5.0. Overview of 3-310

This document summarizes at Level 3 the CDS processing associated with the Phase 2 flight s/w changes. For ease of comparison, the instruments are covered in the same order as in the Level 2 document. For each instrument the data pickup is described, followed by a description of the processing needed to form the packets. Finally there is a brief description of the changes in the instruments' flight s/w.

5.0.1. Use of Fill Data

Whenever the downlink rate exceeds the rate of RT data generation and PB data are not available, CDS will fill the excess capability with high-rate PWS data collected from the PWS LPW buffer, placed into 442-byte packets (including header), and then into RT VCDU's. These are sent to the TLM build process as needed.

5.0.2. Buffer Management

CDS will provide a large (70K) Multi-use Buffer. This buffer will be used to temporarily store raw DMS tape data prior to processing, raw RRCC data, processed DMS tape data (VCDUs) prior to downlink, processed RRCC data (VCDUs), and processed R/T science data (VCDUs) prior to downlink.

CDS will provide a smaller (4K) Priority Buffer. This buffer will be used to temporarily store engineering and OPNAV processed data (VCDUs). Ready data in this buffer will always be given priority over data in the multi-use buffer when constructing the downlink frame. Capability will be provided (via 6TMSED/6TMCHG,FILL) to store the engineering data during periods of no downlink.

CDS will use the 13.5K Imaging ICT line buffer for three purposes. About 5K will be used to process NIMS REC data when this function is active. At other times about 12K will be used to support PPR Burst-to-tape. Finally, when the SSI playback data is being processed, the full buffer will be used to select 8X8 blocks to send to AACS for ICT compression. The ICT data clear in about one RIM, and data left in the buffer at the end of BPT are ignored, but five minutes should be allowed after either of these modes before reusing the buffer.

5.0.3. CDS Internal Fault Protection Changes

CDS will terminate the new phase 2 non-redundant processes after detecting any privileged or non-privileged CDS internal fault if that fault triggers an SFP response. Note that S/C Safing SFP response creates a CDS non-privileged error.

CDS will terminate the new phase 2 non-redundant processes prior to executing any SFP response (other than the new despun bus reset mentioned below) to provide adequate resources to execute the routine. Excepted from this requirement are the "normal" SFP functions of Thruster Firing Imminent (TFI), Thruster Firing All Clear (TFAC), and the LLM SFP Temperature Monitors.

CDS will initiate S/C Safing SFP response after any fault which terminates the new phase 2 non-redundant processes.

CDS will provide for autonomous detection of a despun bus reset and will then recover the affected string, request a new (TBD) SFP response, and will continue the execution of the stored sequence. Functions not compatible with this requirement will be moved out of the despun LLMs.

5.0.4. System Fault Protection Changes

While most of system fault protection is unchanged and provides the same level of single fault tolerances as currently exists, selected system fault protection functions will be changed in order to save memory, and to make responses compatible with the new LGA functions.

5.0.4.1. Algorithm Deletions

Delete system fault protection algorithms not required during the orbital operations period. These include:

a. RPM Overpressure (monitor and response)
b. Celestial Reference Loss (response)
c. AACS_INIT_C (relay/joi sections of the response)
d. UVREC_C (relay/joi sections of the response)
e. DMS Recovery (move non-critical mode portions to UVREC_C)
f. Critical Mode Operation
g. Critical Mode SAFING response
h. DBUM Swap

5.0.4.2. Modifications to Existing Algorithms

Modify system fault protection algorithms as required to support the LGA mission. These will include:

a.	SAFING	Delete VEEGA conditionals (S_TWTA_HI, FPWS_COLD), Bay E heater.
b.	TFAC	PPR commands stored in PGVs
c.	RPMSAFE	Replace RCT-NIMS with Bay E heater
d.	UVREC	Place PLS in a safe condition after a UV-trip (PLS Instrument Power OFF, Replacement Htr ON, Supplemental Htr ON)

5.0.4.3 Algorithm Addition--DBUSR

Add a new SFP algorithm which is requested by CDS internal FP after recovering from a Despun Bus Reset. This algorithm will issue only those commands necessary for S/C and subsystem safety when continuing the stored sequence after the despun bus reset. The response is limited to one command per 15 minor frames to allow concurrent execution with other ongoing activities. Commands are **TBD**.

5.05. Delete Unnecessary CDS Functions

CDS will delete functions which are no longer required in the LGA mission in order to reallocate memory to new functions. Functions to be deleted include Critical Operations Mode (including 6MARK FC).

5.06. Science Instrument Software Changes

The following table summarizes the changes which are being made in the software within the science instruments. The column labled “Class” indicates the size of the reprogramming as a fraction of the initial programming effort: Class 1 = <20%, Class 2 = 20%-40%, and Class 3 = 40%-60%.

Instrument	Class	Description
SSI	2	Slow image readout 2x2 sum at low rate On-chip mosaic
NIMS	2	Edit mirror position, λ Decrease housekeeping data Add realtime data
PPR	0	None
EUV	1	Store spectrum by sector
UVS	1	Change observational modes
MAG	2	Redesign optimal averager
DDS	1	Add lower data rate capability
PLS	3	Decrease cycle time, resolution Compress sensor data Minimize housekeeping data
PWS	0	None
EPD	3	Change to spin-based sampling Reduce channel sampling Minimize housekeeping data
HIC	0	Reconfigure for MRO’s

5.07. Packet Format Summary

The table below shows a typical telemetry packet. The top row gives the names of the fields, the second row the size of the fields in bits, and the third row the data entered.

The first bit of the packet, the Time Included flag, is set to 1 for those packets which include a time field. The rest of the first byte is the Application ID, which is used to identify the source of the data and the format of the rest of the header. The next nine bits give the byte count of the data portion of the packet. Next is the 7-bit Sequence Number, which is a packet counter separately maintained for each Application ID, going from 0 to 127 and then rolling over to 0 again.

Next is the Format ID, which is not present for all Application IDs. It is used to give further information about the contents of the packet, typically giving the data rate or instrument mode.

Following is the optional time field, whose length and content depends on the needs of the ground data processing system. It is always included in packets with a Seq # of 0, and generally also when Seq # modulo some specified power of two is zero, *e.g.*, “every 16th packet.” Time is also included in the first packet of a new “set” of data, such as the start of a record mode or switching a data stream from deselected to selected, indicating a break in the steady collection of data. Time is also included in the first packet after any short packet caused by the playback process reading into the multi-use buffer an amount of data which does not break up into an integer number of whole packets for any given instrument. In the example shown, the time field contains the minor frame count and the least significant two and a half bytes of the RIM count. The time used is the s/c clock value at the time when CDS picked up the first byte(s) of the data included in the packet, unless specifically stated otherwise.

If the time field is missing, the Format ID is in the most significant nybble of a byte by itself. The four bits of the low-order nybble are “don’t care.”

MAG RT Data

Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	Data1	Data2	...	Data 90
1	7	9	7	4	28	16	16	...	16
	MAG1				½R-R-R-mf				

5.1. SSI

5.1.1. SSI Data Pickup

In addition to continuing to supply 94.56 kb/s for HIM, 768 kb/s for IM8 and AI8, and 311.04 kb/s (372.48 including R/S) for IM4, SSI will supply 77.76 kb/s (93.12 including R/S) for HCA and 94.56 kb/s for HMA and HIS. It also will continue to supply 144 b/s (12 bytes per mf) status data for LPW.

5.1.2. SSI Processing in CDS

CDS does a lot of processing for SSI data. In all cases it deletes prepare-cycle data, header and fill. The location of these data is mode-dependent. CDS deletes the Reed-Solomon code from the BARC-compressed modes HCA and IM4. CDS has an editing process for ~~uncompressed~~ SSI science imaging data (windowing) and two types of compression (in AACS): an 8x8 ICT compressor at various commandable target compression ratios and a lossless compressor. To raw SSI data CDS can apply either compressor~~ or neither~~, with or without windowing. Windowing consists of saving only one sequence/PB-table-specified rectangular region of the image for compression and downlink. The window will be a multiple of 8 columns and 8 rows. Another option is a fixed-size “truth window” of losslessly compressed data in an otherwise ICT-compressed image. BARC-compressed data can be windowed vertically only, and no further compression can be done.

The ICT compressor normally operates fast enough to keep up with the 7.68 kb/s tape playback. CDS writes 10 8x8 blocks into AACS memory each minor frame, 5 in RTI 0 and 5 in RTI 1. AACS starts processing in RTI 1, but because of timing considerations is guaranteed not to start processing the second 5 blocks until RTI 2. ICT can accomodate playback-table-selectable target compression ratios ranging from 2:1 to 80:1. CDS will load into AACS optimized Q tables and Huffman tables indicated by the playback table.

The specification of processes and parameters involves the usual PB table algorithm and parameter fields, plus a special second 7-byte PB table entry. The algorithm field of the first entry has two bits to specify the compression type and 1044-byte Huffman table to use (ICT-atmosphere, ICT-satellite, lossless, BARC), two bits to point to which of the 3 128-byte Q tables should be used, and one bit to specify the zigzag pattern. The parameter field is the Q factor.

The extra PB table entry specifies the windowing and truth window (if present). The first byte give the X coordinate of the pixel start of the window to be processed, the next byte gives the Y coordinate, the next byte gives the picture width, and the following byte the height all in 8-pixel units. To indicate that no windowing is being done, the first 4 bytes should be (0, 0, 100, 100)~~ for 800x800 modes~~. The next two bytes give the X and Y coordinates of the upper left corner of the losslessly-compressed 96 x 96 truth window (for ICT-compressed images only) in the same units. If no truth window is desired, both bytes should be 255. Note that there is one unused byte.

In order to reduce the quantity of downlink data, none of the ground-specified (sequence or playback table) processing parameters are included in the data. The science team/MIPL is responsible for using the appropriate parameters from predicts for processing the data.

Each packet contains data from only one image. Except for the grouping of 8 lines of data together for the ICT, data are processed and returned in the same order in which they went onto the DMS. Time is included in the first data packet for each image and every 64th packet.

The data from the ICT and lossless compression consist of a up to 100 slices of compressed data, each slice containing data from 8 lines (up to 100 8x8 blocks). The data for each slice begin with a 25-bit synch code followed by a 7-bit counter giving the slice number (0-99) of the data. Each block of data consists of Huffman codes from the 8x8 region following a 1-bit flag which is set to 1 if the attempted compression resulted in expansion. In the case of expansion, for ICT the data returned are truncated at 64 bytes, while for lossless compression the raw data are returned. The data from the compression blocks within each slice are packed into bytes with no gaps, but with up to 7 bits of zero fill after the last block in the slice. The data field can be split across packet boundaries, but the sync code can not. Each packet for the BARC-compressed modes contains data from one line following a zero-based two-byte line count.

Each image other than those taken for opnav has one special packet with housekeeping measurements associated with that image, along with AACS data (scan platform RA, DEC, and TWIST plus rotor CLOCK) corresponding to the time the SSI shutter was opened, obtained from the LPW recorded with the imaging data, along with the time for the image. For most modes the housekeeping data consist of two bytes per mf collected for 13 mf's, edited into a 15-byte packet. For the 2 1/3 second mode (AI8) it is four bytes per mf for 3 mf's, edited into a 13-byte packet. For details on this and for the subset of the housekeeping data which is put into the engineering telemetry see 3-280~~, paragraphs A2.12.1.1 and A2.12.1.2~~.

SSI imaging and status data can be deselected together from the playback data stream; that is, either both are returned or neither is.

SSI Imaging Data, Non-BARC-compressed Modes

Time inc.	App ID	Size	Seq #	Image #	(Time)	Data1	Data2	...	Data ≤511
1	7	9	7	4	28	8	8	...	8
	SSI1				½R-R-R-mf				

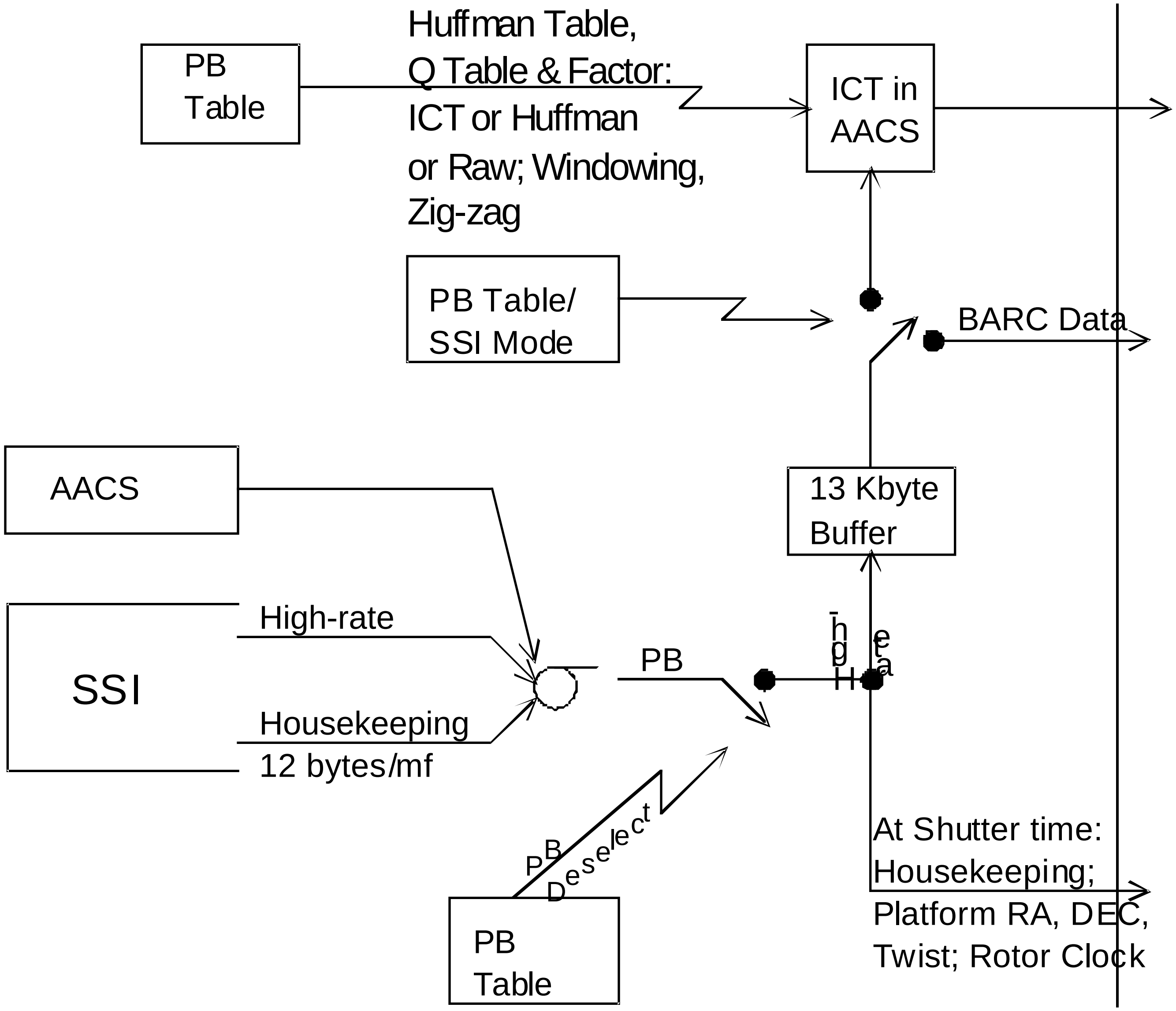
SSI Imaging Data, BARC-Compressed Modes (HCA, IM4)

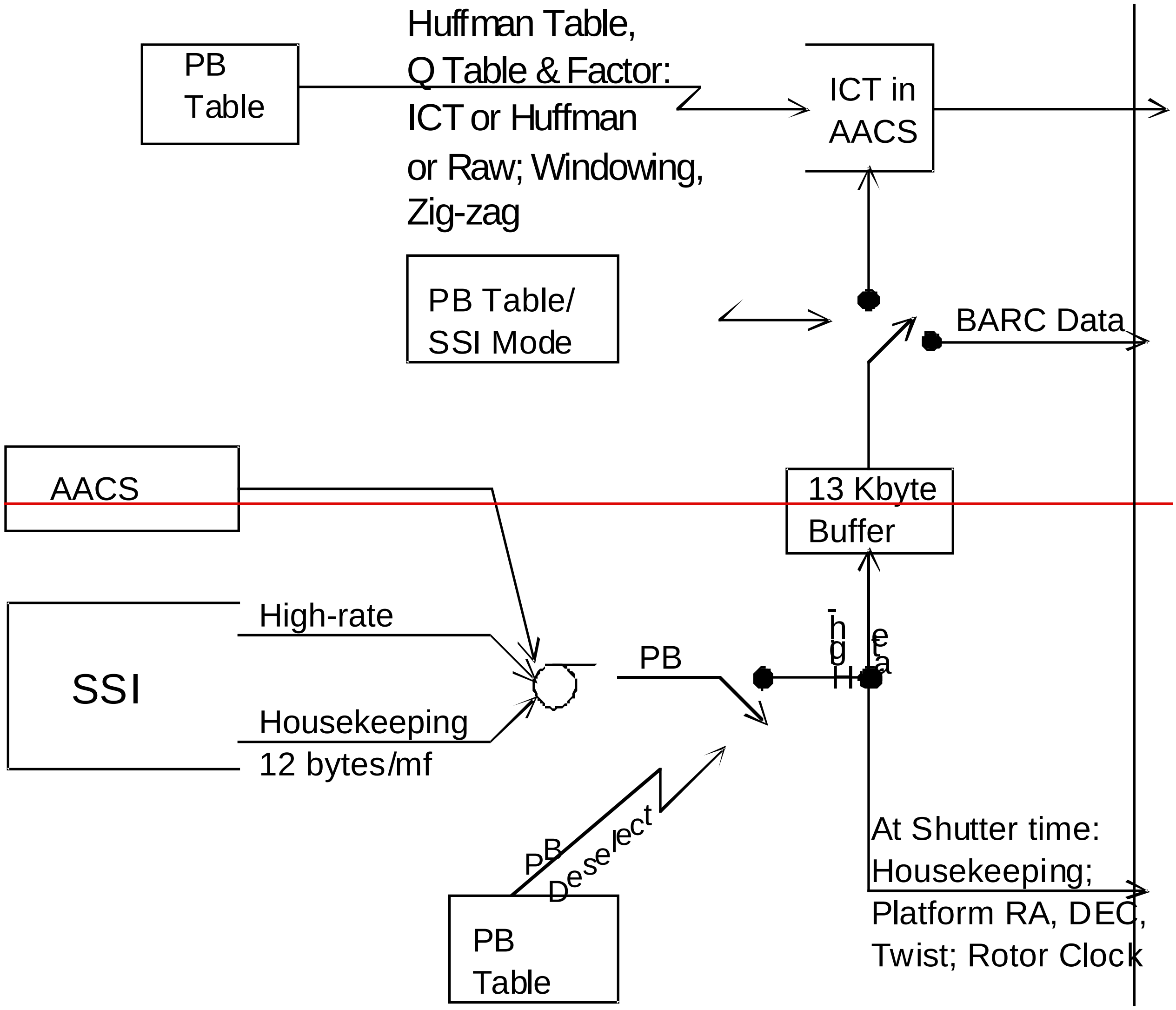
Time inc.	App ID	Size	Seq #	Image #	(Time)	Data1	Data2	...	Data 326~~4~~
1	7	9	7	4	28	8	8	...	8
	SSI2				½R-R-R-mf				

SSI Housekeeping

Time inc.	App ID	Size	Seq #	Image #	Time	Data1	Data2	...	Data 13 or 15
1	7	9	7	4	28	8	8	...	8
1	SSI3				½R-R-R-mf				

SSI Science & Housekeeping Data Flow in CDS


PB Table
Huffman Table, Q Table & Factor: ICT or Huffman or Raw; Windowing, Zig-zag
ICT in AACS
PB Table/ SSI Mode
BARC Data
AACS
13 Kbyte Buffer
SSI
High-rate
Housekeeping 12 bytes/mf
PB
PB Deselect
PB Table
At Shutter time: Housekeeping; Platform RA, DEC, Twist; Rotor Clock


PB Table
Huffman Table, Q Table & Factor: ICT or Huffman or Raw; Windowing, Zig-zag
ICT in AACS
PB Table/ SSI Mode
BARC Data
AACS
13 Kbyte Buffer
SSI
High-rate
Housekeeping
12 bytes/mf
PB
PB Deselect
PB Table
At Shutter time: Housekeeping; Platform RA, DEC, Twist; Rotor Clock

5.1.3. SSI Opnav Processing in CDS

In addition to the science processing for SSI data described above, there is also a different set of algorithms used for optical navigation. These involve finding and returning small windows around the limb/terminator (LT) of one extended body and using its computed position plus uplinked offsets to find and return up to five rectangular windows presumably containing star images. There is also a starless mode in which only LT data are returned, either of multiple bodies for opnav, or for precisely determining the location of a target body in scan platform coordinates to support observation by other instruments. The algorithms can work either with recorded images or by doing realtime readouts of the SSI, using special commands to discard or read out a specified number of lines. The basic routine starts with finding the extended target body by doing 10 to 50 cycles of skip *n* lines and read *m* lines. For each set of *m* lines read, CDS scans each line looking for the two LTs presumed to be present. The first LT is marked by the finding of *i* consecutive pixels greater than a specified high threshold, and the second by finding *a* pixel below a specified dark threshold. In each case the (x,y) location of the LT and the values of the 13~~16~~ pixels surrounding it in the line are saved for downlink (the y value for the second LT data set of a pair is not downlinked, of course, since the two pixel sets are on the same line). The two pixel segments can overlap.

CDS will use locations in the top 1/3 of the LT to determine that the proper bright body has been found (by doing a curve fit to the expected shape), and then to compute the location of the upper cusp of the body. Uplinked offsets from this cusp location determine the locations of the 0 to 5 rectangular windows surrounding star images which will be read out and downlinked along with the LT data. The stars must be below the top 1/3 of the extended body, and the line numbers of the star windows must be non-overlapping, though they may be contiguous. For example, two 33x15 windows could be placed vertically adjacent to each other, making in effect one 33x30 window. The fact that the data from each star window must fit into one packet limits the area of the window to no more than 506 pixels (511 bytes per packet minus 5 bytes overhead), and if the opnav command asks for any more the data will be truncated.

Note that for RT opnav, special commanding of SSI is needed to read out and to skip the specified lines from the CCD. Recorded opnav takes some fancy DMS control to read successive sections of the picture into the multi-use buffer (using many start-stop cycles unless the buffer has much available room), with lines of imaging data then being read from the buffer by the opnav command.

One packet type will be used for the LT data and one for the star windows. No more than six packets will be used for any instance of opnav. The packets can be divided between extended body LT and star windows in any manner, as long as there is at least one LT packet. The packets are turned into VCDUs in the priority virtual channel.

The time field is included in all opnav packets, and contains 1) for RT the spacecraft clock at the time the command starts to execute in the LLM, three mf's after execution of the functional command in the HLM, and 2) for recorded opnav the image time from the PB table.

The locations in the 800x800 image space of the pixel data for the LT packets are included in the packets as indicated in the table below, with multiple datasets per packet:

2-byte pixel #	2-byte line #	Left LT pixels	2-byte pixel #	Right LT pixels

The format of the location data in the star window packets is given in the table below, with one dataset per packet:

2-byte pixel #	2-byte line #	1-byte width	≤506 pixels

Note that no SSI housekeeping data are returned with opnav images.

RT Opnav Extended Body Limb/Terminator Data

Time inc.	App ID	Size	Seq #	Time	Data1	Data2	...	Data≤511
1	7	9	7	32	8	8	...	8
1	OPN1			R-R-R-mf				

RT Opnav Star Window Data

Time inc.	App ID	Size	Seq #	Time	Data1	Data2	...	Data ≤511
1	7	9	7	32	8	8	...	8
1	OPN2			R-R-R-mf				

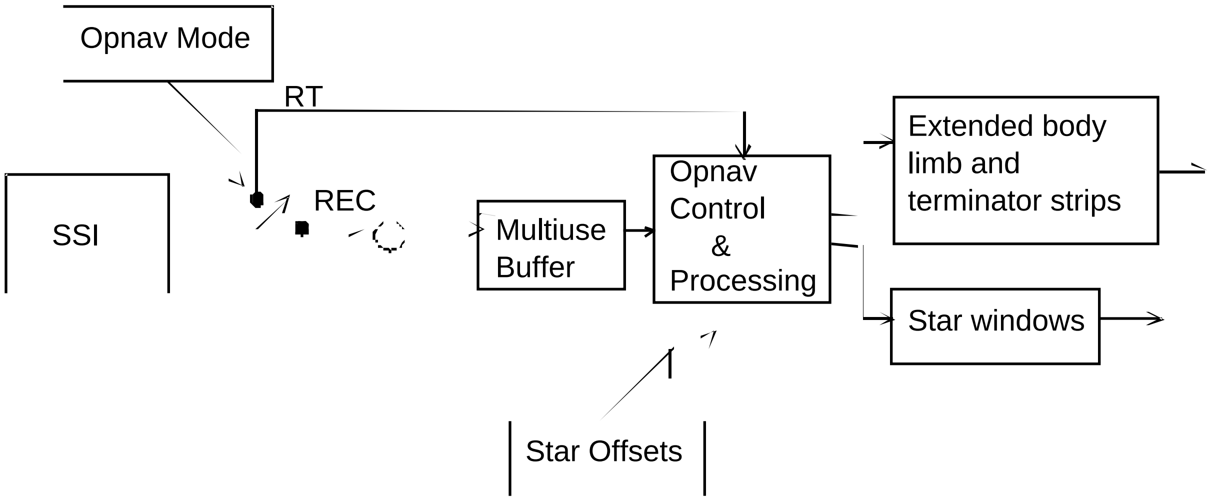
PB Opnav Extended Body Limb/Terminator Data

Time inc.	App ID	Size	Seq #	Time	Data1	Data2	...	Data≤511
1	7	9	7	32	8	8	...	8
1	OPN3			R-R-R-mf				

PB Opnav Star Window Data

Time inc.	App ID	Size	Seq #	Time	Data1	Data2	...	Data_<511
1	7	9	7	32	8	8	...	8
1	OPN4			R-R-R-mf				

SSI Opnav Data Processing in CDS


Opnav Mode
RT
SSI
REC
Multiuse Buffer
Opnav Control & Processing
Extended body limb and terminator strips
Star windows
Star Offsets

5.1.4. Changes in SSI Commands

A 15 1/6 second rate is being added to the mode rate coding.

The 36IP command is being modified to include a READOUT DISABLE function and a FLOOD/ERASE DISABLE utilizing two previously spare bits.

The 36IM command is being modified to include a CONTIGUOUS/SAMPLING READOUT selection and an HGA/LGA MODE selection utilizing two previously spare bits.

The opnav functions to read n lines and skip m lines will be accomplished using four new commands (see 3-290 for details):

36ONSV	SKIP VARIABLE = 5A, CV1, CV2
36ONS1	SKIP ONE = 5B
36ONRV	READ VARIABLE = A4, CV1,CV2,CV3
36ONR1	READ ONE = A5

5.2. NIMS

5.2.1. NIMS Data Pickup

The NIMS RT and record data streams both are derived from the current 96 bytes per RTI pickup.

~~The NIMS RT pickup is 32 bytes per RTI, and the pickup is made whether or not NIMS RT is selected. . The data are read from a dedicated location in NIMS memory.~~

For record modes, in addition to supplying 11.52 kb/s for existing modes MPW, HIM, HPW, IM8, AI8, and IM4, and for new mode HMA, NIMS also supplies 2.592 kb/s (plus 36 b/s status) for LNR, and 6.168 kb/s for LPU.

The three bytes of NIMS status data per mf will continue to be collected as they are now, with four of the subcommutated values returned in fixed engineering. ~~but will always be discarded and never downlinked.~~

5.2.2. NIMS Data Processing in CDS

5.2.2.1. NIMS Realtime

The RT NIMS data as read from the instrument have the same format as explained in detail in section 5.2.2.2, but for RT processing the only important fields are the length byte (the first byte of the 96 per RTI, the 7 LSBs of which tell CDS how many data bytes to keep), the Half Minor Frame Count (HMFC), and the instrument data itself starting at byte 12. When NIMS RT becomes selected, CDS will wait for the next HMFC = 0 in the data stream, and immediately start keeping the indicated number of data bytes from each RTI's data set, continuing until data have been read for 12 mf's or until the 2048-byte buffer is full, whichever comes first. CDS will transmit the saved data, up to 4 packets of up to 511 bytes each (Note that if fewer than 5 data bytes remain, no packet will be generated). The data lengths and the rest of the header each RTI will not be transmitted, only the appropriate number of bytes from the data field. Ground processing will unpack the data according to a model of the commands uplinked.

When transmission is complete, CDS will wait for the *next* HMFC=0 after the pause required to keep the data rate at or below 10 b/s (based on the size of the previous packet), and then repeat the process, continuing in this way until NIMS realtime is deselected.

That is, CDS is responsible for enforcing the RTS data allocation, and new data are kept only after the appropriate pause *and* HMFC=0 has occurred.

~~Each 32-byte RT pickup consists of a flag byte, a length byte specifying the number of valid data bytes, and 0 to 30 data bytes followed by fill if needed, as illustrated below:~~

~~Flag~~	~~Length~~	~~Data / Fill~~

~~The format of the flag byte is given below, with a 1-bit being active:~~

~~MSB~~ ~~LSB~~

~~Start Cycle~~	~~End Cycle~~	~~End Packet~~	~~Start Packet~~	~~Long Cycle~~			

~~The normal situation is that a telemetry packet will contain one to several instrument data cycles. There is one mode, however, in which a cycle must be split into two packets, a situation indicated by having the Long Cycle bit set.~~

~~CDS reads NIMS RT at 32 bytes per RTI, and outputs at 10 b/s. The seeming discrepancy in rates is handled by having CDS throw away most of the data, under the control of the length byte and the bits in the flag byte. When CDS is processing NIMS RT it starts by waiting for a Start Packet flag. It then starts looking for a Start Cycle flag, starting with the current 32-byte block. When it sees one it starts assembling data for a telemetry packet, using the number of bytes indicated in the length byte of the block. and it continues assembling data until it sees an End Cycle flag. If the same block includes an End Packet flag, the packet is sent off to the VCDU maker. Otherwise CDS waits for the next Start Cycle flag to repeat the data assembly process.~~

~~If the nominal 264-byte packet length is passed without an End Packet flag, and the Long Cycle flag was not set, CDS will output a 265-byte packet--the only time a NIMS packet is greater than 264 bytes. CDS will then wait for the End Packet flag before looking for the next Start Packet flag, and will put the time field in the next packet, since the normal ground algorithm for inserting time may not properly function.~~

~~If the Long Cycle flag is set at the start of a cycle, CDS will break up the next 528 data bytes into two packets, ignoring other flag bits, and will then wait for the next Start Packet flag.~~

~~A new NIMS command is being added to tell the instrument how many grating cycles to skip between those kept and how many cycles to put in a packet, so that it can properly set the bits in the flag byte. The proper use of this command and the proper operation of~~

~~NIMS s/w are totally responsible for limiting NIMS to its 10 b/s RT allocation. CDS makes no checks.~~

When the appropriate RT mode is selected, and NIMS is not deselected from RT, CDS will continue to process the RT data stream even if a REC mode is entered which contains NIMS recorded data.

The time will be included in two situations: the first packet of a new data set (after a period of NIMS RT deselect) and every 16th packet. Because of the new way NIMS RT data are taken, the mf/2 byte will always be zero. ~~Note that the mf/2 in the TIME field is a normal minor frame count, but with the high-order bit set if the data taking started in the second half of the minor frame, *i.e.*, in RTI 5-9.~~

NIMS RT Data

Time inc.	App ID	Size	Seq #	(Time)	Data1	Data2	...	Data ≤511~~265~~
1	7	9	7	32	8	8	...	8
	NIMS1			R-R-R-mf/2				

5.2.2.2. NIMS Record/Playback

The format of the NIMS PB table entry is given below, along with the associated tables used in NIMS PB processing:

Algorithm Byte:

MSB							LSB
Select/ Deselect	λ Edit Table Dictionary Entry					Threshold	

Parameter Byte:

MSB							LSB
Dark Removal	Reference Value Vector	Lossy Compression Options (Rate Control)			Reference Countdown		

The λ **Edit Table Dictionary Entry** selects an entry from the 30-entry (0-29) dictionary at the head of the λ Edit Table associated with the current playback table. Each dictionary entry is an index into the 75x3 array of table entries. The table entries consist of a 7-bit repeat count followed by a 17-bit detector mask. The entry pointed to begins a group of entries, which ends at the start of the next group or at the end of the table. The group controls wavelength selection for a particular playback request.

λ Edit Table--each cell a byte (one of four, each associated with a PB table)

DE1	DE2	DE3	DE4	...	DE29	DE30

adr=0	SE1	se1	se1	
adr=3	SE2			
adr=6	SE3			
adr=9	SE4			
	...			
	...			
adr=216	SE73			
adr=219	SE74			
adr=222	SE75	se75	se75	Chksum

Each of the four 256-byte edit tables is linked to one of the Playback Tables via slot number.

CDS will use the 17-bit detector mask in the entry (throwing away data from a detector unless its bit in the mask is a 1) for the number of half-minor-frames indicated in the repeat count field of the entry. It will then move down to the next entry and use it until its repeat count has expired. CDS will continue going down the entries until it sees a Grating Cycle Start flag in the instrument data or until the group has ended. It then resumes at the beginning of the group.

The **Threshold** field, if non-zero, points to one of three 17-entry tables (each entry is two bytes, and the low-order 10 bits are used) which give the values below which data values will~~should~~ be removed from~~suppressed by a mask in~~ the data stream~~replaced with zero~~. The presence or absence of data in each 10-value compression block will be indicated by a block bit and a 10-bit mask preceding the data. If no values are to be sent, only a 0 block bit is sent. Otherwise a block bit of 1 is sent, followed by the 10-bit mask indicating which data values are included.If the field is zero, no thresholding is done and no mask is sent.

Thresholding Table (one of three, selected by PB table entry)

thr1	thr2	thr3	thr4		...		thr17

The **Dark Removal** bit and **Reference Value Vector** bit no longer serve their original purposes, but are reserved for other possible purposes.

~~The **Dark Removal** bit tells CDS whether to use the gain flag value in the instrument data to select one of four 17-byte tables for dark removal. Dark removal consists of subtracting the value in the table from the appropriate detector data, and is done before thresholding. The table values will be prescaled before being uplinked to the CDS by subtracting a constant (approximately 15) from each value, in order to insure that DNs will not be negative after dark removal.~~

~~Dark Removal Table (one of four, selected by gain setting in data)~~

~~dark1~~	~~dark2~~	~~dark3~~	~~dark4~~		~~...~~		~~dark17~~

~~The **Reference Value Vector** bit tells CDS whether each selected channel will begin with a reference value in each RIM-rollover packet.~~

The **Lossy Compression Options** field, if non-zero, points to one of seven tables, each with two two-byte entries, giving the minimum and maximum values of acceptable compression (units are 16-bit words~~TBD~~). If the level of compression is worse than the maximum value at RIM rollover, then 4 fewer mirror positions will be selected for the next RIM. At least 4 mirror positions will be kept. When/if the compression improves so that the level drops below the minimum, 4 mirror positions will be restored. If the field is zero, no lossy compression is done.

The **Reference Countdown** field is 0 for no compression, 1 for Rice compression with reference values inserted at the beginning of each mirror scan, and 3 for Rice compression with reference values inserted at the start of the playback request and at RIM rollover. ~~points to one of eight one-byte table entries which are used by the Rice compressor to tell how many data points can be skipped without putting in a new reference value.~~

Each 96-byte NIMS data set starts with 11 bytes of header/housekeeping data, with format given below.

	MSB							LSB
Byte1	HMF	Contents Length						
Byte2	Spare~~Hskpg Length~~			Cycle	Spare~~Gain Flag~~		Mir.D	M
Byte3	M	M	M	M	M	M	M	M
Byte4	M	M	M	M	M	M	M	M
Byte5	Half Minor Frame Count							
Byte6	Spare~~Housekeeping 1~~							
Byte7	Spare~~Housekeeping 2~~							
Byte8	Spare~~Housekeeping 3~~							
Byte9	Spare~~Housekeeping 4~~							
Byte10	Spare~~Housekeeping 5~~							
Byte11	Spare~~Housekeeping 6~~							

In the table HMF is 1 for RTI 0 and 5 and is 0 otherwise (marking the start of a Half Minor Frame), Contents Length is the sum of the header length (2 if HMF=0 or 5 otherwise) ~~plus the housekeeping length (0-6)~~ plus the data length (0-85), ~~Hskpg Length is the number of valid housekeeping bytes in the packet,~~ the Cycle flag is 1 at a Grating Cycle Start~~the start of a grating cycle~~, Mir.D is set to 1~~0~~ to indicate downward mirror movement, the M bits constitute the detector mask, and the Half Minor Frame Count goes from 0 to 181 during each RIM. The Cycle flag, ~~Gain Flag,~~ Mir.D, M bits, and Half Minor Frame Count are valid only when HMF is 1.

~~The Gain Flag is associated with the instrument gain state according to the following table:~~

~~Gain Flag~~	~~0~~	~~1~~	~~2~~	~~3~~
~~Gain State~~	~~2~~	~~4~~	~~3~~	~~1~~

The header ~~/housekeeping~~ block is followed by 0-85 data bytes.

CDS double buffers the 96-byte data sets as read from the instrument, packs the valid header~~, housekeeping,~~ and data bytes in an intermediate buffer (for rate averaging), and then sends the data to the DMS at the steady rate appropriate for the record mode. If NIMS sends too much data (that is, if the length fields in the data tell CDS to keep too many data bytes) and buffer overflow occurs, CDS will indicate the fact using one of the Error Flag bits in the Subpacket Format 1.

On playback, for each half minor frame of NIMS data CDS will build a table (possibly sparsely-populated) of 20 mirror positions by 17 detectors~~, plus up to 30 bytes of housekeeping data~~. In building the table CDS takes into account the different mirror position offsets of the detectors--0 for detectors 1, 5, 9...; 1 for detectors 2, 6, ...; 2 for 3, 7, ...; and 3 for 4, 8, ...

CDS then determines which additional data to reject in its λ editor, which does a logical AND of the detector map in the data header with one in the current λ edit table entry, keeping only the λ's represented by a 1 bit.

~~In order to make the compression more efficient, CDS will optionally perform dark suppression/normalization by subtracting a detector-dependent value from each number in the table, and will optionally perform thresholding, which is done by suppressing the thresholded values using a bitmask in the data stream.simply zeroing each table value below the threshold value.~~

CDS then sends data to the Rice compressor, 10~~J~~ Mirror Positions at a time, for each kept detector.~~, where J is determined from the number of mirror positions N according to the following table:~~

~~N~~	~~4~~	~~8~~	~~12~~	~~16~~	~~20~~
~~J~~	~~4~~	~~8~~	~~12~~	~~8~~	~~10~~

~~If the compressor does not perform as well as expected, CDS will throw out 4 mirror positions alternately from each end of the scan, making a new choice each RIM rollover. If it is performing better than expected, 4 mirror positions will be restored.~~

~~Since the Rice algorithm uses differences of successive values of each detector as one means of compressing the data, the reference values will optionally be returned once per RIM for robustness.~~

CDS will terminate the current packet and begin a new one (containing the time field) each time the RIM count of the recorded data rolls over.

The packet consists of header information followed by one or more subpackets, each of which has one of the four formats below:

NIMS PB Subpacket Format 0--Zero Count

	MSB							LSB
Byte1	0	0	Zero Count					

where Zero Count is the number of consecutive half-minor-frames without any data~~ or housekeeping~~, up to a maximum of 63.

NIMS PB Subpacket Format 1--Normal Data

MSB							LSB
0	1	Length_{hi}					
Length_{lo}			Mask Present	Error Flags~~Gain Flag~~		Mirror Direction	(Mask_{hi})
(Mask_{mid})							
(Mask_{lo})							
Data_1							
...							
Data_n							
~~Housekeeping_1~~							
~~...~~							
~~Housekeeping_m~~							

where Length is the total number of ~~data+housekeeping~~ bytes in the subpacket (including Length and, if present, Mask)~~half-minor-frame~~, Mask Present is 1 if the 17-bit Detector Mask is present, ~~Gain Flag and~~ Mirror Direction comes from the instrument data, and Mask has 1 bits for those detectors present in the data, Data is the data bytes for a half minor frame, which will optionally be Rice-compressed~~, and Housekeeping is the housekeeping data, whose length is given in the Format 3 subpacket each RIM rollover and at the start of each observation~~. The high-order error flag bit indicates data overflow during the recording of LNR or LPU data, and the low-order bit indicates underflow, both referring to earlier data. NOTE: if the data would extend past the end of a 511-byte NIMS packet, the excess will be put into a single Format 2 subpacket without splitting the two subpacket header bytes containing the subpacket length.

NIMS PB Subpacket Format 2--Remaining Data

MSB							LSB
1	0	~~Remaining~~ Length_{hi}					
~~Remaining~~ Length_{lo}			Spare				
Bytes_1							
...							
Bytes_{Length}							

where ~~Remaining~~ Length is the total ~~of the~~ number of ~~data, housekeeping length, and housekeeping~~ bytes for the half minor frame ~~which did not fit~~ into the ~~preceding Format 1~~ subpacket (including Length), and Bytes is those bytes.

NIMS PB Subpacket Format 3--Observation Start & RIM Rollover

MSB							LSB
1	1	Diagn. Dump Flag~~Spare~~	Rate Control RTI Mask for Downward Mirror Scan in RIM's data				
Spare	Spare	Mir. Dir	Commanded RTI Mask ~~Housekeeping Length~~				

where the RTI Masks are ~~is a~~ 5-bit masks whose bits (left to right) indicate blocking of data from individual RTIs in a mirror scan until the next RIM rollover (a 1 in bit 4 blocks the first~~top~~ 4 mirror positions of the scan, a 1 in bit 5 blocks the next 4, etc.). The Rate Control Mask indicates which RTIs have been blocked for the following RIM by the rate control mechanism (lossy compression). The Commanded RTI Mask is reconstructed by CDS from the data lengths in the data pickup and derives from the 37MB commands sent to the instrument. The Mirror Direction bit (down = 1) is that associated with the First Commanded RTI Mask. Note that the Rate Control RTI Masks for the two directions are mirror images of each other, but the Commanded RTI Masks need not be.~~ and the Housekeeping Length is the number of bytes of housekeeping data in each half minor frame until the next RIM rollover.~~

That is, each subpacket ~~either~~ contains all or part of the data from one half-minor-frame (Format 1 or 2), or gives some information on how to process the data (Format 3), or else tells how many consecutive half-minor-frames there were with no data (Format 0).

Subpackets are not split across packet boundaries, with the exception that the data in a Type 1 subpacket can be continued in a Type 2 subpacket in the next packet~~, with the length field of the Type 1 subpacket including the data of both, as explained above~~.

CDS sends the packet to the PB virtual channel to be made into VCDUs. The time will be included in three situations: the first packet of each set of PB data and those packets which start at a RIM rollover, and when the sequence number rolls over. Even when the full time is not included, the mf/2 byte will be included.

In order to make it easier on the ground to keep track of NIMS data from each mode independently of whether or not it was compressed, NIMS2 and NIMS5 packets share a single sequence number counter, as do NIMS3 and NIMS6, and NIMS4 and NIMS7 (that is, there are three independent sequence number counters for NIMS playback). If the PB table specifies Rice compression, the data are returned Riced even if expansion results.

CDS stores 1024 bytes of wavelength tables for the control of its PB editing, with 256 bytes being active at any given time. The 1024 bytes are divided into 4 λ Edit Tables, each placed into a slot, and each of which is linked to a playback table. Because the PB table is used while the tape is being read, and the λ Edit Table is used when the data are actually being processed (by which time a new PB table may have been already read in), CDS has to be clever to be sure that the NIMS data are processed with the correct λ Edit Table. CDS stores the slot number of the active PB table with the raw NIMS data as they go into the buffer. When CDS later edits the data, if that slot is not active, CDS will move the λ Edit Table in that slot to the active slot.

NIMS 11.52 Kb/s PB Data--Uncompressed

Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	MF/2	Data1	Data2	...	Data≤511
1	7	9	7	4	20	8	8	8	...	8
	NIMS2				_R-R-R	mf/2				

NIMS 6.168 Kb/s PB Data--Uncompressed

Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	MF/2	Data1	Data2	...	Data≤511
1	7	9	7	4	20	8	8	8	...	8
	NIMS3				_R-R-R	mf/2				

NIMS 2.592 Kb/s PB Data--Uncompressed

Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	MF/2	Data1	Data2	...	Data≤511
1	7	9	7	4	20	8	8	8	...	8
	NIMS4				_R-R-R	mf/2				

NIMS 11.52 Kb/s PB Data--Rice-compressed

Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	MF/2	Data1	Data2	...	Data≤511
1	7	9	7	4	20	8	8	8	...	8
	NIMS5				_R-R-R	mf/2				

NIMS 6.168 Kb/s PB Data--Rice-compressed

Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	MF/2	Data1	Data2	...	Data≤511
1	7	9	7	4	20	8	8	8	...	8
	NIMS6				_R-R-R	mf/2				

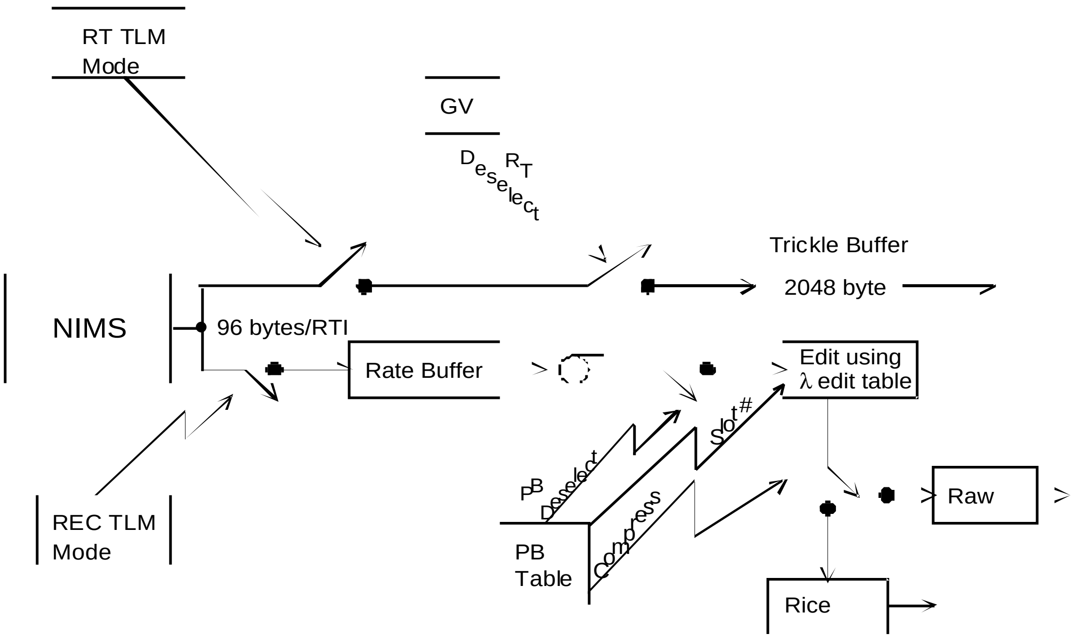
NIMS 2.592 Kb/s PB Data--Rice-compressed

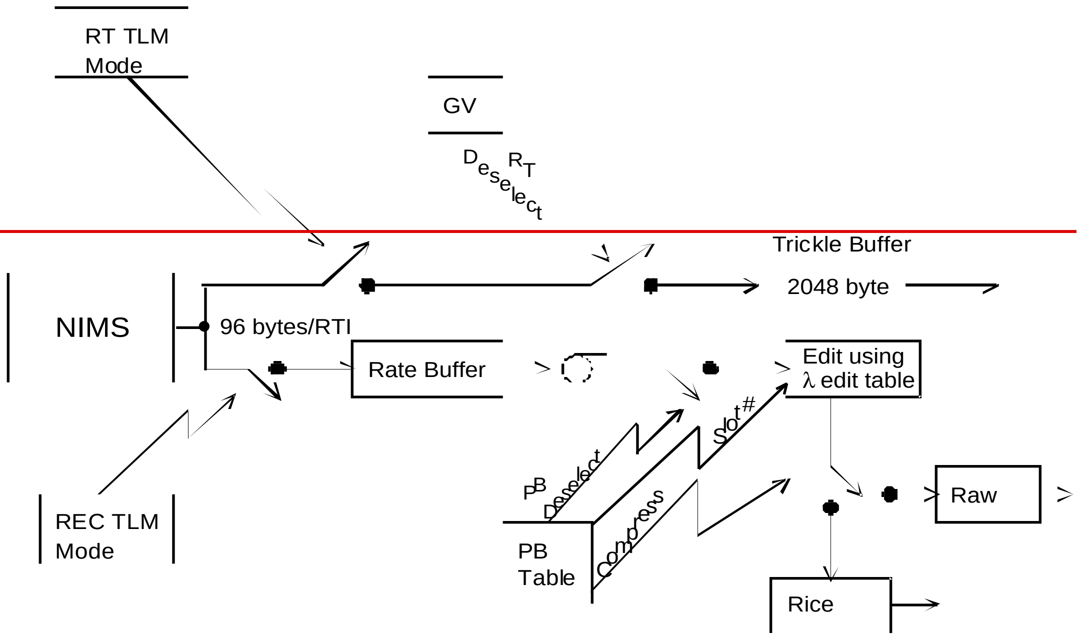
Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	MF/2	Data1	Data2	...	Data≤511
1	7	9	7	4	20	8	8	8	...	8
	NIMS7				_R-R-R	mf/2				

5.2.2.3. NIMS General

Six housekeeping measurements will be picked up from ~~contiguous locations in~~ NIMS ~~memory starting at 1592_{hex}~~ and put into the 91-deck of the engineering telemetry. The first four ~~y~~ are S-1924 (H 19, B 1, M 7), S-1925 (H 20, B 2, M 7), S-1926 (H 21, B 1, M 8), and S-1929~~7~~(H 23, B 1, M 10), where (H *i*, B *j*, M *k*) means NIMS housekeeping word *i*, subcommutated as byte *j* at Mod 91 count *k*. ~~S-1929, and S-1930.~~ The other two bytes are picked up from NIMS memory: S-1931--S/C clock byte 2 at 1596_{hex} and S-1932--S/C clock byte 3 at 1597_{hex}.

NIMS Data Flow in CDS


RT TLM Mode
GV
Deselect RT
Trickle Buffer
2048 byte
NIMS
96 bytes/RTI
Rate Buffer
Edit using λ edit table
Slot #
PB Deselect
Compress
Raw
PB Table
REC TLM Mode
Rice


RT TLM Mode
GV
Deselect RT
Trickle Buffer
2048 byte
NIMS
96 bytes/RTI
Rate Buffer
Edit using λ edit table
Slot #
PB Deselect
Compress
Raw
PB Table
REC TLM Mode
Rice

5.2.3. Changes in NIMS Commands

Two~~Four~~ new command types defined:

37ETB--Edit Table Load. Load up to 125 bytes. Similar to 37DML

~~37RT--RealTime mode--specify map/spectrum mode, mirror position~~

37MB--Mirror position Block for REC modes.

~~37ETS--Edit Spectra~~

S/W loads (editing tables) will be done under sequence control~~needed~~ for switching between RT and REC modes or between different REC data rates. It should be noted, however, that simultaneous RT and REC operation is possible to allow an early look at the recorded data. RT-only observations will in general require RT-specific edit tables.

5.3. PPR

5.3.1. PPR Data Pickup

The LPW pickup for PPR is unchanged at 18 bytes per mf, collected in RTI 5 following a prepare directive in RTI 4, and is also used for LNR and LPU.

The pickup for the PPR burst-to-tape mode BPT is identical to the current LPW pickup--18 bytes per mf, collected in RTI 5 following a prepare directive in RTI 4.

5.3.2. PPR Data Processing in CDS

5.3.2.1. PPR Record/Playback

CDS will block together 20 minor frames of PPR PB data into a precompression packet even if Rice compression is not requested by the PB table. The data are kept in input/time order.

PPR Precompression/Raw Packet Data Field

Data 1 mf 1	Data 2 mf 1	Data 3 mf 1		...		Data 18 mf 1
Data 1 mf 2	Data 2 mf 2	Data 3 mf 2		...		Data 18 mf 2
...	...	...		...		...
Data 1 mf 20	Data 2 mf 20	Data 3 mf 20		...		Data 18 mf 20

If Rice compression is to be done, CDS will consider each of the eighteen data bytes per minor frame to be in a separate channel. CDS will difference and compress the channels independently. The first value of each channel is used by the compressor as a reference value to start off the compression and is returned raw. Subsequent values have the preceding raw value subtracted from them as the first step of the compressor. Compression blocks are 20 values, so there is no carryover of a reference value output from one block to another. The ordering of the data in the compressed packet is:

Compressed Packet Data Field

mf count	Data 1 Reference Value	Data 1 Compressed Data	Data 2 Reference Value	Data 2 Compressed Data	...	Data 18 Reference Value	Data 18 Compressed Data
1 Byte	1 Byte		1 Byte			1 Byte	

~~The pre-compression packets are a fixed size chosen so that, even if compression does not occur, the data will still fit in a single packet without truncation. CDS does not check for data expansion, and sends down a compressed-type packet if the PB table requests compression, even if a raw packet would be smaller.~~
Whenever there are not enough data to make a complete compression block (e.g., end of a tape gulp, end of recorded data, deselect), CDS will autonomously switch to PPR1 for the short packet. In order to make it easier on the ground to keep track of PPR data independently of whether or not it was compressed, PPR1 and PPR2 packets share a single sequence number counter. CDS also switches to PPR1 if attempted compression results in expansion.

The packet is sent to the PB virtual channel to be made into VCDUs. The time will be included in two situations: the first packet of each set of recorded data being downlinked, and every 32nd packet.

PPR Non-BPT PB Data Packet--Raw

Time inc.	App ID	Size	Seq #	(Time)	Data1	Data2	...	Data≤360
1	7	9	7	32	8	8	...	8
	PPR1			R-R-R-mf				

PPR Non-BPT PB Data Packet--Compressed

Time inc.	App ID	Size	Seq #	(Time)	See Compressed Packet Data Field
1	7	9	7	32	
	PPR2			R-R-R-mf	

5.3.2.2. PPR Burst to Tape (BPT)

For BPT, the PPR burst-to-tape mode, CDS collects both PPR and AACS data (scan platform RA & Dec) each minor frame. It edits the data before putting it into the buffer by first deleting repeat PPR data sets (those with bit 24 set) and the AACS data associated with them, then deleting byte two of the kept instrument data sets, and finally adding a 1-byte mf counter in the format given below.

BPT Precompression/Raw Subpacket Data Field

byte 1	mf	ppr1	ppr3	ppr4	ppr5	ppr6	ppr7	ppr8	ppr9	ppr10	ppr11	
	ppr12	ppr13	ppr14	ppr15	ppr16	ppr17	ppr18	Pl RA		Pl Dec		byte 22

CDS puts the instrument data and AACS data into about 12 Kbytes of the SSI buffer. When the buffer becomes almost full it is dumped to tape and then starts refilling with no data being lost. The process is repeated as long as the record mode remains BPT. Any complete frames (14 mf of data) left in the buffer at the end of BPT will be written to tape. It is the responsibility of the ground system to see that a period of five minutes is allowed at the end of BPT before either SSI PB or NIMS PB tries to use the shared buffer, and that a period of 30~~TBD~~ seconds is left before any commands are issued which will move the tape.

On playback CDS then uses one frame (14 of the edited 22-byte AACS/time/instrument packets) as the PPR BPT raw/precompression packet. If Rice compression is to be done, each of the 20 data items is treated as an independent compression block of 14 values. That is, the two AACS measurements are compressed as 16-bit data, resulting in pre-split bits being inserted by the compressor before the reference values, while the mf count and the instrument data are compressed as 8-bit values.

In order to make it easier on the ground to keep track of PPR data independently of whether or not it was compressed, PPR3 and PPR4 packets share a single sequence number counter. If an attempted packet compression results in expansion, CDS will return that packet as PPR3.

BPT Compressed Packet Data Field

mf count	mf Ref value	mf Comp data	ppr1 Ref	ppr1 Data	ppr3 Ref	ppr3 Data
ppr4 Ref		ppr4 Data	ppr5 Ref	ppr5 Data	ppr6 Ref	ppr6 Data
...						
ppr16 Ref		ppr16 Data	ppr17 Ref	ppr17 Data	ppr18 Ref	ppr18 Data
Pl.RA pre-split bits		Pl. RA Ref. value	Pl. RA Comp. data	Pl.Dec pre-split bits	Pl. Dec Ref. value	Pl. Dec Comp. data

CDS then adds header information, and sends the packet to the PB virtual channel to be made into VCDUs. The time will be included in two situations: the first packet of each burst of data, and each time the sequence number rolls over.

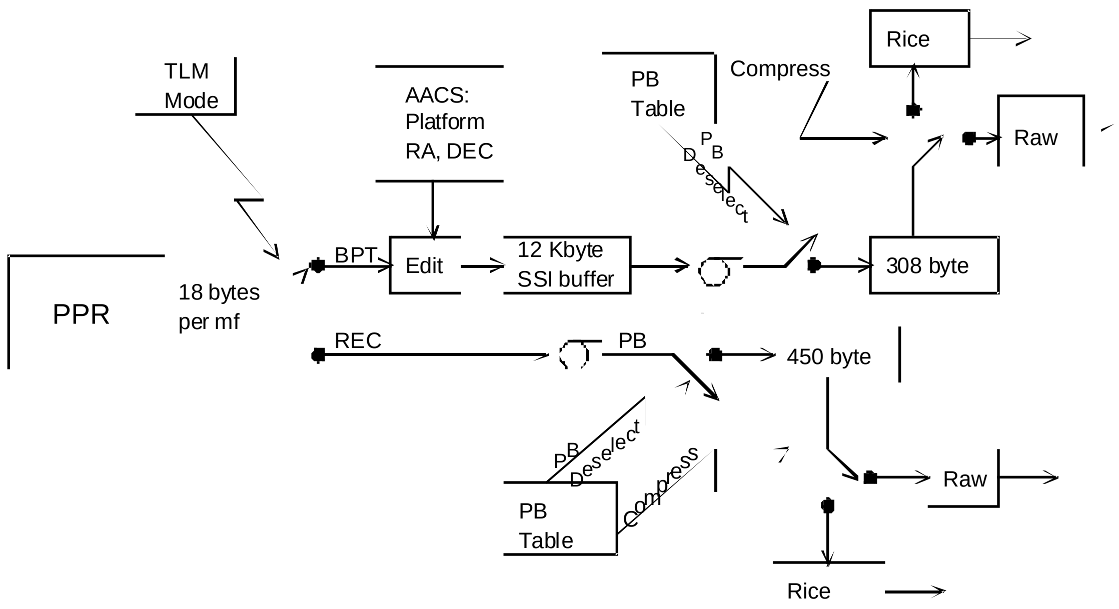
PPR Burst Data (BPT)--Raw

Time inc.	App ID	Size	Seq #	(Time)	Data1	Data2	...	Data≤308
1	7	9	7	24	8	8	...	8
	PPR3			R-R-R				

PPR Burst Data (BPT)--Rice-compressed

Time inc.	App ID	Size	Seq #	(Time)	See Compressed Packet Data Field
1	7	9	7	24	
	PPR4			R-R-R	

PPR Data Flow in CDS



5.3.3. Changes in PPR Commands

None

5.4. EUV

5.4.1. EUV Data Pickup

The EUV RT stream comes from periodic CDS readouts of the instrument's internal 2184-byte buffer. EUV (while it is powered on) is continually summing sensor data into the buffer. CDS periodically reads out the buffer over a period not greater than a RIM, using 56 transactions of 39 bytes each. The readout sets occur at intervals of 2639 mf (29 RIMs) when EUV is at its high rate of 10 b/s or 5369 mf (59 RIMs) at 5 b/s. The transactions take place in RTI 1 and start in minor frame 0 of the RIM which ends the summation period. In mf 57 of the RIM in which the readout occurs CDS tells EUV to zero its buffer by issuing a new 24CLR command.

The EUV REC data pickup remains unchanged from the current 12 bytes per mf in RTI 1. The data are used in LPW whenever EUV is powered on.

Both the RT and REC data streams are available whenever the instrument is on.

EUV does not take part in RRCC.

5.4.2. EUV Processing in CDS

5.4.2.1. EUV Realtime

CDS processing for the EUV RT data is straightforward, since the data part of the packet is one eighth of the 2184-byte buffer, which is read out and cleared as described above, and the only thing to do is add the header information. The time of the start of the summation period is included in the first packet of data from each buffer, and the time of the end of the summation (29 or 59 RIMs later) is included in the last packet from the buffer. The mf is not included in either time because the summation always begins on mf 0 and ends, for the different channels, over an interval starting on mf 0. The packets are formed into VCDU's in the RT virtual channel.

Whenever a 6TMSED/6TMCHG command ~~including an RT mode (other than no-change) is issued,~~ which ~~even if it does not involve~~ switches~~ing~~ the EUV RT allocation is issued, CDS will, if EUV RT is selected, wait for the end of the current RIM, read out the buffer, issue the 24CLR command at mf 57 to tell EUV to zero its buffer, and start a new summation period (appropriate for the new~~possibly for a different~~ rate) at the start of the next RIM. That is, if EUV RT is selected the 6TMSED/6TMCHG acts like the end of a summation period and if EUV RT is not selected the 6TMSED/6TMCHG is ignored except that the new summation interval will be used after the next countdown.

Another way to bring about the end of a summation period is to send a 6UVRT command. This command is acted on whether or not EUV RT is selected. If the command is 6UVRT,PACKET,EUV|BOTH the data from the EUV internal buffer are read and packetized, while if the command is 6UVRT,DISCRD,EUV|BOTH the data are not packetized, but the 24CLR command is sent (at mf 57) to tell EUV to clear the buffer. In both cases a new summation period is begun at the start of the next RIM.

If both a 6TMSED/6TMCHG and a 6UVRT command as described above are issued in the same RIM, the decision whether to packetize or discard the data will come from the 6UVRT command, and the new summation period will come from the 6TMSED/6TMCHG command.

RT deselect has a special meaning for EUV. When EUV is deselected from RT, CDS will keep counting down to the next time when it would perform a readout but will not perform the readout or issue the 24CLR command. CDS will reset the timer so that the next dump opportunity will occur at the same time it would have if the deselect had not been in effect. It will perform a normal packetizing process~~those actions~~ the next time the ~~clock~~ count~~s~~ down timer hits~~to~~ zero while~~and~~ RT is selected. It also will not perform a readout at RT telemetry mode changes during periods of deselect, will not reset the counter, and will not issue the clear command, but will use the new summation interval after the next countdown. ~~will reset the counter.~~

CDS keeps track of whether its EUV buffer has been packetized since it was last filled. If it has not been (because of insufficient downlink rate keeping the multi-use buffer full) at the time when it would otherwise be overwritten with new data, CDS will act as if a deselect were active. That is, it will reset the countdown timer so the next dump time will be the same as it would have been, but will not perform a readout or tell EUV to zero its internal buffer.

EUV RT Data

Time inc.	App ID	Size	Seq #	(Time)	Data1	Data2	...	Data 273
1	7	9	7	24	8	8	...	8
	EUV1			R-R-R				

5.4.2.2. EUV Record/Playback

No editing is performed on the EUV REC/PB data. The packet is simply standard header information followed by 21 12-byte EUV LPW data sets unless fewer contiguous data sets are available. The time will be included in two situations: the first packet of each set of PB data, and every 16th packet. The packets are formed into VCDU’s in the PB virtual channel and the format is:

EUV PB Data

Time inc.	App ID	Size	Seq #	(Time)	Data1	Data2	...	Data≤252
1	7	9	7	32	8	8	...	8
	EUV2			R-R-R-mf				

5.4.2.3. EUV General

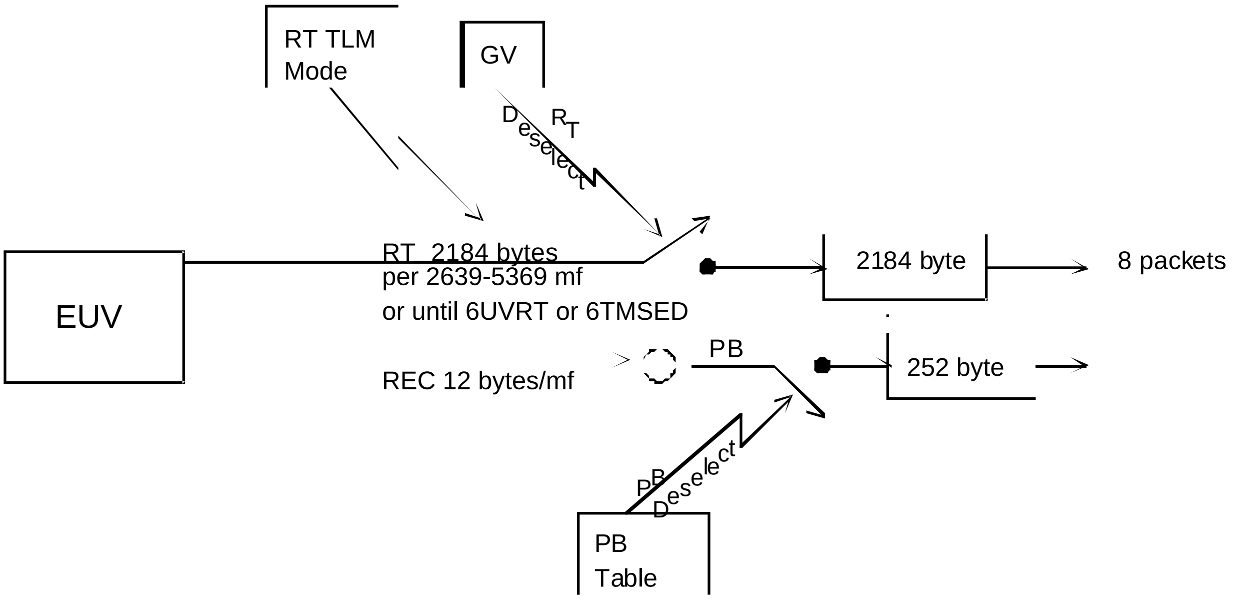
No EUV RRCC.

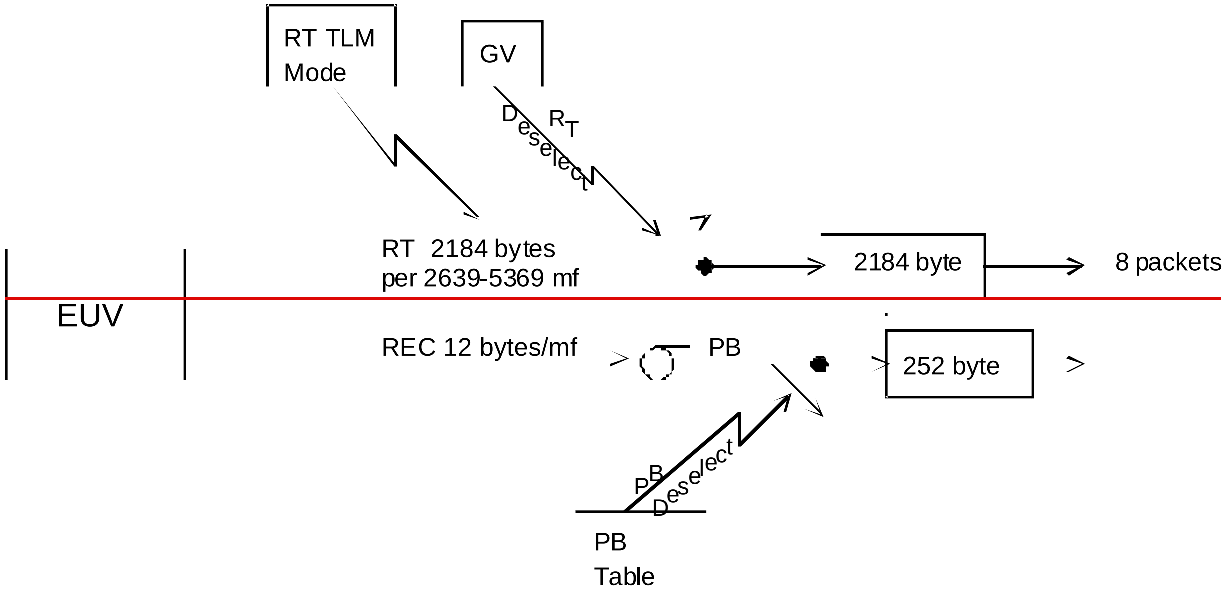
There is no instrument commanding needed for switching between RT and REC modes since EUV can continue the RT mode even when the REC mode is entered, and there is no requirement for a capability to deselect from REC mode.

There are three engineering channels (E-3429, E-3430, E-3431) sub-commutated into E-1680 which each need to be downlinked approximately every ten hours. All instrument housekeeping data are returned in the regular packets.

A period of one RIM~~five minutes~~ must be allowed ~~for clearing the shared buffer~~ when switching between HIC and EUV.

EUV Data Flow in CDS





5.4.3. Changes in EUV Commands

A new 1-byte 24CLR command with a unique address (0FD4 hex) is being added to tell EUV to zero out its buffer. The 24CLR command will supply a non-zero value to the unique address which will tell EUV to zero the buffer.

Memory loads of 750-1000 bytes are required after power on, with the same timing requirements as current memory loads. The loading will be done from the sequence, and not from CDS memory.

Loading of a “fixed pattern noise table” of approximately 128 bytes is~~may also~~ be required.

Also affecting EUV operations is a new CDS command, 6UVRT, which controls the termination of summation periods, as described above.

5.5. UVS

5.5.1. UVS Data Pickup

The RT pickup for UVS is identical to the current LPW pickup--84 bytes per mf, collected in RTI 3, independent of the UVS RT data allocation.

The LPW pickup for UVS is unchanged at 84 bytes per mf, collected in RTI 3, and the same data are also used for LNR and LPU.

~~The RRCC pickup for UVS is identical to the current LPW pickup--84 bytes per mf, collected in RTI 3.~~

The data stream which provides both the RT and REC data is always available whenever the instrument is on.

5.5.2. UVS Data Processing in CDS

5.5.2.1. UVS Realtime

The UVS RT data cycle is 13 mf's, comprising 1092 8-bit measurements at 84 bytes per mf. CDS maintains a 16-bit summation buffer of 2184 bytes and sums the RT data into the buffer (each 8-bit measurement into a unique 16-bit sum, ignoring overflow) for the period appropriate for the UVS RT data allocation--2639 mf (29 RIMs) for 10 b/s, 5369 mf (59 RIMs) for 5 b/s, or 129493 mf (1423 RIMS) for 0.2 b/s.

At the end of the summation period CDS checks to see whether UVS is deselected from RT. If it is, the clock is reset so that its next timeout is when it would have been without the deselect and summing continues with no loss of data. If not, the summation is paused (at mf=0), the buffer is split into 8 packets and sent to the the RT virtual channel to be made into VCDUs. The buffer is zeroed and summation is resumed (at mf=0) one RIM after it was interrupted, with the summation period based on the then-current UVS allocation. When the downlink rate is not sufficient to handle the RT stream and the multi-use buffer is full at what would be a readout time, CDS will reset the countdown clock so as to maintain normal dump timing, zero the buffer, and resume summing. ~~and continue to sum the data rather than throwing any away. In other words, a full multi-use buffer is treated like a deselect.~~

The first mf 0 after a 6TMSED/6TMCHG command which changes the UVS RT data rate~~with an RT field other than No Change~~ will be treated like the end of a summation period, if UVS RT is selected, with the same test for buffer full, and the same subsequent actions. This provides one~~is so that there is a~~ way to interrupt the 24-hour summation when an observation needing higher temporal resolution is to begin. If UVS RT is deselected, the 6TMSED/6TMCHG will be ignored (except that the new summation interval will be used after the next countsown) and summation will continue.

Another way to bring about the end of a summation period is to send a 6UVRT command. This command is acted on whether or not UVS RT is selected. If the command is 6UVRT,PACKET,UVS|BOTH, the data from the buffer are packetized, while if the command is 6UVRT,DISCRD,UVS|BOTH the data are discarded. In either case a new summation period is begun at the start of the next RIM.

If both a 6TMSED/6TMCHG and a 6UVRT command as described above are issued in the same RIM, the decision whether to packetize or discard the data will come from the 6UVRT command, and the new summation period will come from the 6TMSED/6TMCHG command.

No instrument commanding is needed when RT rates are changed.

The time of the start of the summation is included in the first packet for each buffer, and the time of the end of the summation is included in the last packet for the buffer. The mf is not included in either time because the summation always begins and ends on mf 0.

UVS RT Data

Time inc.	App ID	Size	Seq #	(Time)	Data1	Data2	...	Data 273
1	7	9	7	24	8	8	...	8
	UVS1			R-R-R				

5.5.2.2. UVS Record/Playback

CDS will block together the data from 5 minor frames (or fewer if no more data are available in the multi-use buffer) to make up one precompression packet. The data are kept in input/time order.

UVS Precompression/Raw Packet Data Field

Data 1 mf 1	Data 2 mf 1	Data 3 mf 1		...		Data 84 mf 1
Data 1 mf 2	Data 2 mf 2	Data 3 mf 2		...		Data 84 mf 2
Data 1 mf 3	Data 2 mf 3	Data 3 mf 3		...		Data 84 mf 3
Data 1 mf 4	Data 2 mf 4	Data 3 mf 4		...		Data 84 mf 4
Data 1 mf 5	Data 2 mf 5	Data 3 mf 5		...		Data 84 mf 5

If Rice compression is to be done, CDS will treat the entire precompression packet as a single 8-bit channel with 420 measurements, despite the fact that F channel data, if present, are log compressed. The first value is used by the compressor as a reference value to start off the compression and is returned raw. Subsequent values have the preceding raw value subtracted from them before as the first step of the compressor. Compression blocks are fourteen values, with the reference value output from one block being used in the next one within a channel.

In order to make it easier on the ground to keep track of UVS data independently of whether or not it was compressed, UVS2 and UVS3 packets share a single sequence number counter. If an attempted packet compression results in expansion. that packet will be returned as UVS2.

UVS Compressed Packet Data Field

mf count	Reference Value	Comp. Data
1 Byte	1 Byte	$\leq$420+n

~~CDS does not check for data expansion, and sends down a compressed-type packet if the PB table requests compression, even if a raw packet would be smaller.~~
The time will be included for the first packet of each set of PB data and every 16th packet. The packets are formed into VCDU’s in the PB virtual channel and the format is:

UVS PB Data--Raw

Time inc.	App ID	Size	Seq #	(Time)	Data1	Data2	...	Data≤420
1	7	9	7	32	8	8	...	8
	UVS2			R-R-R-mf				

UVS PB Data--Compressed

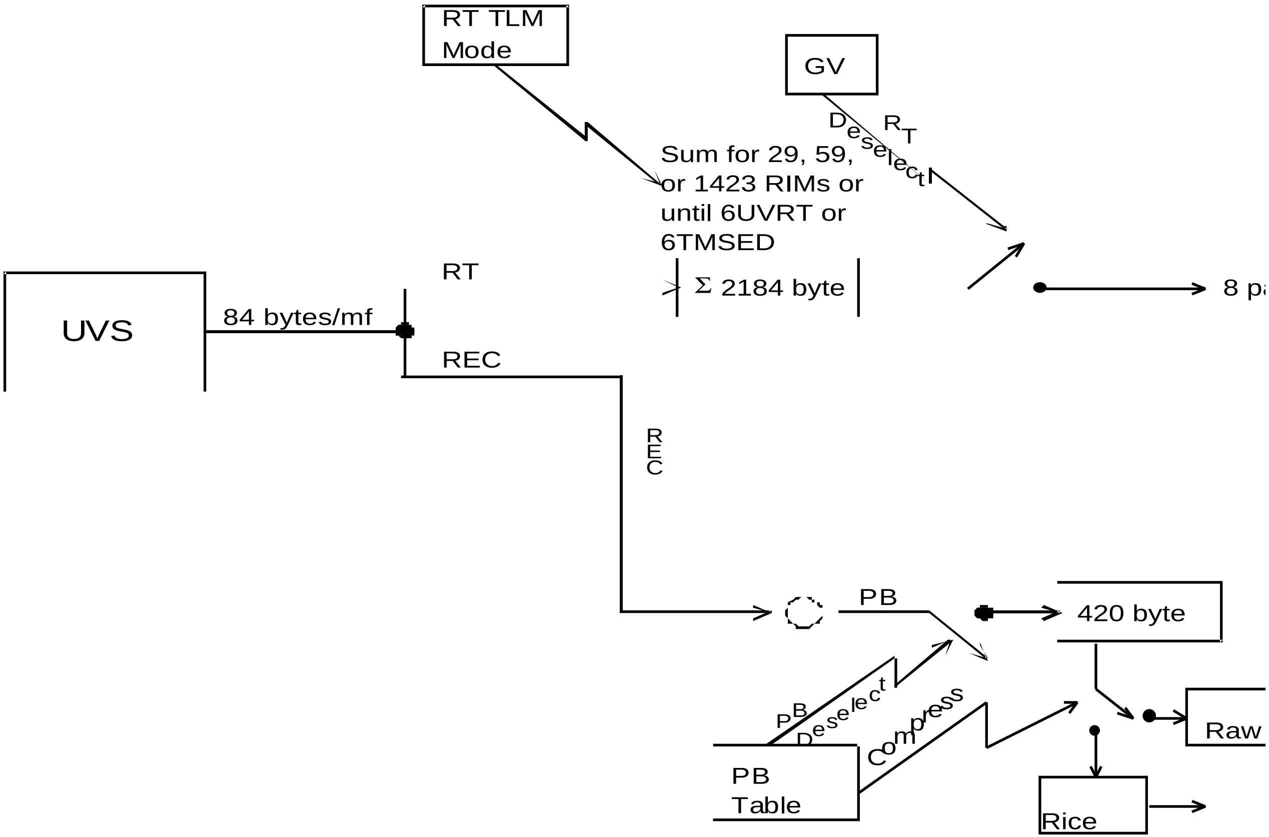
Time inc.	App ID	Size	Seq #	(Time)	Data1	Data2	...	Data≤511
1	7	9	7	32	8	8	...	8
	UVS3			R-R-R-mf				

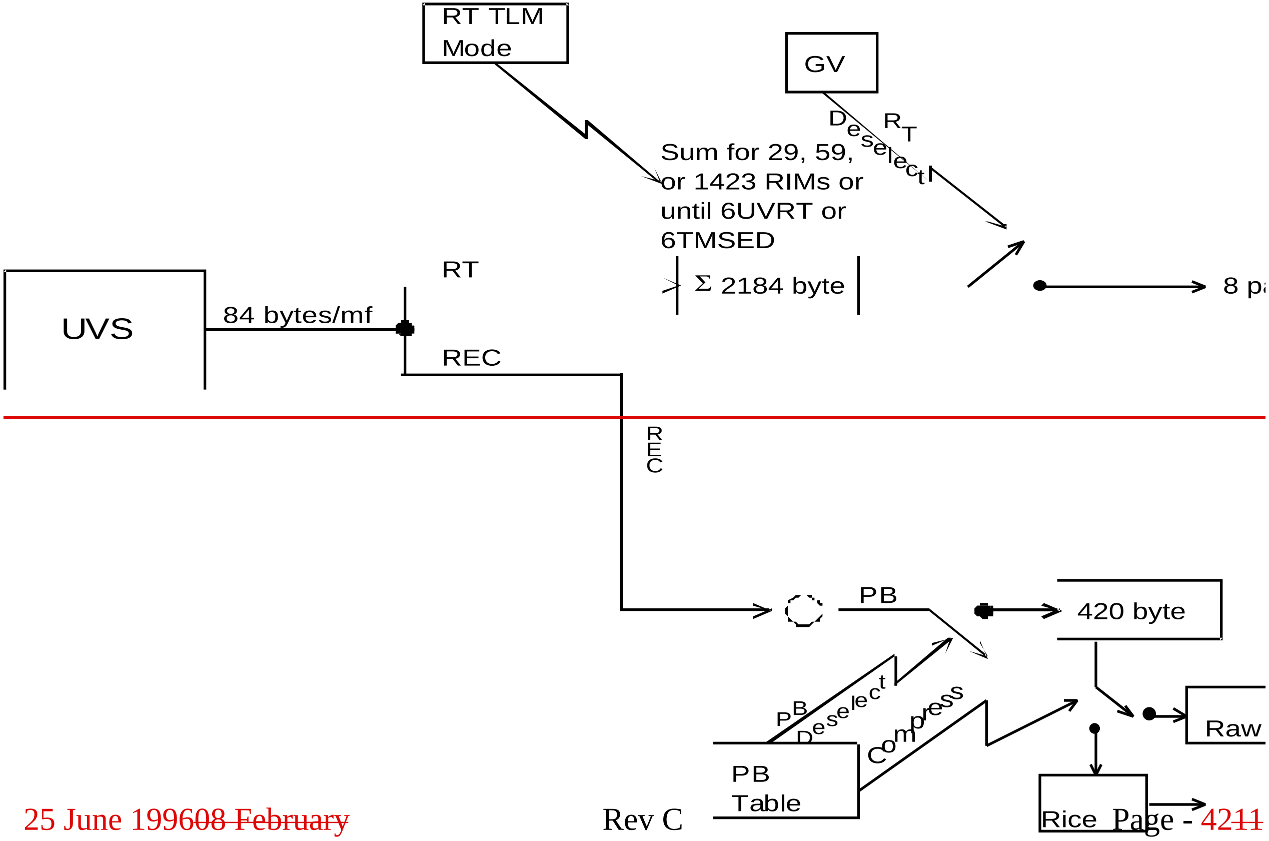
5.5.2.4. UVS General

All instrument housekeeping data are returned in the regular packets. Temperature measurement E1790 will continue to be returned in the 91 deck.

UVS Data Flow in CDS

625-205, 3-310, Phase 2





5.5.3. Changes in UVS Commands

None. There are no new or modified UVS commands.

However. UVS RT operations are affected by the new CDS command 6UVRT.

5.6. MAG

5.6.1. MAG Data Pickup

The MAG RT data pickup is six bytes (a 16-bit time-averaged field strength measurement for each axis) picked up every mf, with CDS keeping only 1-of-n data sets, with the intervals of 4 to 36 mf's corresponding to RT allocations of 16 to 2 b/s, respectively. The pickup can be in RTI 1, 2, 7, or 8 and prepare directives are not used.

Rate 1 keep interval: 36 mf
Rate 2 keep interval: 18 mf
Rate 3 keep interval: 12 mf
Rate 4 keep interval: 9 mf
Rate 5 keep interval: 6 mf
Rate 6 keep interval: 4 mf

The REC LPW pickup for MAG remains unchanged at 20 bytes per mf, with 10 being picked up in RTI 5 and 10 in RTI 1. The data are always available, and are also used for LNR.

The MAG RRCC mode uses the same data pickup as LPW.

MAG supplies three packet types: RT, REC/PB, and RRCC, in addition to standard instrument MRO.

5.6.2. MAG Data Processing in CDS

5.6.2.1. MAG Realtime

The processing for RT MAG data is a straightforward double buffering into 180-byte buffers, packetization (adding header information), and forming into a VCDU in the RT virtual channel. The Format ID field is used to indicate at which RT rate the data were taken. The time will be included in two situations: the first packet of a new rate and every 16th packet. The packet format is:

MAG RT Data

Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	Data1	Data2	...	Data≤90
1	7	9	7	4	28	16	16	...	16

	MAG1				½R-R-R-mf				

When a 6TMSED/6TMCHG command which changes the MAG RT rate~~with an RT field other than No Change~~ is issued, CDS will wait until the next scheduled RT pickup, send the two-byte filter setting appropriate for the ~~(possibly~~ new~~)~~ MAG data allocation to MAG in the same mf in an RTI other than 3-6, and schedule the next RT pickup at the ~~(possibly~~ new~~)~~ appropriate interval. ~~If the MAG allocation does actually change,~~ CDS will also close out the current packet and begin a new one. If MAG is deselected from the RT data stream, CDS will not issue the two filter setting bytes at the 6TMSED/6TMCHG as described, so that MAG can use its special long-term averaging settings.

5.6.2.2. MAG Record/Playback

The format of the raw instrument as read from the DMS follows:

byte 1	St1	St2	X1hi	X1lo	Y1hi	Y1lo	Z1hi	Z1lo	X2hi	X2lo	
	Y2hi	Y2lo	Z2hi	Z2lo	X3hi	X3lo	Y3hi	Y3lo	Z3hi	Z3lo	byte 20

There are two bytes of status data, followed by three sets of sensor data, each of which has X, Y, and Z measurements of two bytes each. CDS will use the data from 24 of these instrument packets to make up one downlink packet. If the active PB table entry for MAG does not specify compression, the (up to) 24 mf's of data are simply concatenated in time order and packetized.

The status data are returned raw even if Rice compression is being done on the sensor data.

If Rice compression is to be done, CDS will difference and compress the nine sensor channels independently. The first value of each channel is used by the compressor as a reference value to start off the compression and is returned raw following the pre-split bits for the first compression block of the channel. Subsequent values have the preceding raw value subtracted from them as the first step of the compression. Compression blocks are eight values, with the reference value output from one block being used in the next one within a channel. Whenever there are not enough data to make a complete 24-mf packet (e.g., end of a tape gulp, end of recorded data, deselect), CDS will autonomously create an 8- or 16-mf compressed MAG3 packet if possible, and switch to MAG2 for a short packet with any remaining data if necessary. In order to make it easier on the ground to keep track of MAG data independently of whether or not it was compressed, MAG2 and MAG3 packets share a single sequence number counter. If an attempted packet compression results in expansion. that packet will be returned as MAG2.

The format of the data in the compressed packet is:

Compressed Packet Data Field

These data columns always present					16 or 24 mf's only		24 mf's only	
mf count (raw)	8, 16, or 24 Status 1, 2 measurement pairs (raw)	X1 pre-split bits	X1 reference	X1 Comp data	X1 pre-split	X1 data	X1 pre-split	X1 data
		Y1 pre-split bits	Y1 reference	Y1 Comp data	Y1 pre-split	Y1 data	Y1 pre-split	Y1 data
		Z1 pre-split bits	Z1 reference	Z1 Comp data	Z1 pre-split	Z1 data	Z1 pre-split	Z1 data
		X2 pre-split bits	X2 reference	X2 Comp data	X2 pre-split	X2 data	X2 pre-split	X2 data
		Y2 pre-split bits	Y2 reference	Y2 Comp data	Y2 pre-split	Y2 data	Y2 pre-split	Y2 data
		Z2 pre-split bits	Z2 reference	Z2 Comp data	Z2 pre-split	Z2 data	Z2 pre-split	Z2 data
		X3 pre-split bits	X3 reference	X3 Comp data	X3 pre-split	X3 data	X3 pre-split	X3 data
		Y3 pre-split bits	Y3 reference	Y3 Comp data	Y3 pre-split	Y3 data	Y3 pre-split	Y3 data
		Z3 pre-split bits	Z3 reference	Z3 Comp data	Z3 pre-split	Z3 data	Z3 pre-split	Z3 data

~~The packet data quantity was chosen so that, even if compression does not occur, the data will still fit in a single packet without truncation. Raw and compressed packets are differentiated by having different App ID's. CDS does not check for data expansion, and sends down a compressed-type packet if the PB table requests compression, even if a raw packet would be smaller.~~ The time will be included in two situations: the first packet of each time-contiguous set of PB data and every 32nd packet. The packets go into the PB virtual channel and the format is:

MAG PB Data--Raw

Time inc.	App ID	Size	Seq #	(Time)	Data1	Data2	...	Data≤240
1	7	9	7	32	16	16	...	16
	MAG2			R-R-R-mf				

MAG PB Data--Compressed

Time inc.	App ID	Size	Seq #	(Time)	See Compressed Packet Data Field
1	7	9	7	32	
	MAG3			R-R-R-mf	

5.6.2.3. MAG RRCC

No editing is performed on the MAG RRCC data. The data for one RRCC instance consists of ~~The pre-compression and decompressed packet is simply standard header information followed by~~ 1~~5~~ to~~-~~31~~18~~ 20-byte MAG LPW packets (from the up to ~20~~12~~-second RRCC period), put into one or two packets, depending on the amount of data. The time will be included in each RRCC packet. The packets are formed into VCDU's in the RRCC virtual channel and the format is:

MAG RRCC Data

Time inc.	App ID	Size	Seq #	Time	Data1	Data2	...	Data≤180
1	7	9	7	32	16	16	...	16
1	MAG4			R-R-R-mf				

5.6.2.4. MAG General

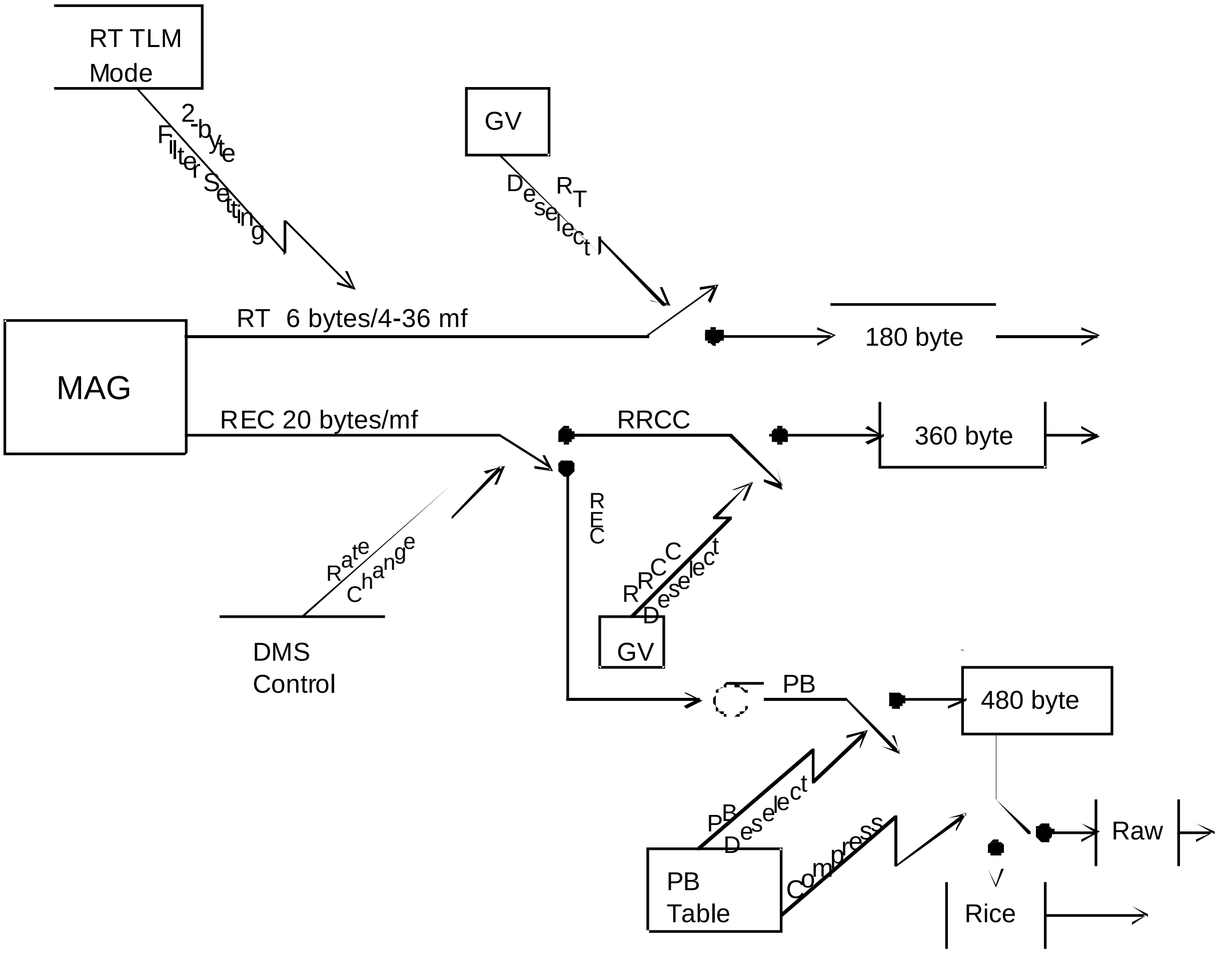
Since both the RT and the REC data streams are available at all times, no commanding is needed for switching from one to the other.

MAG data can be deselected from either the RT or the PB data streams, and the normal usage will be to continue using the RT data while LPW is being recorded.

Housekeeping data will be returned in recorded LPW and by MRO.

Three 2-byte sensor offsets will be added to the 91-deck of the fixed engineering telemetry.

MAG Data Flow in CDS



5.6.3. Changes in MAG Commands

None.

5.7. DDS

5.7.1. DDS Data Pickup

The DDS RT data pickup is simply a less-frequent version of the LPW DDS pickup, with two bytes of data each 7 or 21 mf's. Like the current DDS REC data, pickup occurs in RTI 5 following a prepare directive in RTI 4. No commanding is required for switching between RT rates or between RT and REC modes.

The LPW pickup for DDS is unchanged at 2 bytes per mf, collected in RTI 5 following a prepare directive in RTI 4, and is also used for LNR.

The RRCC pickup for DDS is identical to the current LPW pickup--2 bytes per mf, collected in RTI 5 following a prepare directive in RTI 4.

5.7.2. DDS Data Processing in CDS

5.7.2.1. DDS Realtime

The DDS RT data cycle is 637 or 1911 mf's (at pickup intervals of 7 or 21 mf's, respectively), comprising 182 bytes of data. The raw packet has header data added and is sent the the RT virtual channel to be made into a VCDU. The Format ID field is used to indicate at which RT rate the data were taken. The time will be included in two situations: the first packet of a new rate and every 8th packet.

DDS RT Data

Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	Data1	Data2	...	Data≤182
1	7	9	7	4	28	8	8	...	8
	DDS1				½R-R-R-mf				

When a 6TMSED/6TMCHG command changes DDS's RT rate, CDS will finish the current data cycle (which also completes the packet) and then simply start sending the prepares and picking up the data at the new rate. A short packet will be sent if DDS is switched from RT to REC mode at a time other than just after packet completion.

5.7.2.2. DDS Record/Playback

No editing is performed on the DDS REC/PB data. The packet is simply standard header information followed by 128 2-byte DDS LPW data sets. The time will be included in two situations: the first packet of each set of PB data and every 8th packet. The packets are formed into VCDU's in the PB virtual channel and the format is:

DDS PB Data

Time inc.	App ID	Size	Seq #	(Time)	Data1	Data2	...	Data≤256
1	7	9	7	32	8	8	...	8
	DDS2			R-R-R-mf				

5.7.2.3. DDS RRCC

No editing is performed on the DDS RRCC data. The packet is simply standard header information followed by 1~~5~~ to 31~~18~~ 2-byte DDS LPW data sets (from the up to ~20~~12~~-second RRCC period). The time will be included in each RRCC packet. The packets are formed into VCDU's in the RRCC virtual channel and the format is:

DDS RRCC Data

Time inc.	App ID	Size	Seq #	Time	Data1	Data2	...	Data≤62~~36~~
1	7	9	7	32	8	8	...	8
1	DDS3			R-R-R-mf				

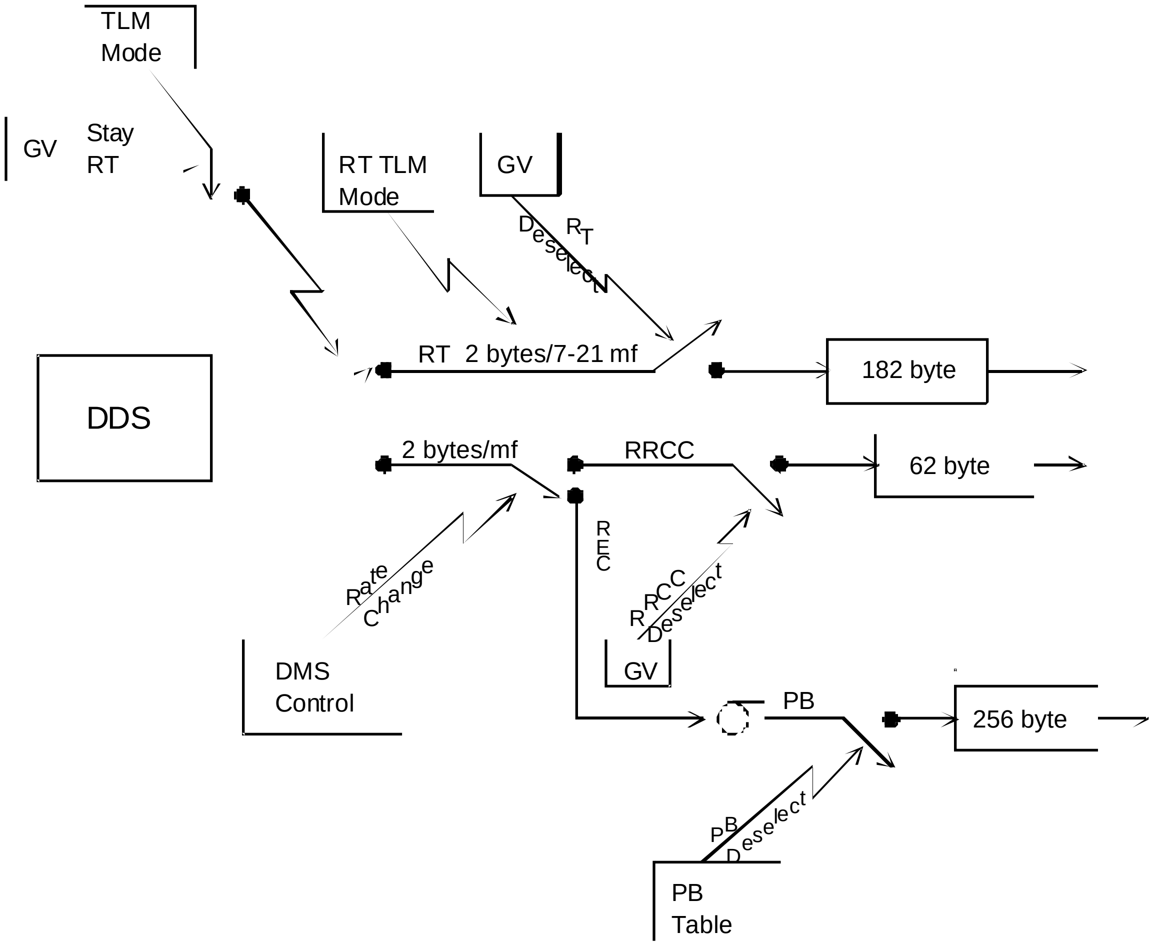
~~CDS will flush the final VCDU of each RRCC instance with dummy data.~~

5.7.2.4. DDS General

The commanding for switching between RT and REC modes is under sequence control, with a bit in a global variable (checked at the beginning of each LPW record mode) telling CDS whether DDS is deselected from a REC mode, in which case the RT stream should continue to be processed.

DDS data can be independently deselected from the RT and PB streams.

DDS Data Flow in CDS



5.7.3. Changes in DDS Commands

No new commands planned. Switching pickup interval (1, 7, 21 mf's) is accomplished simply by sending the prepare directive only when data are going to be read.

5.8. PLS

5.8.1. PLS Data Pickup

The PLS RT pickup is two read transactions of 113 bytes each (totalling 226~~230~~ bytes), performed in RTI 5 following a prepare directive in RTI 4 in sequential mf's. One or the other of t~~T~~he pickups is done every minor frame, even when PLS is not in its RT mode. CDS will throw away the data most of the time, keeping and packetizing it just often enough to fit the PLS RT data allocation, based on a countdown timer maintained by CDS.

The REC LPW data pickup from PLS remains unchanged at 51 bytes per mf, collected in RTI 3 following a prepare command in RTI 2, and is also used for LNR. The pickup and its prepare are done continuously, even when PLS is not in the record mode.

The RRCC data pickup from PLS is identical to the LPW pickup at 51 bytes per mf, collected in RTI 3 following a prepare command in RTI 2.

5.8.2. PLS Data Processing in CDS

5.8.2.1. PLS Realtime

When CDS's timer indicates that it is time to take PLS RT data, CDS will check the first byte (a Data Ready flag) of the PLS even-mf RT buffer (which it reads each even mf). If the Data Ready flag~~it~~ is zero (meaning that the RT data aren't ready), CDS will wait until it reads a non-zero flag~~first byte~~. After it has accepted an RT buffer for packetizing (keeping the 112 data bytes of the even-mf buffer which contains the non-zero flag and the 113 bytes of the next odd-mf buffer it reads), CDS will set the ~~a **TBD**~~ Acknowledge location in PLS memory (address 1108_{hex}) to $0F_{hex}$, so that PLS can start preparing a new RT dataset. When PLS sees the Acknowledge byte set, it will clear both the Data Ready and Acknowledge flags.

The only processing CDS does to PLS RT data is putting one instrument packet of ~~up to~~ 225 bytes ~~(with the exact length being a function of the instrument mode)~~ together with the standard header information and sending the packet to the RT virtual channel. ~~The length byte is the second byte read from the PLS RT buffer, but the first byte returned, since the handshaking byte is discarded.~~ The Format ID field is used to indicate at which RT rate the data were taken. The time will be included in two situations: the first packet of a new rate and every 16th packet.

PLS RT Data

Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	Data 1	Data 2	...	Data≤225
1	7	9	7	4	28	8	8	...	8
	PLS1				½R-R-R-mf				

When a 6TMSED/6TMCHG command changes PLS's RT rate, CDS will complete the current countdown/readout cycle (filling a packet in the process) and will immediately start counting down for picking up data at the new rate.

5.8.2.2. PLS Record/Playback

CDS will channelize and block the PLS PB data into a precompression packet even if Rice compression is not requested by the PB table.

PLS Packet as Read From Instrument

Data 1 1 Byte	Data 2 1 Byte	Data 3 1 Byte		...		Data 51 1 Byte

CDS will use the data from 9 minor frames to make up one precompression packet. The data are treated as 51 independent 8-bit channels. The data in each channel are kept in input/time order.

PLS Precompression/Raw Packet Data Field

Data 1 mf 1 1 Byte	Data 2 mf 1 1 Byte	Data 3 mf 1 1 Byte		...		Data 51 mf 1 1 Byte
Data 1 mf 2 1 Byte	Data 2 mf 2 1 Byte	Data 3 mf 2 1 Byte		...		Data 51 mf 2 1 Byte
...	...	...		...		...
Data 1 mf 9 1 Byte	Data 2 mf 9 1 Byte	Data 3 mf 9 1 Byte		...		Data 51 mf 9 1 Byte

If Rice compression is to be done, CDS will difference and compress the first three channels independently (taking data values vertically). The rest of the data are considered to be one long channel read row-wise. The first value of each channel is used by the compressor as a reference value to start off the compression and is returned raw. Subsequent values have the preceding raw value subtracted from them as the first step in the compressor. Compression blocks are nine values for the first three channels and 12 values for the other channel. The first data byte in the compressed packet is a minor frame count used by the decompressor to tell how many data values to extract in each channel. . Whenever there are not enough data to make a complete compression block (e.g., end of a tape gulp, end of recorded data, deselect), CDS will autonomously switch to PLS2 for the short packet. In order to make it easier on the ground to keep track of PLS data independently of whether or not it was compressed, PLS2 and PLS3 packets share a single sequence number counter. If an attempted packet compression results in expansion. that packet will be returned as PLS2.

PLS Compressed Packet Data Field

mf count	Data 1 Ref. Val. 1 Byte	Data 1 compr. data ≤10 Bytes	Data 2 Ref. Val. 1 Byte	Data 2 compr. data ≤10 Bytes	Data 3 Ref. Val. 1 Byte	Data 3 compr. data <10 bytes	Data 4-51 Ref. Val. 1 Byte	Data 4-51 compr. data ≤448 Bytes

~~The pre-compression packets are a fixed size chosen so that, even if compression does not occur, the data will still fit in a single packet without truncation. CDS does not check for data expansion, and sends down a compressed-type packet if the PB table requests compression, even if a raw packet would be smaller.~~ The time will be included in two situations: the first packet of each set of PB data and every 16th packet. The packets go into the PB virtual channel and the format is:

PLS PB Data--Raw

Time inc.	App ID	Size	Seq #	(Time)	Data1	Data2	...	Data ≤459
1	7	9	7	32	8	8	...	8
	PLS2			R-R-R-mf				

PLS PB Data--Compressed

Time inc.	App ID	Size	Seq #	(Time)	See Compressed Packet Data Field
1	7	9	7	32	
	PLS3			R-R-R-mf	

5.8.2.3. PLS RRCC

No editing is performed on the PLS RRCC data. The packet is simply standard header information followed by up to 5 51-byte PLS LPW data sets . There are up to 7~~4~~ packets per RRCC instance, with the time being included in each~~the first RRCC~~ packet~~ of each RRCC occurence and whenever the sequence number rolls over~~. The packets are formed into VCDU's in the RRCC virtual channel and the format is:

PLS RRCC Data

Time inc.	App ID	Size	Seq #	Time	Data1	Data2	...	Data≤255
1	7	9	7	32	8	8	...	8
	PLS4			R-R-R-mf				

~~CDS will flush the final VCDU of each RRCC instance with dummy data.~~

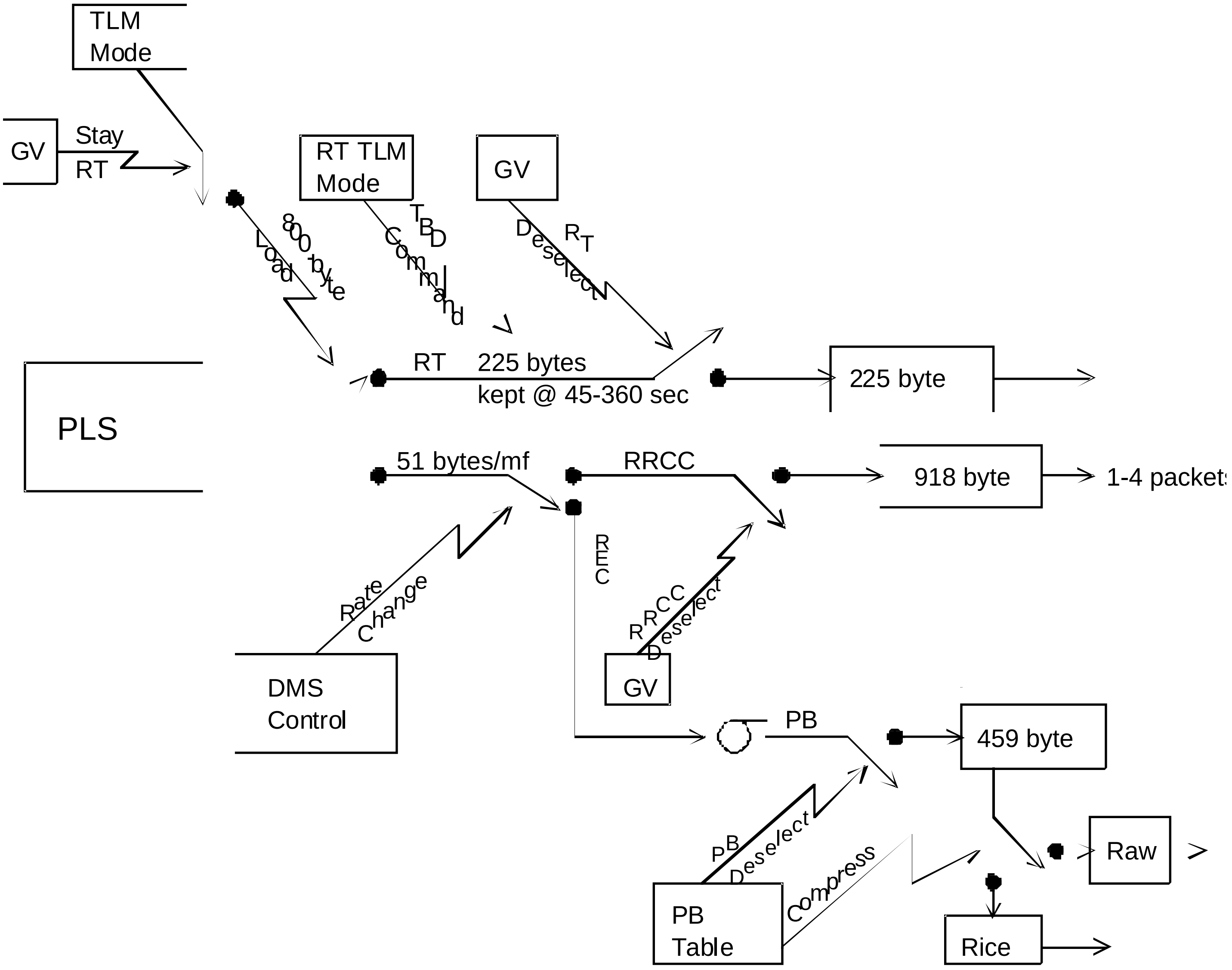
5.8.2.4. PLS General

Commanding (in general from the stored sequence in association with 6TMSED or possibly a realtime command) is required for switching between RT and REC modes, with a memory load of ~800 bytes needed for the latter switch. The switching process involves disabling certain ongoing instrument processes, loading the PLS code memory, updating the links to point to the newly loaded code, and reenabling the stopped processes. CDS will store sets of 32IDM memory load commands (<255 bytes per set) which the sequence will load into PLS memory using 6MCOPY commands. The disable, load, and enable sets of commands will be in separate mf's so that PLS can act on them before the next set is loaded.

The commanding and memory loading for switching between RT and REC modes is under sequence control, with a bit in a global variable telling CDS whether PLS is deselected from a REC mode, in which case the RT stream should continue to be processed. PLS data can also be deselected from the RT stream.

All housekeeping data come from the REC/PB data unless there is an instrument anomaly, in which case MRO's will be used.

PLS Data Flow in CDS



5.8.3. Changes in PLS Commands

New functionality will be implemented by defining pseudonyms of the existing 32IDM command for the memory loads used for switching between RTS and LPW modes, and between different RTS rates.

5.9. PWS High-Rate

5.9.1. PWS High-Rate Data Pickup

There are no RT PWS high-rate modes, although dual-use modes MPW, MPP, and HPW are being kept as REC modes, and PWS data will be used as filler data whenever the downlink rate exceeds the RT rate and PB data are not available or frames of fill data are being used for DSN lockup.

Existing REC modes: PWS supplies data at 2.592 kb/s for LPW, 7.68 kb/s for MPW, 19.2 kb/s for MPP, and 94.56 kb/s for HPW.

5.9.2. PWS High-Rate Data Processing in CDS

5.9.2.1. PWS Fill Data

PWS fill data is taken from the PWS LPW buffer . The data portion of the packet is in the same format as the LPW data but is 3 bytes longer. CDS assembles the fill data from the most recently commanded instrument mode. The PWS fill packets are turned into VCDUs in a dedicated channel. The time will be included in all packets. CDS will be able to generate fill data on demand for use either to fill in when normal downlink is enabled but no other data are available, or to be sent in “dummy” telemetry frames used to allow a DSN station without an FSR to lockup on the signal initially or after a phase/frequency shift.

PWS Fill Data

Time inc.	App ID	Size	Seq #	Time	Data1	Data2	...	Data 435
1	7	9	7	32	8	8	...	8
1	PWH1			R-R-R-mf				

5.9.2.2. PWS High-Rate Record/Playback

CDS processing of PWS high-rate PB data is a 1-of-*n*-line editor (which merely skips *n*-1 RTIs of data after each RTI read for modes other than LPW, for which a line of 432 bytes is the data from two consecutive mf’s, starting with an even-numbered one). The value of *n* can range from 1 to 15. The packets are sent to the PB virtual channel to be made into VCDUs. The Format ID field is used to indicate the RTI of the start of the data (high nybble) and the value of *n* (n=0 is used to indicate the second packet of an HPW frame--

PWH4). The time will be included in two situations: the first packet of each set of PB data and every 16th packet.

REC Mode	PWS Bytes per Frame	Frames per Packet
LPW	432	1
MPW	64	4
MPP	160	2
HPW	788	½

PWS High-rate MPW PB Data

Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	Data1	Data2	...	Data≤256
1	7	9	7	8	32	8	8	...	8
	PWH2				R-R-R-mf				

PWS High-rate MPP PB Data

Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	Data1	Data2	...	Data≤320
1	7	9	7	8	32	8	8	...	8
	PWH3				R-R-R-mf				

PWS High-rate HPW PB Data

Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	Data1	Data2	...	Data 394
1	7	9	7	8	32	8	8	...	8
	PWH4				R-R-R-mf				

5.9.2.3. PWS Golay Replacement Data

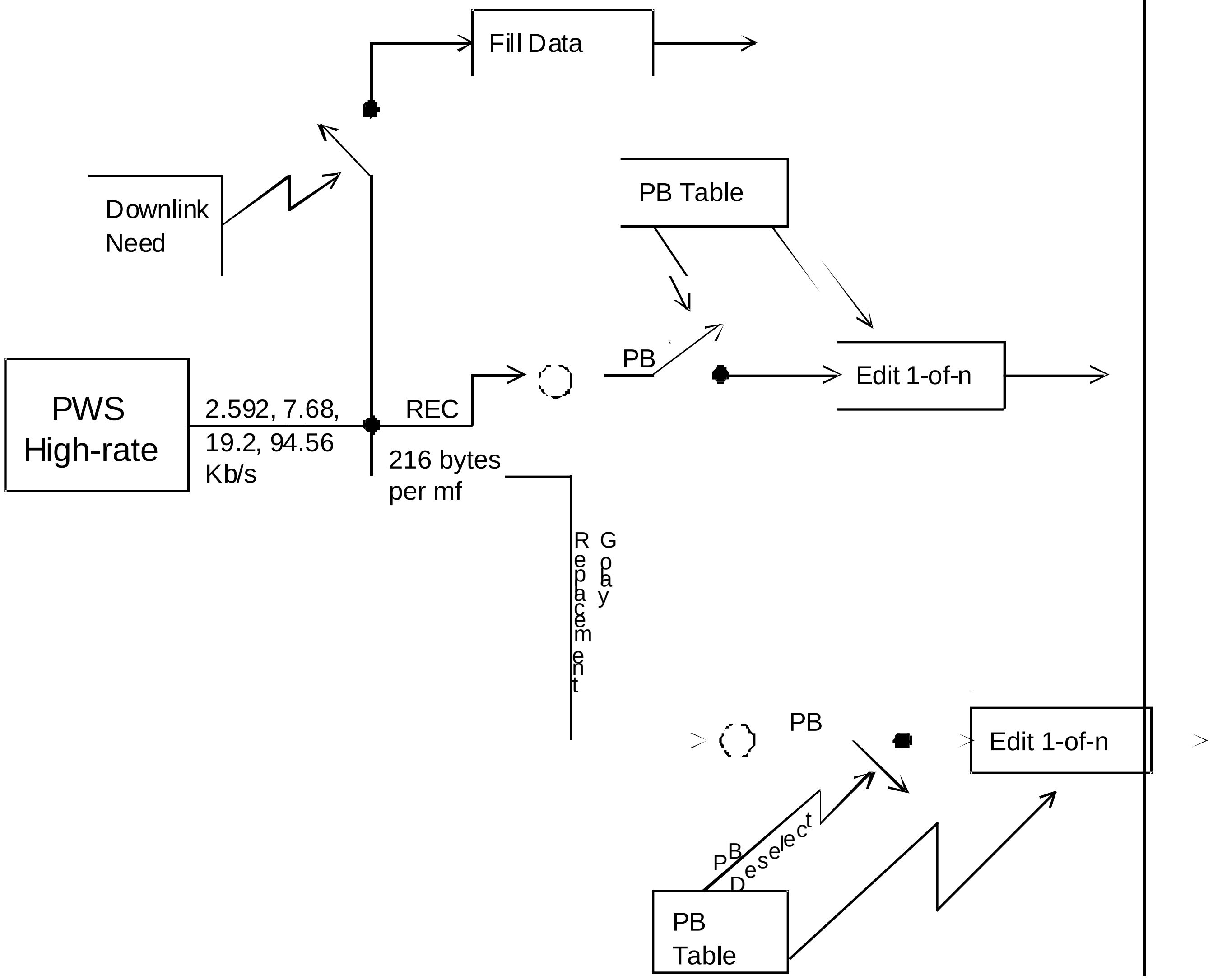
CDS processing of the PWS high-rate used to replace the Golay code (making the old LRS into the new LPW) also uses the 1-of-n-line editor and deselect processes as explained above. The Format ID field uses its low-order nybble to give the n value used in editing. The packets are sent to the PB virtual channel to be made into VCDUs. The time will be included in two situations: the first packet of each set of PB data and every 16th packet.

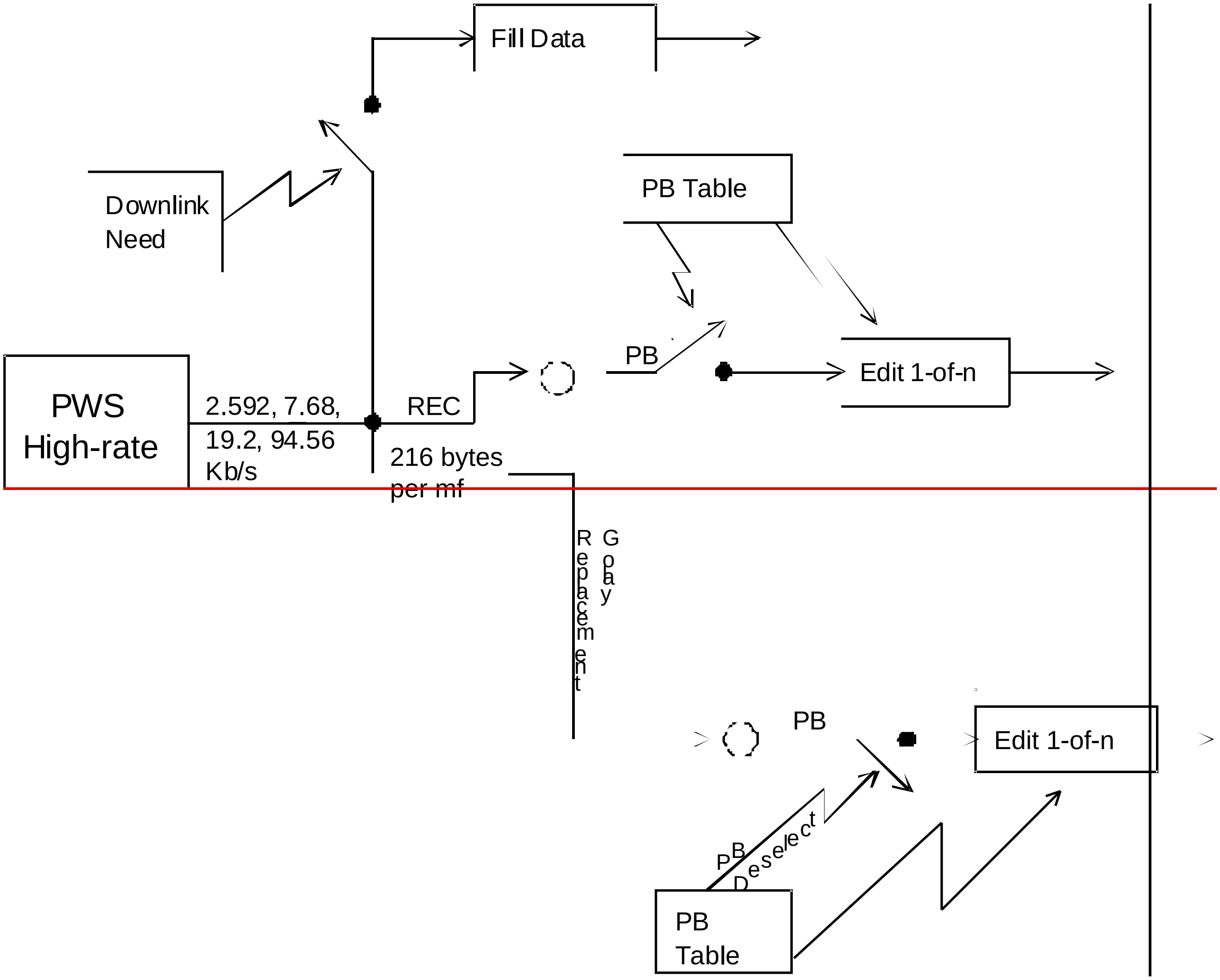
Note that a 6ASSLV command must be sent whenever PWS is changed between Mode 1 or 2 (high-rate) and Mode 3 (low-rate) for the Golay replacement data to be collected correctly.

PWS LPW Golay Replacement Data

Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	Data1	Data2	...	Data ≤432
1	7	9	7	8	32	8	8	...	8
	PWH5				R-R-R-mf				

PWS High-rate Data Flow in CDS




Fill Data
Downlink
Need
PB Table
PWS
High-rate
2.592, 7.68,
19.2, 94.56
Kb/s
REC
PB
Edit 1-of-n
216 bytes
per mf
Replacement
Gap
PB
Edit 1-of-n
PB Deselect
PB
Table

5.10. PWS Low-Rate

5.10.1. PWS Low--Rate Data Pickup

The RT pickup for PWS low-rate data is identical to the current LPW pickup, at 20 bytes per mf in RTI 1 with no prepare command used.

The REC LPW pickup for PWS low-rate data is identical to the current LRS pickup, at 20 bytes per mf in RTI 1 with no prepare command, ans is also used for LNR.

The RRCC pickup for PWS low-rate data is identical to the current LPW pickup, at 20 bytes per mf in RTI 1 with no prepare command used.

5.10.2. PWS Low-Rate Data Processing in CDS

5.10.2.1. PWS Low-Rate Realtime

The RT PWS data are edited, rearranged, and sent through an 8x8 ICT compression in AACS at one of 6 commandable target compression ratios, giving effective post-compression data rates of 5, 10, 15, 20, 30, and 40 b/s. Details elsewhere. The time will be included in ~~two situations: the first packet of a new rate and~~ every 32nd packet. The PWS packets are turned into VCDUs in the RT virtual channel.

Over 28 mf's CDS builds up one line of 152 measurements. When 8 lines have been built up, creating a compression frame of 1216 bytes, the ICT is started. Four different Q factors are used for the 19 8x8 blocks in the strip, with the highest five frequencies using one Q, then the next five, then five more, and finally the lowest four frequencies use the fourth Q.

The level of compression achieved depends on both the data being compressed and the Q factor fed into the ICT algorithm. The high degree of data sensitivity means that if a Q factor is chosen which guarantees a bit rate not greater than the allocation, much of the time the bit rate will be far below the allocation, resulting in less useful data. To avoid this undesirable situation the compression is dynamic.That is, the Q factor is adjusted one step up or down based on the compression performance. Each data rate has a fixed predefined range around it in which the compression is considered acceptable. Compression results outside this range will result in the Q factor being changed for a fixed number of compression frames covering a time interval of 5 to 30 RIMs.

The data from the ICT consists of a strip of 19 8x8 blocks. Each block of data consists of Huffman codes from the 8x8 region following a 1-bit flag which is set to 1 if the attempted compression resulted in expansion. In the case of expansion, the data returned are truncated at 64 bytes, resulting in a post-decompression 8x8 block with the highest spatial frequencies missing. The data from the compression blocks within each strip are packed into bytes with no gaps, but with up to 7 bits of zero fill after the last block in the strip. Each strip's data starts in a new packet. The data field can be split across packet boundaries (necessary only for low compression factors), with continuation packets marked by a 1 in the high-order bit of the Fmt ID.

There is a global variable bit which tells the ICT handler whether or not to use the compression rate specified by the RT part of the telemetry mode. If the bit is set, the Q factor which determines the target level of compression comes from the 5 b/s mode, rather than from the commanded RT mode. When the PWS RT data allocation changes, CDS will finish the current packet before switching to the new Q factor, if the global variable indicates that the RT telemetry mode determines Q.

The Format ID field is used to indicate which set of four Q values was used to compress the data, in addition to the continuation bit mentioned above.

PWS Low-rate RT Data--E Field

Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	Data 1	Data 2	...	Data≤511 ~~256~~
1	7	9	7	4	28	8	8	...	8
	PWL1				½R-R-R-mf				

PWS Low-rate RT Data--B Field

Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	Data 1	Data 2	...	Data≤511 ~~256~~
1	7	9	7	4	28	8	8	...	8
	PWL2				½R-R-R-mf				

5.10.2.2. PWS Low-Rate Record/Playback

The CDS performs no processing on PB PWS low-rate data except blocking 12 20-byte instrument packets together, adding the standard header information, and sending to the PB virtual channel to be made into VCDUs. The time will be included in two situations:

the first packet of each set of PB data and every 32nd packet. This data stream contains all the housekeeping data needed by the PWS team.

PWS Low-rate PB Data

Time inc.	App ID	Size	Seq #	(Time)	Data1	Data2	...	Data≤240
1	7	9	7	32	8	8	...	8
	PWL3			R-R-R-mf				

5.10.2.3. PWS Low-Rate RRCC

No editing is performed on the PWS RRCC data. The data set for an RRCC instance consists of ~~packet is simply standard header information followed by~~ 1~~5~~ to 31~~18~~ 20-byte PWS LPW packets (from the up to ~20~~12~~-second RRCC period). The data will be put into one or two packets. The time will be included in each RRCC packet. The packets are formed into VCDU's in the RRCC virtual channel and the format is:

PWS RRCC Data

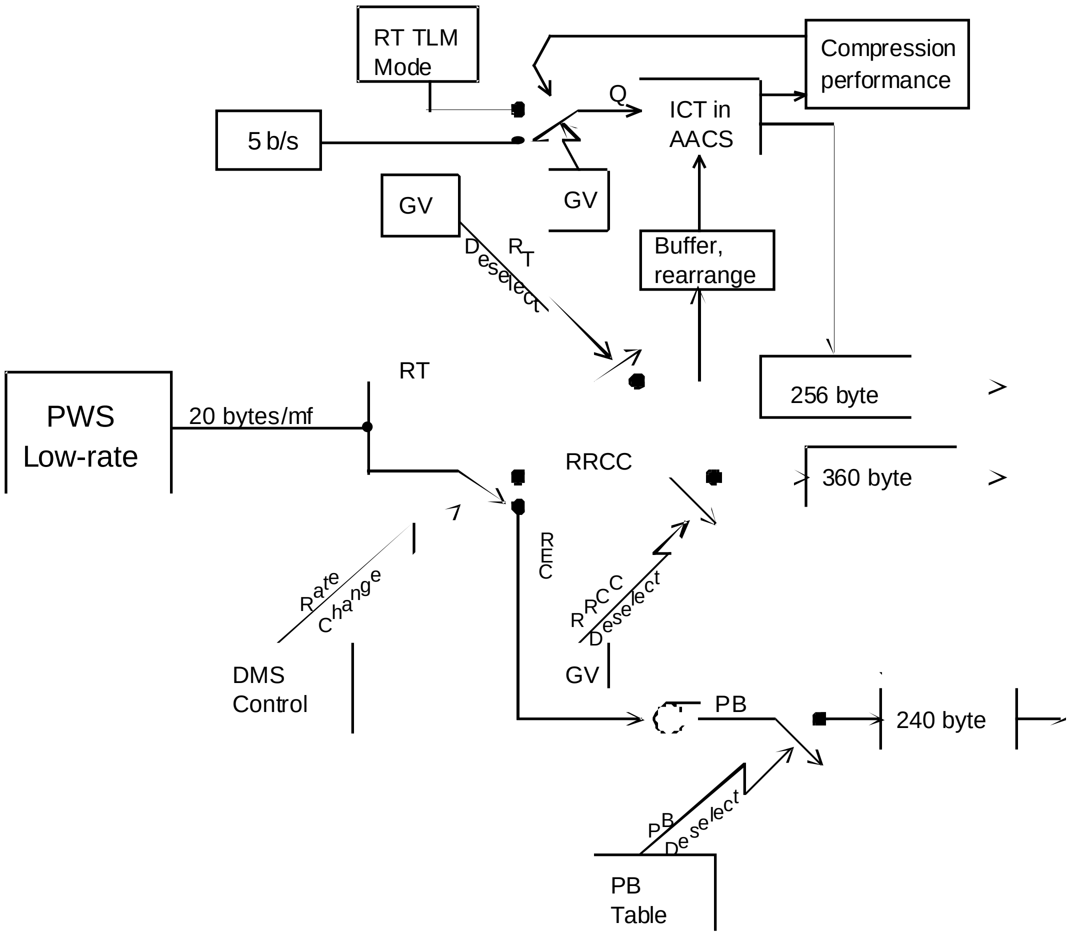
Time inc.	App ID	Size	Seq #	(Time)	Data1	Data2	...	Data≤360
1	7	9	7	32	8	8	...	8
	PWL4			R-R-R-mf				

~~CDS will flush the final VCDU of each RRCC instance with dummy data.~~

5.10.2.4. PWS Low-Rate General

PWS data can be independently deselected from the RT and PB data streams. Normal instrument usage will be to keep processing the RT data even when LPW is being recorded.

PWS Low-rate Data Flow in CDS



5.11. EPD

5.11.1. EPD Data Pickup

EPD supplies 211~~145~~ bytes of RT rate data each mf. CDS will read the data in two transactions (127 & 84 bytes) in RTI 3.

~~EPD supplies a fixed **TBD** number of bytes (≤ 127) of PHA and housekeeping data on command from CDS at intervals ranging from 6 to 72 S/C spins, read in RTI 3.~~

The REC LPW data pickup from EPD remains unchanged at 76 bytes per mf, collected in RTI 5 with no prepare command, and is also used for LNR.

The RRCC data pickup from EPD is identical to the LPW pickup at 76 bytes per mf, collected in RTI 5 with no prepare command.

5.11.2. EPD Processing in CDS

5.11.2.1. EPD Realtime

The processing for EPD RT data is involved. The 145-byte data sets consist of a 1-byte rate channel header followed by 48 24-bit measurements. Many of the 145-byte data sets will contain repeats of data which have already been processed and will be ignored by CDS. They are marked by having the MSB of the header set to zero. The header format is given below:

MSB							LSB
old/new	SP1	SP2	NEWSPIN	MAP ID - 1	M1	M2	M3

where SP1-SP2 give the spin quadrant, NEWSPIN is explained below, MAP ID is 1 or 2, and M1-M3 give the motor position.

Each of the 48 measurements in the non-repeat data sets will be summed by CDS into 32-bit "bins," based on a moderately complicated algorithm. The six different EPD RT data allocations specify both the summation period and which of the two Channel Maps to use. The channel maps assign each of the 48 channels to the High-resolution, the Low-resolution, or the Omni-resolution category. Finally, the resolution category specifies how the spin quadrant and motor position are used to determine the binning. For example, channels which are Omni have only two bins, and the others have 7 and 15.

The summation is for a science record of 1 to 6 data cycles, each of which is 6 S/C revolutions (one revolution at each motor position). The number of spins comprising a science record is determined by the RT data allocation as explained in 3-270. CDS is aided in counting spins by the NEWSPIN~~a **TBD**~~ flag in the EPD data~~, probably~~ which indicates~~in the rate data header byte, indicating~~ the start ~~(or the end, whichever works better)~~ of a spin. When the summation is completed, the bins are log-compressed and packetized, yielding 401 data bytes for Map 1 and 508 bytes for Map 2 .

At the end of each science record, CDS keeps one set of ~~sends a **TBD** command to EPD requesting a set of **TBD** (up to 127) bytes of~~ PHA and housekeeping data, 35 bytes for Map 1 or 65 bytes for Map 2.

Two packets are generated for each science record. The Format ID field of the first packet of the pair, which contains the summed data, is used to indicate at which RT rate the data were taken. The time will be included in every summed data packet, since the spin-controlled data collection would make ground timing of the packets difficult (especially considering that spin rates are different in dual spin and allspin). The housekeeping and PHA data are put in the second packet of the pair, with no time field and the same Format ID.

EPD RT Data

Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	Data1	Data2	...	Data≤508 ~~11256~~
1	7	9	7	4	28	8	8	...	8
	EPD1				½R-R-R-mf				

5.11.2.2. EPD Record/Playback

CDS packs together 3 76-byte EPD PB instrument packets, adds header information, and sends the packet to the PB virtual channel to be made into VCDUs. The time will be included in two situations: the first packet of each set of PB data and every 16th packet.

EPD PB Data

Time inc.	App ID	Size	Seq #	(Time)	Data1	Data2	...	Data 228
1	7	9	7	32	8	8	...	8
	EPD2			R-R-R-mf				

5.11.2.3. EPD RRCC

No editing is performed on the EPD RRCC data. The packet is simply standard header information followed by up to 3 76-byte EPD LPW packets (from the up to ~20~~12~~-second RRCC period). Up to 11 RRCC packets will be made for each RRCC instance. The time will be included in each RRCC packet. The packets are formed into VCDU's in the RRCC virtual channel and the format is:

EPD RRCC Data

Time inc.	App ID	Size	Seq #	Time	Data1	Data2	...	Data≤228
1	7	9	7	32	8	8	...	8
	EPD3			R-R-R-mf				

~~CDS will flush the final VCDU of each RRCC instance with dummy data.~~

5.11.2.4. EPD General

~~The commanding for switching between RT and REC modes (about 39 bytes of pointers), including the instrument s/w load, is under sequence control, with a bit in a global variable telling CDS whether EPD is deselected from a REC mode, in which case the RT stream should continue to be processed.~~ CDS maintains 3 110-byte instrument loads for EPD--one for record mode and two for realtime (each EPD RT data rate is associated with one or the other of the loads). CDS loads them into EPD autonomously as follows (in all cases CDS keeps track of which load it sent to EPD most recently, to avoid sending the same one again unnecessarily):

Whenever a real recording mode 6TMREC command (i.e., one other than NRC or *PB) is issued:

> CDS checks whether EPD record mode is selected. If it is, and EPD is not already in record mode (that is, if CDS most recently sent one of the RT loads), CDS sends the 110-byte record-mode load to the instrument. If EPD record mode is deselected, and EPD is not already in the realtime mode (that is, if CDS most recently sent the record load), CDS will issue the 110-byte realtime-mode load appropriate for the current EPD RT rate to the instrument. In each case, if CDS has most recently sent the load for the desired mode, the instrument load is not performed.

When a no-record or playback 6TMREC command is issued:

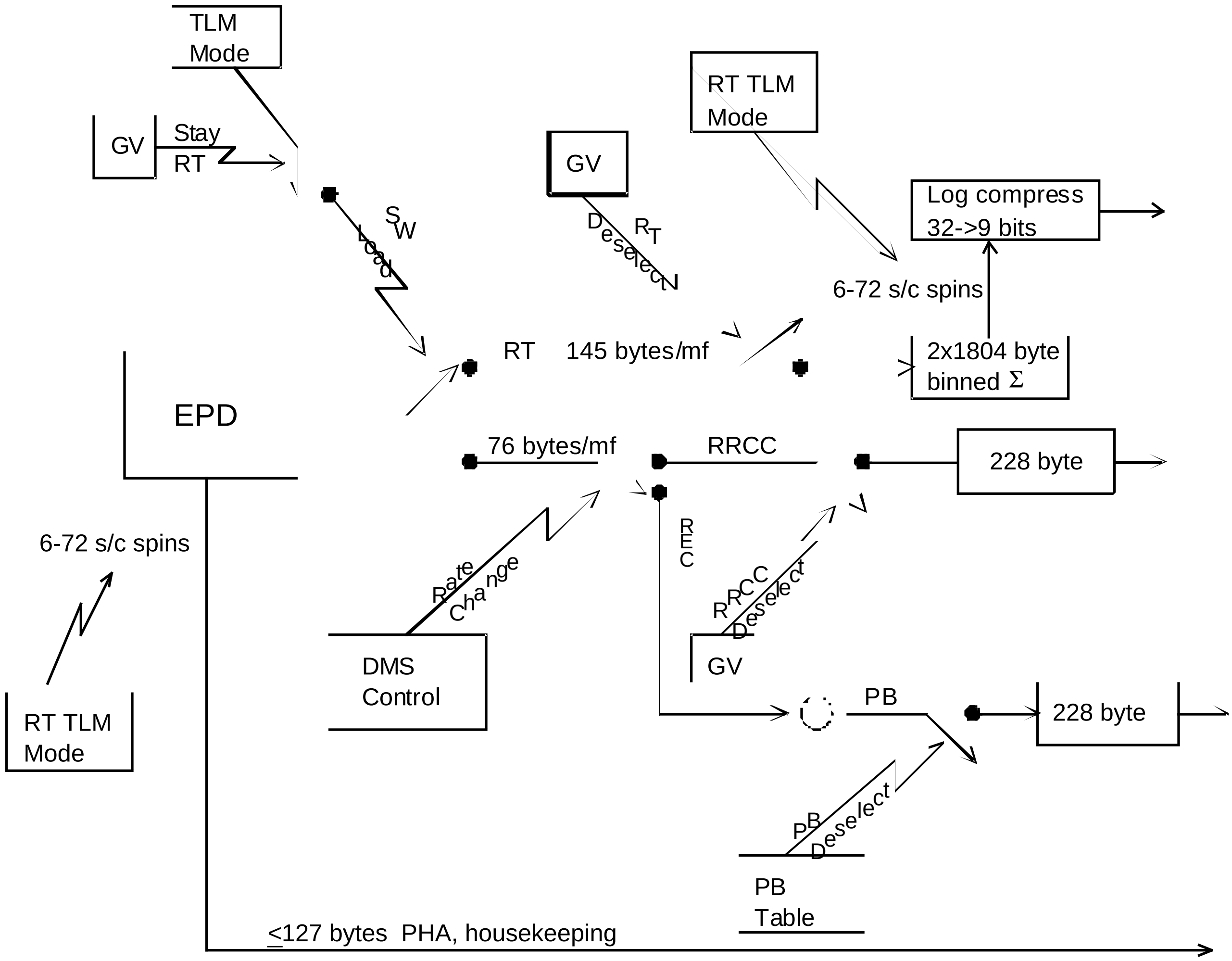
CDS will issue the appropriate realtime load if it most recently sent the record mode load (with the data cycle starting at the next spin boundary), and will do nothing otherwise.

When a 6TMSED command changes EPD’s RT rate:

CDS will finish the current science record, and then switch the summation period to the one appropriate for the new rate. If the new rate requires a new map, CDS will also send EPD the appropriate 110-byte load.

CDS delivers sector data to EPD so that the instrument can de-spin its data.

EPD Data Flow in CDS



5.11.3. Changes in EPD Commands

A new command is being added for CDS to tell EPD to prepare the block of PHA and housekeeping data.

~~Sets of pointers will have to be loaded to switch between RT and REC modes (~39 bytes).~~

5.12. HIC

5.12.1. HIC Data Pickup

The pickup for the HIC RT mode is identical to the LPW pickup--only the CDS processing is different.

The HIC REC LPW data pick up remains unchanged from the current 12 bytes per mf in RTI 1, and is used at all times, although HIC does not see the pickup when EUV is powered on.

The HIC RRCC mode uses the same data pickup as LPW.

5.12.2. HIC Data Processing in CDS

5.12.2.1. HIC Realtime

The format of the HIC RT instrument data set, as read out over three minor frames, is shown below, with all data being in 12-bit trinybbles. Sixteen such data sets constitute an instrument RT data cycle.

1st Rate Area	1st Tag Word	1st PHA Area	1st CRC Word	2nd Rate Area	2nd Tag Word	2nd PHA Area	2nd CRC Word	3rd Rate Area	Status Word	3rd Tag Word	3rd PHA Area	3rd CRC Word
36	12	36	12	36	12	36	12	24	12	12	36	12

.

The rate channel measurements are decompressed (12-to-16 bits), summed into the appropriate one of 57~~52~~ 16-bit bins plus an 8-bit counter, and recompressed (16-to-12 bits). The rare overflows in the summations are ignored. The summing is for a period of 50, 25, or 10 RIMs, for HIC RT allocations of 1, 2, and 5 b/s, respectively. The CRC words are thrown away.

PHA data whose corresponding tag words are zero are ignored. The others are saved in one of fourteen stacks (containing single events, double, triple, and garbage), based on the contents of the tag word. The stacks can hold twenty to fifty-seven~~fifty~~ events each.

The packetizer uses the Format ID to indicate the RT rate at which the data were taken. The data in the packet consist of :

~~1) The most recent 16 bytes of status,~~

1~~2~~) The 57~~52~~ 12-bit summed rate measurements with their 8-bit counters, and

2~~3~~) Data from the stacks (if available, up to the maximum packet size), taking fifteen ~~one~~ each from the stacks in priority order~~single, double, triple events until the stacks are exhausted, followed by any garbage events~~.

3) Ten bytes of PHA counter data--five 12-bit counters and one 16-bit counter.

Each packet has header data added and is sent the the RT virtual channel to be made into a VCDU. The time will be included in two situations: the first packet of a new rate and every 4th packet. Since the data collection begins on a fixed mf and ends at mf = 0, the time field does not include the mf.

HIC RT Data

Time inc.	App ID	Size	Seq #	Fmt ID	(Time)	Data 1	Data2	...	Data≤375
1	7	9	7	4	20	8	8	...	8
	HIC1				½R-R-R				

5.12.2.2. HIC Record/Playback

The HIC PB packet consists of 20 raw HIC LPW 12-byte data sets, along with the usual header information. Time is included in every 16th packet.

HIC PB Data

Time inc.	App ID	Size	Seq #	(Time)	Data 1	Data 2	...	Data 240
1	7	9	7	32	8	8	...	8
	HIC2			R-R-R-mf				

5.12.2.3. HIC RRCC

The HIC RRCC data set~~packet~~ consists of 1~~5~~ to 31~~18~~ 12-byte LPW data sets put into one or two packets with~~plus~~ the usual header information. The time field is included in each packet and the packets are made into VCDUs in the RRCC virtual channel.

HIC RRCC Data

Time inc.	App ID	Size	Seq #	Time	Data 1	Data 2	...	Data≤216
1	7	9	7	32	8	8	...	8
1	HIC3			R-R-R-mf				

~~CDS will flush the final VCDU of each RRCC instance with dummy data.~~

5.12.2.4. HIC General

There is no commanding for switching between RT and REC modes. When a 6TMSED/6TMCHG command which ~~is not no-change in the RT field is issued, whether or not it~~ changes HIC's RT rate is issued, or when HIC RT becomes deselected, CDS will finish processing the current instrument data cycle, re-log-compressing the data and then packetizing. Note that packetizing will not occur if there are still HIC RT packets left over from an earlier cycle. One RIM of data are ignored by CDS while the compression is being done. After the compression is completed, CDS resumes processing the data (unless the change was to Deselect), using the summation period appropriate for the new RT allocation.

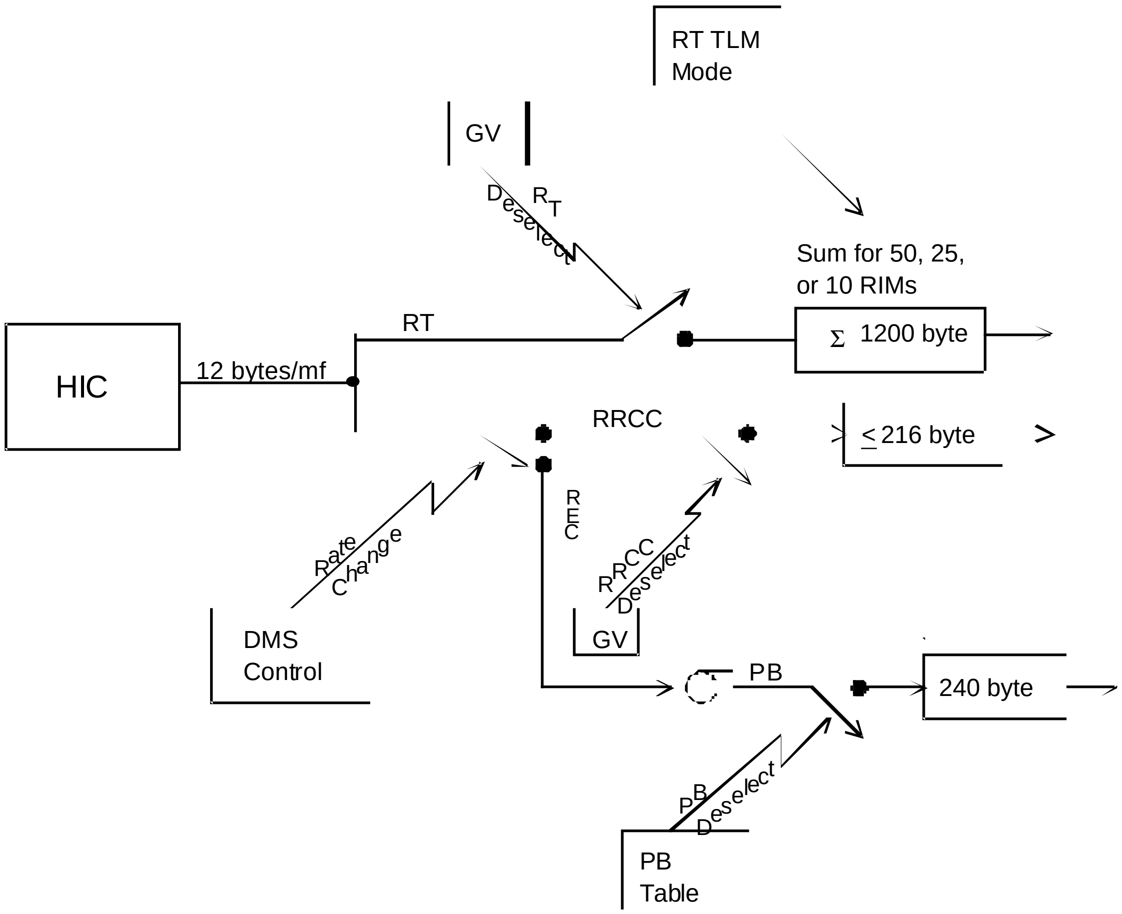
~~The buffer area for HIC RT is shared with EUV RT. The ground sequencing system is responsible for deselecting HIC RT one instrument cycle time before any switch from HIC to EUV, to prevent buffer conflicts.~~

In general, switches from RT to REC for HIC will not come at the end of an RT data collection cycle. CDS will compress, packetize, and downlink the data from any such partial summation period.

A period of five minutes must be allowed for clearing the shared buffer when switching between HIC and EUV.

Except in the case of instrument anomaly, all housekeeping status information will come from the PB data stream.

HIC Data Flow in CDS



5.12.3. Changes in HIC Commands

None.

5.13. AACS

All AACS packets, RT, PB, and RRCC, include six measurements: rotor RA, DEC, and TWIST, and scan platform RA, DEC, and TWIST.

The RT packet also includes spin rate, giving 14 bytes per measurement set.

AACS RT data are sampled the first mf of every fifth RIM. Time is included every 32nd packet.

5.13.1. AACS Realtime

AACS RT

Time inc.	App ID	Size	Seq #	(Time)	Data1	Data2	...	Data 252
1	7	9	7	24	8	8	...	8
	AACS1			R-R-R				

5.13.2. AACS Record/Playback

CDS will channelize and block the AACS PB data into a precompression packet even if Rice compression is not requested by the PB table. CDS will use the data from 40 minor frames to make up one precompression packet. The data in each channel are kept in input/time order.

Precompression/Raw Packet Data Field--All fields are 16-bit

Rotor RA 1	Rotor Dec 1	Rotor Twist 1	Pl RA 1	Pl Dec 1	Pl Twist 1
Rotor RA 2	Rotor Dec 2	Rotor Twist 2	Pl RA 2	Pl Dec 2	Pl Twist 2
...					
...					
Rotor RA 40	Rotor Dec 40	Rotor Twist 40	Pl RA 40	Pl Dec 40	Pl Twist 40

If Rice compression is to be done, the Rice algorithm will difference and compress the six channels independently. The first value of each channel is used by the compressor as a reference value to start off the compression and is returned raw. Subsequent values have the preceding raw value subtracted from them as the first step of the compressor.

Special treatment is given to the Rotor Twist value, since it is the only one in this set which is not expected to remain nearly constant much of the time. Rotor Twist increases by roughly 12° per mf at 3 RPM (actually the number is 2295 in the units used for dualspin and inertial modes). CDS will preprocess the data in this one channel so that the first value (x_0) of each compression block is sent to the compressor raw, and later values (x_n) are converted to $x_n - x_{n-1} + x_0 - 2295$. This works using unsigned arithmetic without special treatment of carry bits or rollover because, in the units chosen, 16-bit values roll over at 360°. Compression blocks are twenty values, with the reference value output from one block being used in the next one within a channel. . Whenever there are not enough data to make a complete packet (e.g., end of a tape gulp, end of recorded data, deselect), CDS will autonomously create a short (20 data sets) AACS3 packet if possible, and put any remaining data in an AACS2 packet if necessary. If an attempted packet compression results in expansion. that packet will be returned as AACS2. In order to make it easier on the ground to keep track of AACS data independently of whether or not it was compressed, AACS2 and AACS3 packets share a single sequence number counter.

The ordering of the data in the compressed packet is:

Compressed Packet Data Field

These columns always present				40-dataset packet only	
mf count	Rotor RA pre-split bits	Rotor RA ref. value	Rotor RA data	Rotor RA pre-split bits	Rotor RA data
	Rotor Dec pre-split bits	Rotor Dec ref. value	Rotor Dec data	Rotor Dec pre-split bits	Rotor Dec data
	Rotor Twist pre-split bits	Rotor Twist ref. value	Rotor Twist data	Rotor Twist pre-split bits	Rotor Twist data
	Pl RA pre-split bits	Pl RA ref. value	Pl RA data	Pl RA pre-split bits	Pl RA data
	Pl Dec pre-split bits	Pl Dec ref. value	Pl Dec data	Pl Dec pre-split bits	Pl Dec data
	Pl Twist pre-split bits	Pl Twist ref. value	Pl Twist data	Pl Twist pre-split bits	Pl Twist data

~~The pre-compression packets are a fixed size chosen so that, even if compression does not occur, the data will still fit in a single packet without truncation. CDS does not check for data expansion, and sends down a compressed-type packet if the PB table requests compression, even if a raw packet would be smaller.~~ The packets go into the PB virtual channel and the format is:

The time field is included every packet.

AACS PB--Raw

Time inc.	App ID	Size	Seq #	Time	Data1	Data2	...	Data≤480
1	7	9	7	32	8	8	...	8
1	AACS2			R-R-R-mf				

AACS PB--Compressed

Time inc.	App ID	Size	Seq #	Time	See Compressed Packet Data Field
1	7	9	7	32	
1	AACS3			R-R-R-mf	

5.13.3. AACS RRCC

An RRCC instance will lead to 1-31 AACS data sets of 6 measurements each, which will be put into one or two packets (up to 18 data sets in the first packet, and any remaining ones in the second). Time is put into each RRCC packet, and the packets go into the RRCC virtual channel

AACS RRCC

Time inc.	App ID	Size	Seq #	Time	Data1	Data2	...	Data≤216
1	7	9	7	32	8	8	...	8
1	AACS4			R-R-R-mf				

5.14. Engineering

5.14.1. Engineering Realtime

The CDS processing of engineering data consists of stripping the 12-byte header from the raw packet, and packing together 4 of the resulting 88-byte data sets (87 data bytes and one spare), each of which is preceded by an Engineering Format Identifier byte specified below:

EFI:
MSB LSB

R2	R1	MRO	CMI2	CMI1	MSN3	MSN2	MSN1

where the RT rate is given by:

R2	R1	Rate, b/s
0	0	2
0	1	10
1	0	40
1	1	1200

and MRO, if set to 1, indicates that the packet includes 32 bytes of MRO data. The two CMI bits are the Commutation Map Identifier, and the three MSN bits are the Map Sequence Number

Time is included every *n*th packet, where *n* is given below:

RT Eng. Rate, b/s	*n*
2	4
10	8
40	32

RT Engineering

Time inc.	App ID	Size	Seq #	(Time)	Data1	Data2	...	Data 356
1	7	9	7	32	8	8	...	8
	ENG1			R-R-R-mf				

5.14.2. Engineering Record/Playback

The only processing done to the engineering data on playback is stripping the 12-byte header and packing 4 of the 88-byte data sets together with their respective EFI bytes as defined above. Time is included in each packet.

Engineering PB

Time inc.	App ID	Size	Seq #	Time	Data1	Data2	...	Data 356
1	7	9	7	32	8	8	...	8
1	ENG2			R-R-R-mf				