSec sample period, 2.5Khz filter.

10.19.2.3 Langmuir Probe, Byte 5, D6

When set, indicates that the Langmuir Probe configuration is in effect for mode-6 (3 channel). The 3 channel mode may select either magnetic sensors (i.e. channels 2, 3, 4) or the electric sensors (0, 1, 2). The 0, 1, 2 sensors may also be connected to the Langmuir Probe sensors (hence the LP designation).

10.19.2.4 Gain select, Byte 5, D5-4, D3-2, D1-0

Gain amplifier setting for magnetic sensors (i.e. channels 2,3,4; bits 5-4), and electric sensors (channel 0 uses bits 1-0, channel 1 uses bits 3-2). Value of 0 to 3 indicates 0dB to 30dB in 10dB steps.

LP bit does NOT change channel assignments.

10.19.2.5 Compression, Byte 6, D7-4

This indicates the type of compression applied to the data (as well as where compression was applied).

10.19.2.6 MSF, Byte 6, D3

More Status Follows. This bit, when set, indicates that an additional 2 bytes of status appear in the status area (data starts 2 bytes later).

Similar to the MSF bit in the WBR, this bit indicates that two additional status bytes are included in the minipacket. These additional status bytes hold the current L/P DAC

settings and are included whenever one of the 5 WFR channels is connected to any of the L/P inputs.

10.19.2.7 Antenna Select, Byte 6, D2-0

Channels 0, 1, and 2 my be connected to electric/magnetic sensors or to the Langmuir Probe with these bits indicating the current selection.

D2 is the select bit for channel ?

D1 is the select bit for channel ?

D0 is the select bit for channel ?

10.19.2.8 Walsh DGF bits, Byte 7, D7-6

These bits are not shown in the above packet diagram as they were added following the construction of the diagram. They indicate ?? happened in the Walsh transform, and therefore are only valid if data has passed through the DCP.

Internally these bits hold the TOGGLE and AGC bits. These bits are used by DCP to accomplish AGC functions and would normally be stripped on the DCP (or replaced with the Walsh DGF bits), but will appear if the data is shipped directly to LRP (either BIU direct or bypassing DCP to achieve higher data rates).

Note that the internal bits will appear in the WFR minipacket when the DCP is bypassed (i.e. in the case of LFDR dual routing or when the DCC hardware is in use).

The normal decompression performed on the WFR data should remove these bits as part of the decompression activity. If these bits appear set in decompressed data, this should indicate TOGGLE MODE or AGC ENABLE. The compression status may be used to determine their validity (i.e. A compression status that indicates the data has passed through DCP would indicate these to bits are Walsh DGF, while any other compression status pattern would indicate these are TOGGLE MODE and AGC ENABLE bits.

10.19.2.9 Toggle/AGC, Byte 7, D7-6 (also Walsh DGF Bits)

These two bits are *overloaded* in that they perform 2 distinct functions. Within the instrument, these bits allow the band toggle mode to manage gain levels independently (i.e. high band vs. low band). This management function is not really necessary for interpreting data on the ground so when we make use of the Walsh compression method, these bits are used to deliver a *digital gain factor* with the dataset.

Keeping in mind that we would, for the vast majority of cases, compress WFR data and these bits will deliver the DGF status. Occasionally, however, we manage to deliver raw WFR data to the ground (dual routing LFDR delivers 12 bit data directly to the LRP, so we will see the data as uncompressed WFR data on the ground). When using one of the unusual

configurations, the bits will contain the gain control bits that we need internally to implement gain control.

Handling of Toggle/AGC bits during Decompression (ground)

It seems like the correct handling of these bits on the ground would be to clear them whenever we encounter uncompressed data (if the data is not compressed, we can't have any shifting that is the result of the Walsh transform and the DGF would be zero).

10.19.2.10 Channel, Byte 7, D5-3

The dataset is typically demultiplexed prior to delivery (this is to accommodate the compression performed on DCP). This field indicates which of the 5 channels this data set corresponds with. Values of 0 to 4 indicate the 5 channels while **a value of 7 is used to indicate multiplexed data** (when using the DCC chip to perform compression).

Note that decompression, performed on the ground, will demultiplex the data and produce individual datasets for each channel. Each dataset will have the channel bits set appropriately.

10.19.2.11 Mode, Byte 7, D2-0

This status field indicates the channel mode the receiver is operating in. There are 5 single channel modes (corresponding to the 5 channels available with this receiver), a 2 channel mode (using channel 0 and 1), two 3-channel modes (the LP bit is used here), and a 5 channel mode.

10.19.2.12 LF-WBR Notes

LF-WBR mode involves operating the WFR in a single channel mode to emulate the operation of the WBR. This mode makes the slower capture rates of the WFR available as a psuedo-wideband receiver. To achieve near continuous coverage, the data set size may be increased to make use of an entire WFR buffer of 10240 samples.

There is a potential hardware issue when operating in low band (i.e. 100Hz sample rate). A data set, when configured in this manner, requires in excess of 100 seconds to acquire. When changing WFR modes, it is necessary to allow any active data set acquisitions to complete prior to reconfiguring the WFR. This is easily accomplished with minimum impact to other data collection activities by idling the WFR a few minutes before sending any other commands to the instrument. As an example, the following command may be used to stop the WFR from acquiring any additional data sets.

73MEM_TWEAK, HRP, BYTE, 0056, 0000, WFRC

The symptoms of failing to observe this restriction is usually a hang in the WFR that may require a reload of flight software (use care as you may be waiting for an acquisition to complete). This can occur when using the DCC hardware and terminating a data collection

cycle early. This seems to place the WFR/DCC into an odd state that (permanently) prevents further use. It looks as though the hardware should provide a means to reset the DCC so if this becomes an issue there should be some method for the software to recover (this should be possible using 00PORT_TWEAK commands).

10.19.3 Data Format

The **RAW** data consists of a series of 16 bit words containing 12 bit samples (upper 4 bits are zero). In most cases the segments (as well as the data sets) are a convenient power-of-2 size (i.e. 1024 or 2048 samples). Also, the multi-channel modes are typically decomutated prior to delivery to accommodate compression activities on the DCP, a complete cycle of data would consist of a series of 5 datasets.

When the DCC hardware is used for compression, all channels are mixed as this is how data is presented to the hardware and cannot be decommutated prior to the decompression step. In this case the data is mixed, in order (i.e. channels 0-4 sample 0 occurs followed by channels 0-5 sample 1, etc.).

10.19.4 DCP Resource Requirements

As with Langmuir Probe, WFR acquires 12 bit samples (storing this in a 16 bit word) so we can approach data compression in several manners.

DCP CPU requirements	Description
N/A	RAW (bypass DCP)
1 second 512 sample packet	PACK
	RICE
	WALSH
	WALSH/RICE

10.19.5 73WFR_ANT_SEL, sensor 0, sensor 1, sensor 2

Three of the five WFR channels make use of a 2 channel multiplexer to select 3 additional receiver channels. Channels 3 and 4 are fixed on the search coil.

sensor 0	Description
exlo	electric dipole antenna, X
lmr	langmuir probe cylinder

sensor 1	Description
ezlo	electric monopole antenna, Z
lmr	langmuir probe cylinder

sensor 2	Description
bx	magnetic search coil
lp	langmuir probe sphere

10.19.6 73WFR_AUTO_CNTL, low set, high set, average interval, time constant

This command is rerouted for processing on the DCP. The WFR hardware does not contain any hardware assists for use with automatic gain control, so the software must examine the data to determine when a gain change is required. This activity, performed on the DCP, is controlled using this command.

10.19.7 00WFR_AUTO_ADJ, direction, channel

This is an internal command used by the AGC software. It is used to adjust the gain setting 10dB in the specified direction for the specified channel. If the command requests an invalid gain setting (i.e. below 0dB or above 30dB) it is ignored.

10.19.8 00WFR_AUTO_SET, agc, channel, gain 0, gain 1, gain 2-3-4

This internal command is used to select a gain settings independently for the primary and secondary bands (this is how we manage gain when operating in a toggle mode). The *agc* field selects manual(MAN) or automatic(AUTO) gain. The channel field selects the primary(PRI) or secondary(SEC) band, and the three gain fields are similar to the gain selects in the **73WFR_GAIN_CNTL** command.

10.19.9 00WFR_RATE_CNTL, direction

This internal command alters the timing period specified at word 0x50. The value specified is expressed in RTI ticks.

10.19.10 73WFR_CHAN_CNTL, channel select

Selects channels to sample.

Channel select	Description
CH0, CH1, CH2, CH3, CH4	single channel (using selected channel)
CH01	dual channel mode
CH012 PROBE	three channel mode using first three channels
CH234	three channel magnetic
CHALL	all channels

10.19.11 73WFR_CMPRS_CNTL, enable, route, word count

This command is used to specify the total number of samples (i.e. 3 channels of 1024 samples each requires 3072 for a sample count). The compression control is used to control use of the hardware compression. Software compression **requires** that hardware compression be disabled. The **route** parameter determines if the packet length is determined before or after compression.

The WFR is double buffered with a maximum sample count of 10240. The sample count is a combined count for all channels so the sample count must be multiplied by the channel count to arrive at the word count.

enable	Description
off	hardware compression not used
on	hardware compression enabled *word count* restrictions

route	Description
wc_out	EOP generated by WCR in 8237 only word count is the number of 16 bit words left in memory (number of samples may be more or less than indicated by word count)
wc_in	EOP generated by WCR in Actel or 8237 word count is the number of 12 bit samples delivered into the DCC hardware. Number of samples may be less if compression is poor.

word count	Description
nn (DCC/wc_out)	number of samples in data set total sample count, **not samples/channel** 10240 samples maximum
nn (DCC/wc_in)	sample count must be a multiple of 1024 samples when using DCC compression.

Note the restriction on the word count when operating the WFR through the DCC compression hardware and controlling the number of samples. When DCC/wc_in is specified, the sample count register, which is loaded with a multiple of 1024 samples, is enable and used to terminate the data acquisition. The lower 10 bits of the word count are truncated/discarded as the sample count register is only 4 bits wide (thus limiting the absolute data set size to 16,384 samples).

Keep in mind that DCC is rarely used in conjunction with the WFR so this problem does not appear often. The symptom of using an invalid word count being a data set with an unexpectedly small number of samples.

Also look in the 73DCP commands for discussions about CPU limitations on the compression processor. This will affect setups where L/P and WFR are attempting to operate at elevated bit rates. The WFR may be commanded into data collection schedules that use excess CPU cycles on the DCP.

10.19.12 73WFR_DATA_CNTL, destination

Sets the route that the data product will take within the instrument. The data may be delivered to the LRP directly (although this is not desirable in flight), through the software compression on the DCP (where automatic gain processing occurs), or directly to CDS using the HRS route.

destination	Description
lrs	WFR data delivered to Low Rate Science stream on Low Rate Processor
hrs	WFR Data delivered to High Rate Science stream on High Rate Processor
dcc_lrs (dcp_lrs)	WFR data delivered to DCP for Walsh/Rice compression

Note that the label used to route data to the compression processor (for software compression) is confusing. The mnemonics were selected very early, prior to code implementation in some cases. Changing the mnemonics involves filing ECR requests with JPL and the possibility that the commands will be mis-interpreted so we have elected to keep the (confusing) keywords.

Refer to the IEB source code for examples of setting compression method correctly.

10.19.13 73WFR_GAIN_CNTL, control, gain 0, gain 1, gain 2-3-4

Selects the gain setting for the 12 bit system. There are 3 separate controls with channel 0 and 1 having independent control and channels 2, 3, and 4 making use of a common gain control.

As with LFDR, V2.6 adds the ***SET*** keyword to the control field. This sets gain levels without changing the AGC setting and is effective only when AGC is set to AUTO. This new keyword is intended to be used internally by the AGC code that runs on DCP.

control	Description
MAN	manual gain control
AUTO	automatic gain control (data must be routed to DCP for AGC to function)

gain0, gain1, gain234	Description
0 10 20 40	gain level, in dB

10.19.14 73WFR_MODE_CNTL, HOLD, band, compression

Selects the sample rate and anti-aliasing filter.

For most setups the WFR will make use of the DCP for compression which dictates that the compression selection in this command be set to ***NoCompress***. Selecting compression with this command and routing data to the DCP will corrupt the compression status bit rendering the data unreadable.

band	Description
lband	10 mSec Sample rate
hband	140 uSec sample rate

compression	Description
nocompress	no **hardware** compression
compress	**hardware** compression (ISFLIP chip)

HEY! This is ***HARDWARE*** compression select **!!!**

These are the two forms that this command should take when sending through the DCP for packing or compression.

73WFR_MODE_CNTL, HOLD, LoBand, NoCompression

73WFR_MODE_CNTL, HOLD, HiBand, NoCompression

10.19.15 73WFR_TOGGLE_CNTL, control, gain 0, gain 1, gain 2-3-4 (V2.6)

Selects the gain setting for the high band toggle mode.

A toggle mode was added in the V2.6 software to allow WFR data, to be collected in alternating low-band and high-band mode. This is accomplished by having WFR flip the band selection with each data acquisition (we are still limited by the single 5 channel 12 bit data acquisition system on the HRP as well as the CPU cycles available on the DCP to perform compression) . To avoid gain problems, a separate gain state is maintained for the alternate band with this command being used to set the alternate gain state.

KEEP IN MIND that using this feature may affect the data collection schedule when operating at elevated data rates due to the increased time required to collect data in lo-band mode. This is noticeable with **Trigger 1A** that appears in the BASE.IEB load that is included with the ALF load, the hi-band acquisition occurs immediately following the lo-band acquisition but is delayed a few RTI periods. In this example, dat acquisition requires 20.47 seconds, leaving barely enough time to get the next acquisition setup before the RTI (seems to be 4 or 5 RTI's late).

The software accomplishes the toggle mode by storing an alternate set of gain values and simply inverting the band control bit, with these actions being taken every other acquisition. It is not possible to alter any other settings (as this is intended to simply be a band toggle mode) although we don't really care if the primary setup is hi-band or lo-band (at least not within the WFR acquisition control code).

Note that this implies that the gain selection in the **73WFR_TOGGLE_CNTL** command refers to the gain of the alternate band (not high band or low band, per se). The primary band is selected in the **73WFR_MODE_CNTL** command and the gain selection in this command refers to the alternate band selection (i.e. the other band).

An AGC assist is also implemented by the *control* keyword **SET**. This control may be used to alter gain settings without affecting the AGC setting. In addition, the gain level will only be changed if AGC is enabled. This control, therefore, may be used by the AGC software on DCP to cause the AGC control bit to become effective. (Internally there is an AGC control command that has primary/alternate bit that is not documented here)

When WFR is being operated exclusively in high-band or low-band mode, the this command would not be used.

control	Description
OFF	Disable toggle mode
MAN	Enable TOGGLE mode without AGC control
AUTO	Enable TOGGLE mode ***with*** AGC control
SET	Change gain level, leaving AGC enable unchanged

gain0, gain1, gain234	Description
0 10 20 40	gain level, in dB

10.19.15.1Compatibility note

This command is implemented as an extension to the WFR gain command (**73WFR_GAIN_CNTL**) in a manner that is reasonably compatible with earlier version of the flight software. This trigger can be used with earlier versions of software as long as the trigger is immediately followed by a gain control command. The reason for this restriction is that older versions of the flight software will decode the **73WFR_TOGGLE_CNTL** command as a **73WFR_GAIN_CNTL** command and make use of the indicated gain levels. By following the toggle command with a normal gain command, the effect in prior versions is nullified while newer version will act accordingly.

In particular, trigger 10 will need to shut down toggle mode in a manner that is as universal as possible, and this is simply accomplished by adding the gain control command immediately prior to the gain control command (this was done around the time of the C32 submission).

Another way to obtain compatibility with older versions is to duplicate the gain selections from the **73WFR_GAIN_CNTL** command in the **73WFR_TOGGLE_CNTL** command. This way the mis-interpreted command duplicates the correct gain control command and the gain ends up being set as desired.

10.19.1673MEM_TWEAK, HRP, WORD, 0x18, 1, WFRC

This tweak may be used to bring the WFR timing control process out of a delayed state when changing instrument modes. Many of the WFR bit rates require rather long cycle periods and changing this period takes effect following the next scheduled data acquisition. If WFR timing control is not currently delayed this tweak will have no effect.

There are two methods of using this tweak, synchronized and unsynchronized.

The unsynchronized method will typically cause an immediate data acquisition to occur and requires a single tweak. This is accomplished by simply placing the tweak following the last WFR command (i.e. the setting of location 0x50 or 0x56).

```
73WFR_ANT_SEL
73WFR_AUTO_CNTL
73WFR_CHAN_CNTL
73WFR_DATA_CNTL
73WFR_GAIN_CNTL
73WFR_MODE_CNTL, HOLD
73WFR_CMPRS_CNTL
73MEM_TWEAK  offset 0x50
73MEM_TWEAK  offset 0x52
73MEM_TWEAK  offset 0x54
73MEM_TWEAK  offset 0x56
73MEM_TWEAK , HRP, WORD, 0x18, 0x01, WFRC
```

The synchronized method requires an additional tweak to location 0x56 to idle the WFR timing control process followed by the tweak to location 0x18 to bring the WFR timing control process out of a delay. These two tweaks should be placed **before** any other WFR commands.

```
73MEM_TWEAK , HRP, WORD, 0x56, 0x00, WFRC
73MEM_TWEAK , HRP, WORD, 0x18, 0x01, WFRC
73WFR_ANT_SEL
73WFR_AUTO_CNTL
73WFR_CHAN_CNTL
73WFR_DATA_CNTL
73WFR_GAIN_CNTL
73WFR_MODE_CNTL, HOLD
73WFR_CMPRS_CNTL
73MEM_TWEAK  offset 0x50
73MEM_TWEAK  offset 0x52
73MEM_TWEAK  offset 0x54
73MEM_TWEAK  offset 0x56
```

10.19.17 73MEM_TWEAK, HRP, WORD, 0x50, *nnnn*, WFRC

Scheduling period expressed in RTI ticks.

10.19.18 73MEM_TWEAK, HRP, WORD, 0x52, *nnnn*, WFRC

Scheduling offset expressed in RTI periods.

10.19.19 73MEM_TWEAK, HRP, WORD, 0x54, *nnnn*, WFRC

Delay (expressed in RTI's) between minipackets. Used to throttle the data delivered to either the DCP or LRP. The total WFR data buffer is 20K words requiring that data be delivered as slowly as it is processed.

10.19.20 73MEM_TWEAK, HRP, BYTE, 0x56, *n*, WFRC

Scheduling mode.

00 Stop

Data acquisition is stopped. ***73WFR_MODE_CNTL, TRIGGER*** is not processed. Internal triggers are not processed. This state is entered whenever sleep is asserted.

01 Idle

Data acquisition is stopped but the process will accept a trigger to perform a single acquisition. After the acquisition the process will return to idle to await further triggers.

12 Run

Data acquisition is continuous and based on the schedule specified at offset 0x50 and 0x52.

23 Trigger

- Single data acquisition.

10.19.2173MEM_TWEAK, HRP, BYTE, 0x5A, *nn*, WBRC

WBR Mx control word. WBR/WFR interference control.

See the WBR section for a discussion of this variable.

10.19.22 73MEM_TWEAK, HRP, BYTE, 0x60, *n*, WFRX

WFRo_SEGMENT_SIZE. Control the size of a WFR minipacket segment.

The nominal value in this location is 1024, giving a minipacket size of 512 samples (this number is expressed as a raw byte count). The nominal value is appropriate for use with the DCP compression routines. This size can be altered ONLY when not making use of the DCP to pack/compress the data. Specifically, when operating in a single channel mode (i.e. psuedo-WBR modes), it is preferred to limit the dataset size to 8192 samples and leave this parameter at it's default setting to maintain the capability of using DCP for compression when operating at lower bit rates.

10.19.23 Byte offset 0x62 PROCESS WFRX

Segment size working area.

10.19.24Byte offset 0x64 PROCESS WFRX

WFRo_DATA_ADDRESS

10.19.25Byte offset 0x66 PROCESS WFRX

WFRo_DATA_COUNT

10.19.26Byte offset 0x68 PROCESS WFRX

WFRo_MPCB_DATA_ADDRESS

10.19.27 73MEM_TWEAK, HRP, BYTE, 0x6A, *n*, WFRX

Minipacket delivery flags.

These values are added to obtain combined results. In particular, this location must be tweaked when high speed WFR data is being delivered to the DCP for compression.

1**1 Checksum enable** (default value)

Setting this bit causes a checksum to be calculated on each IPC packet sent to the IX queue for delivery.

12 Non blocking

Setting this bit will cause minipacket processing to abort whenever the free queue is empty (i.e. a conditional queue read is used to obtain delivery buffers and a read failure causes processing to be aborted).

14 Fast delivery

Setting this bit suppresses the 1 RTI delay between each IPC packet normally imposed by the minipacket assembler.

1**5 Checksum enable and fast delivery** (fast WFR data)

This setting is used to obtain peak WFR data rates through the Data Compression Processor. Eliminating the delay between IPC packets reduces the unrecoverable idle time on the DCP when running WFR data rates above 5,000 bits/second.

10.19.28 73MEM_TWEAK, HRP, BYTE, 0x60, *nn*, DIR_

High Rate Science data movement control. This field controls the HRS data formatting and is, effectively, shared for WBR and WFR delivery activities. Both data sources may not use HRS at the same time.

The only valid value for ***nn*** is ZERO when used with WFR data.

10.19.29 73MEM_TWEAK, HRP, BYTE, 0x50, *nn*, W12J (affects LFDR)

RST-5 Clock Enable flag. This field is normally set to a -1 and should not be altered. When cleared to zero, it causes the sample clock to the 12 bit A/D system to be stopped when data is not being actively acquired. This clock does not appear to interfere with anything within the instrument (i.e. the WBR).

Clearing this flag prevents WBR data acquisition when LBAND data is being acquired.

10.19.30 Byte offset 0x50 PROCESS W12J

Restart Clock. This location is set to 0xFF to keep the RST-5 clock running.

10.19.31 73MEM_TWEAK, HRP, WORD, 0x52, *nn*, W12J (affects LFDR)

Gain change settling time. This field is normally a 2 to allow at least 125 mSec of settling time prior to data acquisition in the event that a gain change has occurred. If both LFDR and WFR are operating, setting this field to zero will probably cause problems.

Gain level changes (i.e. auto gain) and/or antenna selection changes require this field to be set to 2 or greater.

This is intended to allow a Low-frequency WBR mode to be implemented at close to 100% duty cycle. The LFDR, is enabled, must use the same antenna and gain settings. Auto gain is not possible (i.e. gain level changes may trigger the AGC code to continuously change gain levels. Antenna selection is also critical. Even in single channel mode, the WFR must be commanded to use identical antennas as the LFDR for the unused antenna.

10.19.32 73MEM_TWEAK, HRP, BYTE, 0x66, nnnn, DIR_

Minipacket size for WFR data. This field is used to control the size of the minipacket that is formatted for delivery through the HRS stream. This field interacts with the memory map on the HRP so the value must be chosen with care (data from any individual minipacket must not be allowed to cross the memory bank boundary).

10.19.33 WFR Buffer Memory

WFR BUFFER 0 is located at 0x0C400 while WFR BUFFER 1 is located at 0x19400. This results in 0x3C00 (15,360) bytes of buffer 0 being located in bank 0 and 0x1400 (5120) bytes being located in bank 1. Buffer 1 is located entirely in bank 1.

When selecting sizes for minipackets care must be exercised to avoid having a minipacket cross the bank boundary as the data movement code selects the memory bank prior to processing each minipacket (if a minpacket crosses the bank boundary, data from logical addresses above 0x10000 will be taken from the bottom of memory, i.e. Code will be palced in the minipacket rather than WFR data). Note that since WFR buffer 1 is located entirely within bank 1 , that symptoms would be every other dataset has some corrupted data).

Another factor to keep in mind is that the minipacket assembler (does BIU_ do this as well?) decomutates the WFR data when operating in a multi-channel mode. The minipacket size is, effectively, multiplied by the channel count.

Minipacket size	***1 channel***	***3 channel***	***5 channel***
512	valid	valid	valid
1024	valid	valid	valid
1536	valid		valid
2048			
2560	valid	valid	
3072	valid		valid
3584			

10.20 Wrap Command

Provided to allow a subset of the commands listed to be implemented in the various parsers. This command allows any bit pattern (i.e. any command) to be generated.

10.20.1 73WRAP, (value[1..n])

Used to implement any command the instrument is capable of decoding. Commands not documented in 3-291 may be issued using this mechanism. When issuing ***73WRAP*** commands, responsibility for generating parity lies with the command generator. Wrapped commands may be specified with invalid parity.

Also note that because the ***73WRAP*** command simply defines a group of command words, multiple commands may be grouped into a single wrap command.

All RPWS commands may be wrapped, when needed. The command parsing tool, **parser**, located on the RPWS GSE systems is capable of translating all RPWS commands into this format.

The data items are a comma delimited list of hexadecimal numbers containing any number of command patterns that will fit within the limits of the ground system, CDS, and RPWS buffers. A value of less than 121 words is reasonable within the instrument.

10.21 Storage Manager (Special Maintenance)

Commands to this handler are in the form of memory tweaks. All of the Process Descriptor words are listed here for reference. Most of them are located here to allow the use of 73MRO for inspection although a current memory load map is required to determine the appropriate locations.

10.21.1 73MRO, LRP, hsk/tlm, SMGR+0x60 SM_Next_Address

Next Ram Disk Address to be written.

10.21.2 73MRO, LRP, hsk/tlm, SMGR+0x62 SM_Address

Current Ram Disk Address being written.

10.21.3 73MRO, LRP, hsk/tlm, SMGR+0x64 SM_Length

Size of Ram Disk buffer, always 128.

10.21.4 73MEM_TWEAK, LRP, WORD, 0x60, nn, SMHK HK_Flag

This location may be cleared (i.e. set to zero) to cause the housekeeping dump to begin at the beginning of memory.

10.21.5 73MRO, LRP, hsk/tlm, SMHK+0x62 HK_Address

Current Ram Disk buffer being moved to housekeeping

10.21.6 73MRO, LRP, hsk/tlm, SMHK+0x64 HK_Length

Size of Ram Disk buffer, always 128, just like SM_Length. If these two are mis-matched, you're in big trouble!

10.21.7 73MRO, LRP, hsk/tlm, SMST+0x60 Timeout_Count

Timer location to allow error recovery from lost Ram Disk packet.

10.21.8 73MRO, LRP, hsk/tlm, SMST+0x62 Power_Flag

Set to -1 when the Ram Disk is full and power has been removed from the HFR.

10.21.9 73MRO, LRP, hsk/tlm, SMST+0x64 Fault_Bits

Hold completion status bits.

10.22 Maintenance Manager (Special Maintenance)

This code manages the activities triggered by asserting the maintenance discrete bit.

10.22.1 73MEM_TWEAK, LRP, WORD, 0x60, nn, MANT Sleep Delay

This determines the minimum period that will be spent in SLEEP mode when the Special Maintenance software is loaded into the processor. The value loaded with the software (i.e. the default value) is 256 giving a minimum dwell time of 32 seconds.

10.23 Ram Disk Manager (Special Maintenance)

No tweak-able control fields. No data fields keyed to the P.D.

11 Flight Software Patches

As of 2003, it appears we will not be updating our flight software. Most of the problems encountered up until this point in time have been addressed with the V2.6 FSW load. Several small anomalies are still present, but we are able to address them with minor patches that are applied with a small number of MEM_TWEAK commands.

This section discusses the patches we have built, how they are applied, and when they are applied. Note that not all of the patches accomplish their goals (in which case they are noted as non-functional and simply document a potential problem and the remedy that was attempted)

11.1 V2.6 patch list

Patch 5 has undergone extensive testing and is incorporated into the base IEB starting with C39. The idle trigger performs the memory patches required to implement this update.

11.1.1 V2.6 Patch 001

```
#
#       Patch to V2.6 code
#       19 Sep 2002 WTR
#       73POWER_CNTL command in 73IEB_HALT, IDLE
#
#         Examine memory locations prior and subsequent
#       to patching.  This, I hope, will give us a record,
#       in telemetry, of the application of the patch.
#         This patch may be applied more than one time
#       without causing any problems.
#
# Patch 001      19 SEP 2002 73Pwr_Cntl, Pause
#
00:00:10 73mro, lrp, hsk, 3B54, 0                          # Examine the areas
00:00:12 73mem_tweak, lrp, word, 0x0858, 0x0101, IEBC      # Remove
#                                                          # Power_CNTL
#                                                          #  Instruction
00:00:14 73mro, lrp, hsk, 3B54, 0                          # Examine the areas
```

Patch 1 removes the last remnants of the **73POWER_CNTL, PAUSE** command from the internal IEB. The external IEB is updated continuously, so has not had any of this command present for some time.

This patch is not absolutely necessary as we never issue any commands that will cause the problem instruction to be executed.

11.1.2 V2.6 Patch 002

```
#  Patch 002      03 OCT 2002 DMA mode 3 fixup
#                 05 NOV 2002 Mode-3 doesn't even work,
#                         so there is NO point in fixing
#                         the code
#
#00:00:20 73mro, hrp, hsk, 37bd, 0                          # Examine
#00:00:22 00mem_tweak, hrp, byte, 0x37BD, 0x0011            # Change port address
#00:00:24 00mem_tweak, hrp, byte, 0x37C1, 0x0011            # Change other port address
#00:00:26 73mro, hrp, hsk, 37bd, 0                          # Examine result
```

Patch 2 is a failed attempt to address a problem in one of the transfer modes on the HRP. The correct solution requires an update to V2.7 flight software. The patch here is ineffective (note that using DMA MODE 3 on HRP will result in CPU lockup or total loss of WBR data).

11.1.3 V2.6 Patch 003

```
#
# Patch 003      12 DEC 2002 Time fixup on HRP
#                  Change RTI-0 signal to RTI-1.
#                  Allows HRP to slip 1 RTI wothout messing up
#                  Time too bad.
#  Found problem (hope we did anyway), so this
#   patch not really needed either...
#
#00:00:30 73mro, hrp, hsk, 0x0FC0, 0
#00:00:31 73mro, dcp, hsk, 0x0FE8, 0
#00:00:32 73mro, lrp, hsk, 0x4B40, 0
#00:00:35 00mem_tweak, HRP, BYTE, 0x0FC1, 0x01      # HRP RTI-1
#00:00:36 00mem_tweak, DCP, BYTE, 0x0FED, 0x01      # DCP RTI-1
#00:00:37 00mem_tweak, LRP, BYTE, 0x4B43, 0x63      # send 2 RTI delay
#00:00:40 73mro, hrp, hsk, 0x0FC0, 0                #   during RTI-7
#00:00:41 73mro, dcp, hsk, 0x0FE8, 0
#00:00:42 73mro, lrp, hsk, 0x4B40, 0
```

Patch 3 is a failed attempt to remedy a problem we seem to encounter on HRP on occasion (as described elsewhere, the HRP occasionally slips time when an interrupt is lost).

This patch will mask the time regression, limiting it to a single RTI. Note that this will also make it very difficult to locate bad sections of WBR data.

11.1.4 V2.6 Patch 004

```
#
# Patch 004      19 DEC 2002 Time fixup on HRP
#                  biuint_5 seems to be keeping interrupts
#                  off for too long.  stick in a pair of
#                  EI instructions to alleviate the problem
#
00:01:00 73mem_tweak, hrp, byte, 0x15, 0x7F, BIU_      # lower priority (BLOCK)
#                                                      # BIU handler
00:01:01 73mro, hrp, hsk, 29BB, 0                      # look at initial conditions
00:01:02 73mro, hrp, hsk, 2B44, 0                      #
#
#         Load new code fragments
#
00:01:05 00mem_tweak, hrp, byte, 0x3CF0, 0xFB          # EI
00:01:06 00mem_tweak, hrp, byte, 0x3CF1, 0x3A          # LDA
00:01:07 00mem_tweak, hrp, word, 0x3CF2, 0x23D2        # DMA_WORM_HOLES
00:01:08 00mem_tweak, hrp, byte, 0x3CF4, 0xC9          # RET
#
00:01:11 00mem_tweak, hrp, byte, 0x3CF5, 0xFB          # EI
00:01:12 00mem_tweak, hrp, byte, 0x3CF6, 0x3A          # LDA
00:01:13 00mem_tweak, hrp, word, 0x3CF7, 0x3C2C        # DMA_WBR
00:01:14 00mem_tweak, hrp, byte, 0x3CF9, 0xC9          # RET
#
#         We do it in this order so screw-up doesn't crash HRP
#
00:01:21 00mem_tweak, hrp, word, 0x29BC, 0x3CF0        # Patch LDA address
00:01:22 00mem_tweak, hrp, byte, 0x29BB, 0xCD          # Change to CALL
00:01:23 00mem_tweak, hrp, word, 0x2B45, 0x3CF5        # Patch LDA
00:01:24 00mem_tweak, hrp, byte, 0x2B44, 0xCD          # Change to CALL
#
00:01:31 73mro, hrp, hsk, 29BB, 0                      # look at final conditions
00:01:32 73mro, hrp, hsk, 2B44, 0                      # look at final conditions
00:01:33 73mro, hrp, hsk, 3CF0, 3CFE                   # look at final conditions
#
00:01:40 73mem_tweak, hrp, byte, 0x15, 0x6B, BIU_      # return priority to normal
```

Patch 4 is an attempt to remedy the lost interrupt issue. Here we patch in an *Enable Interrupt* instruction into the BIU-DIRECT process in an attempt to get interrupts enabled a little sooner.

11.1.5 V2.6 Patch 007

```
#
# Patch 005      20 MAR 2003 AGC Glitch (W08I)
# Patch 007      24 JUL 2003 (mem_tweak to 0x32CE should be WORD to 32CD)
#                 Gain control code fragment in W08I
#                  was glopping up the antenna select.  Puts glitches
#                  in WFR and MFR.  Found and extra instruction to
#                  delete (talk about careful planning).
#                 Also, the 73WBR_MODE_CNTL and 73DUST_MODE_CNTL
#                  commands are patched such that they no longer
#                  alter the MUX control byte
#
00:02:00 00mem_tweak, hrp, word, 0x32CD, 0
00:02:01 73mem_tweak, hrp, byte, 0x14, 0x40, TWEK
00:02:02 00mem_tweak, hrp, byte, 0x8225, 0x3A
00:02:03 00mem_tweak, hrp, byte, 0x82D4, 0x3A
00:02:04 73mem_tweak, hrp, byte, 0x14, 0x00, TWEK
```

Patch 5 eliminates an antenna switching glitch we have encountered from time to time in the instrument. This patch removes an instruction that incorrectly alters a the contents of the antenna control register for the WBR. By removing the offending instruction, the WBR data acquisition handler operates correctly (as originally intended). This also permits the 73WBR_BURST command to operate as intended.

In flight, we rarely see the effects of this as the offending code fragment is executed only during a change in WBR gain (this occurs very infrequently during cruise).

11.1.6 V2.6 Patch 00**8**

```
#
# Patch 008      04 FEB 2004 HSK RTI Loss Counter (HSK_)
#                  Move RTI LOSS field from data page to HSK page
#
00:02:30 00mem_tweak, lrp, byte, 0x18F0, 0x21 # Lxi H,
00:02:31 00mem_tweak, lrp, word, 0x18F1, 0x0039       #       RPWS_RTI_LOSS
00:02:32 00mem_tweak, lrp, byte, 0x18F3, 0x19 # Dad D
00:02:33 00mem_tweak, lrp, byte, 0x18F4, 0x3A         # Lda
00:02:34 00mem_tweak, lrp, word, 0x18F5, 0x1046       #       RTI_LOSS LSB
00:02:35 00mem_tweak, lrp, byte, 0x18F7, 0x77 # Mov M, A
00:02:36 00mem_tweak, lrp, byte, 0x18F8, 0xC3         # Jmp
00:02:37 00mem_tweak, lrp, word, 0x18F9, 0x17C7       #       "Step"
00:02:38 00mem_tweak, lrp, word, 0x1855, 0x18F0       # Patch it into HSK_
```

Patch 6 is to allow the RTI LOSS counter located in the system data page to appear in the housekeeping stream. It was supposed to happen this way, but it looks like it got missed.

12 Telemetry Modes

Telemetry modes have changed very little over the course of the mission, but with the arrival at Saturn some modes that have been deferred will begin to appear on the Space Craft. This section of the Users Guide will attempt to provide a list of those telemetry modes that the various versions of the software support and methods to work around any unusual telemetry modes that might suddenly appear (any change in telemetry modes appears sudden to us).

Version V2.6 makes some minor changes to the telemetry modes that we implemented in the initial release of the flight software.

As noted elsewhere in the manual, we deliver either LRS or HRS during any given RTI period (never a combination of 1 LRS and the rest HRS). 1 LRS packet or up to 6 HRS packets (or an empty packet). We NEVER deliver both LRS and HRS in the same RTI.

12.1 Spacecraft Telemetry Mode changes

Telemetry modes in the instrument are controlled in several ways. The STM words delivered with the time update (once each second) are used to enable HRS and to set the size of the HRS buffer (i.e. the number of CDS packets that will be delivered with each transaction). Unknown STM patterns are handled by leaving the telemetry mode unchanged (i.e. continuing with the previous mode). When a new telemetry mode is encountered a HRS enable flag and the packet count (number of CDS packets to be delivered during each RTI period from HRP to LRP, the HRS packet count) are updated. The HRS packet count is then used to calculate a word count for BIU direct transactions (packet count • packet size). The word count is later propagated to the HRP.

When changes in BIU direct word count occur, we expect to see a mid-match in word counts to cause an IPC timeout if BIU direct data transfer is active. In most cases, however, a mode change will occur when there is no BIU direct activity and we will not notice the short period of time where LRP and HRP have mis-matched BIU direct word counts. In most cases we activate HRS modes for a short period of time that never overlaps mode changes.

12.2 High Rate Science and Spacecraft Telemetry Mode changes

What happens when we change telemetry modes during a high-rate trigger?

Have we observed some problems on the bench? Seems like the mode information has problems circulating over to the HRP if there is a load of IPC traffic (from the HRS mode). Solved the issue by simply re-issuing the trigger which ends up doing two things: 1) idling the instrument, allowing traffic on IPC to clear and 2) re-commanding the BIU handler to allow it to send new mode information to HRP.

12.3 Mode Table

TLM MODE WORD	LSB of TLM Mode	MODE MNEMONIC	HRS Mode	Comments or MODE Name
xx00 0000 0600 0345 0300 0303 C300 0306 C300 0309 C300 030C C300 030F C300 0312 C300 0315 0300 3309	00	NO SCI		 PCHK IM-40 RTE-5 RTE-10 RET-20 RET-40 RTE-158 RTE-948 RTE-1896 PB&RTE-40
xx02	02	FAST 1	Yes	PC/RTIU mode
xx03 C303 0321	03	FAST 3	Yes	 PRLY
xx06 0306 0324	06	FAST 1		 S&ER-1
xx08	08	FAST 6	Yes	PC/RTIU mode
xx09 0309 0324	09	FAST 3	Yes	 S&ER-2
xx0C 030C 0324	0C	FAST 1	Yes	 S&ER-3
xx0F 030F 0324	0F	FAST 6	Yes	 S&ER-4
xx12 0312 0324 0312 0327 0B12 0324	12	FAST 1	Yes	 S&ER-5 S&ER-10 S&ER5-RPWS ???
xx15 0315 0324	15	FAST 1		 S&ER-6

TLM MODE WORD	LSB of TLM Mode	MODE MNEMONIC	HRS Mode	Comments or MODE Name
xx18 0318 0324	18	FAST 6	Yes	 S&ER-7
xx1B 031B 0324	1B	FAST1		 S&ER-8
xx1E 031E 0324	1E	NO SCI		 S&ER-9
xx21 0321 032A 0321 0330 0321 0336 0321 0339 0321 033C 0321033F	21	FAST 6	Yes	 RTE&SPB-14.22 RTE&SPB-22.12 RTE&SPB-35.55 RTE&SPB-82.95 RTE&SPB-124.5 RTE&SPB-165.9
xx24 0324 033F	24	NO SCI		 SAF-248
xx27 1B27 0342 0327 0327 0324 1327 0342 0B27 0B27 0342	27	NO SCI FAST 1 FAST 6	 Yes Yes	 SAF-248-ALT3 SAF-248-ALT2 SAF-142 SAF-248-ALT1

12.4 0312 xxxx

S&ER-5 and S&ER-10 share a telemetry mode ID and they are not the same pickup rate. V2.6 BIU handler knows of this and configures correctly.

12.5 xx27 xxxx

This mode is decoded in 3 different manners based on the MSB of the telemetry mode. 0327, 0B27, and 1B27 are handled separately.

Is this a mode that no longer appears on the spacecraft (in other words, can it be removed from the flight software)?.

12.6 xx02 xxxx and xx08 xxxx

These are modes left over from early development when we were making use of the PC-based RTIU. Telemetry modes on this S/C simulation consisted of a small number of modes with little flexibility in the selection of STM words.

These patters are relics of the past and simply provide some unused telemetry mode patterns that may be patched in the flight software, a convenient location to patch in a new telemetry mode we may encounter.

12.7 Mode Descriptions

The titles used here are taken from the BIU handler on the LRP. These are the internal names!

As of version V2.6 it appears that we do not have to support the 3.8Kb mode (i.e. 1 packet per second). This means that all science modes are HRS capable with the lowest pickup rate being 30Kb/sec. Assuming we manage to get the HRS word count correct (i.e. the number of packets/RTI) the instrument will automatically manage science data flow to CDS.

In the 30Kb/sec mode, LRP will assert the FLOW control line to HRP every other RTI to limit the flow of HRS data. In other modes (i.e. one or more packets each RTI) LRP will assert the FLOW control line when necessary to deliver LRS packets (if HRP is not producing enough data to require a BIU direct transaction each RTI period, the LRP can move LRS traffic without needing to assert FLOW).

12.7.1 NO SCI: No science telemetry

No Science Telemetry is picked up.

FLOW will be continuously asserted. Any HRS data produced will, after a timeout period, be discarded to allow memory buffers to recirculate (and keep HRP from stalling). Data buffers are statically assigned on HRP (as they are much larger then F5 queue elements) so stopping data with the FLOW line does not impact F5 buffers (so we can continue to send commands and gather status on HRP).

What happens to LRS data? (does BIU handler automagically pitch data when it sees a *NO PICKUP* mode to avoid impacting the F5 queue?)

What to MFR do?

12.7.2 FAST 1: up to 60Kb

This mode covers rates up to 1 packet per RTI. Both LRS and HRS are allowed to flow in this mode with the packet pickup throttling data delivery. Rates below 60Kb will make use of the hardware flow mechanism to limit HRS delivery rate. When an LRS packet is ready for delivery, the hardware flow mechanism disables HRS long enough to release the single LRS packet.

BIU direct packet count is set to 1 (i.e. the word count is set to 1 CDS packet).

12.7.3 FAST 3: 180 Kb

BIU direct packet count is set to 3 (i.e. the word count is set to 3 CDS packets).

LRS delivery will cause the hardware flow to suspend HRS delivery for 1 RTI period (in other words, delivering 1 LRS packet suspends all HRS delivery for an RTI period)

12.7.4 FAST 6: 360Kb

BIU direct packet count is set to 6 (i.e. the word count is set to 6 CDS packets).

LRS delivery will cause the hardware flow to suspend HRS delivery for 1 RTI period (in other words, delivering 1 LRS packet suspends all HRS delivery for an RTI period)

12.7.5 GOOFY: 3.8 Kb (no HRS) or 60Kb

This mode required inspecting the MSB of the telemetry mode as well as PB/RTE mode word to determine the bit rate. The 3.8Kb rate has HRS disabled.

12.7.6 SLO FAST: 3.8 Kb (no HRS) or 60Kb

Not used with Version V2.6.

12.7.7 BIU muted

Used during probe activities.

BIU thinks it is delivering data if CDS starts a transaction.

CDS should be handling this setup the same as the **NO SCI** setup with the collection of all data products suspended (with BIU muted, CDS will mark RPWS as *DEAD* and stop collecting data). Broadcast transactions should continue (RTI Start, Dead Time Start, and Ancillary broadcast). Ancillary data broadcast will probably contain information from the last successful pickup (i.e. immediately prior to muting the orbiters instruments BIU's).

Because of Ancillary data issues, should we always have the instrument idle prior to being muted??? Consider what is broadcast to the rest of the spacecraft if we are muted in the middle of a sounder or L/P sweep).

13 Instrument handlers, etc.

This section contains descriptions of the internal operation of the various instrument handlers and the limitations of the way the handler operate.

There are discussions of some of the support systems as well where they are of interest.

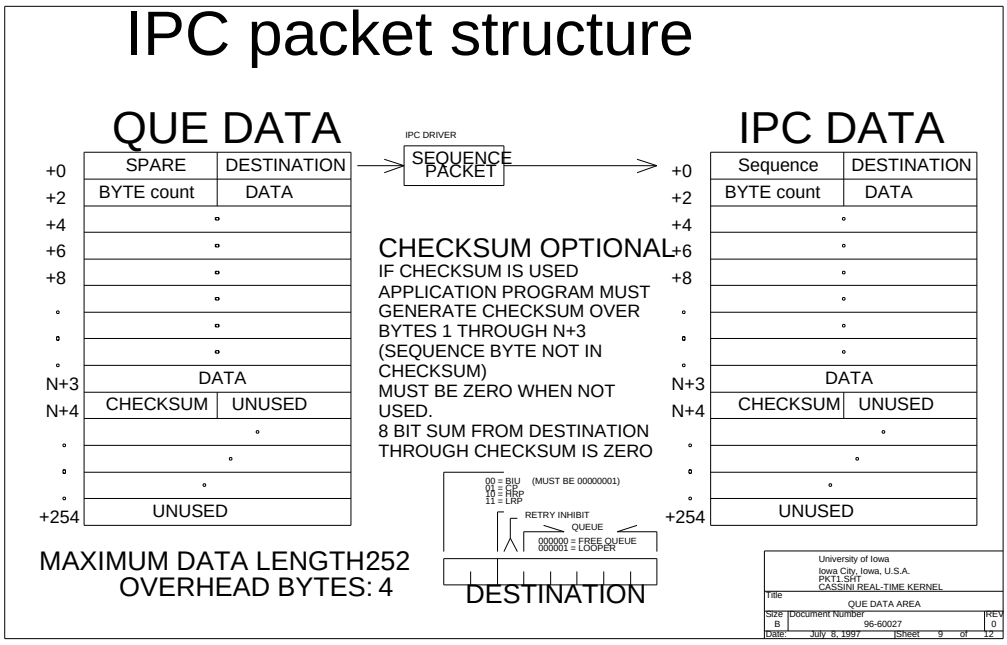
Some of the discussions in this section refer to kernel supported data structures and kernel services. Information on these items may be found in the kernel users handbook.

13.1 IPC inter processor communications

The IPC handler manages communications between the three processors.

Communications between processors is handled by the IPC driver. The IPC driver is also capable of managing communications needs within a single processor. The IPC driver handles data in 256 byte records with longer data items requiring segmentation to be handled. There is a special case for HRS traffic from HRP to LRP to support high data rates (300Kb/s).

The driver is composed of 3 processes and some associated mutual exclusion flags to control access to the communications hardware. The 3 processes are an input process, output process and watch dog process.



The data structure operated on by the driver is a queue element taken from the **F5** free space queue. Although the buffer may contain less than 256 bytes of data, the entire 256 byte buffer is always transferred in order to reduce the CPU cycles required to deliver the data.

13.1.1 Data fields within the IPC packet.

There are several fields that must be present for the IPC driver to function leaving up to 252 byte available to hold user data.

13.1.1.1 Sequence Byte

The IPC driver sequences traffic delivered to each of the other two processors. The sequence field is available for debugging but not used or checked by the IPC driver. An application process can not effectively make use of the sequence field as there is no way to guarantee that a group of packets are contiguous.

13.1.1.2 Destination Byte

This field controls routing of the packet. The upper 2 bits determine the processor that the packet should be delivered to (which may be the local processor). The bottom 5 bits selects 1 of 32 possible destination queues on each processor. And finally, there is a retry inhibit bit to allow suppression of the error recovery mechanism within the driver.

Retry inhibit is provided the support time synchronization of DCP and HRP. The occurrence of a retry can, potentially, cause a time update message to be delayed long enough to render the time update in accurate.

13.1.1.3 Byte Count

This field contains the number of useful data bytes in the block. It may be any value up to 252. This field does **not** control the number of bytes transferred over the IPC bus, but is used in calculating the checksum field.

13.1.1.4 Data

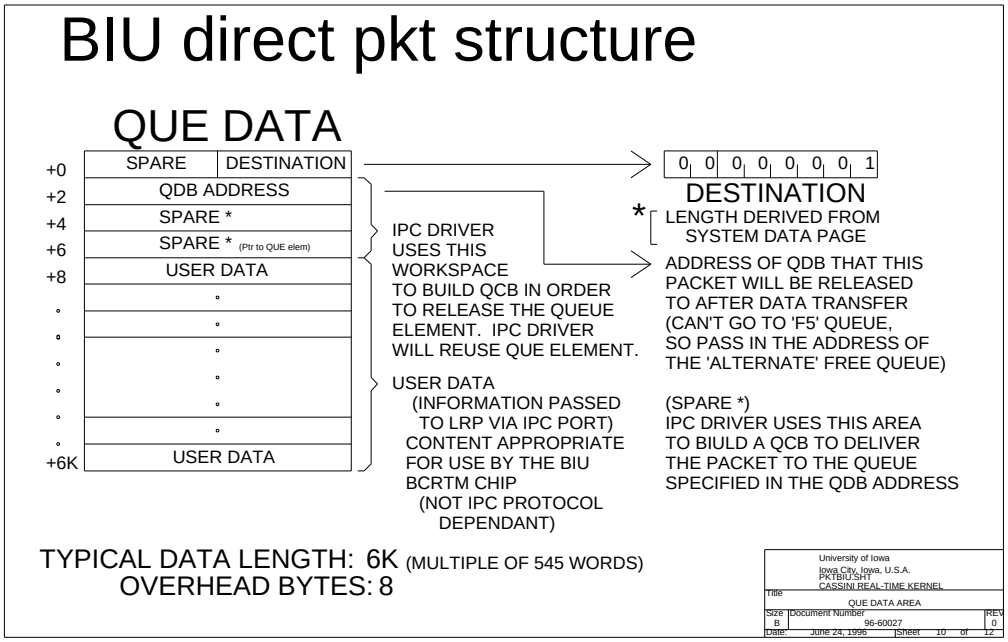
Up to 252 bytes of data.

13.1.1.5 Checksum.

Optional modulo-256 checksum of

13.1.2 BIU Direct

BIU direct is used to transfer HRS packets directly to BIU memory from the HRP. This is a special case for the IPC driver as the address and word count used to effect the transfer are taken from the system data page rather than from the **F5** queue element.



The BIU direct packet is similar to the normal IPC packet for the first 2 bytes. This allows the IPC driver transmit code to determine that this is BIU direct and notify the LRP that an attempt to transfer BIU direct traffic is about to take place. The HRP makes use of a word count stored in the system data page while the LRP uses the word count and data buffer address stored in the system data page. The word count is normally synchronized so that HRP and LRP transfer complete the transaction correctly (if this is not the case, one of the processors will recover using the help of the watch dog process).

The first 8 bytes are not transferred to the LRP but are used to deliver to the data buffer back to the appropriate queue as there are 2 different data sources that can make use of the HRS data channel through the IPC driver to the BIU. Note that the destination field, being 2 bits, selects 1 of 4 destinations, with destination zero being the special case of BIU direct.

The buffer address, stored in the system data page on the LRP, is updated during the 1553 bus dead time to provide double buffering in support of the high data rate. The IPC driver on the HRP synchronizes with the RTI interrupt that follows the dead time to keep the data transfers out of the BIU synchronized with the spacecraft.

Error recovery is not attempted when using the BIU direct destination {a retry operation would not save data when operating at full speed).

13.1.3 Minipacket Assembler

This is a code fragment that is common to all processors. The code fragment is used to break a data set into buffers small enough to be transported through the IPC handler. It is provided with minipacket status information and a separate data buffer. The codefragment then proceeds to request free space, moving status and data to the newly acquired buffer and delivering the buffer to the IPC handler. This is all done in the context of the calling process providing synchronization and multithreading (on the HRP, for example, L/P, WFR, LFDR, DUST, and WBR may all be using the minipacket assembler at the same time).

The HRP is, essentially, deadlock-free when using this system. When the minipacket assembler is blocked, waiting for free space, the IPC handler will, eventually, release free space for use by other processes. There is no high volume user of free space, other than the minipacket assembler (all the free space piles up in between minipacket assembler and the IPC driver, with the IPC driver constantly releasing free space as traffic is delivered).

The LRP is reasonably immune from free space blockage. We haven't identified any potential deadlocks in the LRP architecture at this time. MFR, having a small minipacket, does not require more than 1 free space buffer to hold a minipacket. Although MFR can block for lack of free space, it will resume when data buffers are, again, available. HFR has a dedicated set of buffers that are used to move data into the BIU handler. The remaining free space is available for command delivery and data delivery. Data delivery queues have timers that discard stale data to prevent free space buffer loss. Command delivery will drop commands in the event that data buffers are not available.

The DCP, prior to V2.3 software, had a deadlock in the data handling scheme. Data was accepted from the HRP by the IPC handler and posted to an incoming data queue for one of the compression/data analysis processes. The same process uses free space buffers to deliver data (through the IPC driver) to LRP. If incoming data used all available free space, the output activity would block waiting for free space and the DCP would be rendered deaf, crashed for all intents and purposes. Incoming IPC traffic would be accepted and discarded (so LRP/HRP would not hang or become sluggish due to IPC handler not being able to *see* the DCP).

The solution, for the DCP, is to tell the minipacket assembler that the outgoing data is not too important. Version 2.3 software has a new minipacket assembler that allows the queue read for free space to be performed conditionally. In the event that free space is not available, the minipacket assembler aborts the delivery attempt. This allows the host process to continue on and read the incoming queue and begin processing the next data buffer. As the next data request is processed, the buffers holding the data (accepted from HRP by the IPC process) are released to free space. This processing makes the lack of free space a transient condition that the DCP will recover from without intervention from the ground. Note that it is sensitive to the volume of data being moved through the DCP. As the CPU becomes saturated, the

output rate will fall below the input rate and the blocking condition will occur with the corresponding buffer discard. Also, since the BIU handler on LRP expects complete minipacket segments, the LRP will discard the beginning of the minipacket. The net result is the loss a minipacket.

13.2 BIU Handler

Interesting stuff about the BIU handler.

13.2.1 HRS sequencing

BIU handler has the capability to sequence HRS records prior to releasing them to the spacecraft. This was the standard operating procedure prior to version V2.5 of the flight software. During the summer of 2000 the spacecraft was cycling through several different telemetry modes and it was noticed that the instrument was releasing duplicate HRS records. This was a result of six HRS records being placed in BIU memory as the spacecraft transitioned to a 1 packet per RTI pickup mode. The extra 5 records were left in BIU memory as the HRS feed moved to 1 packet per RTI. Upon the next transition, where S/C again started to pick up 6 packets per RTI, the old records were collected and delivered to the ground. This caused a regression in the packet sequence numbers, somewhat upsetting parts of the ground software.

To solve this problem, the BIU handler no longer sequences HRS traffic (but the handler in the HRP has always assigned sequence numbers) and the BIU handler clears the length field of an HRS packet to zero following collection. With this change, when the spacecraft transitions between telemetry modes (in particular, when changing pickup rates) the lost packets are discarded and extra packets that are collected are zero-length (and discarded).

13.2.2 Alterations to the *BIUH* process.

The BIU handler normally would be used to sequence HRS packets. This method should cause all traffic presented to CDS to be correctly sequenced, even if data is lost between HRP and LRP. The sequencing activity on the LRP may be suppressed using the following **MEM_TWEAK** in order to allow the sequence number generated by the HRP to remain intact. Packets lost within RPWS would, in this case, be a little more evident.

Suppress sequencing of HRS telemetry

```
73MEM_TWEAK, LRP, BYTE,  0x33, 0xFF, BIUH
```

Resume sequencing of HRS telemetry

```
73MEM_TWEAK, LRP, BYTE,  0x33, 0x00, BIUH
```

13.2.3 BIU accommodations

The BIU appears to have some deficiencies that require software accommodation.

13.2.3.1 Write Cycles and BCRTM Registers

In order to accommodate 8 bit processors, the BIU design has an 8 bit holding register that is used to build up a 16 bit transaction when writing to BCRTM registers. The host processor (the 8085) writes the lower 8 bits of the register image to the holding register by writing to the LSB address of the register followed by writing the upper 8 bits to the MSB of the register (causing all 16 bits to be transferred to the BCRTM register).

The holding register is only partially decoded causing it to accept any data written to an even address in BIU address space (i.e. It is sensitive to both register and memory accesses). This implies that any register accesses must be handled as atomic operations (and not be interrupted by writes to BIU memory). In most cases this is not an issue as the BCRTM registers that are accessed on a periodic basis are accessed from within an interrupt routine (i.e. From within the non-maskable interrupt service routine with interrupts disabled, so the 8085 cannot be interrupted). Due to the partial decoding of the holding register in the BIU it is possible for the 8237 to perform a memory write cycle to BIU memory when the 8085 is writing to the BCRTM registers causing register corruption.

In the version 2 implementation of the BIU handler, only R6 and R2 are accessed on a periodic basis, all other registers are initialized and remain un-touched when the 8237 is active. R2, the descriptor table base address register, seems to be impervious to corruption as it is required to be aligned on a 512 word boundary (making the lower 9 bits of the register don't-care, must-be-zero). R6, the interrupt log list pointer register, is sensitive to corruption. Since the upper 8 bits of the register are written without problem, the interrupt log list can appear anywhere within the 256 byte page selected by the upper 8 bits of the write when the write to R6 is corrupted. This version of the BIU handler places the interrupt log list in page 1, the bottom 64 words of which are write protected.

The Version 2.5 BIU handler avoids this problem by disabling the 8237 during R2/R6 update. This is accomplished by reading the command register, setting the disable bit, and rewriting the command register. This is only possible when using the Harris 8237 (Intel/OKI chips do not provide access to the command register). This results in a 40 microsecond period where the 8237 will not transfer data from either IPC or HFR.

13.3 TWEAK/MRO Handler

This process makes use of common code on all 3 processors to reduce the requirements for ALF records.

Using the MRO commands blocks the handler for the duration of the memory dump. Dumps to science telemetry are limited to 1000 bits per second and dumps to housekeeping telemetry are limited to 10 bits per second.

No command is provided to control memory bank selection, one must **73MEM_TWEAK** the process descriptor.

13.3.1 Alterations to the *TWEK* process.

Some alterations may be made to the TWEK process to alter data rates and packet sizes. The housekeeeping stream typically uses a 10 byte dump to keep the micro packet within a single 16 byte dump line (on GSE the display). Dumps to telemetry have a default size of 128 with software version prior to V2.3 while later versions have a default size of 192. The size of the science telemetry MRO record can be easily tweaked when using software V2.4 and later.

In addition the data rate out of the TWEK process is limited and may be changed if required.

In the following examples, the first TWEAK maintains the 10 bit/second data rate through the housekeeping telemetry (this is the RTI delay between successive deliveries to the housekeeping manager). The second TWEAK alters the number of data bytes in each packet delivered.

10 Byte Housekeeping dump, 16 byte packet

```
73MEM_TWEAK, *, BYTE,  0x60, 64, TWEK
73MEM_TWEAK, *, BYTE,  0x62, 10, TWEK
```

26 Byte Housekeeping dump, 32 byte packet

```
73MEM_TWEAK, *, BYTE,  0x60, 128, TWEK
73MEM_TWEAK, *, BYTE,  0x62, 26,  TWEK
```

58 Byte Housekeeping dump, 64 byte packet

```
73MEM_TWEAK, *, BYTE,  0x60, 256,  TWEK
73MEM_TWEAK, *, BYTE,  0x62, 58, TWEK
```

122 Byte Housekeeping dump, 128 byte packet

```
73MEM_TWEAK, *, BYTE,  0x60, 512, TWEK
73MEM_TWEAK, *, BYTE,  0x62, 122, TWEK
```

It is possible, although very version specific, to alter the bit rate for data delivered to the science telemetry stream. This alteration will alter instructions and it's location will change from version to version. The value n, is set to 0009 in the loaded software indicating that 9 RTI periods should elapse between each MRO packet delivered to LRP. Changing this value would allow a bit rate of up to approximately 9,000 bits/second.

Version 2.4 flight software makes use of *TWEK V60* with these values moved to fixed locations.

V2.3 tweak:

73MEM_TWEAK, *, BYTE, 0x01D3, n, TWEK

V2.4 tweak:

73MEM_TWEAK, *, BYTE, 0x0064, n, TWEK

Also in *TWEK V60* (flight version V2.4), the size of the MRO record sent to the science telemetry stream may be altered.

73MEM_TWEAK, LRP, BYTE, 0x0066, 128, TWEK

And, of course, change back to the nominal 192 bye record when done.

73MEM_TWEAK, LRP, BYTE, 0x0066, 192, TWEK

13.4 Time Synchronization Handler

This handler is responsible for updating the system time field on the DCP and HRP. This handler operates under the assumption the, for the most part, S/C is delivering accurate SCLK data (i.e. no time jumps). Every 256 seconds in RTI-0, when the lower 8 bits of SCLK are zero, the RTI0 signal is asserted for 1 RTI to clear the lower bits of SCLK (and zero the RTI) on DCP and HRP. The time lock process sends a copy of the upper 24 bits of the time to DCP/HRP completing the time synchronization process. With the 256 seconds cycle, the timing requirements are considerably relaxed (the upper 24 bits of the time are stable for the entire 256 second period, so a time update does not need to be delivered within a second).

As long as the time update is byte aligned, we can make use of the TWEK process to manipulate the upper bits of SCLK. Since we make use of a common kernel image, the data page is located at the same address on each processor, easing address calculations.

13.4.1 Alterations to the *LOCK* process.

One alterations may be made to the LOCK process to suspend time updates. This may be required when processing long memory dumps.

Suspend time update activities

73MEM_TWEAK, LRP, BYTE, 0x60, 0x00, LOCK

Resume time update activities

73MEM_TWEAK, LRP, BYTE, 0x60, 0xFF, LOCK

Enable new Watch Dog Timer capability.

73MEM_TWEAK, LRP, BYTE, 0x6E, 0xFF, LOCK

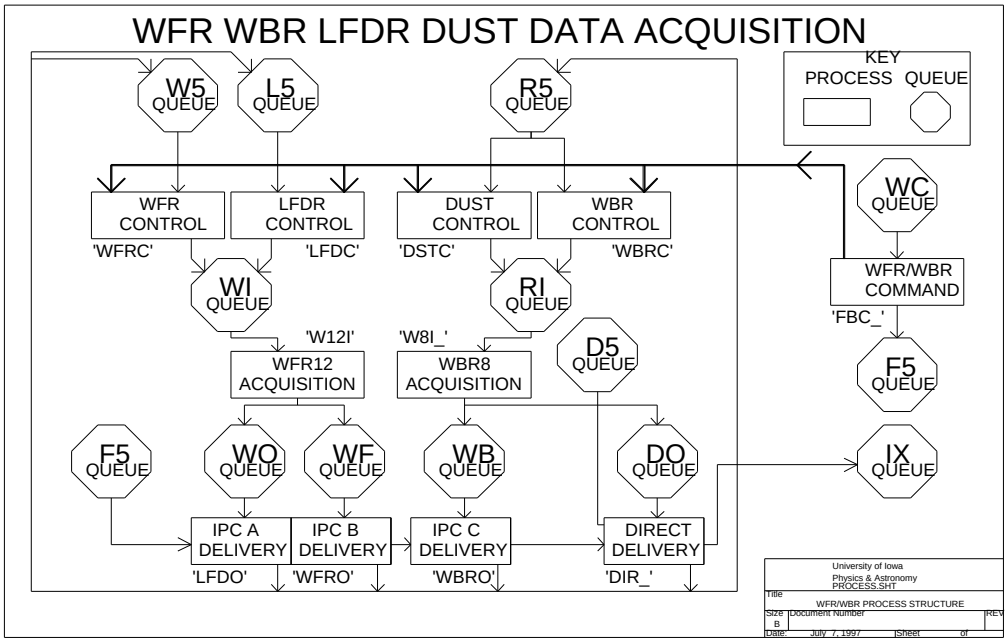
Disable new Watch Dog Timer capability (i.e. resume operating in the 'old' fashion).

73MEM_TWEAK, LRP, BYTE, 0x6E, 0x00, LOCK

This location contains the constant that represents the maximum number of RTI periods that may elapse between looper packets (i.e. time updates). This constant would normally be twice the time between updates and is set to 10 minutes in the V2.4 flight code. When looper packets fail to appear on the LRP (if DCP or HRP software crashes) for this length of time, the watchdog timer hardware in no longer reset (0x6E must be set non-zero).

73MEM_TWEAK, LRP, BYTE, 0x62, nn, LOCK

13.5 WBR, WFR, LFDR, and DUST



The wideband/waveform system makes use of much common hardware so the handler for these two physical instrument is combined. The LFDR and DUST instruments are software derivatives of the WFR and DUST respectively.

13.5.1 Commanding

The command decoding is contained within a single process for all four logical instruments. The command process is blocked only when waiting for incoming commands and should be capable of processing commands at any reasonable rate. The command decoding is broken into 2 levels with the first level responsible for decoding only the 4 bit RPWS destination field. The second level decode is responsible for decoding the individual instrument commands (i.e. the 3 bit instrument command field) and extracting the appropriate control bits (antenna/gain select, etc.).

The command decoder is the only process within the handler that is detected and started a boot time. It then proceeds to create all queues and additional processes.

13.5.2 Alterations to the process descriptors.

Some of the commanding for this set of receivers is accomplished by altering specific memory locations within each process. These memory locations are located in a fixed area that is adjacent to the process descriptor so the locations are, effectively, fixed.

13.5.2.1 Timing Process WFRC WBRC LFDC DSTC

13.5.3 Queues

The handler creates one queue to hold incoming commands, 3 queues to manage DMA buffers, 2 queues to pass *internal acquisition commands*, one queue to manage HRS formatting, and 4 queues to manage data delivery.

The incoming command queue is used by the IPC driver to deliver commands to the 4 logical instruments. The IPC driver is used to combine the commands for the 4 logical instruments into a single queue for command processing. A single command decode process is used to process the incoming commands. The command process waits for a command to arrive, decodes the command and updates the instrument command state table for the specified instrument (although the command are combined into a single queue, the routing field is still present in the command and is used to determine which of the 4 logical instrument is being specified). Since the only action that results from the arrival of a command is updating the instrument state table, the command process cannot block for any reason other than waiting for the next command to arrive, in other words, the command process will not cause the free space queue to become exhausted by failing to process incoming commands.

The DMA buffers are statically allocated to the WFR, LFDR, and WBR/DUST. WBR and LFDR make use of separate buffers to allow formatting and delivery operations to be overlapped with data acquisition. WFR buffers may be in use for long periods when large acquisition buffers are specified and passed to either LRP or DCP.

WBR and DUST do not throttle delivery of their data and share a common buffer pool. Since the output process does not normally throttle the data to provide buffering on the HRP (delays in the WBR/DUST Low Rate Science delivery are present to allow LRP sufficient time for processing), the DUST and WBR can share buffers with minimal impact.

The internal acquisition commands are embedded within the DMA buffer QUE's. These queue elements are passed from the four timing control processes to the two data acquisition processes. WBR and DUST make use of the 8 bit A/D system and therefor share a common acquisition process. Likewise, the LFDR and WFR make use of the 12 bit A/D system and share the other acquisition process. Keeping in mind that the architecture of the operating system is inherently single threaded, coordination of WBR and DUST (as well as LFDR and WFR) is essentially automatic. Although two acquisition commands may be waiting in the input queue for the acquisition process, the can only be processed one at a time. Several hardware conflicts may arise that are handled using the Mutual Exclusion facilities of the operating system (L/P shares a DMA channel with WBR, the compression hardware may be used by WBR and WFR, block moves require both WBR and WFR DMA channels, etc.)

The data delivery queues are used to connect the acquisition processes to the data delivery processes. Two data delivery processes are used to handle WFR and LFDR data to prevent blocking of data. A single data delivery process is used for the WBR and DUST Low Rate Science as the problem of blocking delivery is insignificant. The final data delivery queue is used to route WBR data to *High Rate Science.*

HRS formatting must be handled by a completely separate process as this involves building CCSDS headers rather than IPC headers. In order to meet the desired 300,000+ bit/second data rate there are two buffers dedicated to holding the formatted CCSDS packets. These 2 buffers are described by QCB's that are typically held in a special HRS free queue. The direct delivery process then accepts data from the acquisition process, requests a block of memory to reformat the data into, reformats the data and passes the formatted data to the IPC driver and releases the DMA buffers back to the appropriate free queue.

Low Rate Science data is handled by the three data formatting processes. Each is essentially an identical process that has slightly different timing parameters used to delay an appropriate period of time. This process reformats the data into mini-packet format for delivery through the IPC mechanism to either the LRP or DCP as needed.

Also used by the handlers, but not created is the standard free queue. Incoming commands and LRS delivery require the use of free queue. These buffers are recirculated into the free queue as they are used.

13.5.4 Processes

The WBR/WFR/DUST/LFDR handler is made up of 11 processes.

13.5.4.1 **FBC_** Command and Control

Master process as mentioned above. This process waits on the command queue **WC** and decodes the incoming commands. The command decode involves altering a control block for one of the four logical instruments that contains the hardware state of the logical instrument (antenna, gain, band, etc.). As each command is decoded a few bits in the hardware state table are updated and the command buffer is released to the **F5** queue.

There is also a timeout processor that is located in the command decoder that runs in the context of the IPC watch dog timer. This allows the time-out handler to use CPU cycles each RTI period without having to use an additional process or having to effect a polling loop in the command decoder (i.e. this reduces the CPU cycles required to implement polling).

13.5.4.2 **LFDC** LFDR Timing Control
13.5.4.3 **WFRC** WFR Timing Control
13.5.4.4 **WBRC** WBR Timing Control

Timing control process for the four logical instruments. These processes implement the data acquisition cycle timing for the four instruments. Each of the processes are logically identical although they differ in implementation as a pair is associated with each of the physical instruments.

The timing involves using the hardware state tables (i.e. the delay and offset timing values) to calculate the time of the next data acquisition. The process the releases the CPU by using the kernel delay service. When the requested number of RTI periods has elapsed, the timing process is placed on the computable list and receives the CPU when it becomes available. The timing process then requests a DMA buffer from the appropriate free queue (the diagram above should help in visualizing this activity) and builds the hardware state in the queue element. The DMA buffer is then delivered to the data acquisition process by writing it to the appropriate queue. Note that the timing may be skewed by lack of CPU, assuming that a higher priority processes is using the CPU, or by lack of a DMA buffer if resources are tight (typically due to high data rates).

13.5.4.5 **DSTC** DUST Timing Control

DUST timing control is built as part of the WFRC modules (i.e. using common source code), but is handled slightly different. In order to provide some MOD based timing control, the DUST timing control process uses a synchronous polling scheme. Just as the other three logical receivers use a MOD timing function to collect data, DUST uses a MOD timing function to poll for data acquisition commands. The scheduling controls when DUST will ultimately gather data, although DCP will only set the collection flag when it is ready for data.

13.5.4.6 **W8I_** WBR/DUST data acquisition
13.5.4.7 **W12I** WFR/LFDR data acquisition

The data acquisition process waits an *internal acquisition command* to arrive and then attempts to perform the specified operation. As you will notice in the diagram above each of the acquisition processes has a single input queue that is fed by the two logical instrument timing control processes. The incoming queue effectively serializes the acquisition requests so that only one hardware operation may occur at any given instant.

Once an *internal acquisition command* arrives, the acquisition process request exclusive access to the required hardware (WBR shares the 8237 with L/P, the compression hardware is shared between 8 bit and 12 bit systems) using a kernel service. Once exclusive access has been granted, the hardware state table in the *internal acquisition command* is moved to the hardware and a data acquisition begins.

As expected, the acquisition process releases the CPU using a kernel service to await the completion of the data acquisition. The operation may complete normally or be terminated by a timer in the control process, in either case the data acquisition process regains control of the CPU, notes the abnormal termination, if it occurs, and passes the data buffer on to data formatting using information stored in the hardware state table.

The acquisition process then returns to the top of it's control loop to begin the next operation.

(V2.3) Resource management is performed within these acquisition routines to avoid data acquisition cycles that are invalid (i.e. interfere with each-other). W12I will ask for the Mx on the WBR DMA channel whenever attempting a data acquisition when either of the 8 bit timing processes indicate that high band data may be acquired.

V2.5 adds an additional interlock in **W12I** in an attempt to prevent WBR/WFR interference when both systems are operating in high-band. **W12I** checks to see if either WBR or DUST are set to high band, if so an additional check of the current 12 bit acquisition is performed, and if high-band is indicated, **W12I** will ask for the WBR Mx to preclude both systems form operating at the same time. Due to a mis-typed variable name, however, it doesn't work as easily as planned (an additional mem_tweak is required to **WBRC** to make this code snippet work).

Note that when looking a timing diagrams of data taken when using this feature, the WBR data set may initially appear to occur during the last few WFR samples when, in actuality, the WBR occurs following the WFR acquisition, but later in the same RTI (you have to have been there for this statement to make sense...).

13.5.4.8 **LFDX** LFDR Data Delivery
13.5.4.9 **WFRX** WFR Data Delivery
13.5.4.10 **WBRX** WBR Data Delivery

These processes were renamed in Version 2.3 as some tweak-able locations are now available. The name change provides command compatibility with previous software loads as any MEM_TWEAK's applied using the new names will be ignored by any older software.

The acquired data must be formatted and delivered to the DCP or LRP. The DCP is used to perform data analysis or compression on some of the data products before being passed on to the LRP for telemetering to the ground. These three data delivery processes are build from a single source and differ only in the timing used to provide some buffering on the HRP for the data products.

The formatting involves copying the status from the queue element and the data from the DMA buffer to free space buffers and passing the free space buffers on the IPC driver for delivery to the other processors.

Again, these processes can be blocked when free space (i.e. F5 buffers) becomes unavailable.

Once the acquired data has been reformatted and passed on to the IPC driver, the DMA buffer is released back to the appropriate free queue (again, information is stored in the queue element that describes the DMA buffer) .

As expected, the cycle is repeated endlessly wit the process being blocked by lack of CPU, blocked waiting for free space, or simply being blocked waiting for the next data acquisition to occur.

LFDR/WFR notes

Routing multiple data sources through a single delivery process, specifically LFDR and WFR routing to compression on DCP, should work as long as AGG is off for LFDR. If AGC is enabled on LFDR gain, the LFDR gain changes will be applied to the WFR. Timing on the delivery process may require adjusting to obtain smooth data flow without loss of data on the DCP.

Routing LFDR and WFR through the same delivery process to different queues on DCP (i.e. keeping WFR looking like WFR and LFDR looking like LFDR) will not gain any useful results as there is no issue with mixing LFDR and WFR data in the telemetry stream.

What we want is a way to deliver the dual-routed LFDR data to the DCP for compression while providing data to the LFDR process on the DCP. Can't be done sorry for now (changes to LFDX might accomplish this, but as of V2.6 forget it, we keep trying and stumbling into the AGC issue).

13.5.4.11 **DIR_** High Rate Science formatting

This is logically similar to the other Data Delivery activities, but involves making use of a specially formatted IPC buffer referred to as a **BIU Direct Packet**. This transfer mechanism is optimized to support the higher data rates encountered in High Rate Science modes.

As with the LRS data delivery mechanism, this formatting step takes the status and data and builds minipackets within CDS packets that are ready to be placed in BIU memory for delivery to the ground.

As with the LRS data delivery, incoming data arrives in one input queue and empty CDS buffers arrive in a second. Following the formatting step the data buffers are returned to the queue used by timing control and the CDS buffers are delivered to the IPC handler.

Several hardware assists may be employed by this handler to increase the rate with which data is formatted. Since the 8085 has no intrinsic block move instruction, the 8237 may be employed to move blocks of data from the acquisition buffer to the direct data buffer. The 8237 must, of course, be shared with other activities so there are several sharing schemes that may be employed. In addition, operating the LFDR tends to use one channel of the 8237 for long periods of time (preventing the 8237 from being used for block moves). There is an assist mechanism implemented in the Actel chip to generate an interrupt following each LFDR sample to allow the processor to make use of the 8237 to perform block moves and restore the LFDR channel.

13.5.5 Automatic Gain Control

Both WBR and WFR are provided with a form of automatic gain control. The WBR has a simple hardware assist while the WFR provides a mechanism for a downstream AGC process to adjust the gain.

Both mechanism have shortcomings that will be discussed here.

13.5.5.1 WBR Automatic Gain Control

The WBR receiver contains a peak detector that integrates the peak signal level as seen by the WBR. The peak detector is located close to the output of the WBR so it is detecting the processed waveform (i.e. the amplified and filtered waveform). The integrator has a time constant of ???.

The WBR has a 2 channel multiplexer prior to the A/D converter to allow either the waveform data or the AGC data to be digitized. It is, therefore, impossible to sample the AGC channel while actively collecting data. In addition, the control hardware for the WBR does not provide a means to perform slow sampling of the AGC data so it must be manually sampled, effectively limiting the sample rate to the RTI interval of 8 Hz or slower.

- Version 2.2 Implementation

Following every data capture, a single value from the AGC channel is read and included with the data set. In order to eliminate problems encountered during high speed modes (i.e. 1 data set per RTI or faster), every other AGC sample is ignored resulting in a nominal AGC update rate that is half of the data set acquisition rate.

The pipeline effect is a result of the double buffering used to attain the 300,000+ bit/second modes (it shows up whenever WBR is running at or above 1 data set per RTI). The gain update is posted to the WBR scheduling process rather than to the data acquisition process to allow implementation of the dust detection scheme (also necessary to implement a toggle mode on the WBR). Since there are 2 acquisition buffers available fast scheduling typically results in both buffers being posted to the acquisition process at all times. An AGC update does not take effect until the current buffer is posted back to the scheduling process. Gain updates are processed by the scheduling process and have no effect on buffers already posted to the acquisition process.

- Version 2.3 Implementation

The RTI count between gain updates may now be altered. This change was simply a matter of moving the delay count to an accessible location. When operating the WBR at lower data rates, the gain may be updated with every capture without problems.

13.5.5.2 WFR Automatic Gain Control

The WFR has no hardware assists to aid in making decisions about signal strength. As a result, the data must be examined by on-board software in order to determine when gain changes are required. AGC for this receives is, therefore, strictly a software implementation. Resources, such as memory (for code and buffers) and available CPU cycles drive architecture.

As a further complication, channels 3, 4, and 5 share common gain control bits while channels 1 and 2 each have a dedicated set of gain control bits.

- Version 2.2 Implementation

Resources on the HRP are in short supply so the gain control analysis is performed on the DCP as part of the compression activity. Bypassing the compression step will result in bypassing the AGC analysis.

When sufficient time is available on the DCP, the WFR buffers are examined by the AGC code and a decision is made as to needed gain changes. A single step gain change is implemented by sending an internal gain command back to the HRP. Although commands are available to change gain up or down by a single step, the V2.2 implementation extracts the gain level from the data set and builds an appropriate gain command with a new level specified.

13.6 MFR Handler

Antenna switching shows up in the WBR/WFR data.

Fixed schedule (MOD(SCLK,32)=0)

- Version 2.2 Implementation

The antenna switching is not synchronized with the spacecraft clock bit 6. The first antenna selected in the **73MFR_CNTL** command. Depending on when power is applied to the MFR analog electronics the first sweep may occur on the even or the odd period. This makes attempts to share Ex antenna with HFR Direction Finding rather difficult.

- Version 2.3 Implementation

The antenna switching is now synchronized with the spacecraft clock bit 6. This change in the synchronization scheme allows the antenna selection to be predicted. This change allows the HFR activities to be scheduled such that they do not interfere with the MFR.

13.7 HFR Handler

Don't forget about the register in the actel chip used to control the time between data acknowledge pulses to the HFR.

Sounder notification to both ancillary sub-address in BIU and to HRP.

Dedicated memory area for *scientific programs.* Similar to IEB but dedicated to HFR activities.

Dedicated buffer memory for data acquisition and free space to deliver data to BIU handler.

13.8 Langmuir Probe Handler

Relay issues. Latching relay may need to be switched with each sweep.

LP notification to ancillary address in BIU.

Makes use of DCP for compressing data.

Ancillary data presented through the BIU may limit or constrain the usefulness of synchronizing L/P sweeps with the WFR/LFDR. WFR and LFDR both *notice* when L/P performs a sweep. This couples between the L/P sphere and the electric antenna. One suggestion is to acquire the Mx for the WFR DMA channel, preventing the L/P sweep from occurring during WFR activities (WBR is already blocked due to shared resources).

Using the WFR Mx will, when a conflict occurs, will cause the pending activity information in the ancillary data, to be displayed for several seconds. This extended ancillary status may cause other instruments on the S/C to hold off observations for an unacceptable period of time.

14 Data Formats

This seems like an easy place to keep details of the data products produced by the instrument and processed by the Iowa GSE software. This chapter covers the level-0 data (raw telemetry data in the form of CDS records and reformatted data in the form of RPWS minipackets) as well as Archive data.

Some details of the dataset organization are also kept here.

The Cassini dataset is maintained on the *Space Physics Cluster* at Iowa. The files are segregated into directories according to various mission phases. The base location for mission data is */opt/project/cassini/data* with directories below this level, such as */opt/project/cassini/data/deploy*, */opt/project/cassini/data/venus_1*, */opt/project/cassini/data/ico_m14*, */opt/project/cassini/data/venus_2*, and */opt/project/cassini/data/earth_1* holding data recovered up to Earth encounter.

Subsequent cruise periods should appear in directories such as */opt/project/cassini/data/c16* where the *c16* would refer to cruise sequence 16 data.

14.1Mini packets

General details of the minipacket structure is listed with the commands in the previous section. Additional details are provided here.

Typically the entire dataset is available in the files whose name ends in ***u00***. These files have all data produced by the instrument. Compressed data is decompressed and the data sets are complete, assuming the complete data set reached the ground. WFR and L/P data sets consist of separate groups for each antenna (or sweep/density in the case of the L/P).

In addition, the L/P and HFR data may be segregated into individual files for convenience. Both of the segregated datasets would normally include STIM packets to aid in identifying mode changes.

14.2 CDS packet

The raw CDS records are located in files whose name ends in ***r00***. These files contain the science telemetry stream as well as the housekeeping telemetry stream. The housekeeping data in not propagated beyond this point.

Engineering plots require access to this data.

14.3 SFDU items

SFDU information is not present in the data sets at this time.

14.4 CHDO items

CHDO records of the 92 and 94 are copied into the ***r00*** and ***u00*** files.

14.5 Quaternions

Attitude information.

Queried and kept somewhere convenient.

14.6 RPWS/GSE records

All of the data products use by the GSE software make use of a common record structure that consists of header data followed by instrument data. The header data is common to all record types while the instrument data appears in several formats.

0	**f_length**	record length -4 (number of bytes remaining in the record)
4	**record_type**	record type flag (table follows)
8	**status**	
12	**cds_time_tag_a**	CDS pkt SCLK of the beginning data bit
16		
20	**cds_time_tag_b**	CDS pkt SCLK of the ending data bit (empty for CDS records)
24		
28	**ws_time_tag_a**	BCD time tag from workstation when data first processed
32		
36	**ws_time_tag_a**	BCD time tag from workstation when data last processed
40		
44	**compress_method**	
48	**compress_bit_count**	
52	**compress_result**	
56	**stream_info**	
60	**length_data_start**	length of compressed/raw data packet
64	**length_data_length**	length of uncompressed data packet (may exceed 4K)
68	**segment_count**	
72	**segment_number**	
76	**record_count**	
80	**record_number**	

84	**epoch**	UNIX Epoch. Add this value to SCLK to arrive at UNIX time.
88	**sequence**	

92	**chdo_type_92**	Compressed Header Data Object
		This CHDO contains status information, ERT and RCT
176	**chdo_type_94**	Compressed Header Data Object
		This CHDO contains status information, SCET and SCLK
208	**filler 0**	spare
212	**filler 1**	spare
216	**filler 2**	spare
220	**filler 3**	spare
224	**filler 4**	spare
228	**biust**	BIU status word, when available
232		
236	**f_sclk**	SCLK from CHDO header, when available
240		Spacecraft Clock
244	**f_scet**	SCET from CHDO header, when available
248		Spacecraft event time for the associated SCLK
252	**f_ert**	ERT from CHDO header, when available
256		Earth Receive Time for the associated data record
260	**f_rct**	RCT from CHDO header, when available
264		Record Creation Time for the associated data record

268	**instrument data**	CDS or Minipacket data

f_length	**r_length**	Total record length - 4 (allows working backwards through file)

14.6.1 f length

32bit record length, big-endian.

For a record that contains 1000 bytes, this field will contain 996. This is the number of bytes remaining in the record. To read a variable length record, read the first 32 bits (4 bytes) then use these 32 bits to determine the number of bytes remaining in the data record.

14.6.2 record type

bits 0-7 (1st. byte in the field) indicate minipacket type.
Bits 8-15 indicate data type. (CDS, segmented mini packet, unsegmented minipacket)

The header file, rtiu.h, contains definitions for the data record types that follow.

The first group of definitions cover bits 0-15 (i.e. define data trype and minipacket type)

DATA_MP_packet	packetized products
DATA_MPP_MFR	MFR packet
DATA_MPP_HFR	HFR packet
DATA_MPP_WFR	WFR packet
DATA_MPP_WBR	WBR packet
DATA_MPP_DUST	DUST packet
DATA_MPP_MRO	MRO packet
DATA_MPP_LFDR	WBR packet
DATA_MPP_LP	LP packet
DATA_MPP_HK	Housekeeping packet
DATA_MPP_STIM	STIM packets
DATA_MP_segment	segmented data products
DATA_MPS_HFR	HFR SEGMENT (1K segment)
DATA_MPS_WFR	WFR SEGMENT (1K segment)
DATA_MPS_LFDR	WFR SEGMENT (1K segment)
DATA_MPS_WBR	WBR SEGMENT (1K segment)
DATA_MPS_DUST	WBR SEGMENT (1K segment)
DATA_MPS_MRO	MRO packet
DATA_MPS_LP	LP SEGMENT (1K segment)
DATA_MP_large_segment	segmented data products
DATA_MPL_HFR	HFR SEGMENT (4K segment)
DATA_MPL_WFR	WBR SEGMENT (4K segment)
DATA_MPL_LFDR	WBR SEGMENT (4K segment)
DATA_MPL_WBR	WBR SEGMENT (4K segment)
DATA_MPL_DUST	WBR SEGMENT (4K segment)
DATA_MPL_MRO	MRO packet
DATA_MPL_LP	LP SEGMENT (4K segment)

DATA_MP_complete_segment complete data products

DATA_MPC_HFR	HFR SEGMENT (64K segment)
DATA_MPC_WFR	WFR SEGMENT (64K segment)
DATA_MPC_LFDR	LFDR SEGMENT (64K segment)
DATA_MPC_WBR	WBR SEGMENT (64K segment)
DATA_MPC_DUST	DUST SEGMENT (64K segment)
DATA_MPC_MRO	MRO packet
DATA_MPC_LP	LP SEGMENT (64K segment)

The bottom 8 bits have the following meanings

PACKET_TYPE_invalid
PACKET_TYPE_stim
PACKET_TYPE_mfr
PACKET_TYPE_hfr
PACKET_TYPE_lp
PACKET_TYPE_lfdr
PACKET_TYPE_wfr
PACKET_TYPE_dust
PACKET_TYPE_mro
PACKET_TYPE_wbr
PACKET_TYPE_fill

These indicate CDS records.

DATA_telemetry
DATA_RTIU_telem
DATA_RTIU_hsk

14.6.3 status

14.6.4 cds time tag a

32 bits of SCLK.

This time is obtained from the first CDS record that contains the data. If this is raw data, this will simply be the SCLK extracted from the CDS record. In the case of minipacket data, this is the CDS time tag obtained from the first CDS record containing the minipacket.

14.6.5 cds time tag b

32 bits of SCLK.

This time is obtained from the last CDS record that contains the data. If this is raw data, this will be zero. In the case of minipacket data, this is the CDS time tag obtained from the last CDS record containing the minipacket. When segmented data sets are reconstructed, this time tag should be from the last segment.

14.6.6 ws time tag a

BCD time.

This time tag is placed in the record as it is first created. This is intended to give a clue as to when the data is obtained. This time has nothing to do with when the data was collected on the spacecraft. There is little point in displaying this time tag, it is intended as a diagnostic aid for GSE software.

14.6.7 ws time tag b

BCD time.

This tag is updated whenever the data product is modified or reformartted.

14.6.8 compress method

This field stores the method used to compress data on the spacecraft. It is loaded by the decompression step. Expect this field to be zero for data that has never been compressed of has not been decompressed.

14.6.9 compress bit count

14.6.10 compress result

14.6.11 stream info

14.6.12 length data start

Data length prior to decompression.

14.6.13 length data length

Data length following decompression.

14.6.14 segment count

14.6.15 segment number

14.6.16 epoch

This field contains the difference between UNIX epoch and Spacecraft epoch. This field may be added to SCLK to derive the UNIX clock. This derived time is used by UNIX time functions (i.e. to convert SCLK to text that may be used on a plot)

The calculation used to arrive at this number is only accurate to seconds. Milliseconds are not taken into account. This typically causes this field to jitter by 1 count.

14.6.17 chdo type 92

0	**chdo type**	92
2	**chdo length**	
4	**originator**	
5	**last modifier**	
6	**scft id**	
7	**data source**	
8	**status flags**	
10	**ert**	earth receive time
16	**rs codeword status**	
17	**frame extract count**	
18	**dsn record sequence**	
22	**bet**	
23	**fly**	
24	**decode status**	
25	**decode method**	
26	**sync flags**	
27	**pn errors**	
28	**virtual channel id**	
29	**virtual frame count**	
30	**frame header err flg**	
31	**rs decode errors**	
32	**spare**	
33	**frequency band**	
34	**bit rate**	
38	**spare**	

40	**snt**	
44	**ssnr**	
48	**signal level**	
52	**antennas**	
53	**receivers**	
54	**master antenna**	
55	**master receiver**	
56	**dtm group**	
57	**tlm channel**	
58	**lock status**	
60	**version**	
61	**build**	
62	**orig source**	
63	**curr source**	
64	**rct**	record creation time
70	**anomaly flags**	
72	**lock count**	
74	**lrn**	
76	**pub**	
82	**block v receivers**	

Event time structure used for RCT, ERT, and SCET

0	**days**	days from Epoch
2	**milliseconds of day**	32 bit mSec
4		0..86,400,000

Event clock structure used for RCT, ERT, and SCET

0	**seconds**	seconds from Epoch, 32 bits
2		
4	**milliseconds**	milliseconds, 0..999

14.6.18 chdo type 94

0	**chdo type**	94
2	**chdo length**	
4	**cds error flags**	
6	**src pkt seq count**	
8	**non fill length**	
10	**packet apid**	
12	**super pkt seq count**	
14	**cas sclk**	spacecraft clock
20	**scet**	spacecraft event time
26	**probe rs errors**	
27	**scet flags**	
28	**spare**	

14.6.19 biust

BIU status word collected from the instrument on a regular basis. Currently this field has useful contents only when running on the bench (i.e. with the EM).

(Don't currently know which CHDO type contains this information)

14.6.20 f sclk

SCLK extracted from CHDO record.

Event time structure used for f_scet, f_ert, f_rct

0	**spare**	
2	**days**	days from Epoch (16 bit number)
4	**milliseconds of day**	millisecond of day
6		0..86,400,000 (32 bit number)

Event clock structure used for f_sclk

0	**seconds**	seconds from Epoch (32 bit number)
2		
4	**milliseconds**	millisecond, 0..999 (16 bit number)
6	**spare**	

14.6.21 f scet

SCET extracted from CHDO record.

14.6.22 f ert

ERT extracted from CHDO record.

14.6.23 f rct

RCT extracted from CHDO record.

14.6.24 instrument data

CDS or minipacket data.

14.6.25 r length

same value as f_length.

Useful to work backwards through the dataset.

14.7 Archive Dataset / Wideband and Waveform

Wideband and Waveform data hare a common format in the archive dataset. The only difference being that Wideband data is stored in 8 bit samples and Waveform data is stored in 12 bit samples (each 12 bit sample is stored in a 16 bit word). The header information for each time series is stored using an identical format.

Consult the RPWS_WBR_WFR_ROW_PREFIX.FMT file on the archive volume to determine the size of these fields.

14.7.1 Record Header: SCLK

The record header consists of time information and receiver configuration information.

14.7.1.1 SCLK_SECOND

Seconds is the 32 bit seconds field that is used within the instrument as a time base. This time is inferred from the 13 bit seconds field stored in the minipacket that transported this time series to the ground and the 32 bit clock of CDS packet used to transport the associated minipacket. Also referred to as SCLK.

14.7.1.2 SCLK_PARTITION

The clock on the Cassini was set at launch, and with the seconds portion being 32 bits wide, we can expect the clock **not** to roll over during the life of the mission. In the (unlikely) event that the SCLK does regress, a partition number is externally added to force SCLK to be unique, providing a means to derive time from SCLK.

Nominally this field will contain a 1, although a value of either 1 or 0 refers to partiton 1.

14.7.1.3 SCLK_FINE

Sub-second timing resolution is stored in this field. RPWS stores time information only to a resolution of 1 RTI (the RTI period being 125mSec) in the minipacket time field. This filed, then, contains the RTI number, from the minipacket, in the upper 3 bits. Lower 5 bits should always be zero.

14.7.2 Record Header: SCET

The SCLK field above, is converted the UT which is stored in the SCET field. The SPICE kernel is used for the conversion at the time indicated in the label file. The most current SCLK/SCET file is used to derive SCET.

14.7.2.1 SCET_DAY

Epoch Jan 1, 1958.

14.7.2.2 SCET_MILLISECOND

Milliseconds since start of day (i.e. Midnight ZULU)

14.7.3 Record Header: RECORD_BYTES

Record size, in bytes. Total number of bytes in this record.

14.7.4 Record Header: SAMPLES

Sample count. Number of **samples** in this record.

14.7.5 Record Header: DATA_RTI

RTI field from ninipacket.

14.7.6 Record Header: VALIDITY_FLAG

Validity flags, some collected from the minipacket directly, some generated (derived) from the data.

14.7.6.1 MSF

More Status Follows. Status bit present in both WBR and WFR minipacket. Set to indicate an additional 16 bits of status is present in the minipacket. The archive software passes this bit along and uses it to set assitional status bits and load some of the header data fields.

14.7.6.2 WBR

Set to indicate WBR data. Mutually exclusive of WFR bit.

14.7.6.3 WFR

Set to indicate WFR data. Mutually exclusive of WBR bit.

14.7.6.4 VALID_WALSH_DGF

May be set for WFR data only. Set when data was compressed using the Walsh algorithm and additional gain bits are present. See GAIN field that follows.

14.7.6.5 VALID_SUB_RTI

May be set for WBR data only. Indicates that the SUB_RTI field that follows contains additional timing information. The WBR is frequently run out of synchronization

with the spacecraft RTI signal. This bit, when set, indicates that the SUB_RTI field contains an offset from the above SCLK/SCET time of when the WBR data was acquired.

14.7.6.6 VALID_HFR_XLATE

WBR antenna selection is set to the HFR, which is providing frequency translation.

14.7.6.7 VALID_LP_DAC_0

LP_DAC_0 contains valid data.

14.7.6.8 VALID_LP_DAC_1

LP_DAC_1 contains valid data.

14.7.7 Record Header: STATUS_FLAG

14.7.7.1 AGC_ENABLE

May be set for WBR data produced by flight software V2.6 and later. Set to indicate WBR is using the AGC to control the gain setting. For older software and the WFR this bit is always cleared.

14.7.7.2 FINE_TIME_QUALITY

When set, indicates that SUB_RTI is accurate to no better than 10 milliseconds. When running WBR and LFDR (i.e. WFR low-band) at the same time, the WBR data acquisition is triggered by the next LFDR data acquisition, which occurs every 10 milliseconds. Due to hardware restrictions, we can only determine the point in time that we enable the WBR data acquisition, with the actual data acquisition beginning after then next LFDR sample.

(I think there might be a much larger discrepancy that occurs when switching the 12 bit system to high band and back, where the 10mS sample clock is suppressed... Did I make the WFR/WBR acquisition process sensitive enough to completely suppress WBR... -Willy)

14.7.7.3 TIMEOUT

This bit, present in WBR data produced by FSW V2.6 and later, is set to indicate that this time series was not completely captured (it most likely, didn't capture anything). Rather than dedicate code within the instrument to discard bad data, we simply set the status bit and pass the data to the ground. This doesn't occur frequently enough to be an issue with respect to data volume.

Discard this data.

14.7.7.4 SUSPECT

This bit is a status bit from the archive software that indicates that there is something suspect in the WBR or WFR data.

In particular, is we divide the data into even/odd pairs (i.e. treat it as WFR data) there should be some characteristics that are unique to WBR and WFR. If we logically or the odd column and logically of the even column, the WBR data should have some of the upper 4 bits in both columns set. The WFR, however, should have one of the columns with the upper 4 bits cleared (as it is 12 bit data).

Problems with the ground software seems to occasionally mix data, and for some reason it seems to get WFR data marked as WBR data. Bad or missing records are the cause of this problem.

14.7.7.5 HFR_H2

With WBR connect to the HFR mixer, this bit indicates that the H2 is being used as a mixer for the do\wn convertor.

14.7.7.6 HFR_H1

With WBR connect to the HFR mixer, this bit indicates that the H1 is being used as a mixer for the do\wn convertor.

14.7.7.7 EU_CURRENT

WFR Ex+ connected to Langmuir Probe cylinder, in other words the Ex+ is operating as a current sensor.

14.7.7.8 EV_CURRENT

WFR Ex- connected to Langmuir Probe cylinder, in other words the Ex+ is operating as a current sensor.

14.7.8 Record Header: FREQUENCY_BAND

Band selection index. 2 values are valid for WBR and 2 values are valid for WFR.

14.7.9 Record Header: GAIN

Composite gain setting. Combination of the gain level selection of the receiver and the Walsh gain factor. For WBR there is no Walsh gain as it is not compressed using the Walsh algorithm(therefore the Walsh bits are always 0).

14.7.9.1 Walsh_DGF

Scaling applied to the data during compression by the Walsh algorithm.

14.7.9.2 ANALOG_GAIN

Gain amplifier setting. WBR makes use of the entire range from 0dB to 70dB while the WFR gain range is 0dB to 30dB. Both WBR and WFR step in 10dB steps, so the data here is in 10dB steps.

14.7.10 Record Header: ANTENNA

Receiver antenna selection index.

14.7.11 Record Header: AGC

WBR AGC integrator voltage. This is the value used by the automatic gain control code for the WBR. Always set to zero for WFR.

14.7.12 Record Header: HFR_XLATE

HFR translation frequency. See the HFR section of the commands chapter for specific values that appear here.

14.7.13 Record Header: SUB_RTI

This is the millisecond offset of the start of WBR data acquisition.

14.7.14 Record Header: LP_DAC_0

Langmuir Probe DAC voltage (sphere).

14.7.15 Record Header: LP_DAC_1

Langmuir Probe DAC voltage (cylinder).

14.7.16 Record Header: FSW_VER

The archive software determines the flight software version based on the SCLK of the minipacket.

14.7.17 WBR time series

8 bit samples in 8 bit octets.

14.7.18 WFR time series

12 bit samples in 16 bit words. Big-endian (MSB).

15 Command Parser/IEB Image Builder

Tools exists for the unix environment and for the PC that may be used to translate the commands, as documented in the previous section, into several useful formats. Any commands not listed in the 3-291 document (i.e. those that do not have the instrument identifies ***73*** in their command stem) must be presented in the form of a ***73WRAP*** command. A translation to numerical (i.e. hexadecimal) form is required during the process of building *IEB_LOAD* 's.

This section will briefly describe the capability and function of the RPWS command parser.

15.1 73WRAP Generation

Instrument commands may be translated from 3-291 format to hexadecimal (in 73WRAP form) using the utility that is called "parser" and is typically run on a unix system (the RPWS GSE system *rpwshp2*)

There are 3 control flags for generating wrapped commands. In most cases the *text* parameter would be ***73WRAP***.

-wrap *text*

-lwrap *text*

-pwrap *text*

The first form places the *text* parameter followed by a comma delimited list of hexadecimal words. This form does not contain a length field.

The second form places the text parameter followed by a word count and then the list of hexadecimal words (all comma delimited).

The third form places the text parameter followed by a parenthesis enclosed, comma delimited list of hexadecimal words. This form does not contain a length field. This is the form expected when submitting commands to JPL.

15.2 IEB Generation 1

The first step in generating and **IEB_LOAD** involves translating instrument commands, typically in the form documented in 3-291, into a form that is acceptable for the assembler. The utility that is used for this step is called "parser" and is typically run on a unix system (the RPWS GSE system *rpwshp2*)

There are several control flags that are useful for building **IEB_LOAD** files. This is the first step in a multi-step process that requires translating individual commands into a form that is acceptable for use with an assembler.

15.2.1 MASM Assembler

The assembler supplied with the PC should be adequate to produce an IEB load (although this is not the method currently used at Iowa). The *label* parameter is used to provide a global symbol for the group of commands. This global symbol is used by the linker to reference the group of commands.

parser -masm *label*

15.2.2 AVOCET Assembler

The assembler used for flight software has been used for most IEB's prior to launch. The *label* parameter is used to provide a global symbol for the group of commands. This global symbol is used by the linker to reference the group of commands.

parser -avocet *label*

15.2.3Generic Assembler

parser -ieb

parser -ieb *label*

15.3 IEB Generation 2

The second step makes use of an assembler on the PC (either the 8085 assembler or the 8086 assembler should work for this step).

The assembler is used to resolve address references within the IEB load and allows the load to be modularized. This modularization, it is hoped, will allow many components of the load to be reused.

The commands, translated into an assembler source file in the 1st step, are combined with the timing and looping control to form the load. The assembler/linker produce an *Intel Hex File* containing a binary image of the IEB load.

15.4 IEB Generation 3

The final step of the process involves translating the *Intel Hex File* into the format used by the **JPL** using the ***HEXBUILD*** utility**..** This activity is essentially identical to the operation performed on the instrument software so the same utility is used to gather the *Intel Hex Files* together for the creation of an **IEB module**. As mentioned earlier in this document, the ***HEXBUILD*** utility documentation is usually bound with the users guide.

Sample Build File:

```
!
!     Sample IEB load
!
          /Title="CASSINI/RPWS IEB Load"
          /subtitle="IEB sample"
          /header=RPWS
          /header=RPWS_IEB
          /header=current_date
          /header=User_Name
          /header=RPWS_IEB_x.x
          /header=(not_SSR_image_data)
          /eof=1
          /supress_hex=1
          /unix_flag=xx
          /ieb_version=xx
          /ieb_sequence=0
          /ieb_checksum=1
          /ieb_address_mask=0xC000
          /ieb_size=64
          /ieb_eof_flag=xx
          /ieb_format=xx
Sample
               s:\ieb\Sample.HEX
```

Example Build File (14 month checkout):

```
!
!     IEB load
!
        /Title="CASSINI/RPWS 14 month checkout IEB Load"
        /subtitle="Instrument checkout, Earth-1"
        /header=RPWS
        /header=RPWS_CHK14
        /header=current_date
        /header=William_Kurth
        /header=RPWS_CHK14_0.3
        /header=(not_SSR_image_data)
        /eof=1
        /supress_hex=1
        /ieb_address_mask=0xC000
        /ieb_checksum=1
        /ieb_eof_flag=0
        /ieb_format=0
        /ieb_size=64
CHK_M14
        checkout.HEX
```

Example Batch File (14 month checkout):

```
set sasf_delta_t=3
set unix_flag=0
set ieb_version=2
s:\gse\hexbuild checkout
del chk_m14.sas
ren chk_m14.ssf chk_m14.sas
set unix_flag=3
set ieb_version=2
s:\gse\hexbuild checkout
set unix_flag=0
set ieb_version=1
s:\gse\hexbuild checkout > chk_m14.cks
```

Example Link File (14 month checkout):

```
checkout = checkout, maint, stv_ieb, intcal, baselfdr, lpdeltan,
                bcommon1, bcommon2, hical, base_hfr,
                fast_bas, bl_lbnd, bl_local, wskbase, wskmode,
                mfrnoise, wfrnoise, wbrnoise, lfdrnois, sounder,
                lpcheck, allcal, memdump, mag, venus,
                noisy,  ipcruise, hirate, hfrpat,  hfr
  -PutSeg(IEB_Begin_Memory,  08000h)
  -Order(IEB_Begin_Memory, IEB_Memory,
                IEB_Pointers, IEB_Commands,
                IEB_Com_1, IEB_Com_2,
                IEB_Com_3, IEB_Com_4,
                IEB_mem_dump, code,data)
RecLen=16
-ShowPublics
-Symbols
-ShowMods
```

The bold items should be coded as shown to create the correct header lines in the file. Substitute the italic items as appropriate. The non bolded items are suggested control items that are not strictly necessary.

This sample will produce a file *SAMPLE.IEB* from the file *SAMPLE.HEX*. The example will produce a file in SASF format for post-processing and submission to JPL . The sample above was used to produce the IEB for test and submission. Note that several **HEXBUILD** runs were used to produce files in various formats.

The /ieb_version is set to 1 to generate files for internal use and to a value of 2 to generate in **SASF format** for use with the command building tools on the science workstation. Depending on how the SASF formatted file is transferred to the workstation, setting of the **/unix_flag** may need to be added with a value of 3.

The /ieb_eof_flag value would normally be a 1 to produce a valid 73WRAP record for the end- of-file indicator.

/ieb_format value is normally 1, but may be 0 to generate c style hex numbers records (i.e. /ieb_format=0 yields 0x1234 while /ieb_format=1 yields 1234).

The following is the result of the HEXBUILD operation used for 14 month checkout (data values altered in this example, but the header information is verbatim).

```
#RPWS
#RPWS_CHK14
#1998:09:08
#William_Kurth
#RPWS_CHK14_0.3
#(not_SSR_image_data)
#*---------------------------
#* Tue Sep 08 09:16:39 1998 "CASSINI/RPWS 14 month checkout IEB Load"
#*  "Instrument checkout, Earth-1"
#* CASSINI RPWS module builder, Version 12.3
#*V1 IEB August 1998
#* Output format 0
#* Checksum method 1
73IEB_LOAD, 66,  0, (C000,00D0,0000,0000,0000,0000,0000,0000,
 0000,00D8,0000,0000,0000,0000,0000,0000,0000,00E0,0000,0000,
 0000,0000,0000,0000,0000,00E8,0000,0000,0000,0000,0000,0000,
 0000,00F0,0000,0000,0000,0000,0000,0000,0000,00F8,0000,0000,
 0000,0000,0000,0000,0000,0100,0000,0000,0000,0000,0000,0000,
 0000,0108,0000,0000,0000,0000,0000,0000,0000,38A0)
73IEB_LOAD, 66,  1, (?)
73IEB_LOAD, 66,  2, (?)
73IEB_LOAD, 66,  3, (?)
73IEB_LOAD, 66,  4, (?)
73IEB_LOAD, 66,  5, (?)
73IEB_LOAD, 66,  6, (?)
73IEB_LOAD, 66,  7, (C380,8080,A13D,A12A,A126,3868,0030,00F0,
 504C,495F,9240,6D40,C1AB,A262,8080,A131,A138,A129,A126,B020,
 9240,6367,9240,6D48,80FD,C123,C1A7,B3A1,0444,9240,6363,B031,
 80C7,B3A1,0444,C131,C1A7,A120,A125,9240,6D48,9240,6360,9240,
 6D48,B3A1,0444,C1AD,80D5,0000,0000,0000,0000,0000,0000,0000,
 0000,0000,0000,0000,0000,0000,0000,0000,0000,A8CC)
73WRAP,(4310,0800)
```

```
CCSD3ZF0000100000001NJPL3KS0L015$$MARK$$;
MISSION_NAME = CASSINI;
SPACECRAFT_NAME = CASSINI;
DATA_SET_ID = SPACECRAFT_ACTIVITY_SEQUENCE;
FILE_NAME = CHK_M14_Load;
APPLICABLE_START_TIME = 1999-000T00:00:00;
APPLICABLE_STOP_TIME = 1999-000T00:00:00;
PRODUCT_CREATION_TIME = Tue Sep 08 09:16:35 1998;
PRODUCER_ID = RPWS_Group;
SEQ_ID = C00;
HOST_ID = casrpws;
CCSD3RE00000$$MARK$$HJPL3IF0M01300000001;
$$CAS      SPACECRAFT ACTIVITY SEQUENCE FILE
*PROJECT    CAS
*SPACECRAFT 082
*OPERATOR   RPWS_Group, 700 VAN, 319-335-1696
*FILE_CMPLT TRUE
*DATE       Tue Sep 08 09:16:35 1998
*SEQ_GEN    V23.0 Tue Sep 08 09:16:35 1998
*BEGIN      1999-000T00:00:00
*CUTOFF     1999-000T00:00:00
*TITLE      CHK_M14_Load
*EPOCH_DEF
*IEBLOAD,   1999-000T00:00:00.000
*EPOCHS_END
$$EOH
$$EOD
request(CHK_M14_Load,
            START_TIME, IEBLOAD+000T00:00:00,
            REQUESTOR, "RPWS_Group, UI, 319-335-1696",
            PROCESSOR, "SEQ",
            KEY, "RPWS")
   command(1,
      SCHEDULED_TIME,\00:00:00\,FROM_REQUEST_START,
            73IEB_LOAD( 66,  0, [0xC000,0x314D,0x2034,0x3056,0x372E,
            0x3020,0x5332,0x5045,0x3839,0x4090,0x40EC,0x4128,0x4164,
            0x4182,0x41A8,0x41E4,0x4220,0x40AE,0x423E,0x425C,0x4202,
            0x4684,0x410A,0x40B8,0x40B4,0x40C4,0x40C8,0x40CC,0x40D0,
            0x40D4,0x4090,0x4090,0x4090,0x40D8,0x40DC,0x40E0,0x40E4,
            0x40E8,0x459A,0x462C,0x4662,0x427A,0x42A0,0x42B6,0x42FA,
            0x4318,0x4328,0x4444,0x4090,0x4090,0x4482,0x449E,0x4348,
            0x4392,0x42E0,0x4090,0x4338,0x44E2,0x4502,0x4552,0x4532,
            0x4572,0x457A,0x4588,0x476E,0xEA23])
   ),
---------- lots more stuff here, but you get the idea?
   command(95,
      SCHEDULED_TIME,\00:04:42\,FROM_REQUEST_START,
      73WRAP([0x4310,0x5E00])
   ),
end;
$$EOF
```

The following is an example of a contingency load that is produced by setting the environment variable **ieb_sequence_flag** to **1** and **sasf_delta_t** set to **3,1**. The **ieb_sequence_flag** variable causes the **73IEB_LOAD,0,0** command (i.e. the EOF record that must be "wrapped") to be inserted prior to each load record. Since each load record contains both an address and a checksum, each record is treated as stand-alone with the assumption that the checksum will detect any problems not addressed by the spacecraft data system. The **sasf_delta_t** variable controls the time tag applied to each command. In this example we assume that the command bit dictates a 3 second interval between commands (due to the length of the command). The EOF record, however, being only 2 words long, only requires 1 second (hence the **3,1** is 3 seconds for LOAD and 1 second for EOF).

Since this is intended as a contingency for lost commands, the first command is the EOF in order to clear any outstanding problems in the IEB handler. If, for example, the EOF record was lost, it would be necessary to send an EOF in order to prepare the IEB handler for the upcoming load. Rather than making the determination prior to sending the new command set, we simply send the EOF as the first command.

```
CCSD3ZF0000100000001NJPL3KS0L015$$MARK$$;
MISSION_NAME = CASSINI;
SPACECRAFT_NAME = CASSINI;
DATA_SET_ID = SPACECRAFT_ACTIVITY_SEQUENCE;
FILE_NAME = CHK_VEN_Load;
APPLICABLE_START_TIME = 1999-000T00:00:00;
APPLICABLE_STOP_TIME = 1999-000T00:00:00;
PRODUCT_CREATION_TIME = Mon Nov 02 09:30:11 1998;
PRODUCER_ID = RPWS_Group;
SEQ_ID = C00;
HOST_ID = casrpws;
CCSD3RE00000$$MARK$$HJPL3IF0M01300000001;
$$CAS      SPACECRAFT ACTIVITY SEQUENCE FILE
*PROJECT    CAS
*SPACECRAFT 082
*OPERATOR   RPWS_Group, 700 VAN, 319-335-1696
*FILE_CMPLT TRUE
*DATE       Mon Nov 02 09:30:11 1998
*SEQ_GEN    V23.0 Mon Nov 02 09:30:11 1998
*BEGIN      1999-000T00:00:00
*CUTOFF     1999-000T00:00:00
*TITLE      CHK_VEN_Load
*EPOCH_DEF
*IEBLOAD,   1999-000T00:00:00.000
*EPOCHS_END
$$EOH
$$EOD
```

```
request(CHK_VEN_Load,
        START_TIME, IEBLOAD+000T00:00:00,
        REQUESTOR, "RPWS_Group, UI, 319-335-1696",
        PROCESSOR, "SEQ",
        KEY, "RPWS")
    command(1,
      SCHEDULED_TIME,\00:00:00\,FROM_REQUEST_START,
      73WRAP([0x4310,0x0000])
    ),
    command(2,
      SCHEDULED_TIME,\00:00:01\,FROM_REQUEST_START,
        73IEB_LOAD( 66,  0, [0xC000,0x3256,0x2020,0x3056,0x372E,
        0x3020,0x5332,0x5045,0x3839,0x4090,0x40B8,0x40B8,0x40B8,
        0x40B8,0x40B8,0x40B8,0x40B8,0x40AE,
0x40B8,0x40B8,0x40B8,
        0x40B8,0x40B8,0x40B8,0x40B4,0x40B8,0x40B8,0x40B8,0x40B8,
        0x40B8,0x4090,0x4090,0x4090,0x40B8,0x40B8,0x40B8,0x40B8,
        0x40B8,0x40B8,0x40B8,0x40B8,0x40B8,0x40B8,0x40B8,0x40B8,
        0x40B8,0x40B8,0x40B8,0x4090,0x4090,0x40B8,0x40B8,0x40B8,
        0x40B8,0x40B8,0x4660,0x40B8,0x40B8,0x40B8,0x413A,0x411A,
        0x415A,0x4162,0x4170,0x40B8,0x4A64])
    ),
    command(3,
      SCHEDULED_TIME,\00:00:04\,FROM_REQUEST_START,
      73WRAP([0x4310,0x0000])
    ),
    command(4,
      SCHEDULED_TIME,\00:00:05\,FROM_REQUEST_START,
        73IEB_LOAD( 66,  0, [0xC080,0x40B8,0x40B8,0x40B8,0x40B8,
        0x40B8,0x40B8,0x40B8,0x40B8,0x0182,0x0004,0x0186,0x0004,
        0x018A,0x0025,0x018A,0x0002,0x0192,0x0002,0x0196,0x0002,
        0x019A,0x0000,0x4090,0x01A2,0x0040,0x40AE,0x019E,0x0000,
        0x0182,0x0004,0x0186,0x0004,0x018A,0x0025,0x018A,0x0002,
        0x0192,0x0002,0x0196,0x0002,0x019A,0x0000,0x4090,0x3862,
        0x0032,0x00FF,0x4942,0x4855,0x34F2,0x10C6,0x1650,0x3879,
        0x0050,0x0001,0x4257,0x4352,0xC1E5,0xC298,0x0800,0xC131,
        0x3879,0x0050,0x0000,0x4257,0xAFDD])
    ),
    command(5,
      SCHEDULED_TIME,\00:00:08\,FROM_REQUEST_START,
      73WRAP([0x4310,0x0000])

---------- lots more stuff here, but you get the idea?

    command(25,
      SCHEDULED_TIME,\00:00:48\,FROM_REQUEST_START,
      73WRAP([0x4310,0x0000])
    ),
```

```
        command(26,
          SCHEDULED_TIME,\00:00:49\,FROM_REQUEST_START,
              73IEB_LOAD( 66,  0, [0xC600,0x9240,0x6800,0x9240,0x6D07,
              0x9240,0x6F00,0x9240,0x8401,0x9240,0x80DE,0x9240,0x8203,
              0x9240,0x9001,0x9240,0x8CDE,0x9240,0x8E03,0x9240,0x860A,
              0x9240,0x8801,0x9240,0x8A46,0x9240,0x9C01,0x9240,0x98DC,
              0x9240,0x9A03,0x9240,0x9228,0x9240,0x9404,0x9240,0x9646,
              0x34F2,0xCC4E,
0x0000,0x34F2,0xCC50,0x0000,0x3483,0xCC40,
              0x0000,0x3483,0xCC80,0x0000,0x4090,0x0000,0x0000,0x0000,
              0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
              0x0000,0x0000,0x0000,0x0000,0x133A])
        ),
        command(27,
          SCHEDULED_TIME,\00:00:52\,FROM_REQUEST_START,
          73WRAP([0x4310,0x1A00])
        ),
      end;
      $$EOF
```

15.5 IEB loaded with ALF load

Available in Flight Software Version 2.4.

Although BULK Memory is not large enough to hold the flight code and the IEB load, it should not be necessary to store the IEB load in BULK Memory as it is kept in an area or LRP memory that is not scrubbed during a reset and, hopefully, not affected by software or hardware problems on the LRP. Code on the LRP normally leaves the area of memory that holds the IEB load deselected. It is hoped that this will prevent corrupting the IEB Memory if the processor suffers a SEU or software failure. Note that his would be the case when loading the IEB separately from the ALF load.

15.5.1 Restriction on ALF loads

ALF loads are restricted to loading the lower part of memory on the 8085. Specifically, the ALF records are allowed to load 0000-7BFF and 7F00-7FFF. The area from 7C00-7EFF and all of memory above 8000 are restricted by the flight ROM on all 3 processors. Also keep in mind that the memory from 0000-7FFF is scrubbed (zero-filled) immediately following a reset.

This restriction in the flight ROM's prevents direct access to IEB memory by the ALF loading mechanism. As of version V2.3 of the flight software, the code that is resident on the LRP requires about 20K bytes of the 31K bytes available to the ALF loading mechanism, leaving about 9K bytes of unused memory.

15.5.2 Workaround to ALF restriction

In order to work around the load restriction imposed by the ALF mechanism, the IEB load is restricted to about 8K-9K bytes. This is about half of the available IEB memory and should be sufficient to establish a reasonable set of science operational modes. The IEB load must be offset from it's normal location to a location specified in the IEB handler. In addition

the load that is generated will include a checksum table that is located at an fixed address, again defined by the IEB handler.

The IEB Handler may then examine the area of memory that is expected to contain the IEB load and move it to IEB memory if, in fact, the load was delivered. If the IEB load is not included with the ALF load, the handler may proceed to checksum verification in the expectation that IEB memory still contains a good copy of the IEB load.

15.5.3 Workaround to limits on IEB size

IEB memory consists of a 16K byte block of memory that is controlled by a bank switching mechanism on the 8085. In normal operation, the IEB bank is disabled by activating an alternate bank of memory at the same location. The size of the IEB memory is determined by the hardware bank switching mechanism.

Since the memory available on the LRP to temporarily hold the IEB load is smaller than the available IEB memory, it is reasonable to expect that a partial IEB load would be included in the ALF load. As a result the beginning of IEB memory that contains the dispatch table or vector table may be partially populated. As an example, the IEB load used for the first instrument checkout contains 16 triggers that define some basic science modes that occupy the first 48 bytes of IEB memory. Assuming that the vector table contains sufficient place-holders to fill out to trigger number 7E or 8E, it becomes clear that we will require some of the 73IEB_LOAD commands, used to fill out the IEB load following the ALF load, to be odd lengths. As an alternative, we can require that the entire IEB memory be reloaded (i.e. no partial reloads).

15.5.4 IEB "load-ability' issues

The IEB handler in V2.3 requires that a complete IEB load be presented to the instrument in order to correctly load. Each 73IEB_LOAD command contains address, checksum, and sequence information. The checksum must be valid and the sequence must be montonically increasing in order for the IEB record to be loaded into memory.

If any record is damaged in transmission to the spacecraft, the damaged command along with successive commands will be discarded.

If we expect to deliver IEB loads directly from the ground, perhaps a better scheme would be to discard invalid records (i.e. those with invalid checksum) and load all valid records. A second load could be attempted to fill in any gaps that occur. A record can be kept of breaks in the sequence that could be obtained on the ground through the use of MRO commands or perhaps the error report could be delivered to the housekeeping process through a micro-packet.

As part of the new scheme, we may consider an additional 73IEB_TRIGGER, MASK command to be used to clear IEB memory prior to loading. This would eliminate all traces of the IEB contained in an ALF load. This would eliminate the possibility of a false checksum

(i.e. a checksum in the old load that matches one in a new load, making it difficult to determine if the IEB is properly loaded.

15.6 IEB sequence issues

In the event that problems delivering a good *IEB Load* in real-time occur, a workaround exists to bypass some of the error checking mechanisms in the IEB handler. This change effectively disables the sequence checking implemented in the handler by presenting each *IEB Load record* as a standalone load to the IEB handler.

The IEB handler expects to see the EOF record, **73IEB_LOAD, 0, 0** at the end of a load. The handler does not, however, have any expectations about the length of the load. If each record is marked with a sequence number of zero and followed by an EOF record, the loader will place the record in IEB memory and recalculate the checksums. If any record is dropped the entire load may be resent to the instrument with the expectation that the problem record will be successfully received during a subsequent transmission. If, during the subsequent transmission, any records are lost that are already in memory, this is of no consequence.

The HEXBUILD utility has the capability to insert an EOF record prior to each IEB Load record. This placement provides an EOF record to the IEB handler in order to reset the expected sequence number in the event that the EOF record was not received. The timing control is also independently adjustable for the IEB_LOAD and the EOF record. Since the EOF record is only 2 words long, it should require less time to transmit to the spacecraft. Load times can be improved by reducing the time allocated to the EOF record.

16 Flight Rules

This section contains discussions of the various flight rules established for the RPWS instrument.

16.1 73FRC1 RPWS ALF Load Restriction

This rule clarifies the limited processing capability when in sleep mode.

The instrument performance is significantly reduced while operating in sleep mode. This rule simply documents the fact the RPWS must be operating out of sleep in order to perform a download. A suggested work-around involves sending ALF records at a reduced rate. When operating in sleep, the instrument will process ALF records, but the processing time required to handle each record increases to several RTI periods. Adding an idle time of 4 to 8 RTI periods between each group will allow the instrument to successfully perform the download.

Failure to honor the spirit of the rule will cause the instrument to fail to perform the download.

A corollary to this rule would indicate that the 73ALF commands (i.e. memory load commands) are ignored by any downloaded software. In other words, it is **not** possible to reload software without resetting the instrument.

16.2 73FRC2 RPWS Recovery time following reset

This rule attempts to meet the minimum idle time following a reset.

The processors perform a memory scrub operation following reset. During this scrub operation the processors ignore incoming commands and BIU Discrete bits. The processor clocks are all set to the slowest state to guarantee the instrument is on it's lowest power state.

As the reset signal used by the instrument (i.e. BIU Discrete 0) is edge sensitive, the rising edge triggers the reset activity. The signal may remain high indefinitely or be brought low almost immediately (i.e. in the following RTI period). This flight rule is intended to prevent commands, other than the reset control commands, from reaching the instrument during the initialization period.

Failure to honor the spirit of the rule will cause the instrument to fail to perform the download.

We have specified a minimum idle time following any **73RT_RESET** command of 5 seconds. This results in a minimum of 10 seconds following the actual reset before any commands may be delivered to RPWS. Bear in mind that the instrument will start an internal IEB load approximately 23 seconds after the **73RT_RESET, RESET** command is delivered and that you should schedule the **6SSR_MEM_LOAD** command close enough to the **73RT_RESET, RELEASE** command to avoid having the instrument processing an internal load.

10 seconds between commands is marginal for operations on the bench, it seems to work better if you make use of some delta-t of 6-8 seconds between the **73RT_RESET** and **6SSR_MEM_LOAD** commands.

For Example:

00:00 73RT_RESET, RELEASE
00:06 73RT_RESET, RESET
00:12 73RT_RESET, RELEASE
00:18 6SSR_MEM_LOAD

16.3 73FRB3 RPWS Antenna Motor on time limit

This rule attempts to minimize the window during which power may be applied to the antenna mechanism.

The antenna mechanism is controlled through 3 separate control paths. The mechanism is powered separately from the main electronics. The power routing relay is enabled through a BIU Discrete bit assigned to each mechanism. And finally, the processor selects an element and direction to move.

The *on time limit* is intended as another interlock to reduce the opportunity for unintended element movement to occur. The processor removes power when a limit switch is encountered, so leaving motor power on, although not desirable, is not immediately fatal.

As an operational interlock, this rule may be interpreted as referring to the time the discrete bits are allowed to be on. As long as the software is functioning normally, deasserting the BIU discrete bit to meet this rule will not interfere with successive deploy operations (whereas removing power from the deploy electronics would obviously prevent any successive activities).

The rule is intended to time the period from the issue of a ***73ANT_CONTROL*** command until the corresponding ***73RT_Ex_x_CNTL, DISABLE*** command.

16.4 73FRC4 RPWS Antenna Discrete bit

This rule clarifies the function of the 3 antenna discrete bits.

This rule makes a note of the fact that the discrete bit is required to be asserted before antenna element movement will occur. Three of the eight BIU discrete command bits are used as interlocks to the antenna control electronics. The corresponding discrete bit must be asserted before power is routed to the antenna deploy mechanism

Failure to honor the rule will not result in any permanent damage, the selected element will simply not move. The situation may be remedied by asserting the appropriate discrete bit and re-sending the antenna deploy command.

Note that the discrete bits control latching relays within the antenna control electronics. The discrete bit must be enabled to allow the respective relays to be enabled. The discrete bits do not directly control power to the motors. As a result, the discrete bits do not directly stop antenna element movement once it has started (the software monitors the BIU discrete bits and stops antenna element movement if required).

16.573FRC5 RPWS Power up Restriction

This rule explains the internal power up sequence requirements that are intended to prevent RPWS from exceeding it's power allocation.

This rule is aimed at cases where the HFR would be powered on when L/P and/or ME02 are already powered. When the instrument subsystems are powered on in a particular order, the associated current transients never exceed the peak power allocated to RPWS. If, however, the specific order is not followed, the turn-on transient will exceed the allocation, possibly driving the S/C into a bus undervoltage condition.

The simple method to avoid this problem is to place the instrument into *SLEEP* and then apply power in the prescribed sequence, leaving any unused portion of the instrument powered off (i.e. simply delete it's power-on command).

Also note that any software reload will cause power to be removed from the receivers. In other words, a software reload will have the same effect as sending a sleep command.

All versions of the flight software (i.e. those later than version 2.0) have been modified to slow the internal power sequencing to avoid a case where two internal supplies were switched during the same RTI period.

To clarify the rule, any internal power change requires that the instrument be in a sleep-like state (i.e. internal subsystems powered off) prior to the power change. The simple method to accomplish this is through the 73RT_SLEEP, 73POWER_CNTL, SLEEP or the 73RT_RESET commands.

Acceptable Power Sequences	Comments
73PS_RPWS, ON	Initial power on does not apply power to receivers
73RT_RESET, RELEASE 73RT_RESET, RESET 73RT_RESET, RELEASE	ROM code removes power from receivers This executes ROM code
73RT_SLEEP, SLEEP 73RT_SLEEP, ACTIVE	Discrete SLEEP state removes power from receivers
73RPWS_POWER, SLEEP, SLEEP 73RPWS_POWER, SLEEP ACTIVE	Commanded SLEEP state removes power from receivers
6EXT_MEM_LOAD	Requires prior 73PS_POWER or 73RT_RESET

Note that the 6EXT_MEM_LOAD command does not, by itself, trigger any power activity within the instrument. This command always appears following an instrument reset (either power-on using 73PS_RPWS or explicitly using 73RT_RESET). If this command is issued without having the instrument in a reset state (i.e. executing out of ROM) the memory load will be ignored and the power switches will remain unaltered.

16.6 73FRB6? RPWS Antenna Command Restriction

This rule restricts use of antenna control commands.

Following successful deploy on 25 October 1997, no additional antenna movement is planned during the mission. To this end, the set of **73ANT** commands as well as the **73PS_ANT_MOTOR** command have been listed as restricted or *not to be used*.

The intent, of course, being that the antenna elements are not to be retracted at any time during the mission. Once deploy software is purged from the S/C (planned for late 1997 or early 1998) no software will exist on the S/C to effect antenna movement. In other words once the deploy software is gone, the **73ANT** commands will be ignored by the instrument (ROM, science, and deploy software).

In addition, as long as the antenna control electronics remain un-powered, no hardware command from the instrument processors can cause antenna element movement. Even a software crash/fault will not cause a problem.

Therefore, consider the **73PS_ANT_MOTOR** as the critical item to observe (i.e. DMD channel X-????) is the antenna control electronics power switch.

73ANT commands, although they should not appear, will not cause antenna element movement.

17 Consumable Items

In normal operations there are no consumable resources within the RPWS instrument. The design was not conceived with any restrictions on power cycling or power on hours in mind. Within the DPU section, conservative timings were used in the design to allow for reasonable/expected degradation due to radiation effects. The software used for science and deploy operations is also configured to reduce power dissipation by reducing bus activity during idle periods (8085 HLT instruction) In addition Power dissipation within the receivers is spread across a large board area.

The ***antenna deploy mechanism***, being a mechanical unit, has a limited life. The mission profile calls for the three antenna mechanisms to be deployed once during the mission so this is not viewed directly as a consumable item. Note that the mechanism specifically avoided the use of a brush type motor to enhance reliability in the event that the motor is called on after a long period of dormancy.

18 Power Consumption

Power consumption tables.

18.1 Telemetry Collection Schedules

When observing telemetry from the power supply (i.e. 12 voltage monitors and 4 current monitors) keep in mind that the ME01 current measurement is instantaneous and will depend on the collection schedule used by CDS. Various activities that occur on LRP will affect when housekeeping data is collected (i.e. Activities on LRP that have a higher priority than housekeeping may cause a housekeeping reading to occur a little later in the RTI period than what would nominally occur. Also, CDS may schedule high priority activities before telemetry pickups in the RTI period.

This can have a significant effect on the ME01 current number we see, as the BIU current draw, as seen by the ME01 power supply, changes by about 100mA when the BIU is transmitting. In a 360Kb mode the BIU transmits for slightly over 57 mSec.

The net effect of this is that for a portion of the RTI period the BIU is drawing an additional 100mA of current that may be visible in housekeeping. It seems to be typical that the bench model will show the additional current while the instrument on the spacecraft will not. We think that is explained as follows:

The bench model exists as a single instrument on the PPCRTIU, thus requiring no 1553 bus bandwidth to support engineering and science subsystems. We do not have models of the other subsystems and cannot have the 1553 traffic present for the other subsystems. The 1553 traffic to support our instrument occurs at the beginning of the RTI period where the housekeeping task would trigger the A/D system. This causes the current spike to be visible on the bench.

The flight model, on the other hand, coexists with the full complement of science and engineering subsystems on the spacecraft. The level of traffic is much greater, with typical 1553 usage being at a sustained level (around 75% to 80% ???). The traffic from RPWS (i.e. the 6 science telemetry records) will, most likely, occur later in the RTI period than they do on the bench.

Also keep in mind that the collection schedule is dictated by CDS, we will transmit as much data as CDS demands, but simply mark most of it as empty when there is not sufficient data to be delivered.

Also keep in mind that the 1553 traffic is, effectively, prioritized by the order in which CDS arranges pickups and deliveries. Science telemetry is (probably) the lowest priority activity and would, therefore, occur as the last transactions in any given RTI.

Typical CPU usage levels on the LRP are on the order of 25%-30% so we might expect to completely miss seeing the BIU traffic on the current monitor altogether. In addition, it is probably not unreasonable to see the BIU current when RPWS is in a slow data collection mode.

The monitoring limits we have established on the ground seem to be reasonable, based on several years of operation. We see occasional excursions into the yellow limits and rarely see a sample that is just into the red overcurrent.

18.2 Suggested Power-On Sequence

The following sequence presents the optimal current load to the spacecraft. The HFR presents the largest power on surge followed by the Iowa receivers (ME02). Both of the power switches located on the HRP (L/P digital and HRP A/D) have slow turn-on to minimize the turn-on spike. The L/P digital must be powered on before the L/P analog to avoid damage to the L/P A/D converter (this is enforced with a hardware interlock).

HFR Electronics

ME02 MFR, WFR, WBR receiver

Langmuir Probe Digital

Langmuir Probe Analog

WFR/WBR A/D section on HRP

18.3 Instrument power-on current plot

18.4 Instrument idle power

Measurements of instrument power requirements

18.4.1 Flight Electronics

	SLEEP		RUN		MAINTENANCE	
	SLEEP	ACTIVE	HALT	NOP	ON	OFF
ROM MODE			N/A			
RAM MODE				N/A	N/A	N/A

18.4.2 Flight Spare Electronics

	SLEEP		RUN		MAINTENANCE	
	SLEEP	ACTIVE	HALT	NOP	ON	OFF
ROM MODE			N/A			
RAM MODE				N/A	N/A	N/A

18.4.3 Engineering

	SLEEP		RUN		MAINTENANCE	
	SLEEP	ACTIVE	HALT	NOP	ON	OFF
ROM MODE	102ma	116ma	N/A	116ma	172ma	116ma
RAM MODE	98ma	98ma	104ma	N/A	N/A	N/A

18.5 Low Rate Science

Measurements of instrument power requirements

18.5.1 Flight Electronics

SAF 142	POWER CONSUMPTION	
	NET(NO BIU)	GROSS (WITH BIU)
BASE		
HFR		
HRP		
ME02		
L/P DIGITAL		
L/P ANALOG		
BASE		
HFR		
BASE		
HRP		
ME02		
BASE		
L/P DIGITAL		
L/P ANALOG		

18.5.2 Flight Spare Electronics

SAF 142	POWER CONSUMPTION	
	NET(NO BIU)	GROSS (WITH BIU)
BASE		
HFR		
HRP		
ME02		
L/P DIGITAL		
L/P ANALOG		
BASE		
HFR		

SAF 142	POWER CONSUMPTION	
BASE		
HRP		
ME02		
BASE		
L/P DIGITAL		
L/P ANALOG		

18.5.3 Engineering Model

SAF 142	POWER CONSUMPTION	
	NET(NO BIU)	GROSS (WITH BIU)
BASE	<100ma	
HFR	292ma	
HRP	376ma	
ME02	478ma	
L/P DIGITAL	512ma	
L/P ANALOG	552ma	
BASE	<100ma	
HFR	292ma	
BASE	<100ma	
ME02	242ma	
HRP	328ma	
BASE	<100ma	
L/P DIGITAL	118ma	
L/P ANALOG	122ma	

18.6 High Rate Science

Measurements of instrument power requirements

18.6.1 Flight Electronics

SAF 248	POWER CONSUMPTION	
	NET(NO BIU)	GROSS (WITH BIU)
BASE		
HFR		
HRP		
ME02		
L/P DIGITAL		
L/P ANALOG		
BASE		
HFR		
BASE		
HRP		
ME02		
BASE		
L/P DIGITAL		
L/P ANALOG		

18.6.2 Flight Spare Electronics

SAF 248	POWER CONSUMPTION	
	NET(NO BIU)	GROSS (WITH BIU)
BASE		
HFR		
HRP		
ME02		
L/P DIGITAL		
L/P ANALOG		
BASE		
HFR		

SAF 248	POWER CONSUMPTION	
BASE		
HRP		
ME02		
BASE		
L/P DIGITAL		
L/P ANALOG		

18.6.3 Engineering Model

SAF 248	POWER CONSUMPTION	
	NET(NO BIU)	GROSS (WITH BIU)
BASE		
HFR		
HRP		
ME02		
L/P DIGITAL		
L/P ANALOG		
BASE		
HFR		
BASE		
HRP		
ME02		
BASE		
L/P DIGITAL		
L/P ANALOG		

19 Software Changes

This section contains details of the changes applied to the software following launch.

19.1 Version 2.7

This version is planned for 3Q/4Q of 2003 (looks like we may not even bother).

Version V2.7 finally has a working implementation of RST-5 memory move assist. This is demonstrated in trigger 32 of the base IEB where we run WBR and LFDR concurrently with WBR producing data are a rate of about 250K bits/sec.

The 73POWER_CNTL, PAUSE command is also rendered harmless in this release (the code to pulse the SLEEP line has been removed).

CTLWBR will not pass compression along if WBRC offset 0x58 is set to 0xFFFF (i.e. will not allow compression in LFDR SYNC mode as this will cause HRP to hang).

BIU Direct handler seems to have had the interrupts disabled for long periods of time. At certain points this can block a time update on HRP, causing the SCLK to be retarded by 256 seconds when RTI-0 occurs. Enabling interrupts an a slightly earlier point should allow RTI interrupt to occur correctly.

DUST analysis on the DCP was operating at the same priority as the idle process. This would starve DUST analysis of CPU cycles. The low priority definitions for DCP were altered (increased by 1).

73WBR_BURST requires an appropriate setting of the W08I MUX byte.

19.2 Version 2.6 / Patch level 005

This is a set of patches that are used to address the following problem (this patch set was used with C39 and later IEB loads). C39/C40 implemented the removal of the MUX control byte setting only for trigger 10, It did not eliminate the MUX control byte setting in other triggers. The full implementation of this fix id in the *base26_5* directory.

19.2.1 WBR AGC glitch.

Fix a problem in the gain control code in W08I that was causing an invalid AGC reading to occur. This resulted in gain settings that were incorrect. This patch also eliminates the need to use the W08I MUX control byte.

```
# Patch 005       20 MAR 2003 AGC Glitch (W08I)
#                   Gain control code fragment in W08I
#                    was glopping up the antenna select.  Puts glitches
#                    in WFR and MFR.  Found and extra instruction to
#                    delete (talk about careful planning).
#                   Also, the 73WBR_MODE_CNTL and 73DUST_MODE_CNTL
#                    commands are patched such that they no longer
#                    alter the MUX control byte
00:00:00 00mem_tweak, hrp, byte, 0x32CE, 0
00:00:01 73mem_tweak, hrp, byte, 0x14, 0x40, TWEK
00:00:02 00mem_tweak, hrp, byte, 0x8225, 0x3A
00:00:03 00mem_tweak, hrp, byte, 0x82D4, 0x3A
00:00:04 73mem_tweak, hrp, byte, 0x14, 0x00, TWEK
```

19.3 Version 2.6 / Patch level 006

This is a set of patches that can be used to address the following problems.

19.3.1 73POWER_CNTL, PAUSE

This patch group removes an instruction that can cause the HRP to hang. Not of significant importance as we don't use the code that can cause the problem.

19.3.2 Time fixup on HRP

This patch re-enables interrupts in the BIU-direct handler. This is intended to eliminate a problem we have seen where time on HRP regresses. 73WBR_MODE_CNTL and 73DUST_MODE_CNTL commands are patched so that they no longer alter the MUX control byte in W08I.

19.3.3 WBR AGC glitch.

Fix a problem in the gain control code in W08I that was causing an invalid AGC reading to occur. This resulted in gain settings that were incorrect.

19.3.3.1 Listing of Patch Level 006

Patch 005 is incorporated into the IEB loads starting with C39.

```
#       Patch to V2.6 code
#       19 Sep 2002 WTR
#       73POWER_CNTL command in 73IEB_HALT, IDLE
#
#         Examine memory locations prior and subsequent
#       to patching.  This, I hope, will give us a record,
#       in telemetry, of the application of the patch.
#         This patch may be applied more than one time
#       without causing any problems.
#
# Patch 001     19 SEP 2002 73Pwr_Cntl, Pause
#
00:00:10 73mro, lrp, hsk, 3B54, 0           # Examine the areas
00:00:12 73mem_tweak, lrp, word, 0x0858, 0x0101, IEBC  # Remove Power_CNTL
                                                       #   Instruction
00:00:14 73mro, lrp, hsk, 3B54, 0           # Examine the areas


# Patch 002     03 OCT 2002 DMA mode 3 fixup
#               05 NOV 2002 Mode-3 doesn't even work,
#                       so there is NO point in fixing
#                       the code
#
#00:00:20 73mro, hrp, hsk, 37bd, 0                  # Examine
#00:00:22 00mem_tweak, hrp, byte, 0x37BD, 0x0011    # Change port address
#00:00:24 00mem_tweak, hrp, byte, 0x37C1, 0x0011    # Change other port address
#00:00:26 73mro, hrp, hsk, 37bd, 0                  # Examine result
```

```
# Patch 003       12 DEC 2002 Time fixup on HRP
#                   Change RTI-0 signal to RTI-1.
#                   Allows HRP to slip 1 RTI wothout messing up
#                   Time too bad.
# Found problem (hope we did anyway), so this
#   patch not really needed either...
#
#00:00:30 73mro, hrp, hsk, 0x0FC0, 0
#00:00:31 73mro, dcp, hsk, 0x0FE8, 0
#00:00:32 73mro, lrp, hsk, 0x4B40, 0
#00:00:35 00mem_tweak, HRP, BYTE, 0x0FC1, 0x01      # HRP RTI-1
#00:00:36 00mem_tweak, DCP, BYTE, 0x0FED, 0x01      # DCP RTI-1
#00:00:37 00mem_tweak, LRP, BYTE, 0x4B43, 0x63      # send 2 RTI delay
#00:00:40 73mro, hrp, hsk, 0x0FC0, 0                #   during RTI-7
#00:00:41 73mro, dcp, hsk, 0x0FE8, 0
#00:00:42 73mro, lrp, hsk, 0x4B40, 0
```

```
# Patch 004       19 DEC 2002 Time fixup on HRP
#                   biuint_5 seems to be keeping interrupts
#                   off for too long.  stick in a pair of
#                   EI instructions to alleviate the problem
#
00:01:00 73mem_tweak, hrp, byte, 0x15, 0x7F, BIU_# lower priority (BLOCK)
#
00:01:01 73mro, hrp, hsk, 29BB, 0                  # look at initial conditions
00:01:02 73mro, hrp, hsk, 2B44, 0                  #
#
#         Load new code fragments
#
00:01:05 00mem_tweak, hrp, byte, 0x3CF0, 0xFB  # EI
00:01:06 00mem_tweak, hrp, byte, 0x3CF1, 0x3A  # LDA
00:01:07 00mem_tweak, hrp, word, 0x3CF2, 0x23D2        # DMA_WORM_HOLES
00:01:08 00mem_tweak, hrp, byte, 0x3CF4, 0xC9  # RET
#
00:01:11 00mem_tweak, hrp, byte, 0x3CF5, 0xFB  # EI
00:01:12 00mem_tweak, hrp, byte, 0x3CF6, 0x3A  # LDA
00:01:13 00mem_tweak, hrp, word, 0x3CF7, 0x3C2C        # DMA_WBR
00:01:14 00mem_tweak, hrp, byte, 0x3CF9, 0xC9  # RET
#
#         We do it in this order so screw-up doesn't crash HRP
#
00:01:21 00mem_tweak, hrp, word, 0x29BC, 0x3CF0        # Patch LDA address
00:01:22 00mem_tweak, hrp, byte, 0x29BB, 0xCD # Change to CALL
00:01:23 00mem_tweak, hrp, word, 0x2B45, 0x3CF5        # Patch LDA
00:01:24 00mem_tweak, hrp, byte, 0x2B44, 0xCD  # Change to CALL
#
00:01:31 73mro, hrp, hsk, 29BB, 0                  # look at final conditions
00:01:32 73mro, hrp, hsk, 2B44, 0                  # look at final conditions
00:01:33 73mro, hrp, hsk, 3CF0, 3CFE               # look at final conditions
#
00:01:40 73mem_tweak, hrp, byte, 0x15, 0x6B, BIU_# return priority to normal


# Patch 005       20 MAR 2003 AGC Glitch (W08I)
#                   Gain control code fragment in W08I
#                    was glopping up the antenna select.  Puts glitches
#                    in WFR and MFR.  Found and extra instruction to
#                    delete (talk about careful planning).
#                   Also, the 73WBR_MODE_CNTL and 73DUST_MODE_CNTL
#                    commands are patched such that they no longer
#                    alter the MUX control byte
00:02:00 00mem_tweak, hrp, byte, 0x32CE, 0
00:02:01 73mem_tweak, hrp, byte, 0x14, 0x40, TWEK
00:02:02 00mem_tweak, hrp, byte, 0x8225, 0x3A
00:02:03 00mem_tweak, hrp, byte, 0x82D4, 0x3A
00:02:04 73mem_tweak, hrp, byte, 0x14, 0x00, TWEK
```

19.4 Version 2.6

This version was released to PSL on **02-OCT-2002**.

Version 2.6 adds some capabilities that started out as a work-around to replace a failing MFR-2. Significant changes occurred on the HRP with code moving about and some internal commands being deleted (early portions of the AGC code that are not used).

This version also has some updates to the STM tables (Spacecraft Telemetry Mode). In particular, S&ER-5 and S&ER-10 had mod words that were identical and they are handled correctly with this software revision.

WBR/WFR high-band lockout corrected. We can now schedule high-band WBR activities without needing to stop WFR/LFDR, the lockout works correctly now. W12J sample clock now reverts to 100Hz following data acquisition (helps LFDR synch mode work smoothly). MMISR has some fixes to the RST-5 code (we haven't used this too much yet). WFROUT correctly handles large segment counts now. W08I AGC timing works correctly now. W08I sub-RTI status byte now appears to be accurate. CMPX (on DCP) now looks for AGC enable bit and skips forwarding packets that have AGC disabled to the AGC process (saves some DCP CPU cycles). IEB handler is a little more robust with respect to handling **73IEB_LOAD** commands.

New Commands

- 73WBR_BURST
- 73WFR_TOGGLE_CNTL
- 73LFDR_TOGGLE_CNTL
- 00WFR_AUTO_SET
- 00LFDR_AUTO_SET

Minipacket Status Changes

- WBR minipacket now has a bit that indicates when a timeout has occurred. This bit, when set, indicates that the WBR data is invalid (stale data, from a previous acquisition cycle).

- WFR minipacket has 2 status bits that are overloaded. Their meaning depends on how the packet was handled in the instrument.

Commands to look out for in IEB's that will cause problems with this version.

- 73MEM_TWEAK, HRP, BYTE/WORD, 0X5A, NN, WBRC
  This will alter the burst count for 73WBR_BURST commands. Was used to force WBR/LFDR sync mode to work.

- 00MEM_TWEAK, HRP, BYTE/WORD, 87C9, 0000
- 00MEM_TWEAK, HRP, BYTE/WORD, 87CB, 0000
  Was used to force WFR mode to work correctly. Will alter instructions and possibly cause a crash.
- 00MEM_TWEAK, LRP, BYTE/WORD, 0x29xx-0X31xx, nn
  Was used to fix a pattern used to control clock speed on all processors. Will alter an instruction and possibly cause a crash.
- Setup sequences may have new commands separated from similar commands to allow for V2.5 compatibility. May want to regroup these following V2.6 upload.
- **73POWER_CNTL, PAUSE** command is being phased out. It is undesirable because it causes power to be removed from Langmuir Probe for approximately 2 RTI periods.

  The **73IEB_HALT, IDLE** command has a **73POWER_CNTL, PAUSE** command that may need to be removed if this command is to be used. The following MEM_TWEAK may be used to remove the offending power control command:

  73MEM_TWEAK, LRP, WORD, 0x0858, 0x0101, IEBC

19.5 Version 2.5

WPV found. BIU has a deficiency that allows corruption of BCRTM R6. V2.5 has a simple work-around for the problem (we disable the 8237 when accessing BCRTM registers).

Change to WBR auto-gain from 1-of-N to holdoff-N.

Watch-Dog timer defaults to ENABLED.

Another ISA addressed: HRS sequencing occurs on HRP and BIU handler zeros record length on HRS packets after delivery. This eliminates a problem with duplicate packets when major mode changes occur (i.e. When telemetry mode on S/C changes).

19.6 Version 2.4

Following delivery of the 2.3 modifications, we begin work on the next group of improvements and fixes.

19.6.1 TWEAK/MRO

As a result of housekeeping issues raised with activities in support 14 month checkout, the memory read out function has been slightly modified to provide access to the constant that defines the size of a minipacket delivered to the science telemetry stream. This change allows

the housekeeping buffer to fit within a single MRO minipacket of 192 data bytes. This make both Science and Housekeeping parameters easily tweakable.

19.7 Version 2.3

This version is planned to be uploaded prior to 14 month checkout. Since we have very little operating time with the prior version, minimal changes are planned.

19.7.1 Power and SLEEP Control

Added a method to control the HRP instruments through the SLEEP control line. This change allows a method of shutting down data collection on the HRP without using the IPC mechanism. IPC is susceptible to packet loss under high load situations, such as when operating at high data rates. This change alters the handling of the SLEEP line breaking it down into a 2 step activity.

When HRP detects that the SLEEP line is asserted, it's first action is to idle all data collection on the HRP. Only when 4 successive RTI periods with SLEEP asserted occur, will power be switched to L/P and the WBR/WFR A/D system.

19.7.2 TWEAK/MRO

As a result of issues raised with activities in support of Venus-1, the memory read out function has been slightly modified to provide cleaner access to the timer that controls the bit rate for memory read out mini packets. In the previous version, the delay between MRO packets was fixed at 9 RTI's. Altering this value required a code patch. This change simply moves the timer to a fixed location in the process descriptor area so that any future changes will not require changes to IEB or other commands.

19.7.3 IPC/F5/Miniproc

Bugfix in IPC: Error recovery timer would occasionally loose an interrupt. Setup changed to improve immunity to this condition. Upgrade to MiniPkt: Non-blocking flag and Fast-delivery flag added. F5 restructured to make room for MiniPkt changes (lost a buffer on LRP)

19.7.4 MAINTENANCE

During the first RAM Maintenance activity, it was noticed that the discrete status bit was not correctly set to indicate that maintenance mode was active. Changes to the code prior to launch moved the code fragment that sets the status bit to a point in the control flow that was too early to allow the bit to be properly asserted. The code fragment to set the status bit was moved to the appropriate position to allow the status to appear correctly.

19.7.5 MFR

In prior versions, the MFR operated asynchronously with respect to antenna switching. Although an individual sweep was synchronized with the SCLK, the antenna selection was not locked to the SCLK in any way.

Changing the antenna selection code to make it synchronous with SCLK allows other receivers to avoid activities that may cause interference with the MFR.

19.7.6 WBR

The AGC update rate is now programmable (previously it was fixed at ½ of the data set capture rate). When operating in higher data rates, the AGC activity must not occur any faster than before, but when operating the WBR at lower data rates (i.e. 2 RTI's between captures and slower), the ACG may be updated every RTI if desired.

Also note that the AGC update rate may also be set to slower speeds, if desired.

19.7.7 WFR/WBR interference

Adds Mx to 12 bit acquisition process to prevent WBR Hband interference during WFR Hband when using LFDR sync (affects SKR science mode).

This appears to not work as intended, but no adverse side-effects have been detected. Will look at this more closely following V2.3 submission.

19.7.8 HFR

Remove the meander table patch from the HFR handler, reducing ALF requirements.

Determined that the cross-correlation sig problem fix didn't work well, so it was removed. This 1 instruction patch can be included in the IEB if required.

Add code to HFR handler to allow downloading RICE code from the HFR-IEB area of memory.