HSO / Planck

SUBJECT:	**CDMS Simulator Software Specification (SSD)**	
PREPARED BY:	D.J.Parker	
DOCUMENT No:	**SPIRE-RAL-DOC-002298**	
ISSUE:	Draft **3**	**Date: 5 July 2001**
APPROVED BY:		**Date:**

Change Record

ISSUE	DATE	
draft 3	5/7/01	editorial

Contents

1. Introduction

1.1 Purpose of the document

1.2 Definitions, acronyms and abbreviations

1.3 References

1.4 Overview of the document

This document follows the layout recommendations of ESA's "Guide to applying the ESA software standards to small projects", RD1.

2. Documents

2.1 Applicable Documents

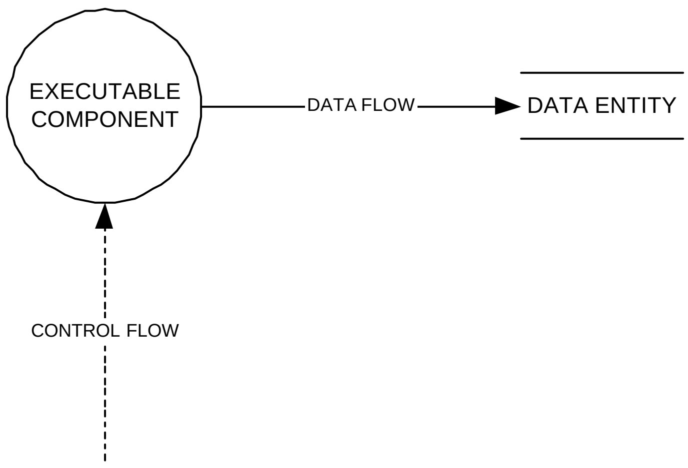
AD1	FIRDST/Planck CDMS Simulator requirements SRON-U/HIFI/SP/2000-004
AD2	Router specification

2.2 Reference Documents

RD1 BSSC(96)2 iss 1: Guide to applying the ESA software engineering standards to small software projects.

3. SOFTWARE MODEL DESCRIPTION

This section describes the logical model of the CDMS Simulator software. Yourden notation is used; symbols used in the diagrams are as follows:



3.1 Context

The Herschel/Planck CDMS simulator is used in instrument-level testing. It replaces the spacecraft data system and EGSE with a single unit, having a spacecraft-representative interface on one side (1553 bus) and a network connection on the other. Figure 3-1 shows the external interfaces with the CDMS simulator software.

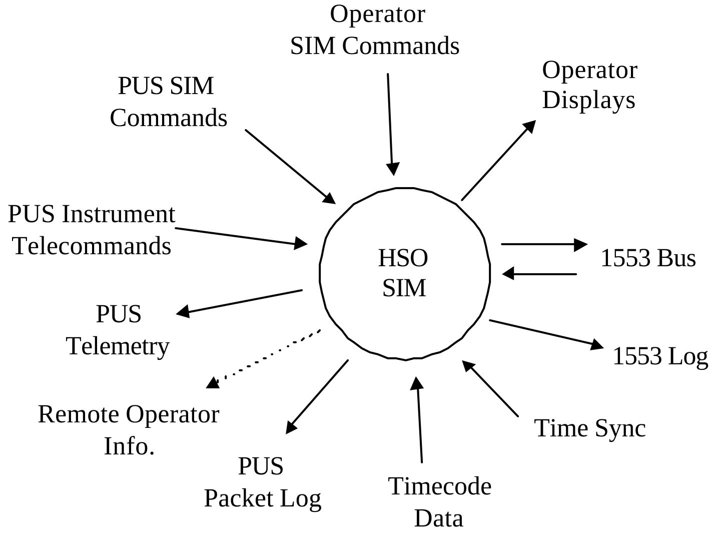


Figure 3-1: CDMS Simulator Context Diagram

The principal functions illustrated in Fig.3-1 are expanded in the following sections.

3.2 Operator Interfaces

Operator control is required in four main areas.

a) Simulator operating mode selection - related to the intended use of the simulator.

- Instrument testing, stand-alone (no other EGSE)
- EGSE testing with no instrument available
- Full system testing, with instrument plus full EGSE

b) Control of the 1553 bus.

- A or B bus selection.
- Message sequence: It should be possible to select between predefined bus polling sequences. As a minimum those corresponding to the predefined CDMU modes (normal, burst mode, memory load).

c) Data display and logging

- A log of messages on the 1553 bus must be available for examination in real-time
- Telemetry and command displays should be available to monitor data packet traffic.

d) Local Command Generation

- The operator should be able to generate and send commands from the simu lator console.

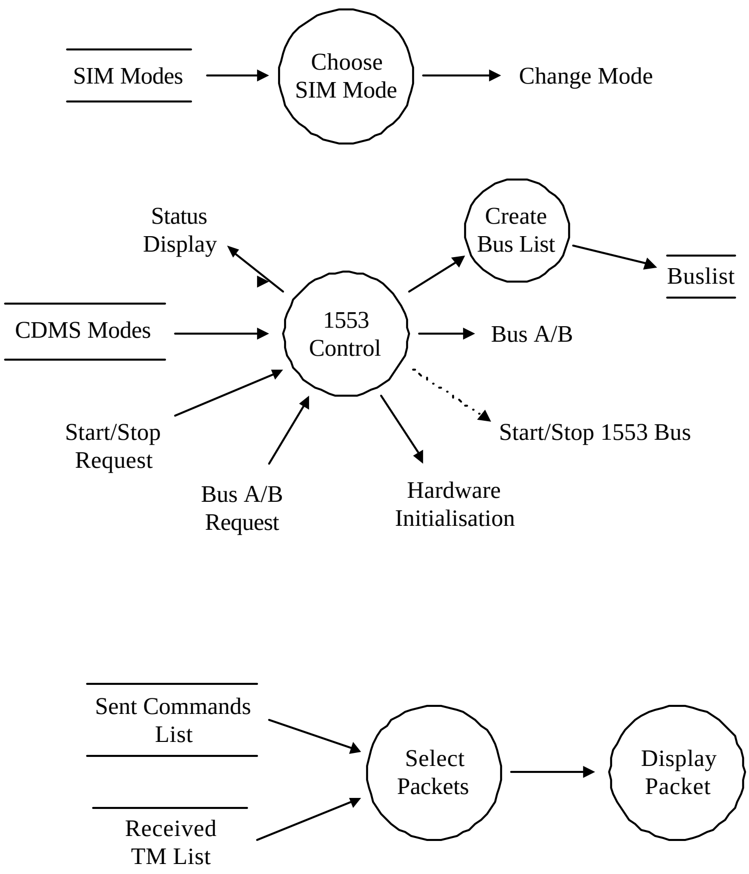


Figure 3-2: Operator Interfaces

3.3 Telecommand Functions

Instrument commands are packetised according to the PS-ICD. The operator should be able to generate these locally. Locally generated commands may be either valid or invalid; the simulator should not check them for errors. The operator will make a selection from a command list at the time of transmission to the instrument. Locally generated commands may need to be assembled from their component data and the appropriate header.

Alternatively commands may be received via the network from another EGSE machine. These commands could be addressed to the simulator itself for remote control purposes. They are recognised by a special Application ID.

Commands are logged, and a local display of processed commands is available for the operator. Optionally, the complete packet contents may be displayed. Next the packet is placed on a queue, ready for 1553 transfer-layer processing.

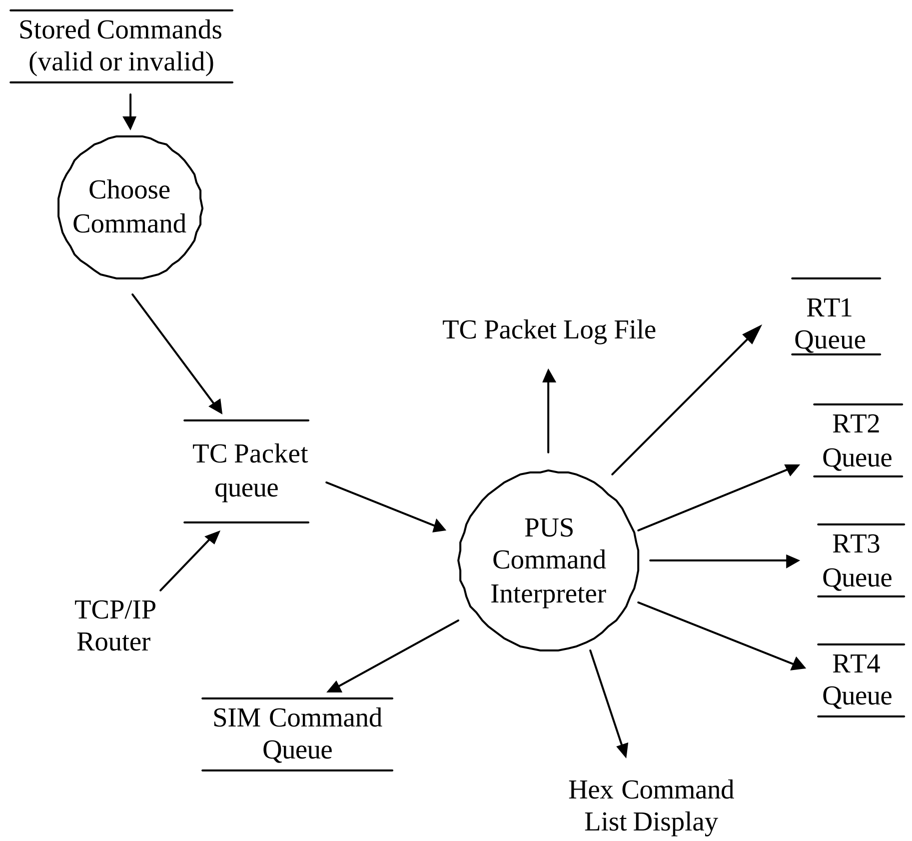
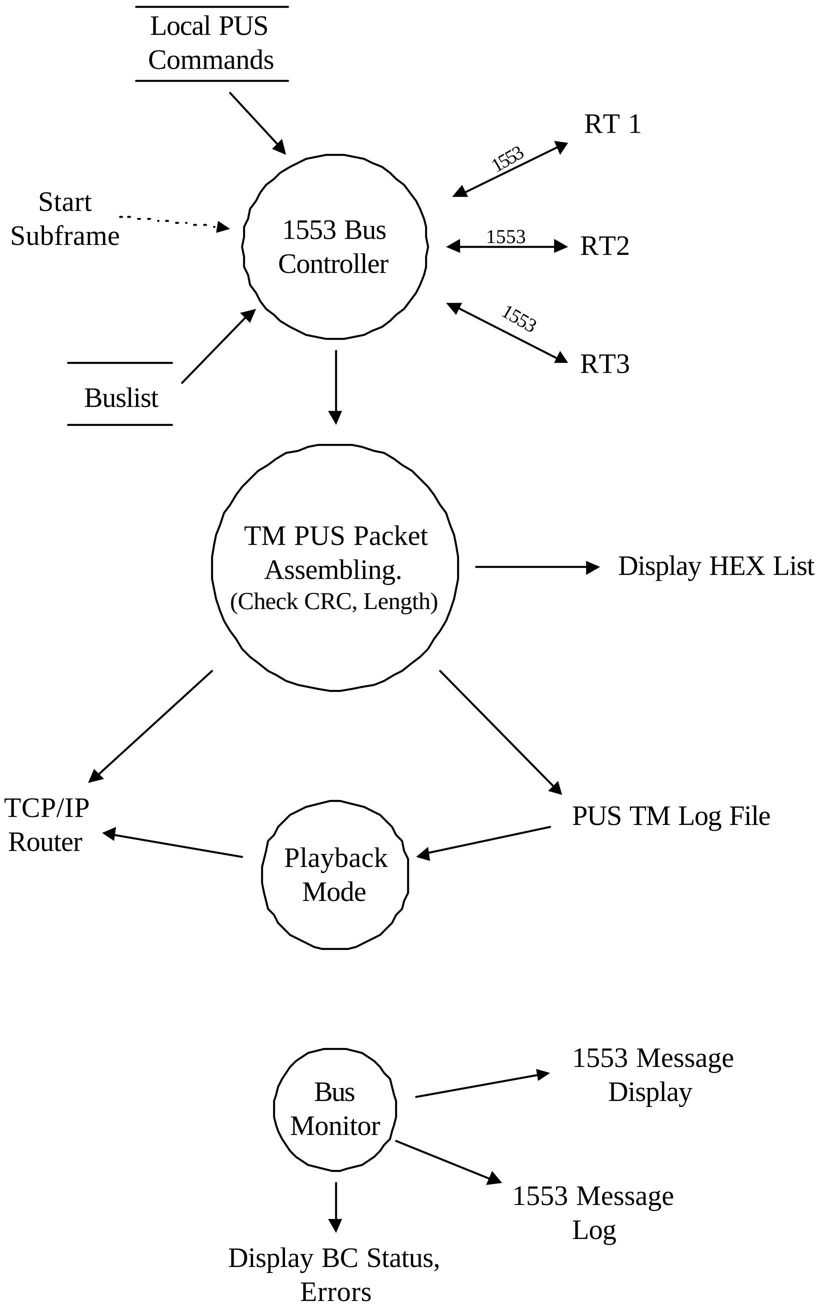
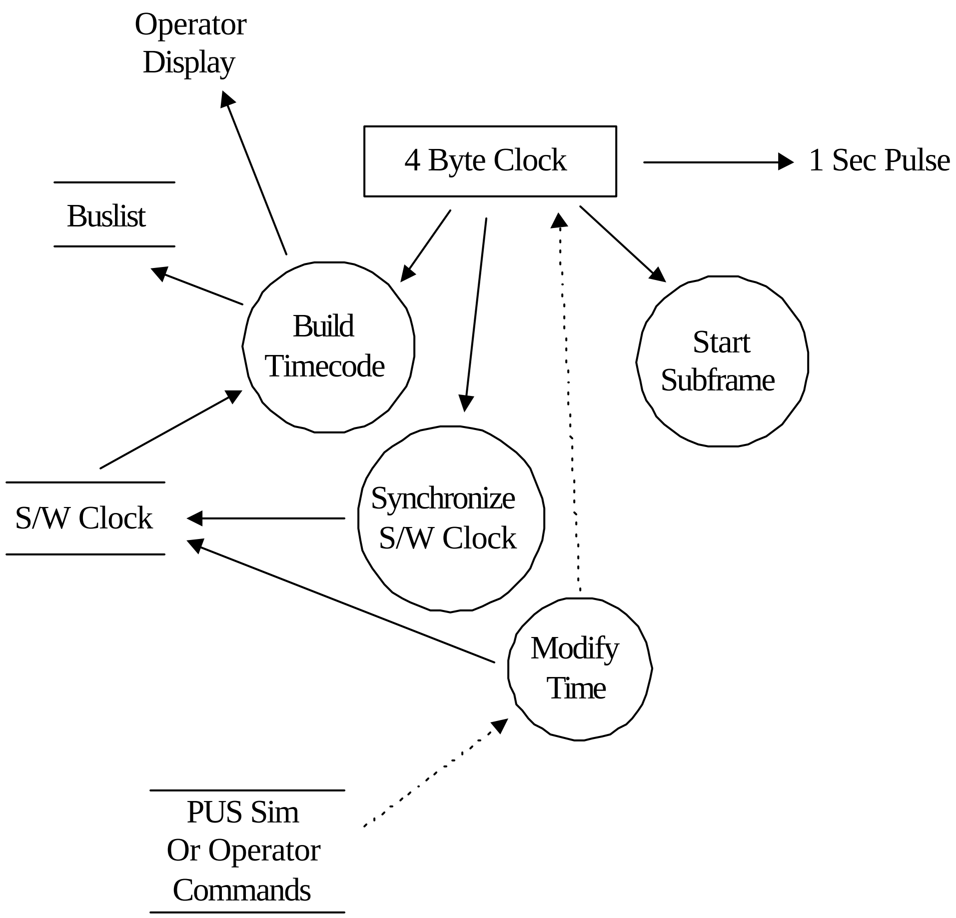


Figure 3-3: Command Functions

3.4 1553 Bus and Telemetry Functions



3.5 Timing Functions



4. SPECIFIC REQUIREMENTS

4.1 General

Functional requirements
Performance requirements
Interface requirements
Operational requirements
Resource
Verification
Acceptance testing
etc

4.2 Bus Controller

4.3 Bus Monitor

	Requirement	Comment
4.2.1	Display all messages transferred over the 1553 bus	
4.2.2	Record 30 minutes of bus messages	
4.2.3	Display BC-generated errors	
4.2.4	Display RT-generated errors	RT status word – decode error bits display in red

4.4 Instrument Simulator (Remote Terminal)

	Requirement	Comment
4.3.0	General	
4.3.0.1	The RT “instrument simulator” implements a simplified version of the Appendix 9 protocol specification, suitable for early stages in software integration.	Other more complete versions may be needed for CDMS SIM validation.
4.3.1	Telemetry Requirements	
4.3.1.1	The RT shall generate fixed length telemetry packets; length to be specified at the start of session.	
4.3.1.2	The Application ID for the simulated instrument shall be specified at the start of session.	
4.3.1.3	The packet counter shall increment for each packet	

	generated.	
4.3.1.4	The length field in the packet header is calculated from the actual length specified in (1.1).	
4.3.1.5	The data field shall consist of incrementing byte values, starting at 6 (representing the position in the overall packet including header).	
4.3.1.6	The packet shall be split up into 64 byte messages and placed in SA11..26, starting at 11 and using as many as necessary for the specified length.	
4.3.1.7	A telemetry Packet Transfer Request message shall be assembled as per PS-ICD section 4.6.1.1 and placed in SA10T.	After the packet data is inserted (see 6). Not used initially by BC.
4.3.1.8	The transfer request packet count shall be the same as that in the packet header	8 LSBs
4.3.1.9	The use of a CRC is TBD	
4.3.1.10	The RT monitors SA10R for a confirmation of reception by the BC, see PS-ICD section 4.6.1.2	Not used initially by BC.
4.3.1.11	The RT shall wait for a packet transfer confirmation before building a new packet with the counter incremented.	
2	Telecommand Requirements	
4.3.2.1	The RT shall return the contents of a TC packet as a TM packet which can be checked by the originator of the command.	
4.3.2.2	The RT shall monitor SA27R for a TC Packet Descriptor as described in PS-ICD section 4.5.1	
4.3.2.3	The RT shall read the messages comprising the packet data from SA11R..14R as defined by the Packet Descriptor.	
4.3.2.4	The RT shall insert a TC Packet Transfer Confirmation message in SA27T, see PS-ICD section 4.5.1	
4.3.2.5	The RT shall build a telemetry packet identical to the TC received, except with the Type field in the Packet ID set to zero.	
4.3.2.6	The command echo packet shall be sent as telemetry data instead of the next packet generated according to the telemetry requirements above.	

5. System Design

5.1 Design Method

The physical model of the design described in this section was created using the Yourdon method.

5.2 Simulator State Diagram

The simulator runs two separate executables under NT as shown in Figure 5-1 : Simulator State Diagram.

- "ACE Windows menu" : (DDC software provided with the BU65549 card) is run in bus monitor mode (simulator Normal mode only).
- RAL Simulator Program: provides other real-time functions.

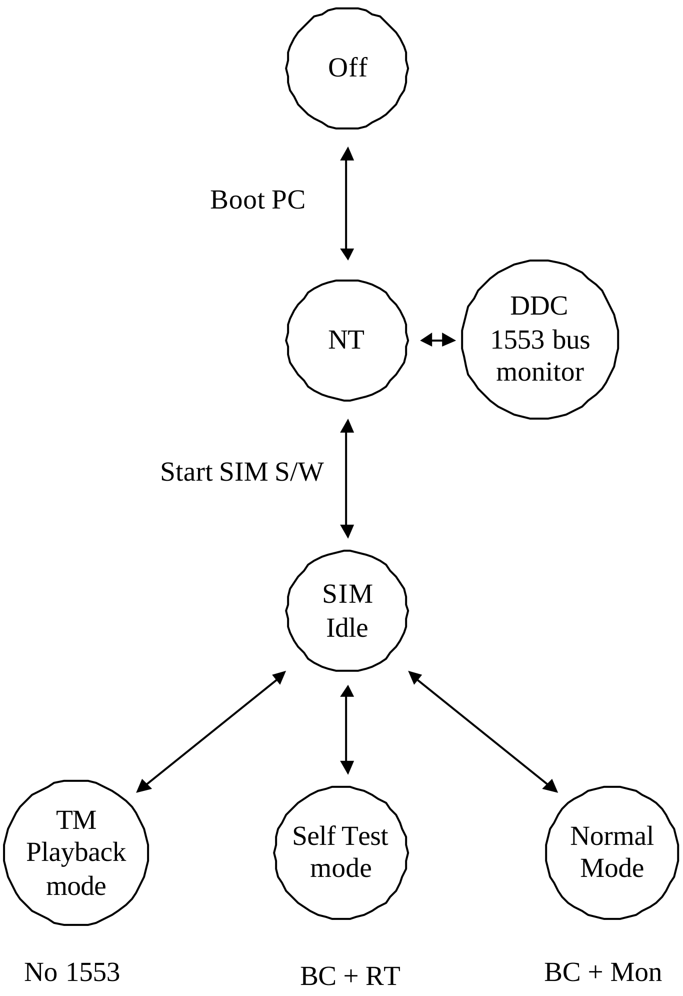


Figure 5-1 : Simulator State Diagram

5.2.1 Normal mode

ACE channel 0 acts as a Bus Controller, as the CDMS will do on the spacecraft.
ACE channel 1 is a Bus Monitor; the DDC "ACE windows menu", set to monitor mode, provides this function.

5.2.2 Self Test mode

ACE channel 0 acts as a Bus Controller, as the CDMS will do on the spacecraft.

ACE channel 1 is a dummy Remote Terminal. It gives responses to the 1553 messages for packet telecommand and telemetry transfers over the bus.

5.2.3 TM Playback mode

ACE channels are unused. The simulator plays back previously recorded telemetry files to the router.

5.3 Data Flow

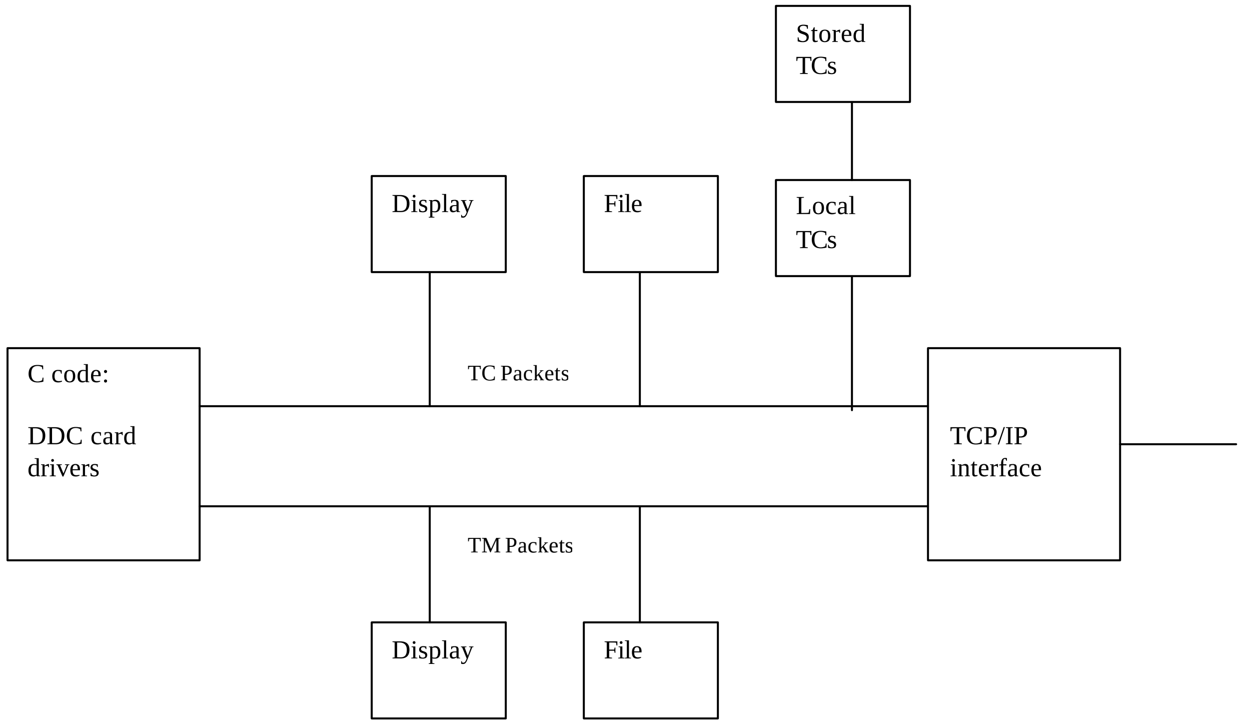


Figure 5-2

All of the functions illustrated in Figure 5-2 are written in Labview, with the exception of the C-code module which handles the transfer-layer protocol. Processing of TC and TM packets is carried out in parallel, with queues at the interface with the C code.

5.4 Real-Time Operation of 1553 Bus

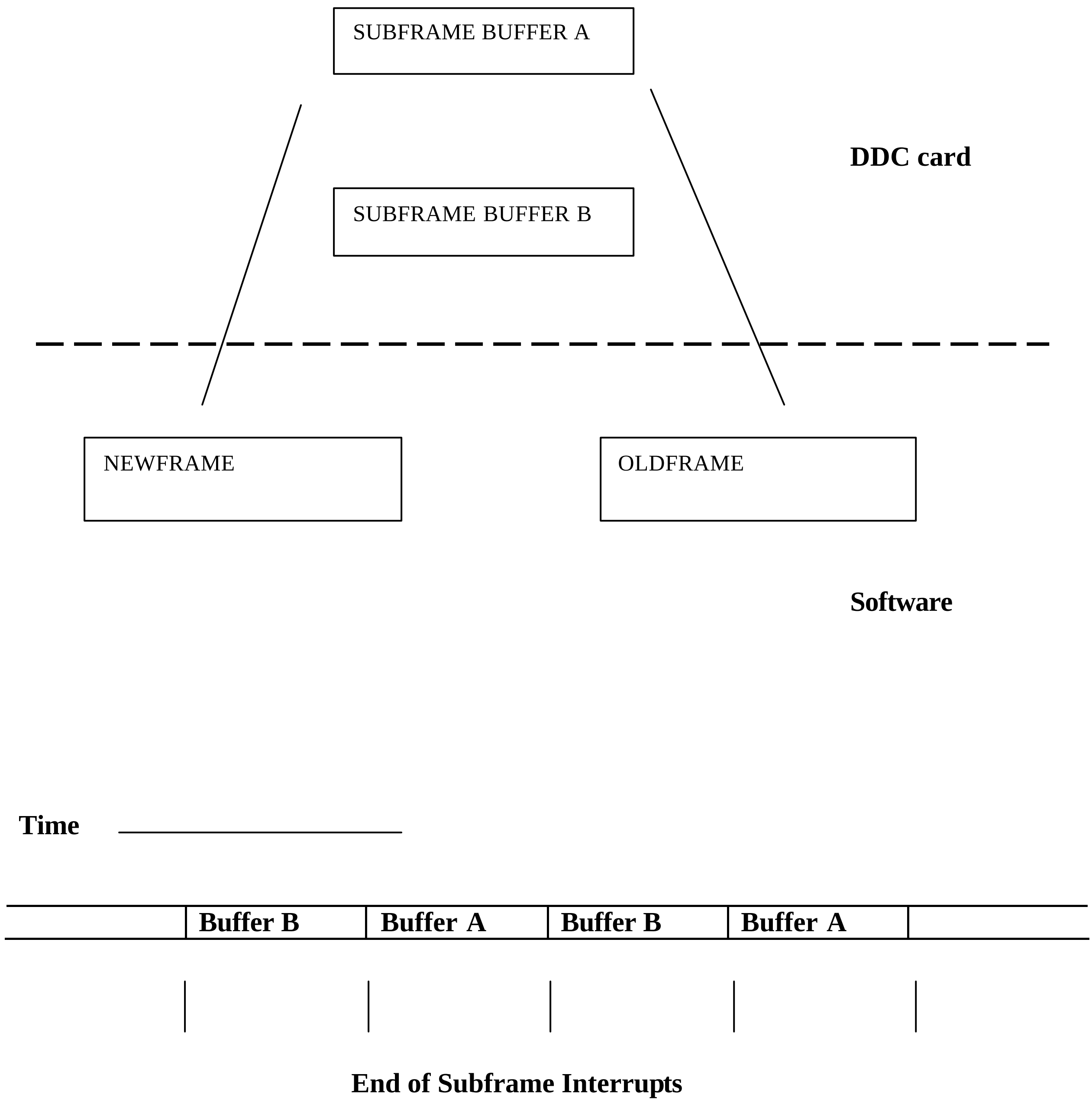
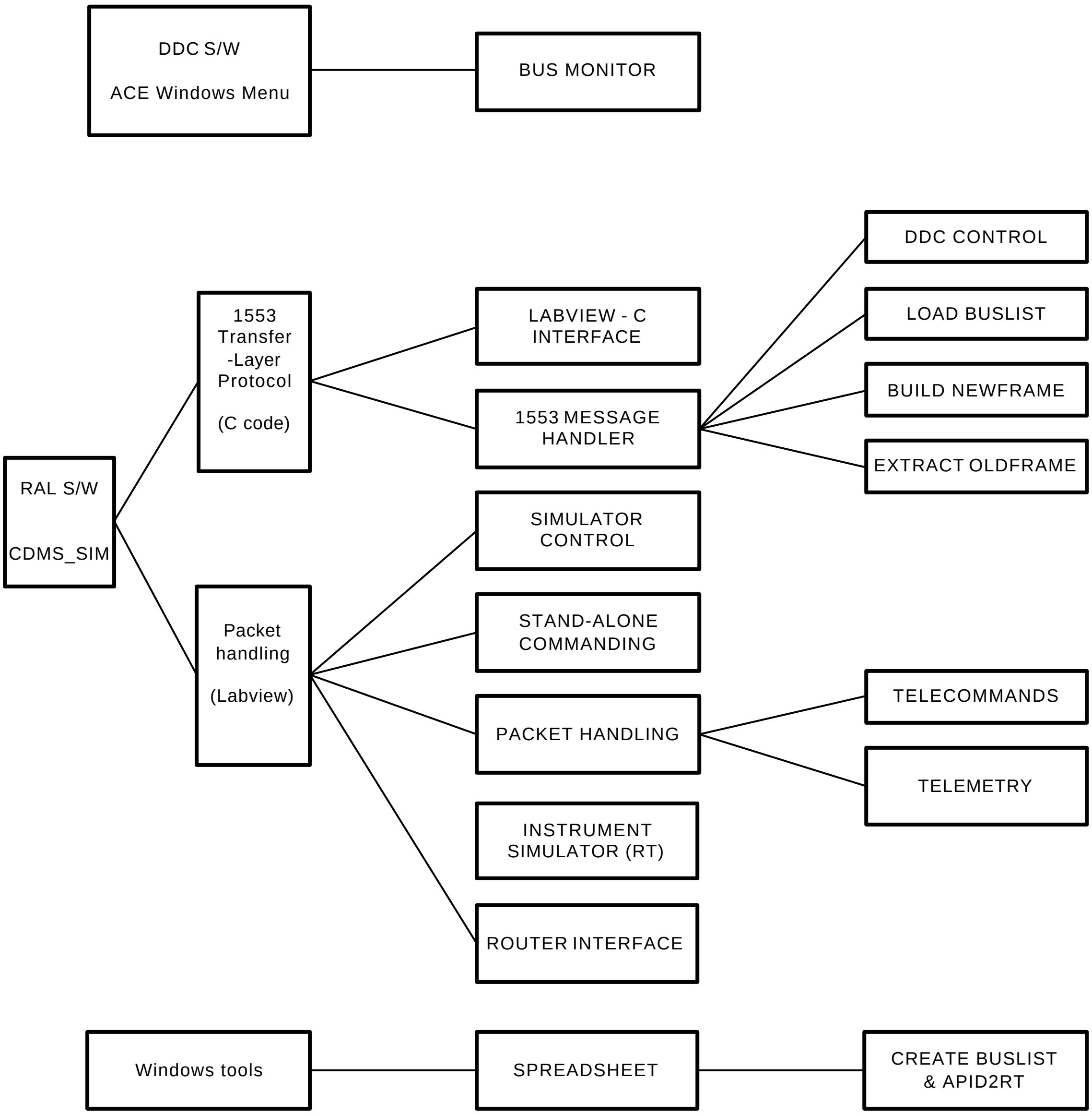


Figure 5-3: DDC Card data exchange

The 1553 card used in the simulator has a kind of processor (the ACE hybrid) which handles the transfer of messages over the bus. Message timing, and the overall length of a subframe is set up by software parameters before initiating the running of a subframe. Then the hardware takes over using the message types and data contents previously defined by the host PC.

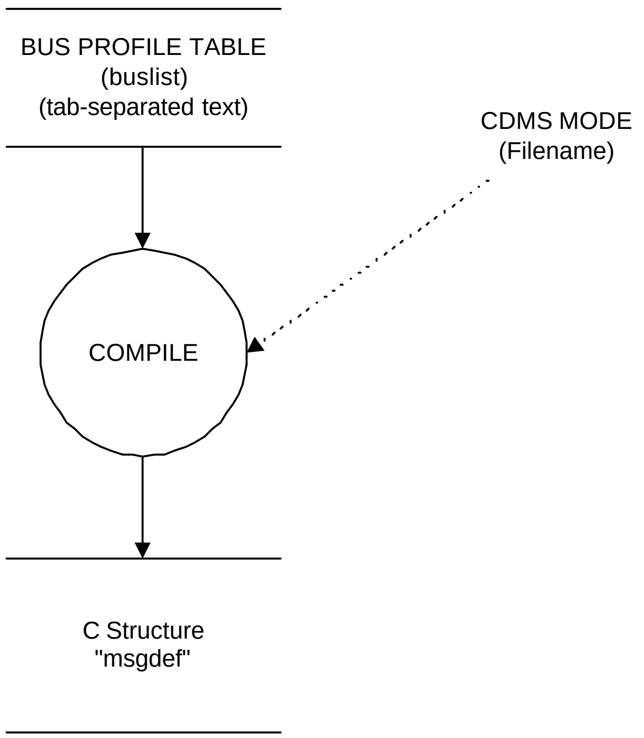
At the end of a 15.6 mS subframe an interrupt is generated to signal this fact to the host. Due to the limitations of a PC running NT messages cannot be exchanged with the card within the running period of a subframe . All of the data exchange takes place after the end of the subframe. For this reason the DDC library software has a double buffering system as illustrated in Figure 5-3. During the running of a subframe using the B buffers, data is extracted from the A buffers (from the subframe which has just run). Then the A buffers are filled with the messages required in the next subframe. At the end of subframe B control switches to the A buffers again.

5.5 Decomposition description



5.5.1 Load Buslist

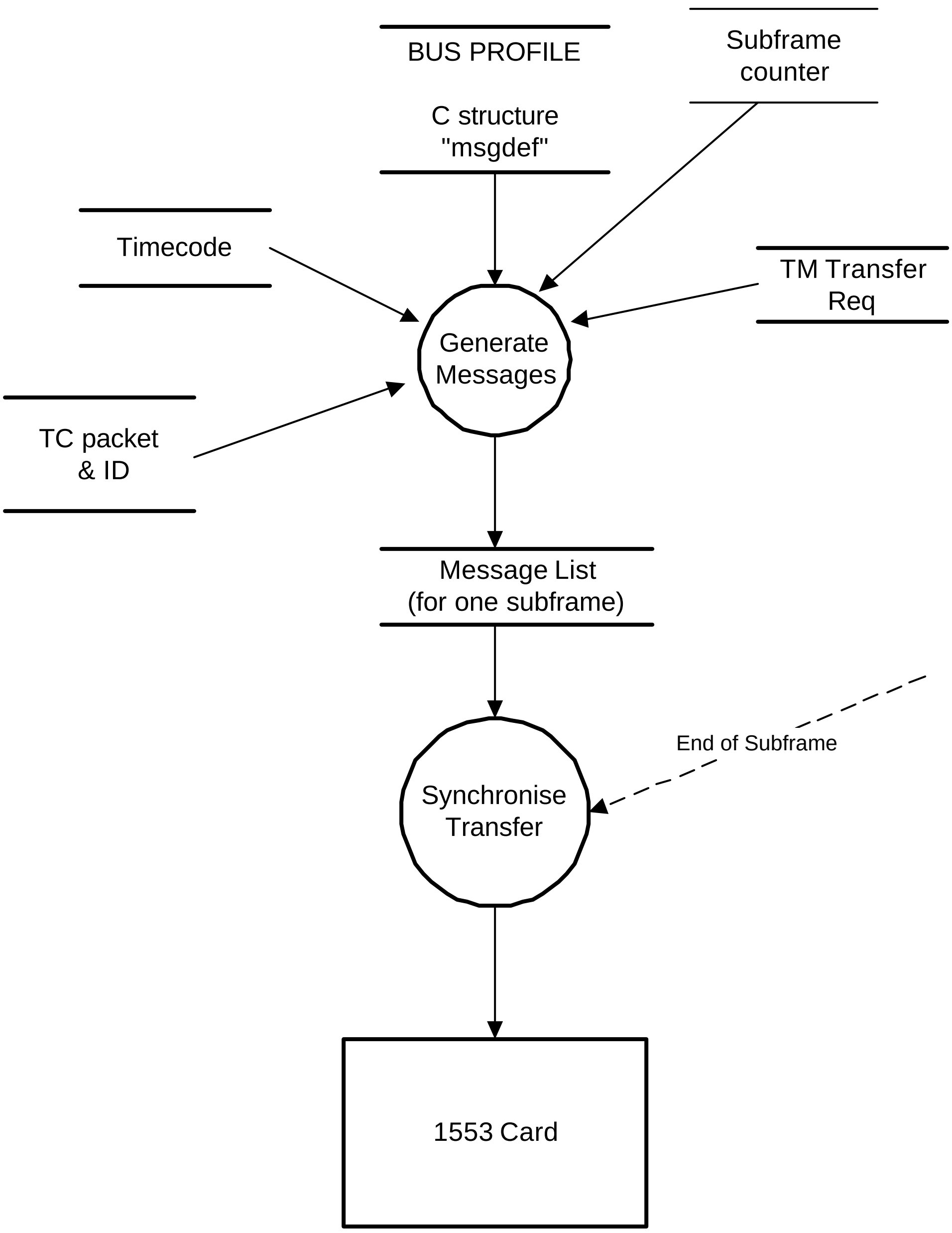
This function converts the Bus Profile source file, created as tab-separated text, into a C structure for use by the program. The structure is used to build a message list for each new subframe (build_newframe) and to extract the messages transferred over the bus (extract_oldframe).



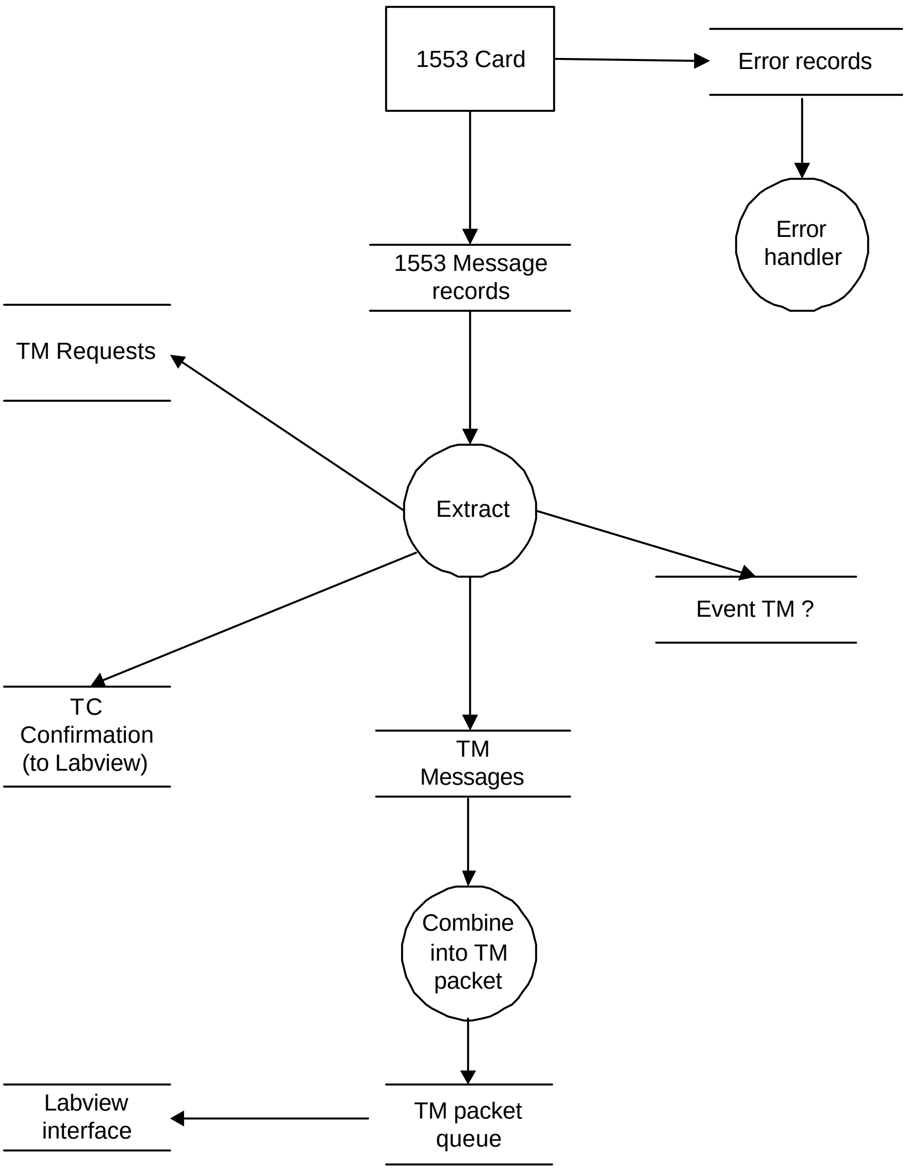
5.5.2 Build_Newframe

Build_Newframe takes the C structure “msgdef” as its starting point, and depending what telecommand and telemetry transfers are pending, builds the list of 1553 messages for the next subframe.

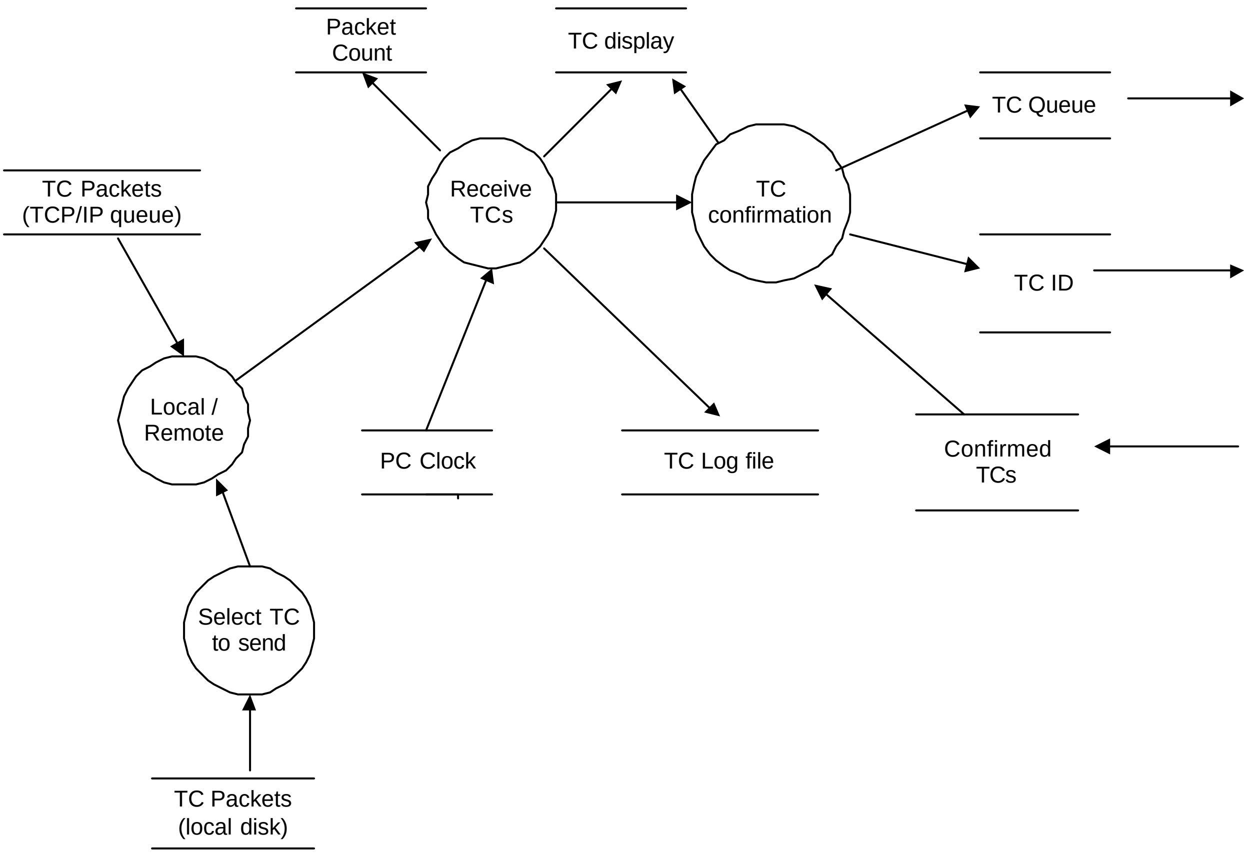
BUILD_NEWFRAME



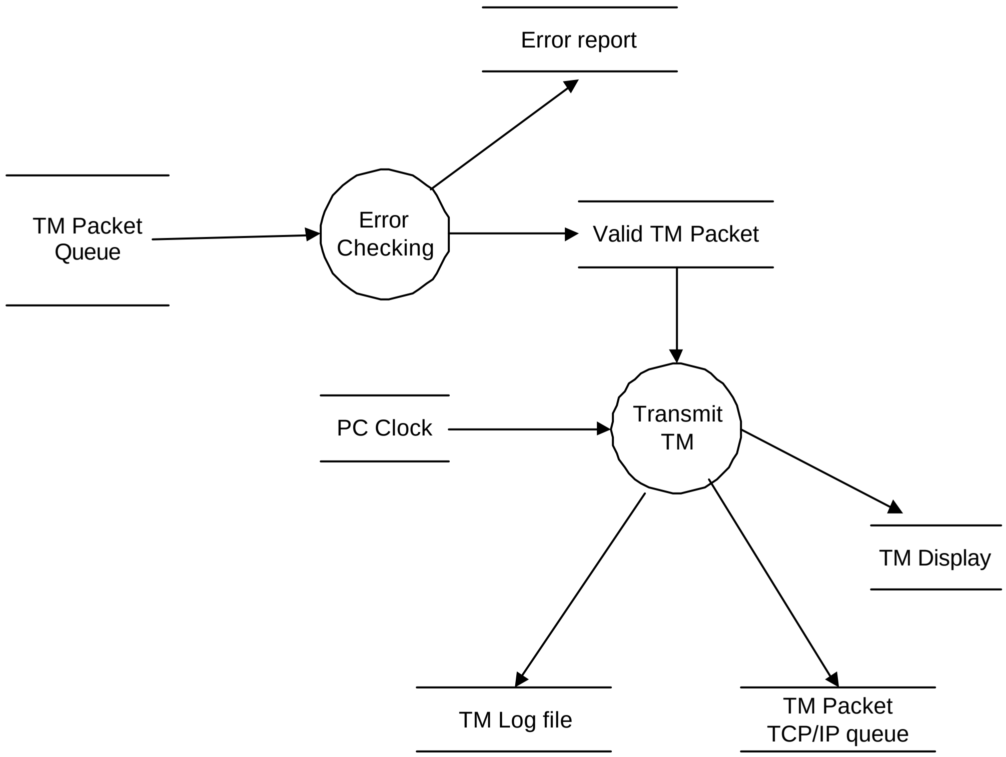
5.5.3 Extract_Oldframe



5.5.4 Packet Telecommand Handling

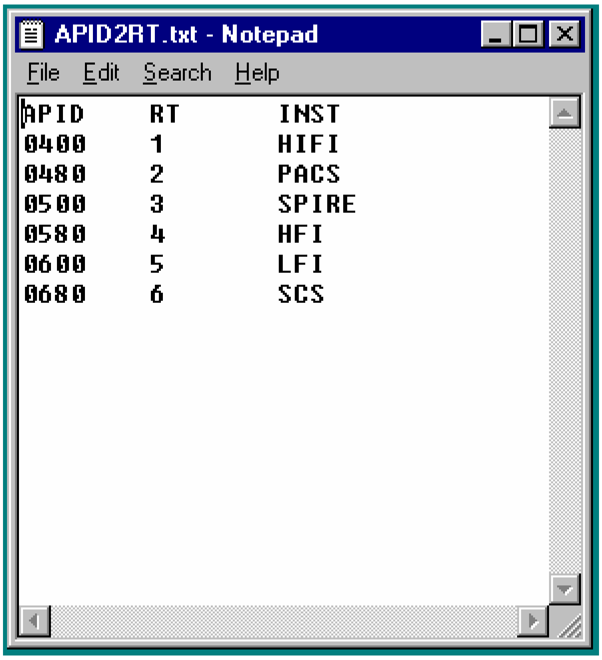


5.5.5 Packet Telemetry Handling



6. COMPONENT DESCRIPTION

This is the text file used as a lookup to see which Remote Terminal number corresponds to the Application ID in an incoming TC packet.



APID	RT	INST
0400	1	HIFI
0480	2	PACS
0500	3	SPIRE
0580	4	HFI
0600	5	LFI
0680	6	SCS

TM Transer Subframes

Message No.	0	1	2	3	4		19	20	21	22	23
Message Type	Broadcast: Sync with data Mode code 17	Status poll Or L.L cmd	Event TM	Event TM	Packet Transfer	Packet transfer continues	Packet Transfer	TM Confirmation	TM Request	TC Confirmation	regulatio
Message Duration (uS)	150	750	750	750	750		750	150	150	150	775
RT	31										
SA	0R				11T	to	26T	10R	10T	27T	

TC Transfer Subframes:

Subframe 0 – with sync

Message No.	0	1	2	3	4 - 7		20	21	22	23
Message Type	Broadcast Sync (no data) Mode code 1	Status poll Or L.L cmd	Event TM	Event TM	TC Packet Transfer	Unused	TC Descriptor	TM Request	TC Confirmation	regulatio
Message Duration (uS)	150	750	750	750	750		150	150	150	775
RT	31									
SA	0R		5T	6T	11R – 14R		27R	10T	27T	

Subframe 32 – with timecode

Message No.	0	1	2	3	4 - 7		20	21	22	23
Message Type	Broadcast: Sync with data Mode code 17	Status poll Or L.L cmd	Broadcast Time	Event TM	TC Packet Transfer	Unused	TC Descriptor	TM Request	TC Confirmation	regulatio
Message Duration (uS)	150	750	750	750	750		150	150	150	775
RT	31		31							
SA	0R		8R		11R – 14R		27R	10T	27T	

Subframes 16, 48

Message No.	0	1	2	3	4 - 7		20	21	22	23
Message Type	Broadcast: Sync with data Mode code 17	Status poll Or L.L cmd	Event TM	Event TM	TC Packet Transfer	Unused	TC Descriptor	TM Request	TC Confirmation	regulatio
Message Duration (uS)	150	750	750	750	750		150	150	150	775
RT	31									
SA	0R						27R	10T	27T	

Figure 6-1: Bus message sequence-CDMS normal mode

Microsoft Excel - Astrium buslist.xls

File Edit View Insert Format Tools Data Window Help

L24 =

	A	B	C	D	E	F	G	H
1	**Subframe**	**Message slot**	**Start time**	**Message type**	**RT**	**SA**	**Data type**	
2		(0 - 23)	(uS)	MCSync	(0 if TC)		None	
3				MCData			SyncFC	
4				BCtoRT			Timecode	
5				RTtoBC			PacketTC	
6							TCDesc	
7							TCConf	
8							PacketTM	
9							TMReq	
10							TMConf	
11							EventTC	
12							LLCmd	
13							EventTM	
14							StatusTM	
15								
16	0	0	0	MCSync	31	0	None	
17	0	4	2400	BCtoRT	0	11	PacketTC	
18	0	20	14400	BCtoRT	0	27	TCDesc	
19								
20	1	0	0	MCData	31	0	SyncFC	
21	1	4	2400	RTtoBC	1	11	PacketTM	
22	1	20	14400	BCtoRT	1	10	TMConf	
23	1	21	14550	RTtoBC	3	10	TMReq	
24								
25	2	0	0	MCData	31	0	SyncFC	
26	2	4	2400	RTtoBC	2	11	PacketTM	
27	2	20	14400	BCtoRT	2	10	TMConf	
28	2	21	14550	RTtoBC	1	10	TMReq	
29	2	22	14700	RTtoBC	0	27	TCConf	
30								
31	3	0	0	MCData	31	0	SyncFC	
32	3	4	2400	RTtoBC	3	11	PacketTM	
33	3	20	14400	BCtoRT	3	10	TMConf	
34	3	21	14550	RTtoBC	2	10	TMReq	
35								
36	4	0	0	MCData	31	0	SyncFC	
37	4	4	2400	RTtoBC	1	11	PacketTM	
38	4	20	14400	BCtoRT	1	10	TMConf	
39	4	21	14550	RTtoBC	3	10	TMReq	
40								
41	5	0	0	MCData	31	0	SyncFC	
42	5	21	14550	RTtoBC	1	10	TMReq	
43								
44	6	0	0	MCData	31	0	SyncFC	
45	6	21	14550	RTtoBC	2	10	TMReq	

Example of a buslist source file generated by Excel

```
0     0     0     MCSync      31    0     None
0     4     2400  BCtoRT      0     11    PacketTC
0     20    14400 BCtoRT      0     27    TCDesc

1     0     0     MCData      31    0     SyncFC

2     0     0     MCData      31    0     SyncFC
2     4     2400  RTtoBC      2     11    PacketTM
2     20    14400 BCtoRT      2     10    TMConf
2     22    14700 RTtoBC      0     27    TCConf

3     0     0     MCData      31    0     SyncFC
3     21    14550 RTtoBC      2     10    TMReq

4     0     0     MCData      31    0     SyncFC

5     0     0     MCData      31    0     SyncFC

6     0     0     MCData      31    0     SyncFC
6     21    14550 RTtoBC      2     10    TMReq

7     0     0     MCData      31    0     SyncFC

8     0     0     MCData      31    0     SyncFC

9     0     0     MCData      31    0     SyncFC
9     21    14550 RTtoBC      2     10    TMReq

10    0     0     MCData      31    0     SyncFC

11    0     0     MCData      31    0     SyncFC

12    0     0     MCData      31    0     SyncFC
12    21    14550 RTtoBC      2     10    TMReq

13    0     0     MCData      31    0     SyncFC

14    0     0     MCData      31    0     SyncFC
14    4     2400  RTtoBC      2     11    PacketTM
14    20    14400 BCtoRT      2     10    TMConf

15    0     0     MCData      31    0     SyncFC

16    0     0     MCSync      31    0     SyncFC
16    4     2400  BCtoRT      0     11    PacketTC
16    20    14400 BCtoRT      0     27    TCDesc

17    0     0     MCData      31    0     SyncFC

18    0     0     MCData      31    0     SyncFC
18    22    14700 RTtoBC      0     27    TCConf

19    0     0     MCData      31    0     SyncFC
19    21    14550 RTtoBC      2     10    TMReq
```

20	0	0	MCData	31	0	SyncFC
21	0	0	MCData	31	0	SyncFC
22	0	0	MCData	31	0	SyncFC
23	0	0	MCData	31	0	SyncFC
24	0	0	MCData	31	0	SyncFC
25	0	0	MCData	31	0	SyncFC
25	21	14550	RTtoBC	2	10	TMReq
26	0	0	MCData	31	0	SyncFC
27	0	0	MCData	31	0	SyncFC
28	0	0	MCData	31	0	SyncFC
28	21	14550	RTtoBC	2	10	TMReq
29	0	0	MCData	31	0	SyncFC
30	0	0	MCData	31	0	SyncFC
31	0	0	MCData	31	0	SyncFC
31	21	14550	RTtoBC	2	10	TMReq
32	0	0	MCData	31	0	SyncFC
32	2	900	MCData	31	8	Timecode
32	4	2400	BCtoRT	0	11	PacketTC
32	20	14400	BCtoRT	0	27	TCDesc
33	0	0	MCData	31	0	SyncFC
34	0	0	MCData	31	0	SyncFC
34	22	14700	RTtoBC	0	27	TCConf
35	0	0	MCData	31	0	SyncFC
36	0	0	MCData	31	0	SyncFC
37	0	0	MCData	31	0	SyncFC
38	0	0	MCData	31	0	SyncFC
38	21	14550	RTtoBC	2	10	TMReq
39	0	0	MCData	31	0	SyncFC
40	0	0	MCData	31	0	SyncFC
40	4	2400	RTtoBC	2	11	PacketTM
40	20	14400	BCtoRT	2	10	TMConf
41	0	0	MCData	31	0	SyncFC
41	21	14550	RTtoBC	2	10	TMReq
42	0	0	MCData	31	0	SyncFC
43	0	0	MCData	31	0	SyncFC

44	0	0	MCData	31	0	SyncFC
44	21	14550	RTtoBC	2	10	TMReq
45	0	0	MCData	31	0	SyncFC
46	0	0	MCData	31	0	SyncFC
47	0	0	MCData	31	0	SyncFC
47	21	14550	RTtoBC	2	10	TMReq
48	0	0	MCData	31	0	SyncFC
48	4	2400	BCtoRT	0	11	PacketTC
48	20	14400	BCtoRT	0	27	TCDesc
49	0	0	MCData	31	0	SyncFC
50	0	0	MCData	31	0	SyncFC
50	22	14700	RTtoBC	0	27	TCConf
51	0	0	MCData	31	0	SyncFC
51	21	14550	RTtoBC	2	10	TMReq
52	0	0	MCData	31	0	SyncFC
53	0	0	MCData	31	0	SyncFC
54	0	0	MCData	31	0	SyncFC
54	21	14550	RTtoBC	2	10	TMReq
55	0	0	MCData	31	0	SyncFC
56	0	0	MCData	31	0	SyncFC
57	0	0	MCData	31	0	SyncFC
57	21	14550	RTtoBC	2	10	TMReq
58	0	0	MCData	31	0	SyncFC
59	0	0	MCData	31	0	SyncFC
60	0	0	MCData	31	0	SyncFC
60	21	14550	RTtoBC	2	10	TMReq
61	0	0	MCData	31	0	SyncFC
62	0	0	MCData	31	0	SyncFC
63	0	0	MCData	31	0	SyncFC
63	21	14550	RTtoBC	2	10	TMReq

Prototype operator display