

AD-A076 481

TEXAS INSTRUMENTS INC DALLAS EQUIPMENT GROUP

F/G 17/7

GLOBAL POSITIONING SYSTEMS (GPS) HIGH DYNAMIC USER EQUIPMENT. (--ETC(U)
AUG 79

F04701-75-C-0180

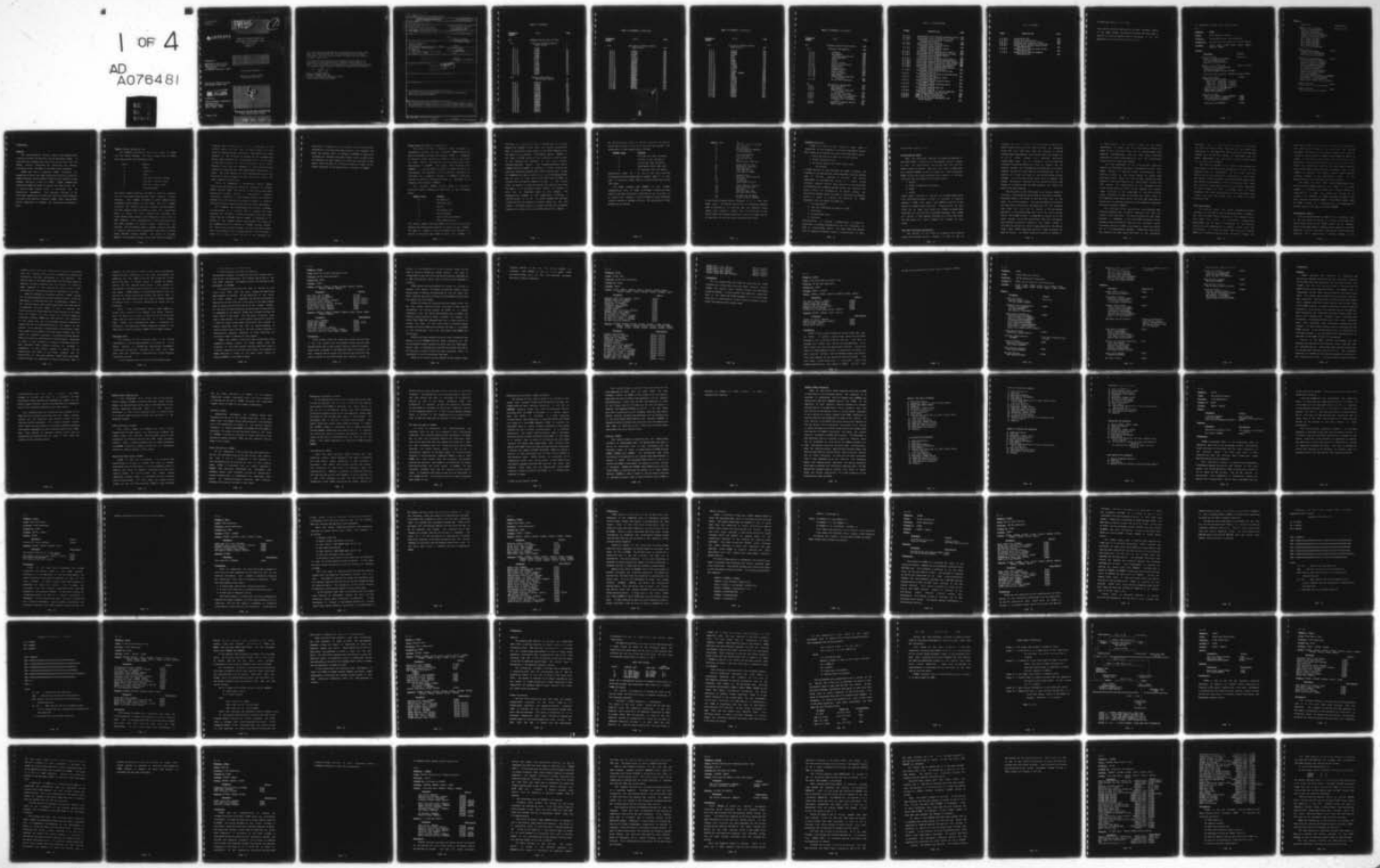
SAMS0-SD-TR-79-12-VOL-3

NL

UNCLASSIFIED

1 OF 4

AD
A076481



AIR FORCE REPORT
SD TR-79-12
VOLUME III

~~LEVEL III~~



AD A076481

GLOBAL POSITIONING SYSTEMS (GPS)
HIGH DYNAMIC USER EQUIPMENT (HDUE)
SET DESCRIPTION
VOLUME III

DDC
RECEIVED
NOV 14 1979
REGISTRY
E

Prepared for

DEPARTMENT OF THE AIR FORCE
Space and Missile Systems
Organization
Los Angeles, California 90009

Distribution Statement A

Approved for public release;
Distribution Unlimited

Contract No. F04701-75-C-0180
Data Sequence Number A003

DDC FILE COPY

Prepared by

TEXAS INSTRUMENTS INCORPORATED
Equipment Group
8001 Stemmons Freeway
Dallas, Texas 75266



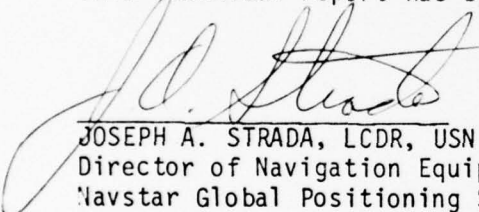
**TEXAS INSTRUMENTS
INCORPORATED**

1 August 1979

79 11 09 138

This final report was submitted by Texas Instruments Inc, Dallas, Texas, under Contract F04701-75-C-0180, with the Space and Missile Systems Organization, Air Force Systems Command, Los Angeles Air Force Station, Los Angeles, California.

This report has been reviewed by the Information Office (OIS) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations. This technical report has been reviewed and is approved for publication.



JOSEPH A. STRADA, LCDR, USN
Director of Navigation Equipment & Avionics
Navstar Global Positioning System

18 SAMSO

UNCLAS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER SD-TR-79-12 - Vol 3	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Global Positioning System (GPS) High Dynamic User Equipment (HDUE) Final Report, Volume III. Set Description, (CONT)	5. TYPE OF REPORT & PERIOD COVERED Final Report, Vol III June 1975-Aug 1979	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)	8. CONTRACT OR GRANT NUMBER(s) F04701-75-C-0180	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Texas Instruments Inc. Equipment Group 8001 Stemmons Freeway Dallas, TX 75266	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS F63421F	
11. CONTROLLING OFFICE NAME AND ADDRESS HQ SAMSO/YEN P.O. BOX 92960, Worldway Postal Center Los Angeles, CA 90009	12. REPORT DATE 1 Aug 1979	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 129375	13. NUMBER OF PAGES 367	
	15. SECURITY CLASS. (of this report) UNCLAS	
15a. DECLASSIFICATION DOWNGRADING SCHEDULE		
16. DISTRIBUTION STATEMENT (of this Report) "A" <i>This document has been approved for public release and sale; its distribution is unlimited.</i>		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Global Positioning System High Dynamics User Equipment Software and Module Description and Hardware.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This Volume Contains descriptions of both software/hardware modules in the HDUE system, which includes receiver software, master state control, navigation software, executive software., receiver LRU and navigation LRU descriptions.		

405 076

TABLE of CONTENTS

PARAGRAPH NUMBER	TITLE	PAGE
5.0	Software Module Descriptions	1
5.1	Receiver Software Module Description	2
5.1.1	R1MRC	2
5.1.2	R1AID	21
5.1.3	R1FMT	24
5.1.4	R1MTM	26
5.1.5	R1SRC	28
5.1.6	R1BSN	44
5.1.7	R1CAL	46
5.1.8	R1CC	48
5.1.9	R1DDT	51
5.1.10	R1NSE	55
5.1.11	R1PCK	56
5.1.12	R1PIN	61
5.1.13	R1RNG	64
5.1.14	R1RRM	72
5.1.15	R1SCH	73
5.1.16	R1SET	76
5.2	Master State Control Module Description	78
5.2.1	M1ADIS	78
5.2.2	M1CCIO	81
5.2.3	M1CMSC	85
5.2.4	M1CRNC	90
5.2.5	M1DBPR	92
5.2.6	M1IIUO	95
5.2.7	M1PDBR	97
5.2.8	M2BLNK	98
5.2.9	M2BUSO	99
5.2.10	M2CRUS	102
5.2.11	M2CVNM	104
5.2.12	M2DBS1	105
5.2.13	M2DBS2	107
5.2.14	M2DBS3	109
5.2.15	M2HPFP	111
5.2.16	M2IIUT	114
5.2.17	M2MOVE	117
5.2.18	M2STIN	118

TABLE of CONTENTS (continued)

PARAGRAPH NUMBER	TITLE	PAGE
5.3	Navigation Software Module Description	122
5.3.1	N1INIT	122
5.3.2	N1MCNS	123
5.3.3	N1MITK	125
5.3.4	N1NFLT	129
5.3.5	N1SVPN	130
5.3.6	N1XFRM	132
5.3.7	N2CGOQ	136
5.3.8	N2COVR	137
5.3.9	N2FCPO	140
5.3.10	N2FOTP	142
5.3.11	N2OPMO	146
5.3.12	N2HMTR	149
5.3.13	N2IONO	152
5.3.14	N2MCNI	155
5.3.15	N2MOGP	159
5.3.16	N2MOLD	162
5.3.17	N2MSLH	164
5.3.18	N2NEGS	166
5.3.19	N2NVIN	168
5.3.20	N2OSMG	172
5.3.21	N2SVEC	173
5.3.22	N2SVSL	179
5.3.23	N2LPCM	183

Accession For	
NTIS GMAI	<input checked="" type="checkbox"/>
DGC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

TABLE of CONTENTS (continued)

PARAGRAPH NUMBER	TITLE	PAGE
5.4	Executive Software Module Description	185
5.4.1	B1SLIO	185
5.4.2	B2MSTR	187
5.4.3	DATAN2	189
5.4.4	DCOS	190
5.4.5	DEXP	191
5.4.6	DSIN	192
5.4.7	DSQRT	193
5.4.8	EASHFT	194
5.4.9	IASHFT	195
5.4.10	X1COMM	196
5.4.11	X1COMN	197
5.4.12	X3ACT	198
5.4.13	X3CANC	200
5.4.14	X3ERR, X3ERRA	202
5.4.15	X3REL	204
5.4.16	X3REG	205
5.4.17	X3STOP	206
5.4.18	X3STOR	208
5.4.19	X3TIME	209
5.4.20	X3TIMM	210
5.4.21	X3WAIR	211
5.4.22	X3WAIT	212
5.4.23	X9JOIN	213
5.4.24	X9MCSP	214
5.4.25	X9MPMB	217
5.4.26	X9NAVP	218
5.4.27	X9RCVP	221
5.4.28	X9TRAP	223

TABLE of CONTENTS (continued)

PARAGRAPH NUMBER	TITLE	PAGE
6.0	Hardware Module Descriptions	224
6.1	Receiver LRU Modules	224
6.1.1	Wideband	224
6.1.2	Narrowband	230
6.1.3	Frequency Synthesizer	237
6.1.4	Code Generator	243
6.1.5	Output	251
6.1.6	Clock	259
6.1.7	Built-in-Test	264
6.1.8	Signal Distribution	271
6.1.9	Oscillator	274
6.1.10	Preamplifier	277
6.1.11	Antenna Switch	279
6.1.12	Power Supply	282
6.1.13	Microprocessor	287
6.1.14	Data Memory	296
6.1.15	Communication Register Interface	301
6.1.16	Serial Bus Interface	308
6.2	Navigation LRU Modules	319
6.2.1	Microprocessor	319
6.2.2	Data Memory	319
6.2.3	Communication Register Interface	319
6.2.4	Serial Bus Interface	319
6.2.5	Driver/Receiver Interface	320
6.2.6	Internal Bus Interface	326
6.2.7	Floating Point Arithmetic Unit	334
6.2.8	Data-Bus Extender Module	347
6.2.9	Power Supply	361

LIST of ILLUSTRATIONS

FIGURE	DESCRIPTION	PAGE
6. 1. 1-1	Functional Block Diagram Wideband Module	225
6. 1. 2-1	Functional Block Diagram of the NBM	231
6. 1. 3-1	Functional Block Diagram Frequency Synthesizer Module	238
6. 1. 4-1	Functional Block Diagram	244
6. 1. 5-1	Functional Block Diagram Output Module Digital Board	252
6. 1. 5-2	Functional Block Diagram Output Module, Analog Board	253
6. 1. 6-1	Functional Block Diagram Clock Module	260
6. 1. 7-1	Functional Block Diagram Built in Test	265
6. 1. 8-1	Functional Block Diagram Signal Distribution Module	272
6. 1. 9-1	Functional Block Diagram Clock Module	275
6. 1. 11-1	Functional Block Diagram Antenna Switch	280
6. 1. 12-1	Functional Diagram Receiver Power Supply	283
6. 1. 13-1	Functional Block Diagram Microprocessor	288
6. 1. 14-1	Functional Block Diagram Data Memory	297
6. 1. 15-1	Functional Block Diagram Communication Register Interface Module	302
6. 1. 16-1	Functional Block Diagram Serial Bus Interface Module	310
6. 2. 5-1	Functional Block Diagram Driver/Receiver Interface Module	321
6. 2. 6-1	Functional Block Diagram 1-Bus Interface Module	327
6. 2. 7-1	Simplified Diagram Floating Point Arithmetic Unit	335
6. 2. 7-2	Functional Diagram Floating Point Arithmetic Unit	336
6. 2. 8-1	Functional Diagram Data Bus Extender Module	348
6. 2. 8-2	Simplified Diagram of Typical Data Bus Extender Module Applications	349
6. 2. 8-3	DBEM to DBEM Configuration	350
6. 2. 8-4	DBEM to TILINE Configuration	351
6. 2. 9-1	Functional Diagram Navigation LRU Power Supply	362

LIST of TABLES

TABLE	DESCRIPTION	PAGE
6.1.13-1	Instruction Set	289
6.1.16-1	Status Bit Description	316
6.2.7-1	Floating Point Status Format	338
6.2.7-2	Overflow and Underflow Conditions	339
6.2.8-1	Remote Cycle Address Upper Bound Determination	353
6.2.8-2	Remote Cycle Address Lower Bound Determination	354
6.2.8-3	Remote Cycle Address Segment Determination	355

5.0 SOFTWARE MODULE DESCRIPTIONS

This section contains descriptions of each software module in the HDUE system. The Receiver modules are described in section 5.1 with the Master Control following in 5.2, the Navigation in 5.3 and the Executive in 5.4.

5.1 RECEIVER SOFTWARE MODULE DESCRIPTIONS

5.1.1

Mnemonic: RIMRC

Title: Multi Receiver Control

Priority: 160 MS Background (Non-reentrant)

Invoked By: Executive during power-up initializations.

Invokes: R1AID, R1FMT, R1MTM, R1SRC, X3ACT, X3WAIT,
B1SLID, X3ERRA

Inputs:

<u>Parameter</u>	<u>Source</u>
Executive Data	Executive
Power-up Semaphore(XGPRUP)	
FTF Mod 65536(XC0022)	
Reentrant Offset(XDATA)	
Master Control Messages:	Master Control
Buffer Protect Flag(RCV3B1)	
Command Word(RCV3B3)	
SV Queue(RCV3B5 through RCV3B9)	
Velocity Aiding Buffer (RCV3C3 through RCV3C7)	
Calibration Data:	R1MTM
Measured Pseudo-ranges	
Upper Ant, L1(RPL1A1 5 Words)	
Upper Ant, L2(RPL2A1 5 Words)	
Lower Ant, L1(RPL1A2 5 Words)	
Lower Ant, L2(RPL2A2 5 Words)	
Measured Range Rate(RPRATE)	
Measurement time(RPFTF)	
R1MTM Status(RPMTST)	
Acquisition Data	
Remaining Search Length(RSRMAN)	R1SCH
C/A Code Step Size(RSCINC)	R1SRC
P Code Step Size(RSPINC)	R1SRC
Response Word(RMSRCR)	R1SRC

Outputs:

<u>Parameter</u>	<u>Destination</u>
Master Control Message	Master Control
Buffer Protect(RCV2B1)	
Update Flag(RCV2B2)	
RCVR LRU Status(RCV2B3)	
Command Acknowledge(RCV2B5)	
Command Complete(RCV2B6)	
SV 1 Status(RCV2B8)	
SV 2 Status(RCV2B9)	
SV 3 Status(RCV2BA)	
SV 4 Status(RCV2BB)	
SV 5 Status(RCV2BC)	
Precision RCVR(RCV2BD)	
R1SRC Control	R1SRC
Command Word(RMSRCD)	
Acquisition Mode(RMSVMD)	
Calibration Term(RMMTD)	
Acquisition Parameters	R1SRC
Antenna Selection(RSANT)	
Code Tap Selection(RSCACD)	
Doppler Frequency(RSDOPP)	
Aiding Semaphore(RSFLAG)	
L1/L2 selection(RSFREQ)	
GT Selection(RSINVR)	
Doppler Bin Offset(RSOFST)	
Code Phase Reset(RSPHRS)	
Time Bias(RPBIAS)	
Estimated Range(RSRNG1 and RSRNG2)	
Aided Search Command(RSSCHM)	
Code Search Length(RSSLTH)	
Generic SV Identification(RSVID)	
R1AID Interface	R1AID
Total Aiding Inhibit(RPADST)	
R1MTM Interface	R1MTM
Delay Table Pointer(RPMTPT)	

Processing:

General

The Multi-Receiver Control task is the highest level controller within the Receiver Control Subsystem (RCSS). It receives mode commands and acquisition aiding data from the Master Control Subsystem and supervises the resource allocation within the RCSS to implement these commands.

R1MRC and the 5 reentrant R1SRC processes run sequentially within the same background priority so that all R1SRC processes receive R1MRC directives and respond to them between execution cycles of R1MRC. R1MRC commands the individual R1SRC processes to acquire and track an SV, GT, or Built-in-test signal using a particular set of acquisition parameters. This sequence consists of an individual Doppler bin of specified code chip width centered at a particular Doppler frequency. R1MRC takes appropriate action depending on whether the acquisition succeeds or fails.

Master Control Communications

All commands from Master Control are routed to R1MRC via the RCV3B message. The basic command word is RCV3B3 which may assume the following values:

VALUE	MEANING
1	Cancel
2	Total Kill
3	TTDC
4	Conc Sch: C/A Acq, C/A Rng
5	Conc Sch: C/A Acq, P Rng
6	Conc Sch: P Acq, P Rng
7	Precision Mode

The cancel command instructs the RCVR to cancel all channels not in a legitimate tracking mode. Other channels continue tracking. This command precedes all other commands which the Master Control sends to the RCVR. The kill command instructs the RCVR to cancel all channels unconditionally. The TTDC command causes the RCVR to enter the calibration mode in which it will measure and calculate the inter-channel delay terms at each frequency (L1/L2) and on both antennas. This sequence also confirms the ability of the RCVR hardware to execute signal acquisition and tracking. The Concentrated Search command instructs the RCVR to acquire and track the constellation specified in the SV queue (RCV3B5 through RCV3B9). The queue is reviewed against the present tracking status when the new command is

received. Any SV numbers which do not correspond to the current status are put into the search queue. If a zero value is received the SV previously associated with that location in the SV queue is dropped and its corresponding channel made available for concentrated search. The SV queue is used only on entry to the Concentrated Search state. Precision mode may be commanded if four SV'S are being tracked on P-code and the fifth RCVR channel health word is valid. The fifth channel will then sequentially acquire and track the L1 and L2 signals for each of the SV'S which the other channels are tracking.

Prior to commanding a Concentrated Search, Master Control ensures that valid aiding is available in buffers RCV3C through RCV3F. This aiding is used during the signal acquisition process. During a reacquisition sequence (following completion of Concentrated Search) this data is assumed to be of sufficient accuracy to support a direct-p acquisition with a search window of plus/minus 1000 meters.

All commands are acknowledged upon receipt in RCV2B5. An acknowledge of -1 implies that a command has been rejected by the RCVR. A normal acknowledge echoes the same value received in RCV3B3. When a command is completed it is echoed in the command complete word RCV2B6. The status of all commanded SV'S is contained in RCV2BB through RCV2BC. For that array a positive number indicates the RCVR channel which is assigned to the corresponding SV in the SV queue. A value of 0 implies that the SV is not selected for

acquisition. A negative value indicates that the SV has been selected for a Concentrated Search which is in progress. The RCVR LRU health word is returned in RCV2B3. A value of 0 indicates all channels have good health. A bit is set to one corresponding to each channel which has been declared in bad health. When the RCVR is in Precision Mode the number of the channel assigned to the sequencing is returned in RCV2BD.

Single Receiver Control Communications

The control of the 5 reentrant RISRC processes is implemented through the two variables RMSRCD (Directive Word) and RMSVMD (SV Acquisition Mode). RISRC is commanded to execute a single acquisition sequence based on the values of these control variables and the SV related acquisition data passed in RSCOMM. The Single RCVR Response Word (RMSRCR) is used by RISRC to report to RIMRC the status of the sequence. All sequences run to successful completion as indicated by a Ranging response in RMSRCR or they fail because of system or hardware problems as indicated by a negative value passed in RMSRCR.

The variable RMSRCD directs RISRC to execute a particular type of sequence as described in the following table:

<u>RMSRCD VALUE</u>	<u>MEANING</u>
0	No Operation
1	Absolute Cancel
2	New SV
3	New Doppler Bin
4	Switch Antenna
5	Switch Freq and Antenna
6	Start Reacquisition

The Cancel command instructs RISRC to drop all activity and prepare the hardware and software to receive a new command. The New SV command is used when RIMRC first switches to a new SV in the queue. When RISRC receives this command, it

retrieves all acquisition data in RSCOMM data set. The New Doppler Bin command is used when the present acquisition data is valid and a rapid switch of the VCXO frequency to a new search bin is desired. The Antenna Switch command is not used by R1MRC since all such switching is done as part of the reacquisition sequence or the more general Frequency/Antenna switch. The Frequency/Antenna Switch command instructs R1SRC to perform a direct acquisition using the specified antenna and L-band control logic. It may be commanded only when a source is in a valid Ranging state, and as such is used only in the TTDC state of R1MRC. It differs from the New SV command in 2 ways: 1. The current value of Doppler is read and used as the starting value for the search. 2. If the built-in-test module is running it will not be restarted based on this command. The Reacquisition command is used immediately following a loss-of-signal on an SV. It causes hardware and software initializations to expedite a direct-P search for the lost signal. The R1SRC process is stepped through the subsequent reacquisition search bins by the New Doppler Bin command.

The SV Acquisition Mode word (RMSVMD) specifies the type of source and the type of code to be acquired and tracked. The values which it may assume are as follows:

<u>RMSVMD VALUE</u>	<u>MEANING</u>
1	C/A Code Acq, Rng, SV Source
2	C/A Acq, P Rng, SV Source
3	P Code Acq, Rng, SV Source
4	RCVR Health Check (Not used)
5	X1 Code Acq, Rng, BIT Source

Acquisitions modes of 1, 2, and 3 are used with the corresponding Concentrated Search command. Mode 3 is also used in Precision Mode. Mode 5 is used exclusively in the TTDC state.

The R1SRC response word RMSRCR is the primary communication word to R1MRC from R1SRC. A positive value usually indicates successful completion of all or a part of an acquisition sequence, while a negative value indicates either a system or hardware failure. The values which it may assume are as follows:

<u>RMSRCR VALLE</u>	<u>MEANING</u>
8	Absolute Cancel Complete
7	BIT Ranging
6	RHC Complete
5	Initialization Complete
4	P-code Ranging
3	C/A code Ranging
2	P Carrier Acq
1	C/A Carrier Acq
0	Null
-1	Output Module ADC Fail
-2	Range-rate Ctr Fail
-3	Data Clock Inactive
-4	VCXO DAC Fail
-5	Code Search Fail
-6	C/A Loss-of-signal
-7	Handover Fail
-8	Excessive Parity Errors
-9	Not Used
-10	P Code LOS
-11	Weak Signal Hold-on
-12	Barker Search Time-out
-13	VCXO Calibrate Fail
-14	Not Used
-15	Data Bit Sync Fail
-16	Bit Sync Verify Fail
-17	SV Data Out of Range
-18	Directive Word Ignored

In the Initialization state a response of 5 is the only legal value. In TTDC an acquisition is considered complete after a response of 7 is received and any negative value causes that channel's health word to be set bad. In the other states the response handling is done via table look-up as described in the individual state sections below.

Software Structure

R1MRC is a state driven controller whose state is determined by the commands it receives from Master Control. Each state has a similar structure with three basic parts.

1. Entry section with data base initialization.
2. R1SRC response handler.
3. Directive handler and other state unique processing.

A common section of code processes all R1SRC responses and branches to the appropriate state dependent response handler when a non-zero response is received. This section terminates by transferring control to a state unique directive handler section. Each execution cycle terminates by releasing the incoming message buffer and clearing the output buffer for transmission to Master Control. All transitions between states are made in the Directive Handler section in order to simplify the handling of R1SRC responses. The five states of R1MRC are:

1. Initialization
2. Test and Time Delay Calibration (TTDC)
3. Idle/Track
4. Concentrated Search
5. Precision

Initialization is entered automatically at power-up. Idle/Track is entered automatically at the completion of TTDC or Concentrated Search. The other modes are entered only on command from Master Control. A description of each

of the R1MRC states follows.

Initialization State

When the Executive executes its power-up sequence it activates R1MRC, initializes all designated volatile memory to zero, and synchronizes the system time (FTF) with the Master Control Processor. When this has been accomplished the variable XQPRUP is set to a value of 1. Upon activation R1MRC initializes those variables which require a non-zero value and then activates the following tasks:

1. B1SLIO
2. R1SRC (5 Reentrant Processes)
3. R1FMT
4. R1AID

These tasks continue to execute on a steady-state basis until the RCVR Processor is reset or executes a power-up sequence. R1MRC then waits until XQPRUP is set to +1 and begins its steady state execution. Transition out of this state can be made only after all 5 R1SRC processes have responded with Initialization Complete messages. Then R1MRC will accept a TTDC command preceded by a cancel and enter the TTDC state. All other commands are illegal in this state.

Test And Time Delay Calibration

The purpose of this state is to measure the relative delay terms between various channels in both L1 and L2

tracking and with respect to both antenna preamplifiers. Pseudo-range correction terms (RPL1A1 array) are generated which are such that when they are applied to raw SV pseudo-range measurements, coincidentally arriving signals (L1 or L2 on either preamp) will generate equivalent pseudo-range data. This is necessary since the RCVR has different processing delays on L1 and L2 signals and the cabling between the preamps and the RCVR may be different lengths. The Built-in-test module is modulated with the X1 portion of P-code at both L1 and L2 and cabled to both preamps. It is the synchronous source used to measure the delays. Successful execution of this sequence gives a high degree of confidence that the RCVR hardware can handle SV acquisition and tracking.

The execution of the TTDC state is as follows. R1AID is inhibited from updating aiding data to individual channels. Estimated range is set equal to zero and time bias is set equal to 6.5 milliseconds. This forces R1PIN to initialize the data epoch sufficiently far away from an FTF boundary that R1MRC will not have to resolve a 20 millisecond ambiguity in the pseudo-range data from the built-in-test module. Each R1SRC process is commanded to acquire L1 on the upper antenna via an MTD acquisition. When all 5 processes have reported successful ranging on the signal, R1MTM is activated to provide the first 5 measurements for the RPL1A1 array. When R1MTM completes execution, R1MRC processes the data as follows. All delay terms are relative to channel 1,

L1, Upper Antenna. This raw data is saved as the master measurement in MTNULL and the first entry in RPL1A1 set equal to zero. Since the first set of pseudo-ranges were measured simultaneously, the next four delay terms are generated by simply subtracting the four raw pseudo-range values from the master measurement.

The code generator in the built-in-test module is started simultaneously with and in alignment with the first RISRC process to execute P-code initialization. This start-up is based on the frozen values of aiding data described above. Once the built-in-test module has been started it must be allowed to run continuously during the TTDC process. The relationship of its alignment to the frozen aiding data will drift in time, however, as a nearly linear function of the range-rate of the built-in-test signal. Therefore in order to minimize the search time of the remaining three acquisitions, the time bias term used in acquisition is adjusted according to the measured range-rate recorded by R1MTM in RPRATE.

The remaining three acquisitions and subsequent data handling sequences are virtually identical and only the second is described here. The second acquisition (L2, upper antenna) is commanded on all five RISRC processes via the Frequency Switch directive. When all five processes report successful tracking, R1MTM is activated to record the second set of five measurements (RPL2A1). R1MRC must then modify this data for the same reason described in the preceding

paragraph. That is, when two pseudo-ranges are subtracted for use as calibration data, they must both be effectively referenced to the same time. After the drift term has been removed, each of the five values is subtracted from the master measurement and the result is the five delay terms for L2 on the upper antenna.

The TTDC will restart up to 5 times if any problems are encountered in the acquisition sequences or if any of the measured data is unreasonable. On the fifth cycle any channel which fails acquisition is declared in bad health and Master Control notified via the LRU Health Status word. Any channel so designated in TTDC or during any other state may be reset to good health only by a successful completion of a TTDC sequence. If TTDC is aborted prior to normal completion the most significant bit of the LRU Health word is set to one and the system will be unable to enter Precision Mode.

Idle Track State

The Receiver enters this state automatically after completion of the TTDC or the Concentrated Search sequence. Any time the RCVR receives a Cancel or Kill command it will also enter this state. In Idle/track steady-state measurement and SV data is recovered for all tracking sources and this data passed to Master Control each 320 milliseconds. If a loss of signal (LOS) occurs on a source RIMRC will initiate a reacquisition procedure through the

associated R1SRC process. No other channels are involved in the reacquisition sequence. The precision time aiding variables described in the R1PIN section are used to initialize the P-code once. R1MRC directs R1SRC to search a window of plus/minus 1000 meters on both antennas, first searching on the opposite antenna from the one on which the source was previously tracking. The total search sequence takes approximately 8 seconds and is repeated until a new command is received from Master Control. Master Control normally commands a new Concentrated Search if the Reacquisition is not successful within 30 seconds.

The RCVR will honor TTDC, Concentrated Search, and Precision Mode commands while in this state. If a cancel command is received, the RCVR will drop all sources in reacquisition. A total kill causes all sources to be dropped. The handling of R1SRC responses is based on the table starting at address IDLRSP in the R1MRC program. When an R1SRC process is in reacquisition, its responses are handled based on the table at address REARSP.

Concentrated Search

The Concentrated Search command is the mechanism used to add and delete SV sources. It is only in this mode that the SV status words in the RCV2B buffer can be changed. Three Concentrated Search commands may be passed to the RCVR depending on which code type is desired for acquisition and tracking. The P acquisition P ranging command is not

normally useful even when a Navigation solution is available with four sources since a lack of ephemeris on the new SV prevents an accurate calculation of its position. The control logic in the RCVR associated with the three types of searches is nearly identical and, as a result, all three are legal commands to the receiver and there is only 1 state associated with Concentrated Search regardless of the type.

When the command is initially received, R1MRC reviews the SV queue against the current tracking status. If any SV PRN number is different than its corresponding element in the previous queue, the channel associated with that source is cancelled. If an SV number of zero is received, the source previously associated with that location in the queue will be dropped. At this point all SV'S are tagged as tracking or queued. R1MRC will then start at the beginning of the queue and use all available receiver channels to search for the first untracked source. The nature of the pattern to be searched is determined by calculating a table entry based on the number of sources being tracked and the number of channels available. The patterns become smaller after 2 sources are tracking since the Navigation Subsystem is able to use 2 sources to produce a degraded solution. Each available RCVR channel (process) is assigned a subpattern of the overall pattern to search. All SV acquisitions are done on the upper antenna and GT acquisitions on the lower antenna. R1MRC steps each R1SRC process through its subpattern as acquisitions fail. When 2

channels are idle after completing their entire subpatterns without success, R1MRC moves to the next non-tracking SV repeating all the steps it did for the initial SV. At the end of the queue, a status report is passed to Master Control and the CDU and source status is then updated for the operator. Processing then resumes at the beginning of the queue. R1MRC remains in Concentrated Search until all sources are successfully being tracked. Any source that is acquired and then lost during this state is merely requeued for search since a direct-P reacquisition would not be practical.

While in the Concentrated Search state, R1MRC will accept only a cancel or kill command from Master Control, each of which will cause a transition to the Idle/track state. When all SV'S in the queue are in a valid ranging mode, R1MRC will automatically make a transition to Idle/track. The handling of R1SRC responses is based on the table which starts at address CONSCH in the R1MRC program.

Precision Mode

The purpose of the Precision Mode is to provide simultaneous L1 and L2 measurements on all SV'S for use by Master Control in determining appropriate Ionospheric corrections for each SV. The RCVR will enter this state only from the Idle/track state and only if the following conditions are met:

1. Four channels are in a valid P-code ranging mode.

2. All channels have good health.

3. TTDC has been completed successfully.

The available channel is designated the fifth channel and is used for all the sequencing. This channel may be any of the five RCVR channels, and Master Control is notified of the assignment via RCV2BD.

The sequencing for Precision Mode is driven by the local clock (FTF) with the basic sequencing time being approximately 10 seconds. At the beginning of the cycle, the fifth channel is commanded to perform a Direct-p L1 acquisition on the first SV in the queue on the same antenna that is being used to track the SV on the primary channel. At the next 10 second boundary an equivalent L2 acquisition is commanded on the same SV. R1MRC then proceeds through the other 3 SV'S in the queue. Thus the total Precision cycle time is 80 seconds. If a fifth channel acquisition fails or if loss-of-signal occurs after acquisition, the channel is simply cancelled until the next 10 second boundary. If loss-of-signal occurs on any of the four primary channels, a reacquisition sequence identical to that described for Idle/track state is started on that channel.

R1MRC will remain in Precision mode indefinitely until commanded by Master Control to change states. With the exception of the fifth channel tracking data this state is virtually identical to the Idle/track state. The handling of R1SRC responses is based on the table which starts at address PRERSP in the R1MRC program.

5.1.2

Mnemonic: R1AID

Title: Receiver Aiding Computation Task

Priority: 20 MS (Non-reentrant)

Invoked By: R1SRC

Invokes: X3WAIT

Inputs: XC0022, XCM016, RCV3E3, RCV3D3, RCV3C3, RCV3F3,
RPSVIX, RSFLAG, RSOFST

<u>Parameter</u>	<u>Source</u>
FTF count LSW (XC0022)	X1IT20
FTF count mod 16 (XCM016)	X1IT20
Range aiding (RCV3E3)	Master control
Alternate range aiding (RCV3F3)	Master control
Velocity aiding (RCV3C3)	Master control
Alternate velocity aiding (RCV3D3)	Master control
SV oriented index (RPSVIX)	R1MRC
R1AID update flag (RSFLAG)	R1MRC
Doppler bin offset (RSOFST)	R1MRC

Outputs: RSRATE, RSRNG1, RSRNG2, RPBIAS, RPZ01, RPZ02, RPN01,
RPN02, RPADST

<u>Parameter</u>	<u>Destination</u>
R1AID status (RPADST)	R1MRC
Range-rate (RSRATE)	R1SET, R1SCH
Range MSW (RSRNG1)	R1PIN
Range LSW (RSRNG2)	R1PIN
Precision time bias (RPBIAS)	R1PIN
Precision time Z-count (RPZ01, RPZ02)	R1PIN
Precision time FTF count (RPN01, RPN02)	R1PIN

Processing:

R1AID accepts range and range-rate aiding data for each of the five generic SV's from master control and provides scaled aiding data to the single receiver control processes. Aiding data is passed to the RCSS each 320 milliseconds. On even numbered 320 ms frames the velocity and precision time data is placed in the RCV3C buffer and position aiding is

placed in the RCV3E buffer. On odd numbered frames the the data is placed in RCV3D and RCV3F buffers. The data is transmitted during the first two 20 millisecond subframes of a 320 millisecond period and is valid on the subframe 11 boundary.

R1MRC passes the array RPSVIX to R1AID to provide a mapping from generic SV number to Receiver channel. Input aiding data is ordered corresponding to the SV queue but RCVR channels may have arbitrary SV assignments during both the search and track modes.

R1AID calculates an acceleration term based on the old and new values velocity aiding. This term is then used to extrapolate the aiding passed to R1SCH in a manner compatible with the 80 ms VCXO update cycle in R1SCH. The velocity aiding passed to R1SCH includes the doppler bin offset that R1MRC has commanded and thus is used directly to control the VCXO. If the final value of velocity aiding exceeds plus/minus 2000 meters/second the data is truncated to the appropriate limit and the status flag RPADST set accordingly.

R1AID also takes updated precision time information and passes it to the RPCOMM data set, after correcting for the 10 microsecond offset between the 20 ms and 5 ms interrupts. Aiding data is referenced to the 20 millisecond time marks but the data is used to start the code generator which is referenced to the 5 millisecond time mark.

R1AID will inhibit aiding updates to any channel whose

variable RSFLAG is set true. All aiding updates are suspended when RPADST is set to +1 by R1MRC. After activation R1AID will run until the Receiver processor executes a power-up sequence.

5.1.3

Mnemonic: R1FMT

Title: Format Task

Priority: 20 MS (Non-reentrant)

Invoked by: R1MRC

Invokes: X3WAIT

Inputs: XDATA, RMRDYF, RMRSTA, RMSVID, RMCMD, RMFREQ, RMANT,
RMMDR, RSRDYF, RSSFID, RSWCNT, RSMSW1, RSMSW2, RSDTVL

<u>Parameter</u>	<u>Source</u>
Receiver data base lengths (XDATA)	X9RCVP
RMMCOMM ready flag (RMRDYF)	R1SRC
Receiver status (RMRSTA)	R1SRC
Satellite ID (RMSVID)	R1SRC
Commanded mode (RMCMD)	R1SRC
Frequency (RMFREQ)	R1SRC
Antenna (RMANT)	R1SRC
Pseudo-range correction (RMMDR)	R1SRC
RMMCOMM ready flag (RSRDYF)	R1PCK
SV subframe ID (RSSFID)	R1PCK
SV message word number (RSWCNT)	R1PCK
SV data word number 1 (RSMSW1)	R1PCK
SV data word number 2 (RSMSW2)	R1PCK
SV data validity (RSDTVL)	R1PCK

Outputs: RCV2AE, RCV2A2, RCV2A3, RCV2A4, RCV2A5, RCV2A6, RCV2AD,
RMRDYF, RCV2A7, RCV2A8, RCV2A9, RCV2AA, RCV2AB, RSRDYF,
RCV2A1, RCV2C1, RCV2D1, RCV2D1, RCV2E1, RCV2C2, RCV2E2

<u>Parameter</u>	<u>Destination</u>
Update flag (RCV2AE)	Master control
Receiver status (RCV2A2)	Master control
Satellite ID (RCV2A3)	Master control
Commanded mode (RCV2A4)	Master control
Frequency (RCV2A5)	Master control
Antenna (RCV2A6)	Master control
Receiver time delay (RCV2AD)	Master control
RMMCOMM ready flag (RMRDYF)	Master control
SV subframe number (RCV2A7)	Master control
SV message word number (RCV2A8)	Master control
SV data word number 1 (RCV2A9)	Master control
SV data word number 2 (RCV2AA)	Master control
SV data validity code (RCV2AB)	Master control
RMMCOMM ready flag (RSRDYF)	R1SRC
RCV2A status word (RCV2A1)	Master control

RCV2C status word (RCV2C1)	Master control
RCV2D status word (RCV2D1)	Master control
RCV2E status word (RCV2E1)	Master control
Pseudo-range MSW (RCV2C2)	Master control
Pseudo-range-rate MSW (RCV2E2)	Master control

Processing:

On an appropriate time frame for each receiver, R1FMT transfers all output data from the single receiver control (R1SRC) and parity check (R1PCK) routines into the RCV2A outgoing message. It also presets the range and range-rate data in the RCV2C and RCV2E messages to -1 to insure that they will be interpreted as invalid values by master control unless the Ranging task updates them.

5.1.4

Mnemonic: R1MTM

Title: Time Delay Data Processor Task

Priority: 20 MS (Non-reentrant)

Invoked by: R1MRC

Invokes: X3WAIT, X3STOP

Inputs: XCM016, RCV2C2, RCV2C3, RCV2B3, RCV2E3, RPMTPT

<u>Parameter</u>	<u>Source</u>
FTF count (mod 16) (XCM016)	XCOUNT
Pseudo-range MSW (RCV2C2)	R1RNG
Pseudo-range LSW (RCV2C3)	R1RNG
Receiver LRU status (RCV2B3)	R1MRC
Range-rate LSW (RCV2E3)	R1RNG
Pointer into time delay tables (RPMTPT)	R1MRC

Outputs: RPMTST, RPRATE, RPFTF, RPL1A1

<u>Parameter</u>	<u>Destination</u>
R1MTM run status (RPMTST)	R1MRC
Pseudo range-rate data (RPRATE)	R1MRC
FTF time mark (RPFTF)	R1MRC
Delay table (RPL1A1)	R1MRC

Processing:

R1MTM collects pseudo-range and pseudo-range-rate data for R1MRC. It gathers pseudo-range data from the RCV2C message on four consecutive 320-ms periods. The data is averaged and rounded, then checked for reasonableness. It is then placed into the time delay table (RPL1A1). During the second of these 320-ms periods, it determines the health of each receiver channel from the RCV2B message, and obtains five valid samples of raw range-rate data from RCV2E. From this data, pseudo-range-rate is calculated, scaled, and summed algebraically, then passed to RPRATE. The FTF count

at which the range-rate data was read is passed to RPFTF.

5.1.5

Mnemonic: R1SRC
Title: Single Receiver Control
Priority: 160 MS Background (Reentrant)
Invoked By: R1MRC during power-up initializations.
Invokes: R1CAL, R1SET, R1NSE, R1SCH, R1CC, R1BSN, R1PIN,
 R1RNG, R1RRM, R1DDT, R1PCK, X3WAIT, X3ACT, X3CANC,
 X3ERRA

Inputs:

<u>Parameter</u>	<u>Source</u>
Executive Data: Receiver Number (R1) FTF Mod 65536 (XCO022)	Executive
Command/Control Parameters: Command Word(RMSRCD) Acq Mode(RMSVMD) Calibration Term(RMMTD)	R1MRC
Acquisition Parameters Antenna Selection(RSANT) Code Tap Selection(RSCACD) Doppler Frequency(RSDOPP) L1/L2 Selection(RSFREQ) Code Phase Reset(RSPHRS) Aided Search Command(RSSCHM) Code Search Length(RSSLTH) Generic SV Identification(RSVID)	R1MRC
Code Search State Remaining Length(RISLTH)	R1SCH
Process Communications (RIEMOD)	From each interface task invoked.
Tracking Status Weak-hold Flag(RIHOLD) Task Status(RIRERR)	R1RNG
Data Recovery Monitor Task Status(RIDDST)	R1DDT
SV Data Monitor Task Status(RIFERR)	R1PCK

Resource Status Information	From other R1SRC processes
BIT Power-up(RPBIRS)	via RPCOMM
BIT Allocation(RPBMD)	
BIT Run Status(RPBMS)	
Bit Sync Status(RPBSRS)	
Data Recovery Status(RPDRRS)	
BIT Initialization(RPITRS)	
P-code Init Status(RPPIRS)	

Outputs:

<u>Parameter</u>	<u>Destination</u>
R1MRC Communications Response Word(RMSRCR)	R1MRC
Code Search Status Remaining Length(RSRMAN) C/A Step Size(RSCINC) P Code Step Size(RSPINC)	R1MRC
Channel Status Data Antenna Selection(RMANT) L-band Selection(RMFREQ) Commanded Mode(RMCMDE) Calibration Term(RMMTDR) Channel Status Word(RMRSTA) Generic SV Ident(RMSVID)	R1FMT
Hardware Control Signals	Frequency Synthesizer, Code Module, Output Module, Wideband Module, Narrowband Modules, and B. I. T. E. Module.
Code Search Parameters Dwell Time(RIDWEL) Search Mode(RISCHM) Code Phase Reset(RIPHR)	R1SCH
Code Search Length(RISLTH) Search Step Size(RIPHIN)	
Channel and Locking Parameters Code Type(RICODE) Centering Type(RICTYP) Channel Config(RICHC) Handover Mode(RICPHM) Narrowband Number(RIFSCE)	R1CC
Acquisition Doppler Frequency(RIFREQ)	R1SET
Bit Sync Control	R1BSN

Verification Semaphore(RIVRFY)

P-code Init Control R1PIN
Acquisition mode(RISVMD)
Code Tap Value(RSPCDE)
Built-in-test Init(RIPNX1)

Data Recovery Control R1DDT
Mode Word(RIDDMD)

Data Monitor Control R1PCK
Parity Check Inhibit(RIPCIN)

Tracking Control R1RNG
Channel Control(RICHC)
Time Delay Calibration(RIMTD)
Narrowband 1 ID(RINBC1)
Narrowband 2 ID(RINBC2)
Output Module Config(RISWES)

Processing:

General

R1SRC performs the function of initiating and monitoring interface functions for the individual receiver of the HDUE under the supervision of the R1MRC process. One command code and one acquisition code is received from the R1MRC process every execution period. The command code is interpreted during that period using the transition block table. This causes the initiation of any required actions. Each acquisition code specifies a unique carrier environment which, together with the command code and the current state of the R1SRC process, is used to identify a unique stream of subroutine calls by which the required actions are defined. Each subroutine call represents a process control state for which both hardware and software setups, interface task activation, in-progress monitoring of the given interface task, error reporting, and interruption capabilities are defined. For each control state a code is displayed for the R1MRC process. In the event that an error is reported by an interface process, it is translated into a message which is returned to the R1MRC process for evaluation.

Timing in the HDUE receiver environment and the hardware architecture of the receiver itself requires that bit synchronization, direct mode data recovery activation, p-code generator starting, and the B.I.T.E. code generator starting, be treated as resources allocatable to only one of the following at a time: a single receiver control process

or an interface task under such a process. The R1SRC program is written such that in a reentrant run-time environment these resources are managed and shared by the active single receiver control processes both for themselves and for the interface processes which they control.

The responsibility for controlling the hardware of the HDUE receiver resides exclusively with the single receiver control and the interface task process levels of the receiver control subsystem. The single receiver control level bears the main burden in providing extensive hardware environment management for the interface tasks of the HDUE. Only that portion of hardware control which must be determined at interface task run time is left under the control of the interface tasks.

Control State Descriptions

In the paragraphs which follow each of the various control states of R1SRC is briefly described. These control states are the building blocks used in the execution of the various sequences described later in this section. Activation of one or more interface tasks is normally part of a control state and those modules are identified in the paragraph headers.

VCXO Calibration (R1CAL)

This control state is normally the first in each sequence and for that reason the appropriate antenna and L-band signal are selected. The Calibrate control in the synthesizer is turned on forcing the VCXO under processor control. R1CAL is then activated with its status monitored via RIEMOD. A failure to obtain a valid set of calibration terms will cause a failure in this state.

Range Rate Measurement (R1RRM)

R1RRM is used for 2 purposes: 1. In built-in-test acquisitions it is used to obtain the value of doppler associated with the BIT module. 2. In the Frequency switch it obtains the value of doppler associated with the local replica signal. For built-in-test the output module range-rate counter input is configured to read a special built-in-test signal. For both modes the output module timing is set for 5 milliseconds. R1RRM is then activated

and its status monitored via RIEMOD. If the measured range-rate exceeds reasonable limits, this error condition is reported. At the end of R1RRM execution the range-rate counter input is reset to its normal configuration.

VCXO Set (R1SET)

Appropriate narrowband and wideband module time constants are set up in this control state and, since this state does not always follow VCXO Calibrate, the Calibrate bit in the synthesizer is turned on. The desired Doppler frequency is moved from RSDOPP to RIFREQ and then R1SET is activated. Its status is also monitored based on the value of RIEMOD. A failure to converge or a data overflow for the digital-to-analog converter (DAC) are the possible failure modes in this state.

Noise Calibration (R1NSE)

At the beginning of this state the code selection is set for the late/early flop to be used during code search. The code is C/A or P depending on the sequence. A non-correlating code is selected for use in this control state. R1NSE is activated with its status reported in RIEMOD. When execution terminates the narrowband and wideband module time constants are set to 1 second and 2 seconds respectively in preparation for sequential code search. An analog-to-digital converter (ADC) time-out hardware error may be reported in this state.

Sequential Code Search (R1SCH)

At the beginning of this control state the correct type of code is selected: C/A, P, or X1. If C/A or P the code tap value passed from R1MRC is selected. The output module is set up for 5 millisecond timing with the 2 narrowband correlation signals routed to the sum amplifier. R1SCH reads the output of the sum amplifier each 5 milliseconds. The narrowband AGC is set for 1.0 second time constant. Code search length and initial phase reset are passed in R1SLTH and R1PHRS. R1SCH is activated and its status monitored through R1EMOD. R1SCH reports a search fail if it searches the entire bin with no successful correlations. It also may report an ADC fail or a VCXD DAC fail. At the end of the state, code phase information is returned to R1MRC through RSRMAN.

Code Centering (R1CC)

There are many execution paths through the code centering control state because of the types of centering and code types. Coarse centering is used following a successful code search and includes carrier loop closure. Fine centering is used prior to executing data bit synchronization. Handover centering is used when tracking C/A code and the carrier loop feedback is to be switched to P code. R1CC operates on both C/A and P codes with X1 treated as P-code. R1SRC initializes the output module for

20 millisecond timing and AGC controls are set for fast time constants. The appropriate code flop mechanism is set up in the code module and R1CC is activated. If it reports a failure to lock in the coarse centering mode, the synthesizer is switched to open loop operation (calibrate on) and R1SCH is activated to search the remaining portion of the commanded search bin. If the initial handover attempt fails, a portion of the sequence will be tried a second time. An ADC fail may also be reported in this state.

Bit Synchronization (R1BSN)

In order to accomplish Bit Synchronization the processor must be set up to operate on the 1 millisecond interrupt. This involves enabling the code module tri-state bus so that its C/A epoch controls the interrupts. Since this can be done with only one channel at a time, all other bit sync processes must be locked out while one of them executes. The output module timing is set up for 1 millisecond reading of the data signal. If a built-in-test sequence is being executed, semaphore RIVRFY is set so that R1BSN will only verify that the hardware can properly execute the sequence. Then R1BSN is activated and its status monitored through the value passed in RIEMOD. An A/D Converter hardware fail may be reported in this control state. A uniqueness test failure may occur as described in the R1BSN section. A verification test fail may be reported when RIVRFY is set.

Data Recovery Activation (R1DDT and R1PCK)

The purpose of this control state is to initialize the system time counters (frame synchronization) and begin the steady-state SV data recovery mechanism. In the C/A or HANDOVER sequence frame sync must be determined from the recovered data, so R1DDT is activated in a barker search mode and R1PCK is activated in a mode requiring initialization. When frame sync is achieved R1PCK sets R1ZOK to a value of -1 and R1SRC commands R1DDT to execute the save-data mode in which it saves a snapshot of the system counters for use in P-code initialization. In a Direct-P acquisition, the frame synchronization data is derivable from the precision time information and R1DDT is activated in a special mode in which it initializes all the system counters. For either mode steady-state data recovery then begins with R1SRC using the Data Recovery Monitor subroutine to monitor the status of R1DDT and R1PCK. R1DDT may report a failure of the local data clock or other errors related to the recovered data. R1PCK reports an occurrence of 8 consecutive words with parity errors. It also reports any discrepancy between recovered GPS time and the local time counters. These tasks are not deactivated unless R1SRC receives a cancel directive from R1MRC or a reacquisition sequence is begun.

P Code Initialization (R1PIN)

This control state is used for the initialization of the code generator at other than its reset state. The only hardware control by R1SRC in this state is for the special case when the built-in-test code generator is to be started simultaneously. In this case the BIT code generator is reset to an X1 epoch and the tri-state bus for the code module is enabled. The semaphore RIPNX1 is set to -1 so that R1PIN will also initialize on an X1 boundary. When the code module begins running the BIT code generator will be aligned with it. For other modes R1PIN monitors the variable R1SVMD to determine where to obtain the precision time information and what type of start-up to use. There are no failure modes associated with this control state.

Ranging (R1RNG)

This control state is concerned with the steady-state recovery of measurement data. The appropriate code type is set up on early/late flop on each narrowband and the tracking status of them is communicated to R1RNG through RINBC1, RINBC2, and RISWES. The appropriate time delay correction to be applied to all pseudo-range measurements is passed in RIMTD. The narrowband AGC control is set to 1.0 second and R1RNG is activated. Its status is polled through 2 variables: RIHOLD and RIRERR. When RIHOLD has a non-zero value R1MRC is notified that the channel is in Weak Signal Hold-on. RIRERR is used to report a loss-of-signal condition or hardware failures. When a loss-of-signal occurs R1MRC is

notified via RMSRCR and R1SRC prepares to enter a reacquisition sequence.

this data, pseudo-range-rate is calculated, scaled, and summed algebraically, then passed to RPRATE. The FTF count

PAGE 26

Control State Sequences

Each of the control state sequences executed by R1SRC is described in the following tables. The sequence to be executed is determined from the command word (RMSRCD) and acquisition mode word (RMSVMD) passed from R1MRC. At the beginning of each sequence general hardware and software initialization is accomplished. This is normally done with the Directive 1/2 Handler which is called either internally or as a result of a cancel command from R1MRC. References to the Lock Monitor and Handshake are made in the tables. The Lock Monitor is activated during an acquisition sequence as soon as carrier lock is achieved. Its purpose is to provide a rapid indicator of loss-of-signal before the steady-state lock monitor in R1RNG is activated. This makes the channel resource available to R1MRC more quickly. The Handshake is the technique used by a process to secure a resource which can be allocated to only one of the R1SRC processes at a time. The special resources include P-code Initialization, Data Bit Synchronization, Special Data Recovery modes, and Built-in-test Module Initialization. When a process obtains one of these resources, it locks out the other processes until it has completed use of the resource. The Frequency Switch and Reacquisition Sequences are special sequences which may be entered only following a Ranging state. The New Doppler Bin sequence may be entered only after a search state has failed acquisition. It is used primarily in the reacquisition mode by R1MRC.

PAGE 40

MASTER TIME DELAY SEQUENCE

1. VCXD Calibration
2. Handshake: Built-in-test Initialization
3. Range Rate Measurement
4. VCXD Set
5. Handshake: R1BSN, R1PIN
6. P-code Initialization
7. Noise Calibration
8. Code Search
9. Coarse Code Centering (If loop closure fails
go to step 8)
10. Link Lock Monitor
11. Fine Code Centering
12. Handshake: R1BSN, R1PIN
13. Data Bit Synchronization
14. Data Recovery Activation
15. Disconnect Lock Monitor
16. Ranging

C/A ACQUISITION SEQUENCE

1. VCXD Calibration
2. VCXD Set
3. Noise Calibration
4. Pass Search Parameters
5. Code Search
6. Coarse Code Centering (If loop closure fails
go to step 5)
7. Link Lock Monitor
8. Fine Code Centering
9. Handshake: R1BSN, R1PIN
10. Data Bit Synchronization
11. Data Recovery Activation
12. Disconnect Lock Monitor
13. Ranging

C/A TO P HANDOVER SEQUENCE

1. VCXD Calibration
2. VCXD Set
3. Noise Calibration
4. Pass Search Parameters
5. Code Search
6. Coarse Code Centering (If loop closure fails
go to step 5)
7. Link Lock Monitor
8. Fine Code Centering
9. Handshake: R1BSN, R1PIN
10. Data Bit Synchronization
11. Data Recovery Activation
12. Handshake: R1BSN, R1PIN
13. P-code Initialization
14. Disconnect Lock Monitor
15. Handover Centering (On first Handover fail
go to step 7)
16. Ranging

DIRECT-P ACQUISITION SEQUENCE

1. VCXD Calibration
2. VCXD Set
3. Handshake: R1BSN, R1PIN
4. P-code Initialization
5. Noise Calibration
6. Pass Search Parameters
7. Code Search
8. Coarse Code Centering (If loop closure fails
go to step 7)
9. Link Lock Monitor
10. Data Recovery Activation
11. Disconnect Lock Monitor
12. Ranging

FREQUENCY SWITCH SEQUENCE

1. Link Lock Monitor
2. Cancel R1RNG, R1DDT, R1PCK
3. Range Rate Measurement
4. Disconnect Lock Monitor
5. VCXD Set
6. Handshake: R1BSN, R1PIN
7. P-code Initialization
8. Pass Search Parameters
9. Noise Calibration
10. Code Search
11. Coarse Code Centering (If loop closure fails go to step 10)
12. Data Recovery Activation
13. Disconnect Lock Monitor
14. Ranging

REACQUISITION SEQUENCE

1. Cancel R1RNG, R1DDT, R1PCK
2. VCXD Calibration
3. VCXD Set
4. Handshake: R1BSN, R1PIN
5. P-code Initialization
6. Noise Calibration
7. Pass Search Parameters
8. Code Search
9. Coarse Code Centering (If loop closure fails go to step 8)
10. If tracking P code go to step 9 of Direct-P sequence.
If tracking C/A code go to step 7 of C/A sequence.

NEW DOPPLER BIN SEQUENCE

1. Select Commanded Antenna
2. VCXD Set
3. Code Search
4. Resume original sequence following Code Search

5.1.6

Mnemonic: R1BSN
Title: Bit Synchronization
Priority: 1 MS (Reentrant)
Invoked by: R1SRC
Invokes: X3WAIR, X3STOR
Inputs:

<u>Parameter</u>	<u>Source</u>
1 mS Data Samples	Output Module
Receiver Number	GPS HDUE RCVR Executive
Verification Semaphore(RIVRFY)	R1SRC

Outputs:

<u>Parameter</u>	<u>Destination</u>
Synchronizing Reset Pulse	/20 counter in the Output Module
Exit Mode (RIEMOD)	R1SRC

Processing:

R1BSN is executed after a C/A acquisition when an ambiguity may exist in the knowledge of which 1 millisecond local C/A epoch is coincident with the data transition on the received signal. The local data clock is then synchronized with the received data stream and Frame Synchronization may be attempted.

This transition interval is detected by accumulating differences between successive 1 MS readings of the data signal over the 20 MS period. Data is taken over a total of 75 of these 20 MS periods. The 20 MS interval is partitioned into summations of differences across C/A epochs, S(1) through S(20), each of which represents the sum

of 75 differences between 1 MS samples before and after that particular C/A epoch.

Once the summations are accumulated, the transition epoch is identified by a peak search over S(1) to S(20). If 2 of these bins have the same value, an error is reported. If the detected maximum S(i) is less than 1.5 times the next earlier bin S(i-1), the transition epoch is identified as S(i-1). This test is required since the transition may be wider than 2 MS. Once the correct bin is identified the Divide by 20 counter in the Code Module is reset accordingly.

For Built-in-test operation (RIVRFY = -1) the same sequence is executed except that the Code Module counter is not reset. Then the calculated transition is compared to the actual transition in the hardware, as determined by reading the data clock in the 1 MS priority. A Verification failure is reported if they do not agree. This test is intended to verify the integrity of the hardware to execute data bit synchronization with the special 1 MS interrupts active.

5.1.7

Mnemonic: RICAL

Title: VCXO Calibration

Priority: 5 MS (Reentrant)

Invoked by: R1SRC

Invokes: X3STOP, X3WAIT

Inputs: XC0020

<u>Parameter</u>	<u>Source</u>
Current FTF count (XC0020)	XCOUNT

Outputs: RIINCP, RISLPE, RIEMOD, RIFTF

<u>Parameter</u>	<u>Destination</u>
Calculated intercept of linear approx. (RIINCP)	R1SET
Calculated slope of linear approx. (RISLPE)	R1SET
Exit mode flag (RIEMOD)	R1SET
FTF count at VCXO cal. (RIFTF)	R1SET

Processing:

RICAL is the task which calibrates the voltage controlled oscillators (VCXO) in the HDUE receiver. Calibration is accomplished in the following manner: the output frequency of the VCXO is measured at each of five input voltages. A least squares linear approximation is generated for these five points. Five voltages are calculated from the original range-rate points, and then compared to the original voltages. If any error exceeds the voltage equivalent of 300 Hz, a second calibration is attempted. If this calibration fails, an error is noted and the procedure terminates. Upon successful calibration, the slope and intercept of the approximation along with the FTF

time at calibration are returned to global memory.

5. 1. B

Mnemonic: RICC

Title: Code Centering

Priority: 20 MS (Reentrant)

Invoked by: R1SRC

Invokes: X3WAIT, X3STOP

Inputs: RICODE, RICTYPM, RICHC, RICPHM, RIFSCE

<u>Parameter</u>	<u>Source</u>
Code type (P or C/A) (RICODE)	R1SRC
Centering type (coarse or fine) (RICTYP)	R1SRC
Channel configuration (RICHC)	R1SRC
C/A to P handover mode (RICPHM)	R1SRC
Narrowband channel number (RIFSCE)	R1SRC

Outputs: RIEMOD

<u>Parameter</u>	<u>Destination</u>
Exit mode word (RIEMOD)	R1SRC

Processing:

RICC is responsible for fine tuning the alignment of the P and C/A codes generated by the satellite and by the receiver code module. This alignment is important in making the transition from open to closed-loop operation. There are three modes in this task:

1. A coarse mode used during initial acquisition
2. A fine (C/A) mode prior to Data Bit Synchronization
3. A mode used in Handover itself

The coarse mode is a single shot correction designed to improve the phase alignment to within approximately 1/4 of a code chip. For C/A code, speed in alignment is essential since there is only time for one correction. In the case of

P-code, speed is not so essential, but only one correction is necessary since the code tracking loop of the ranging task will achieve and maintain final alignment.

After the coarse centering operation, the transition from open to closed carrier loop is made. The sequence is as follows:

- 1) Disable code flop
- 2) Select prompt code phase injection
- 3) Loop control = 3RD ORDER WIDE (26 Hz) FLL
- 4) Wait 400 milliseconds
- 5) Loop control = 3RD ORDER WIDE (26 Hz) PLL
- 6) Test for successful loop closure

A successful loop closure occurs when the lock detector is high for three consecutive 20 ms samples. If this does not occur within one second a loop lock failure is reported to RISRC.

The fine mode is used to align the C/A-code replica closely with its satellite counterpart (within 1/17 of a chip). This mode is used during closed loop operation with C/A code and improves the ability to execute subsequent data bit synchronization. This centering is a four-step process as opposed to the one-shot method in the coarse mode.

In the handover mode, R1CC is activated with C/A prompt code tracking on narrowband channel two with C/A fine centering having been previously accomplished. P-code late/early is flopped on narrowband channel one. An open loop P-code coarse centering (tau-dither) is accomplished on

NB channel one while tracking C/A code on channel 2. Then the following steps are taken once centering has finished: C/A prompt code (aligned to within 1/17 chip with satellite code) is injected into narrowband channel two. After a 100 ms delay, four correlation samples are read from the ADC at 20 ms intervals and summed. The injected code is then changed to P prompt code and the process is repeated. The ratio of P to C/A correlation is computed and if greater than 0.45, handover is declared successful and the carrier feedback is switched from narrowband channel two to one. If the ratio test fails, a handover failure is reported to R1SRC.

5.1.9

Mnemonic: R1DDT

Title: Data Detect Task

Priority: 10 MS (Reentrant)

Invoked by: R1SRC

Invokes: X3WAIT, X3STOP

Inputs: RIRCVR, XCO010, XCO020, XCO022, RIM075, RIDRDY, RIDDMD

<u>Parameter</u>	<u>Source</u>
Receiver number (RIRCVR)	R1SRC
10 ms count (mod 2) (XCO010)	X1IT10
20 ms count, MSW (XCO020)	X1IT20
20 ms count, LSW (XCO022)	X1IT20
Mod 75 data bit counter (RIM075)	R1DDT
Data ready flag (RIDRDY)	R1PCK
Data detect mode flag (RIDDMD)	R1SRC

Outputs: RIBRKR, RIBRSN, RIDRDY, RIM075, RIDDMD, RIANBO, RIANAO, RIANCO, RI5IN, RI20IN, RIZIN, RI75IN, RIM030, RIMSWD, RIDDST, RIZX1, RIZCNT, RIWCNT, RISFID, RIZOK, RINTID

<u>Parameter</u>	<u>Destination</u>
State flag (RIBRKR)	R1PCK
Barker sync complete flag (RIBRSN)	R1PCK
Data ready flag (RIDRDY)	R1PCK
Mod 75 data bit counter (RIM075)	R1DDT
Data detect mode word (RIDDMD)	R1SRC
Pseudo-range 17f0 counter (RIANBO)	R1RNG
Pseudo-range / 17 state (RIANAO)	R1RNG
Pseudo-range 4 ms subframe count (RIANCO)	R1RNG
Saved 5 ms frame count (RI5IN)	R1PIN
Saved 20 ms frame count (RI20IN)	R1PIN
Saved value of RIZX1 (RIZIN)	R1PIN
Saved value of RIM075 (RI75IN)	R1PIN
Mod 30 sync counter (RIM030)	R1DDT
SV message word buffer (RIMSWD)	R1PCK
Data detect status (RIDDST)	R1SRC
1.5 second code epoch counter (RIZX1)	R1RNG, R1PIN
HOW ID code (RIZCNT)	R1PCK
SV message word counter (RIWCNT)	R1PCK
6 second epoch ID (RISFID)	R1PCK
Z-count ok for handover flag (RIZOK)	R1RNG
Data transfer subframe ID (RINTID)	R1RNG

Processing:

R1DDT checks for transitions in the SV data clock and, depending on the commanded mode of operation, executes direct P-sync, barker code search, or steady-state SV data output. Also, on the first half of each 20 ms frame, it reads and saves the three components of pseudo-range for use by R1RNG. There is also a save-data mode that saves enough information to establish the relationship between system time and GPS time for use by R1PIN in the handover P-code initialization sequence.

Execution begins on the first half of a 20 ms frame when the three components of pseudo-range are retrieved and saved for use in R1RNG. The SV data clock is checked for a transition, and if one has occurred, the data bit is transferred into an accumulator. If no clock transition is detected in three consecutive frames, an error indicating a hardware failure is noted and the process is deactivated.

Based on the value of R1BRKR, one of the three modes is entered following the detected transition. Following C/A acquisition, the data from the satellite is searched for the barker code. If it or its complement is found, the system counters (R1M030, R1M075, and R1ZX1) are initialized appropriately. Once the barker code is found, the initial two 30-bit words are passed to R1PCK for verification of frame synchronization. If frame sync is not valid, R1PCK will reset R1BRKR to force R1DDT back into the barker search mode. If 330 consecutive data bits are recovered without a barker code match, then an error is noted in R1DDST and the

search continues.

After a successful frame sync, R1SRC commands R1DDT to enter the save-data mode, which is executed once as outlined above. The steady state mode now changes to one in which data from the satellite is output 30 bits at a time to R1PCK, which checks for valid parity and Z-count. During the first 10-ms subframe in each 20-ms period the system counters RIZX1 and RIM075 are updated based on the transitions detected in the present and previous 10-ms subframes. This data, along with RINTID, which indicates the 10-ms interval in which the transition has been detected, allow R1RNG to properly identify the epoch associated with each pseudo-range measurement passed to Master Control.

If direct-P acquisition is attempted, the direct-P sync mode is executed once following the initial detected data clock transition. The system counters are initialized based on the information saved by R1PIN. The following quantities are computed:

```
DELFTF = XC0020 - R1Z0IN
RIM075 = rem ((R175IN + DELFTF)/75)
RIZX1 = RIZIN + int((R175IN + DELFTF)/75)
BITCNT = RIM075 + rem(RIZX1/4) * 75
RIWCNT = int(BITCNT/30) - 1
RIM030 = rem(BITCNT/30)
RIZCNT = int(RIZX1/4)
```

RISFID = rem(RIZCNT/5)

Notes: If RISFID = 0, set RISFID = 5

If RIWCNT = -1, set RIWCNT = 9

If RIWCNT <> -1, set RIZCNT = RIZCNT + 1

"int" means the quotient from an integer divide operation

"rem" means the remainder from a integer divide operation

Following the counter initialization R1DDT and R1PCK
begin normal data recovery procedures.

5.1.10

Mnemonic: R1NSE
Title: Noise Calibration
Priority: 5 MS (Reentrant)
Invoked by: R1SRC
Invokes: X3WAIT, X3STOP
Inputs:

<u>Parameter</u>	<u>Source</u>
5 mS Noise Correlation	Output module
Receiver Number	GPS HDUE RCVR Executive

Outputs:

<u>Parameter</u>	<u>Destination</u>
Average Noise Correlation(RINSC)	R1SCH
Exit Mode Message(RIEMOD)	R1SRC

Processing:

The purpose of R1NSE is to estimate the level of the cross-correlation between the environmental radiation and a GPS PRN code. This estimate is referred to as the noise level by the code search process. The estimate is computed by taking the simple mean of 64 5 mS cross-correlations between the environmental radiation and a GPS PRN code not transmitted by either the GPS space vehicles or the GPS ground stations. This result is placed in a data set for use by the R1SCH process. Status is returned to the activating single receiver control process upon termination. This status is used to indicate one of the following three things: In Progress, Normal Termination, or A/D Converter Failure.

5.1.11

Mnemonic: R1PCK

Title: Parity Check Routine

Priority: 140 MS (Reentrant)

Invoked by: R1SRC

Invokes: X3WAIT

Inputs: RIDRDY, RIBRSN, RIWCNT, RIZCNT, RISFID, RIMSWD, RIPCIN,
RSRDYF, RIHOLD, RPZ01, RPZ02

<u>Parameter</u>	<u>Source</u>
Data ready flag (RIDRDY)	R1DDT
Barker sync complete flag (RIBRSN)	R1DDT
SV message word counter (RIWCNT)	R1DDT
HOW ID code (RIZCNT)	R1DDT
Six-second epoch ID (RISFID)	R1DDT
SV message word buffer (RIMSWD)	R1DDT
Parity check inhibit (RIPCIN)	R1SRC
Precision time Z-count (RPZ01, RPZ02)	R1DDT
RSCOMM data ready flag (RSRDYF)	R1FMT
Weak-hold flag (RIHOLD)	R1RNG

Outputs: RIDRDY, RIBRKR, RIZX1, RIZOK, RIFERR, RSSFID, RSWCNT,
RSMSW1, RSMSW2, RSDTVL

<u>Parameter</u>	<u>Destination</u>
Data ready flag (RIDRDY)	R1DDT
State flag for R1DDT (RIBRKR)	R1DDT
1.5 second code epoch counter (RIZX1)	R1DDT
Z-count OK for handover flag (RIZOK)	R1SRC
SV message word number (RSWCNT)	R1FMT
SUBFRAME number (RSSFID)	R1FMT
SV data word 1 (RSMSW1)	R1FMT
SV data word 2 (RSMSW2)	R1FMT
Frame count error flag (RIFERR)	R1SRC
Data validity word (RSDTVL)	R1FMT

Processing:

R1PCK has the responsibility of computing and verifying parity of the data words transmitted from the satellites. In the C/A acquisition mode, R1PCK waits for R1DDT to declare a successful barker search and to pass the HOW and

TLM words. Parity is computed on both words and, if good, the recovered subframe ID is tested to see if it is consistent with the recovered Z-count. If this test also passes, frame sync is declared to R1SRC through RIZOK. The HOW data is used to correct the value of RIZX1 initialized by R1DDT, and the subframe counter (RISFID) and word counter (RIWCNT) are also initialized. If any of the tests fail, R1PCK will notify R1DDT through RIBRKR to resume barker search.

The steady state mode is entered upon completion of frame sync or in direct-P acquisition. R1PCK receives the recovered SV data words from R1DDT once every 600 ms (the received data consists of the last two bits of the preceding word along with the 30 bits of the current word) and computes and verifies parity. The subframe, word count, and Z-count are updated and verified against the received subframe and Z-count. Any disagreement is reported by passing an invalid word count (RSWCNT). The data word is passed to RSMSW1 and RSMSW2, along with the current subframe count (to RSSFID) and word count (to RSWCNT). During the steady state mode, if there are eight consecutive parity failures on the SV data words, R1SRC is notified of the problem. R1PCK also checks the Roll-Momentum Dump and SV Sync flags and sets the subframe ID (RSSFID) to an invalid value if either flag is set.

Parity check is inhibited (RIPCIN = -1) during built-in-test sequences, but the data is still blocked and

passed along as usual. During weak-signal hold-on (RIHOLD = 1), the counters RIZCNT, RIWCNT, and RISFID are maintained as usual but no data is passed externally.

The parity check algorithm is as follows: If the last bit of the preceding word (E30) is '1', the bits d1-d24 are inverted, otherwise left as is. They are then substituted into the parity equations below. If the computed bits D25-D30 match the received D25-D30 then the parity check passes, else a failure is reported.

Following are the ground segment parity encoding equations used in RIPCK:

Equations for words 3-9

$$D1 = d1 * E30$$

$$D2 = d2 * E30$$

$$D3 = d3 * E30$$

$$\dots$$

$$D24 = d24 * E30$$

$$D25 = E29 * d1 * d2 * d3 * d5 * d6 * d10 * d11 * d12 * d13 * d14 * d17 * d18 * d20 * d23$$

$$D26 = E30 * d2 * d3 * d4 * d6 * d7 * d11 * d12 * d13 * d14 * d15 * d18 * d19 * d21 * d24$$

$$D27 = E29 * d1 * d3 * d4 * d5 * d7 * d8 * d12 * d13 * d14 * d15 * d16 * d19 * d20 * d22$$

$$D28 = E30 * d2 * d4 * d5 * d6 * d8 * d9 * d13 * d14 * d15 * d16 * d17 * d20 * d21 * d23$$

$$D29 = E30 * d1 * d3 * d5 * d6 * d7 * d9 * d10 * d14 * d15 * d16 * d17 * d18 * d21 * d22 * d24$$

$$D30 = E29 * d3 * d5 * d6 * d8 * d9 * d10 * d11 * d13 * d15 * d19 * d22 * d23 * d24$$

where

$d1, d2, \dots, d22$ are the raw data bits

$D25, \dots, D30$ are the parity bits and equations

$E29, E30$ are the last two bits of the previously transmitted word

$D1, D2, \dots, D29, D30$ are the bits uploaded by the control segment and are subsequently transmitted by the satellite

'*' indicates the exclusive-OR operation

Equations for words 1, 2, and 10

$$D1 = d1 * E30$$

$$D2 = d2 * E30$$

$$D3 = d3 * E30$$

$$\vdots$$

$$D22 = d22 * E30$$

$$D23 = E29 * d1 * d7 * d8 * d11 * d13 * d14 * d16 * d17 * d18 * d19 * d21$$

$$D24 = d1 * d3 * d5 * d6 * d7 * d9 * d10 * d14 * d15 * d16 * d17 * d18 * d21 * d22$$

$$D25 = E30 * d2 * d3 * d5 * d6 * d7 * d8 * d10 * d12 * d16 * d19 * d20 * d21$$

$$D26 = d1 * d2 * d4 * d5 * d9 * d10 * d11 * d12 * d13 * d16 * d17 * d19 * d22$$

$$D27 = E29 * d1 * d3 * d4 * d5 * d7 * d8 * d12 * d13 * d14 * d15 * d16 * d19 * d20 * d22$$

$$D28 = E29 * d1 * d2 * d4 * d5 * d6 * d7 * d9 * d11 * d15 * d18 * d19 * d20$$

$$D29 = 0$$

$$D30 = 0$$

where

$d1, d2, \dots, d22$ are the raw data bits

$D23, \dots, D30$ are the parity bits and equations

$E29, E30$ are the last two bits of the previously transmitted word

$D1, D2, \dots, D29, D30$ are the bits uploaded by the control segment and are subsequently transmitted by the satellite

'*' indicates the exclusive-OR operation

5.1.12

Mnemonic: R1PIN

Title: P-code Initialization Task

Priority: 5 MS (Reentrant)

Invoked by: R1SRC

Invokes: X3WAIT, X3STOP, X3ERRA

Inputs: RISVMD, RSPCDE, XCM016, RSRNG1, RSRNG2, RPBIAS, RPZ01,
RPZ02, RPN01, RPN02, XCO020, XCO022, RI75IN, RI20IN,
RIZIN, RI5IN, RIPNX1

<u>Parameter</u>	<u>Source</u>
SV acquisition mode (RISVMD)	R1SRC
P-code taps - byte quantity (RSPCDE)	R1SRC
FTF count mod 16 (XCM016)	X1IT20
Estimated range (RSRNG1, RSRNG2)	R1AID
Precision time bias (RPBIAS)	R1AID
Precision time Z-count (RPZ01, RPZ02)	R1AID
Precision time FTF (RPN01, RPN02)	R1AID
Current FTF count (XCO020, XCO022)	X1IT20
Saved mod-75 data bit counter (RI75IN)	R1DDT
Saved 20 ms frame count (RI20IN)	R1DDT
Saved 1.5 second code epoch count (RIZIN)	R1DDT
Saved 5 ms frame count (RI5IN)	R1DDT
X1 code semaphore (RIPNX1)	R1SRC

Outputs: RIEMOD, RI75IN, RI20IN, RIZIN, RI5IN

<u>Parameter</u>	<u>Destination</u>
Exit mode flag (RIEMOD)	R1SRC
Saved mod-75 data bit counter (RI75IN)	R1DDT
Saved 20 ms frame count (RI20IN)	R1DDT
Saved 1.5 second code epoch count (RIZIN)	R1DDT
Saved 5 ms frame count (RI5IN)	R1DDT

Processing:

The purpose of R1PIN is to initialize and start the P-code generator in either the direct-P or C/A to P handover mode. This is accomplished after the relationship between the local clock (FTF) and the GPS time has been established. In the direct mode this relationship is

defined by the precision time parameters from Master Control: time bias (RPBIAS), precision FTF (RPN01 and RPN02), GPS time tag (RPZ01 and RPZ02), and the estimated range to SV (RSRNG1 and RSRNG2).

In the handover mode analogous parameters are developed within the receiver subsystem by R1DDT: GPS 1.5 Sec Z-count ID (RIZIN), GPS 20 MS Bit Epoch count (RI75IN), corresponding local time ID (RI20IN), and 5 ms ID (RI5IN).

The processing in R1PIN is concerned with translating this time information into P-code register states valid when the code generator will be started. Since this start will always occur on a 20 ms data bit epoch, the X1A state is 0. The general equations for the states of X1B, X2A, and X2B are as follows:

Let A = number of 10.23 MHz cycles in 20 ms (204600)

B = 20 ms epoch count

Z = 1.5 sec Z-count

then,

$$X1B = \text{rem} (A * B / 4093)$$

$$X2A = \text{rem} ((A * B - 37 * Z) / 4092)$$

$$X2B = \text{rem} ((A * B - 37 * Z) / 4093)$$

where "rem" indicates the remainder from an integer divide.

If the desired starting time coincides with one of the stopped states of any of the P-code registers, the start time is delayed until the following 20 ms epoch. If the semaphore RIPNX1 is set, indicating that the built-in-test X1 code generator is being concurrently initialized, the

start time is delayed until the next 1.5 second epoch.

After all the P-code registers have been initialized the code generator is armed to start. For the handover sequence, a data bit epoch is identified on which the register states are valid. Approximately 10 ms prior to that epoch, the generator is armed to start off the next epoch. For the direct sequence the 5 ms period in which the starting epoch is scheduled to occur is identified. During the preceding 5 ms period the delayed start count is loaded and the generator is armed to start.

In the direct mode, the precision time information is saved for R1DDT in R15IN, R175IN, R120IN, and RIZIN. R1DDT subsequently initializes all system counters based on this data. Execution terminates after the code generator is running.

5.1.13

Mnemonic: RIRNG

Title: Ranging Task

Priority: 20 ms (Reentrant)

Invoked by: RISRC

Invokes: X3STOP, X3WAIT

Inputs: XC0020, XC0022, RICHC, RIZX1, RIM075, RINTID, RIANAO,
RIANBO, RIANCO, RIMTD, RISWES, RINBC1, RINBC2

<u>Parameter</u>	<u>Source</u>
20-ms FTF count (XC0020)	X1IT20
Mod-16 FTF count (XC0022)	X1IT20
Channel configuration (RICHC)	RISRC
Narrowband Configuration	RISRC
RINBC1	
RINBC2	
Output Module Config(RISWES)	RISRC
1.5 second Z-count (RIZX1)	RIDDT
Mod-75 epoch count (RIM075)	RIDDT
Interval det. flag (RINTID)	RIDDT
Raw pseudo-range data	RIDDT
17 f0 count state (RIANAO)	
f0 counter (RIANBO)	
C/A epoch / 4 (RIANCO)	
Master time delay offset (RIMTD)	RISRC

Outputs: RIRERR, RIHOLD, RCV2C2, RCV2C3, RCV2C1, RCV2D2, RCV2D3,
RCV2D1, RCV2E2, RCV2E3, RCV2E1, RIFREQ

<u>Parameter</u>	<u>Destination</u>
Ranging error flag (RIRERR)	RISRC
Weak-hold flag (RIHOLD)	RISRC
Pseudo-range, word 1 (RCV2C2)	Master Control
Pseudo-range, word 2 (RCV2C3)	Master Control
RCV2C message ready (RCV2C1)	Master Control
Sync, word 1 (RCV2D2)	Master Control
Sync, word 2 (RCV2D3)	Master Control
RCV2D message ready (RCV2D1)	Master Control
Range rate, word 1 (RCV2E2)	Master Control
Range rate, word 2 (RCV2E3)	Master Control
RCV2E message ready (RCV2E1)	Master Control

Processing:

General

The ranging task obtains pseudo-range and range-rate data from a working receiver and makes it available to the navigation tasks. GPS time tag identification is provided for all pseudo-range measurements in order to remove range ambiguity. Correlation data is measured for use in the code tracking loop and in the calculation of the vernier range correction to measured pseudo-range. The carrier loop is maintained in the Weak-hold monitor section.

The processing in R1RNG is divided into 2 categories: 1. Processing done each 20 milliseconds and 2. 320 ms processing based on the mod 16 value of the local clock (FTF). The former is referred to as common processing and the latter as frame unique with a frame identifier in the range 0-15, for example Frame 8 unique. Table 5.1.13-1 lists all frame unique processing.

Common Processing

The task first initializes its data base and begins steady-state execution on the first frame 8. In steady-state operation, the analog-to-digital converter (ADC) is started and the first correlation voltage read after 100 microseconds. This value is stored and, if in a 2-channel (phase-lock loop) mode, the ADC is cleared and started again on the second sample and hold circuit. The data from the ADC is summed into the appropriate

accumulators for use in code-tracking and vernier range computations.

The Weak-hold monitor is a state driven subroutine with 5 legal states as shown in the following table. All transitions are made based on the value of the hardware lock detector and the slip filter described below. The same data is presented graphically in figure 5.1.13-1.

WEAK HOLD STATES

STATE	CARRIER LOOP	CODE LOOP	LEGAL TRANSITIONS
1	PLL 26HZ E/L	E/L TOGGLE	2
2	PLL 26HZ PROMPT	TAU DITHER	1, 3
3	FLL 15HZ PROMPT	2ND ORDER	3A, 4
3A	FLL 26HZ PROMPT	2ND ORDER	2
4	LOSS-OF-SIGNAL		

State 3A is implemented because of hardware restrictions in switching between loop bandwidths. When state 4 is entered, RIRNG terminates.

The carrier slip detector is checked for slips in the carrier phase, and the slip filter is updated according to the following equation:

$$\text{FILTER}(t) = 63/64 \text{ FILTER}(t-1) + 1/64 \text{ SLIP}(t)$$

The value of the slip filter, along with the loop lock detector, drive the weak-hold processing. If the lock detector has a value of 0, an immediate transition to state 2 is made. After 300 milliseconds in state 2 the lock detector monitor is resumed and if a value of 0 is read, an immediate transition to state 3 is made. When the lock detector is used for state transitions, the slip filter is

preset to a value to provide some hysteresis in the weak-hold loop. The lock detector is not used in state 3 (FLL). The slip filter keys all transitions to lower numbered states. When entering state 2 from state 1, R1RNG modifies the code generator registers so that the tracking narrowband receives the prompt code and the other narrowband early/late. Code tracking is with the tau-dither scheme in state 2 and with the second order loop in state 3. In state 3 RIHOLD is set to +1 to inhibit all SV data recovery. When returning to state 1, the full code flop is restored to both narrowbands.

Code tracking is accomplished in the following manner:

During phase-lock operations the code phase is alternately advanced and retarded by $1/17$ chip in odd-numbered subframes to provide correlation data at two different code phases for the adjustment task. Into the correlation accumulators (SUM11, SUM12, SUM21, SUM22) go the "late" and "early" correlation information, the values depending on whether normal operation (state 1) or tau dither (state 2) is the current mode. The net adjustment, ($+1/17$ chip, $-1/17$ chip, or 0 each 320 ms), is determined in the frame 8 processing from the sign of each phase's discriminator (L-E) function. No net phase adjustment is made when the two phases have opposite signs. This technique yields a highly accurate estimate of vernier range, but provides a maximum code phase slew rate of only 0.18 chips per second.

In the frequency-lock mode (state 3), the phase adjustment task is based on the following equations which are propagated each 60 milliseconds:

$$B(x) = B(x-1) + S(x-1) + C * K1 * DELTAT$$

$$S(x) = S(x-1) + C * K2 * DELTAT**2$$

where:

C = correlation data

DELTAT = change in time (in this case, 0.06 sec)

K1 = 10246.5

K2 = 9133.5

B = Code Track Accumulator

S = Second Order Accumulator

The adjustments are accomplished over a period of 60 ms. During the first 20 ms subframe, the early or late correlation data is read from the ADC and stored. During the next subframe, correlation data again is read, this time being late or early (opposite of the first read). This value is subtracted from the first quantity and negated, if necessary, to obtain the quantity L-E. This is the C term in the above equations. Code phase adjustments are then made per the following table:

<u>B value</u>	<u>Phase adj.</u>	<u>B correction</u>
B > 512	rt'd 2/17 chip	-512
256 < B < 512	rt'd 1/17 chip	-256
-256 < B < 256	none	0
-512 < B < -256	adv 1/17 chip	+256

B < -512

adv 2/17 chip

+512

During the last subframe, no action is taken to allow time for the phase adjustment to affect the data read from the correlators.

This scheme of code phase adjustment is much more responsive, allowing adjustments of up to 1.96 chips/second but no vernier range estimate is made. In the frequency-lock mode, more sustained slips occur between the local replica and satellite-generated signals in the carrier loop than occur during phase-lock. These slips accumulate as misalignments in code phase and are corrected by the code loop as outlined above.

R1RNG executes until a loss-of-signal occurs or until it is deactivated by R1SRC.

FRAME UNIQUE PROCESSING

- Frame 5 -- The range rate counter is armed to start.
- Frame 7 -- If the task is in a mode where vernier ranging may be performed, the vernier ranging flag is cleared.
- Frame 8 -- In addition to the code phase adjustment outlined above, the vernier range correction to measured pseudo-range is computed, provided vernier ranging is enabled.
- Frame 10 -- The range rate counter is armed to stop.
- Frame 11 -- Pseudo-range is output provided that its components are within prescribed ranges. If not, a value of -1 is output.
- Frame 12 -- Sync timing information is calculated and output.
- Frame 13 -- Range-rate data is read from the counter and, if in prescribed ranges, output to the RCV2E bus message. Otherwise, -1 is passed.

TABLE 5.1.13-1

5.1.14

Mnemonic: R1RRM
Title: Range Rate Measurement
Priority: 5 MS (Reentrant)
Invoked by: R1SRC
Invokes: X3WAIT, X3STOP

Inputs:

<u>Parameters</u>	<u>Source</u>
Receiver Number(RIRCVR)	R1SRC
Range-Rate Data	Output Module
Commanded Mode(RIR RTP)	R1SRC

Outputs:

<u>Parameters</u>	<u>Destination</u>
Range-Rate Data(RIFREQ)	R1SET
Exit Mode(RIEMOD)	R1SRC

Processing:

R1RRM is used by the MTD and frequency switching sequences to measure the doppler before acquiring or dropping a GPS signal, respectively. Exit mode information is provided for the invoking single receiver control process via the exit mode word: In Progress, Normal Termination, and Range-Rate Counter Failure.

5.1.15

Mnemonic: RISCH

Title: Code Search Task

Pripritu: 5 MS (Reentrant)

Invoked by: R1SRC

Invokes: X3WAIT, X3STOP, X3ERRA

Inputs: RIDWEL, RIRCVR, RISCHM, XCOO22, RISLPE, RIXSET, RIPHRS,
RIPHIN, RISLTH, RINSC, RSRATE

<u>Parameter</u>	<u>Source</u>
Threshold dwell time (RIDWEL)	R1SRC
Receiver CRU base address	R1SRC
Search task mode word (RISCHM)	R1SRC
FTF count LSW	XCOUNT
VCXD calibrated slope (RISLPE)	R1CAL
Last setting of VCXD by R1SET (RIXSET)	R1SET
Code phase reset (RIPHRS)	R1SRC
Search step size (RIPHIN)	R1SRC
Search length (RISLTH)	R1SRC
Noise calibration (RINSC)	R1NSE
Range-rate (RSRATE)	R1SRC

Outputs: RIXSET, RIPHRS, RIEMOD, RISLTH, RINSC

<u>Parameter</u>	<u>Destination</u>
Last setting of VCXD by R1SET	R1SRC
Code phase reset (RIPHRS)	R1SRC
Exit mode (RIEMOD)	R1SRC
Search length (RISLTH)	R1SRC
Noise calibration (RINSC)	R1SCH

Processing:

The purpose of RISCH is to align the locally-generated code (P or C/A) with that being received from the satellite. The program initializes the code generator to the phase corresponding to the beginning of a search bin (a fixed-size group of consecutive code phases), and proceeds to search for phase alignment starting there and retarding

the code phase toward the end of the bin until the end of the bin is reached or until alignment is achieved. The number of 5 millisecond periods in which correlation data is to be gathered at each step is passed in RIDWEL. This value is 2 for all HDUE sequences. During this process the ambient noise estimate (RINSC) is updated at each code phase step.

In addition, if R1SRC is in a reacquisition mode or if range-rate and acceleration data are meaningful at the current time, R1SCH updates the VCXD setting during the search process. This is to compensate for the doppler shift caused by changes in operating temperature and in relative velocity between the user and the satellite.

Initial code phase alignment is achieved whenever the ratio of the correlation to the average ambient noise exceeds 1.25. Alignment is verified based on the following criteria:

Two counters are used. One, initially zero, counts the total number of measurements taken and the other, initially one, is incremented or decremented as the measurement is above or below the same noise threshold ratio. Success is defined as the second counter reaching 8 or the first reaching 20 before the second reaches 0. A failure is defined as the second counter reaching zero before either of the above conditions occur. If successful, R1SCH leaves the code in such a state that code centering can make further adjustments and attempt to close the carrier loop. A

failure to verify will cause the process to repeat until either success is achieved or until the limit passed by R18RC (R18LTH) is reached, at which time failure is indicated and the task terminates.

5.1.16

Mnemonic: R1SET

Title: VCXO set

Priority: 5 MS (Reentrant)

Invoked by: R1SRC

Invokes: X3STOP, X3WAIT

Inputs: R1FREQ, R1SLPE, R1INCP

<u>Parameter</u>	<u>Source</u>
Commanded VCXO frequency (R1FREQ)	R1SRC
VCXO slope (R1SLPE)	R1CAL
VCXO intercept (R1INCP)	R1CAL

Outputs: R1ITER, R1EMOD, R1XSET

<u>Parameter</u>	<u>Destination</u>
R1SET iterations (R1ITER)	R1SRC
Exit mode flag (R1EMOD)	R1SRC
VCXO set voltage (R1XSET)	R1SCH

Processing:

R1SET has the responsibility of setting the voltage-controlled oscillator (VCXO) using the calibration information provided by R1CAL and a target doppler supplied by R1SRC. An initial estimate of the control voltage is calculated using the slope and intercept computed in R1CAL. The range-rate counter is then read to determine the actual VCXO frequency. A correction to the VCXO voltage is calculated using the slope and the difference between the desired and measured frequency. This process is repeated until either the difference between the desired and measured frequency is less than 10 Hz @ 14.72 MHz or after six iterations. If the complete six iterations are performed

without success, an error is noted, indicating either a hardware problem or a need for recalibration.

5.2 MASTER STATE CONTROL MODULE DESCRIPTION

5.2.1

Mnemonic: M1ADIS

Title: Control Display Unit Command Processor

Priority: 640 ms

Invoked by: Activated by M1CMSC

Invokes: M2CVNM, M2BLNK, X3WAIT, IBSET

Inputs: From Data sets: MMALRT, MNXXXX, MWDMSD

<u>Parameter</u>	<u>Source</u>
CDU Inputs (MNOCDU)	M1CCIO
Waypoint Output Data (MNOWYP)	N2WPCM
Present Position (MNNDIS)	N1XFRM
Processor Status Data (MNSTAT)	M1CMSC, M2STIN, M1CCIO, N2WPCM
Epoch and Week Number (MNBSIO)	M1CRNC, M1CMSC
Memory Read/Write Data (MNMEMQ)	M1CMSC, M2STIN
Satellite ID's (MMSVID)	N2SVSL
Assigned Receiver No. (MMRECN)	M2STIN
Nav Status (MMSTUS)	N2SVSL, N1SVPN, N1MITK, N2FOTP
SV Messages (MWMSSG)	M1PDBR
Receiver Data (MNCRCV)	M1CRNC, N2SVSL

Outputs: To Data Set MNXXXX

<u>Parameter</u>	<u>Destination</u>
CDU Outputs (MNOCDU)	M1CCIO
Waypoint Input Data (MNIWYP)	N1XFRM, N2WPCM
Waypoint Control Data (MNCWYP)	N2NVIN, N1XFRM
Processor Status Data (MNSTAT)	M1CCIO
Receiver Data (MNCRCV)	M1CRNC, N2SVSL
Epoch and Week Number (MNBSIO)	M1CRNC, M1CMSC
Memory Read/Write Data (MNMEMQ)	M1CMSC, M2STIN

Processing

M1ADIS processes operator and system inputs for display on the CDU and for use by other tasks in the Master Control and Navigation systems. The task will detect processor

errors and enable the appropriate warning, as well as determine the need for other types of warnings (e.g. "REACQ FAIL"). It also services the Freeze (store present position in Freeze location), Mark (store Freeze location in selected waypoint), and Update (initialize present position from selected waypoint) functions. It buffers input from M1CCIO and uses this and other data to determine the inputs and outputs appropriate to each display switch position (e.g. RCVR, ERR, etc.). Finally, it detects operator input errors, such as inputting a latitude greater than 90 degrees.

The processing flow for M1ADIS is as follows:

Processor error buffers are checked to see if any processor has reported an error. If so, the status flag for that processor is set. Next, the "INIT TIME" and "INIT POS" flags are checked, and the corresponding status flags are set appropriately.

The vector of status flags (MNSTAT(1-20)) is checked to see if the caution light should be turned on. The vector is in descending order of priority and is checked bottom to top. If any of the flags is -1, the caution light is turned on (the light flashes for a processor fail) and a warning number is set corresponding to the highest priority flag. This warning number is for use in M1CCIO.

The "Mark" function is then serviced. The "Freeze point" is stored in the selected waypoint (in MNIWYP(1-10,*), where "*" represents the waypoint number)

provided the "FZ" key has been pushed sometime prior to the "MK" key. The freeze point is found in MNOCDU(169-178).

Next is the "Freeze" function. If the "FZ" key has been pressed, present position (MNNDIS(1-10,*), where "*" indicates which buffer M1ADIS is accessing at that time) is stored at the freeze point. The freeze flag is then set to communicate to M1CCID that the freeze operation is complete and that the "MK" key is now valid.

The "Update" function will initialize present position to a selected waypoint. Provided that time has been initialized and the display switch is at the position (LAT or UTM) into which the waypoint was entered, the waypoint number will be passed to the navigation subsystem and the nav initialization and nav go/no-go flags are set.

The remaining section of M1ADIS processes the CDU I/O requests. Tests are done to determine if the display is complete or that action has been requested. If no request, this task is finished, and it returns control to the EXECUTIVE system. Otherwise, a local buffer is initialized, and one of fifteen independent subsections to execute is computed, using the value of the display switch setting. In each of these subsections, the displays are normally updated every 640-ms, and tests are then made for input requests. Refer to Paragraph 3.2.5, Control Display Unit Support Function, for a comprehensive description of the CDU inputs and displays.

5.2.2

Mnemonic: MICCID

Title: Control Display Unit Message Control Task

Priority: 100 ms

Invoked by: Activated by M1CMSC

Invokes: M2CRUS, X3WAIT

Inputs: From data set MNXXXX, Block Data CKXXXX

<u>Parameter</u>	<u>Source</u>
CDU input parameters (MNOCDU)	M1ADIS, M2CRUS
Display constants (CKXXXX)	M9MSCD

Outputs: To data set MNXXXX

<u>Parameter</u>	<u>Destination</u>
CDU output parameters (MNOCDU)	M1ADIS, M2CRUS

Processing:

First, M2CRUS is called to retrieve information concerning switch positions, keys, and keyboard flags. Also, the CDU displays from the previous 100-ms period are output. The inputs are compared to an array containing the same information for the previous 100-ms period. If any changes have occurred, the input templates for the current display switch setting (CKITUD for the upper display and CKITLD for the lower display, found in the common block CKXXXX) are retrieved, then unpacked into 10-member arrays (one for each display character) and the displays are blanked.

Next, the "waypoint range" is checked. There is an upper and a lower waypoint limit for each display switch

position, contained in the arrays CKWYLL and CKWYUL. If, for the current display switch position, the waypoint switch is at a value outside of this range, the warning "SWITCH RNG" is displayed.

The "invalid almanac" flag (MNCRCV(20)) is checked to see if an SV has been entered that has no almanac. If so, the alert "NO ALMANAC" is displayed.

Next, display data from M1ADIS is buffered, provided that M1ADIS has completed the display, as indicated by MNOCDU(9). Then, if a key press was detected by M2CRUS, the key is decoded, using the column and row numbers of the key (variables MNOCDU(14) and MNOCDU(15)), and whether it is a lower case, upper case left, or upper case right entry. The equivalent hexadecimal and ASCII value of the key is determined from the tables CKDHEX and CKDASC. If none exists, the value is set to -1.

Checks are made to see if various command keys have been pressed. First, the "IN1" and "IN2" keys are checked. If one of these has been depressed, a flag is set to indicate from which display to expect inputs. Also, other quantities are initialized to prepare for the inputs.

The "CLR" key is next to be serviced. If it has been pressed, the displays are blanked momentarily. If a warning (e.g. "REACQ FAIL) is currently pending, the status flag corresponding is cleared.

The mark key is next in line to be serviced. If it has been pressed, the freeze flag is checked to see if the "MK"

has been preceded by a "FZ". If so, the mark waypoint is set and the freeze flag is cleared. If not, the alert "NO FREEZE" is displayed.

The freeze key is checked and if it has been pressed, the freeze flag is set. The update key is serviced in the same manner. The warning key, if pressed, displays the highest priority warning currently active.

The shift keys ("upper left" and "upper right") are checked and a flag set to indicate which has been pushed. Also, the decimal (".") and the minus ("-") keys are scanned to see if a memory address increment command should be issued.

Next, the "Enter" command is serviced. If the key has been pressed, the current display is checked to see if it is full. If not, the alert "ENT 2 SHORT" is displayed. Also, if more keystrokes have been made than are required, the alert "ENT 2 LONG" is displayed. If the proper number of keys have been pressed, the command is accepted.

If none of the command keys have been pushed, then it is assumed that a data switch is to be serviced. Hence, the keystroke is compared against the mask for the display in question to see if it is of the proper type (decimal, hex, or alpha). If not correct, an alert indicating which entry should have been made is displayed. If correct, the entry is accepted. The individual numerical entries are then gathered and converted into input values for use in MIADIS.

Lastly, the outputs are serviced. The normal displays

are placed into the working displays. Then a check is made to see if any alerts are pending, in which case they are written over the normal displays. Then the working displays are converted to the proper 16-segment format through a table lookup for display on the CDU.

5.2.3

Mnemonic: M1CMSC

Title: Master State Control Task

Priority: 20 ms

Invoked by: Executive, at power-up

Invokes: B2MSTR, X3TIMM, X3TIME, X3ACT, X3ERR, X3WAIT

Inputs: From Data Sets: MNXXXX, MXXXXX, MZDIU, XMNAVE, MKXXXX

<u>Parameter</u>		<u>Source</u>
Read pointer for nav. double bfr	(MNCRCV(243))	M1CMSC, M2STIN
Write pointer for nav. double bfr	(MNCRCV(244))	M1CMSC, M2STIN
FTF	(MNCRCV(241))	M1CMSC
Routing indicator matrix	(MKBTBL)	Block Data
BUS action I/O flags	(MKIDFG(1-9))	Block Data
Relative addr table for RCVR3 msg	(MKOUTS)	Block Data
Vector of BUS I/O flags	(MNBSCN(1-10))	M1CMSC
Vector of BUS routing indicators	(MNBSCN(11-20))	M1CMSC
RCVR3A BUS message	(MNBSIO(313-328))	M1CMSC, M1CRNC
RCVR2A BUS message	(MNBSIO(145-215))	M2STIN
RCVR2C BUS message	(MNBSIO(229-242))	M1CMSC
RCVR2D BUS message	(MNBSIO(243-256))	M1CMSC
RCVR2E BUS message	(MNBSIO(257-270))	M1CMSC
IIU block transfer flag	(MNCIIU(20))	M1CMSC
Multi-processor control array	(MNCPRC(1-6))	M1CMSC
FTF mod(32) at 160-MS boundary	(MNCRCV(163))	M1CMSC
NAVp sync/reset flag	(MNCRCV(245))	M1CMSC
RCVP sync/reset flag	(MNCRCV(246))	M1CMSC
RCVR3 message history flag	(MNCRCV(247))	M1CMSC
IIU transmission attempt counter	(MNCRCV(248))	M1CMSC
Time flag	(MNCRCV(251))	M1CMSC
Memory read/write control array	(MNNMEQ(1-20))	M2STIN
RCVR availability flag	(MNSTAT(103))	M1CMSC
IRIG time from external source	(MZDIU(259-260))	(IIU)
IIU status word	(MDZIIU(258))	(IIU)
Week number	(MNBSIO(5-6))	M1ADIS
Reference time epoch	(MNBSIO(1-4))	M1CRNC
Epoch at next 320-MS mark	(MNCRCV(249-250))	M1CRNC
NAVp running flag	(XMNAVE)	M1CRNC

Outputs: To Data Sets: MNXXXX, XMNAVE, MZXXXX, MXXXXX

<u>Parameter</u>		<u>Destination</u>
Read pointer to nav. double bfrs	(MNCRCV(243))	(MSCP, navp)
Write pointer to nav. double bfrs	(MNCRCV(244))	(MSCP, navp)
Raw range-rate measurements	(MNBSIO(51-70))	N2MCNI
FTF	(MNCRCV(241-242))	M1CMSC
FTF when FTF mod(16) = 11	(MNCRCV(169-170))	M1CMSC

FTF when FTF mod(16) = 0	(MNCRCV(63-64))	M1CMSC
IRIQ when FTF mod(16) = 11	MNCRCV(173-174))	M1CMSC
Update command for MSCP	(MPUPDA(1-6))	M1CMSC
Vector of BUS I/O flags	(MNBSCN(1-10))	M1CMSC
Vector of BUS routing indicators	(MNBSCN(11-20))	M1CMSC
RCVR3A bus message	(MNBSIO(313-328))	(RCVP)
RCVR parameters for IIU blk 206	(MNBSIO(97-106))	IIU)
Multi-processor control array	(MNCPRC(1-6))	M1CMSC
NAVP sync/reset flag	(MNCRCV(245))	M1CMSC
RCVP sync/reset flag	(MNCRCV(246))	M1CMSC
FTF mod(16) at 160-MS boundary	(MNCRCV(156))	M1CMSC
FTF mod(32) at 160-MS boundary	(MNCRCV(163))	M1CMSC
FTF mod(192) at 160-MS boundary	(MNCRCV(164))	M1CMSC
RCVR3 message history flag	(MNCRCV(247))	M1CMSC
IIU transmission attempt counter	(MNCRCV(248))	M1CMSC
Epoch at next 320-MS mark	(MNCRCV(249-250))	M1CMSC
Time flag	(MNCRCV(251))	M1CMSC
Memory read/write control array	(MNNMEG(1-20))	M1ADIS
RCVR availability flags	(MNSTAT(103-108))	M1CMSC, M1ADIS
NAVP fail flag	(MNSTAT(2))	M1ADIS
BUS fail flag	(MNSTAT(13))	M1ADIS
RCVP fail flag	(MNSTAT(3))	M1ADIS
IIU command word	(MZDIIU(257))	(IIU)
WEEK number	(MNBSIO(5-6))	N2MCNI
Pseudo-range measurements	(MNBSIO(11-20))	N2MCNI
RCVR2C bus message	(MNBSIO(229-242))	N2MCNI
RCVR2D bus message	(MNBSIO(243-256))	N2MCNI
Adjusted time epoch	(MNBSIO(1-4))	N2MCNI
Epoch by RCVR	(MNBSIO(31-40))	M1CMSC
NAVP running flag	(XMNAVE)	M1CMSC

Processing:

M1CMSC is the main software control element of the Master State Control Processor (MSCP). It executes the following functions:

- (1) MSCP start-up;
- (2) Compute timings;
- (3) MSCP task scheduling;
- (4) Receiver Processor (RCVP) start-up;
- (4) Navigation Processor (NAVP) start-up;
- (5) Service Serial Time Division Multiplex (STDM) bus;
- (6) Service Instrumentation Interface Unit (IIU);
- (7) Service receiver measurements.

For MCSP start-up, the following sequence is executed, provided that the MCSP has not already been initialized. The data pointers are initialized as follows.

- 1) divide FTF modulo(32) by 16
- 2) the integer result determines the appropriate buffer

<u>Result</u>	<u>0</u>	<u>1</u>
Bread	1	2
Bwrite	2	1

For buffer read and write, a '1' is for the upper buffer and a '2' is for the lower buffer. The NAVP and RCVP reset/sync flags are set to sync, and the FTF (MCSP) is set to zero. It also sets the initialization flag, the NAVP running flag, and it sets all receivers available.

The next section computes timing by calling X3TIME, which returns the FTF, and by calling X3TIMM, which returns the modulo-X of the FTF. Several tests are made to determine actions based on these times. When appropriate, these actions change upper/lower data pointers, store the FTF count, store the 320-ms count, store the IRIG time and FTF count for the navigation data validity time mark.

MSCP task scheduling is done one time by setting the ready flag and calling the Executive service routine, X3ACT, with one task name as an argument at each call.

The RCVP start-up is done when the BUS frame number is equal to 6, setting the routing indicator for the RCV3A message, as well as setting several flags for that message.

The next section performs the NAVP start-up, once, using the reset/sync flag and the initialization flag.

AD-A076 481

TEXAS INSTRUMENTS INC DALLAS EQUIPMENT GROUP

F/G 17/7

GLOBAL POSITIONING SYSTEMS (GPS) HIGH DYNAMIC USER EQUIPMENT. (--ETC(U)
AUG 79

F04701-75-C-0180

UNCLASSIFIED

SAMSO-SD-TR-79-12-VOL-3

NL

2 OF 4

AD
A076481



The STDM BUS service is executed in four parts. First, determine the I/O flag and routing indicator for the message. Since the STDM BUS is table driven, the table controlling the BUS is MKBTBL. The contents of MKBTBL are the routing indicators for each of the 9 types of messages, at every 20-ms frame within two 320-ms intervals. An entry of -1 is found when there is no message to service for a particular device in the appropriate time frame. If a message is to be processed, the master BUS control requires three pieces of information; for the i-th message this data is:

Select word : MNBSCN(1-9)
Input/output flag : MKIOFG(1-9)
Routing indicator : MKBTBL(1-144)

For control of bus message flow, the row of the BUS table containing desired routing indicator is calculated, the I/O flag for this message is set to zero, and the appropriate entry from the BUS table is retrieved. If the routing indicator from the table is greater than zero, the appropriate BUS action for this message is determined, whether incoming or outgoing, and the routing indicator is stored as the first word of the message, if outgoing.

Secondly, the BUS action for RCV3 messages consists of setting the RCV3 flag if a message is ready to be transmitted.

Thirdly, the BUS is serviced with a subroutine call to B2MSTR, the Master BUS Service Routine.

Fourthly, the results of BUS service are tested. If a RCV3 message was transmitted, the update flag is reset. A test is made for a BUS error, and, if one is detected at transmission time, the Executive subroutine, X3ERR is called.

The IIU section is serviced next by setting up the transmission, by checking the IIU hardware status. Also, additional data for Block 206 is processed and stored in array, MNBSIO.

The receiver measurements are processed last in MICMSC. If the BUS frame number is 14, the measurement data is ready. The last time is propagated ahead by 16 FTF'S, and the minimum time is set equal to the predicted time. For five measurements, the week differences are corrected, and the minimum time is calculated. If a pseudo range or pseudo range rate is invalid, no processing for that data occurs. The difference between the predicted time and measured time is calculated; if the difference is greater than 3/4 week, one week is added; if the difference is less than -3/4 week, one week is subtracted. The minimum time is then stored, and the epoch is calculated. Finally the corrected pseudo ranges, epoch, and pseudo range rates are stored in array, MNBSIO for use by the Navigation Subsystem.

5.2.4

Mnemonic: M1CRNC

Title: Receiver And Navigation Processors Control Task

Priority: 160 Milliseconds

Invoked by: Activated by M1CMSC, scheduled by executive

Invokes: M2BUSO, M2IIUT, M2STIN, X2WAIT

Inputs: From Data Sets: MSXXXX, MNXXXX

<u>Parameter</u>		<u>Source</u>
Last operator action command	(MSOPCB)	M2BUSO, M2STIN
RCVP initialization flag	(MNCPRC(5))	M1CMSC
Epoch when FTF mod(16) = 11	(MNBSIO(1))	M1CMSC
Operator CDU command flag	(MNCRCV(2))	M1ADIS
Operator command	(MNCRCV(3))	M1ADIS
FTF mod(16) at 160-MS boundary	(MNCRCV(166))	M1CMSC
FTF when FTF mod(16) = 11	(MNCRCV(169))	M1CMSC

Outputs: To Data Sets: MSXXXX, MNXXXX, MMALRT

<u>Parameter</u>		<u>Destination</u>
Start of RCVR3 message buffer	(MNBSIO(317))	M2BUSO
Start of RCVR3C message buffer	(MNBSIO(345))	M2BUSO
MRC action command	(MSMRCA)	M2BUSO, M2STIN
Pending MRC action command	(MSMRCP)	M2BUSO, M2STIN
Last operator action command	(MSOPCB)	M2STIN
Sole access/release flag	(MMFLAG)	X3REQ
Source selection flag	(MMSELM)	N2SVSL
Soft cancel command flag	(MSCANC)	M2BUSO
Command acknowledge/complete flag	(MSMCMP)	M2BUSO
Operator CDU command flag	(MNCRCV(2))	M1ADIS

Processing

M1CRNC provides the status and control functions for the receiver processor and the navigation processor.

The processing flow for M1CRNC is as follows: (1) Initialize receiver command and receiver bus message routing indicators; (2) call subroutine, M2IIUT, to transmit Instrumentation Interface Unit (IIU) data blocks; (3)

process operator commands; (4) call subroutine, M2STIN, to process bus messages from the receiver processor; (5) call subroutine, M2BUSO, to transmit bus messages to the receiver processor; (6) calculate six-Second epoch and FTF, and 1.5-Second Z-count.

5.2.5

Mnemonic: M1DBPR

Title: Data Collection Routine

Priority: 160 ms

Invoked by: Activated by MICMSC

Invokes: X3REQ, X3REL, LAND, IBCLR, IBTEST

Inputs: From Data Sets: MBDBPR, MMALRT, MIEPHM, MTTCD, MNXXXX, MFDATA

<u>Parameter</u>	<u>Source</u>
Cold start almanac	(MFADAT) Block Data
Receiver control table	(RCVCTL) M2STIN
Satellite data (2A bus msg.)	(MNRINA) B2MSTR

Outputs: To Data Sets: MBDBPR

<u>Parameter</u>	<u>Destination</u>
Satellite data	(MCDATA) M2DBS1, M2DBS2, M2DBS3

Processing

M1DBPR collects the data words received from the SV's or QT's and passes data blocks to M1PDBR for processing. The first time through M1DBPR, the cold start almanac is copied into the almanac data array for all 24 satellites.

Each time that the RCV2A bus message update flag is set, M1DBPR will attempt to collect a 24 bit data word, and store it in the array MCDATA. The word is accepted and stored provided: (1) The source has a valid ID; (2) The word and frame numbers are valid; (3) Data validity is good; (4) The roll momentum dump or SV sync fail is not set; (5) There is no parity error; or (6) The data block is not already being processed. The bits in the array MCDWPC are used to keep track of which data words have been

successfully collected from each subframe for each receiver channel.

After word 10 in each subframe is copied, the data block validation logic is executed. If all the data for subframe 1 is collected and either the TOC or AODC is different from the last subframe 1 which was processed, then the block is marked okay for processing by M2DBS1. This block is used for either SV data or GT data and is marked to indicate which type of source it is from. Subframes 2 and 3 must both be complete before Data Block II can be processed. This is the ephemeris data so the AODE on both subframes must be the same. Again if the AODE is the same as the last processed data then processing is not needed. Subframe 4 contains the satellite messages, so when all the data is collected, and if the message has changed, then it is passed on for processing. Subframe 5 contains the almanac data and once all the almanac is collected then it is passed if the time of almanac has changed.

The array MCDBFL in data set MBDBPR is used to control communication for each receiver channel with the background task MIPDBR. It takes on the following values:

- MCDBFL(NR) = 0 No processing pending;
- 1 Process Data Block I for SV data
- 2 Process Data Block II;
- 3 Process Data Block I for GT data;
- 4 Process the Message Subframe;
- 5 Process Data Block III;

Where NR is the receiver channel designation, and it takes values from 1 to 5.

5.2.6

Mnemonic: M1IIUO

Title: Instrumentation Interface Unit Data Conversion Task

Priority: background

Invoked by: Activated by M1CMSC, scheduled by Executive

Invokes: M2HPPF

Inputs: From Data Set(s): MNXXXX, NINTRF

<u>Parameter</u>		<u>Source</u>
User equipment ID	(MNCIIU(28))	(operator)
Data for IIU Block 002	(MNZBLK(1- 63))	N1XFRM
Data for IIU Block 007	(MNZBLK(105-239))	N2MCNI
Data for IIU Block 010	(MNZBLK(265-323))	N2MCNI
Data for IIU Block 011	(MNZBLK(325-383))	N2MCNI
Data for IIU Block 201	(MNZBLK(385-544))	N2MCNI
Data for IIU Block 202	(MNZBLK(545-658))	N2MCNI
Saved FTF for IIU Block time tag	(NGIFTF)	N2MCNI
Saved IRIG for IIU Block time tag	(NGIFTF)	N2MCNI
Option number for Block 202	(NGZOPT)	N2MCNI

Outputs: To Data Set(s): MNXXXX

<u>Parameter</u>		<u>Destination</u>
IIU Block 002 transmit flag	(MNCIIU(2))	M2IIUT
IIU Block 007 transmit flag	(MNCIIU(6))	M2IIUT
IIU Block 010 transmit flag	(MNCIIU(8))	M2IIUT
IIU Block 011 transmit flag	(MNCIIU(9))	M2IIUT
IIU Block 201 transmit flag	(MNCIIU(4))	M2IIUT
IIU Block 202 transmit flag	(MNCIIU(5))	M2IIUT
Data for IIU Block 002	(MNZBLK(1- 63))	M2IIUT
Data for IIU Block 007	(MNZBLK(105-239))	M2IIUT
Data for IIU Block 010	(MNZBLK(265-323))	M2IIUT
Data for IIU Block 011	(MNZBLK(325-383))	M2IIUT
Data for IIU Block 201	(MNZBLK(385-544))	M2IIUT
Data for IIU Block 202	(MNZBLK(545-658))	M2IIUT

Processing

M1IIUO performs the Instrumentation Interface Unit (IIU) Block formatting and data conversion required prior to transmission to the Hewlett-Packard HP21MS IIU data acquisition computer. The specific data blocks converted are:

(1) Block 002 - Baseline block: HDUE Motion Accuracy (2) Block 007 - Baseline block: Nav Covariance Matrix (3) Block 010 - Baseline block: Range Residuals (4) Block 011 - Baseline block: Range-rate Residuals (5) Block 201 - Nav Filter Inputs And Filter States (6) Block 202 - SV Motion And Nav Filter Gains.

The output of the IIU data is complicated by two factors: (1) some data must be tagged with a particular time during the filter cycle; (2) data is "captured" in one task priority, converted in a second priority, reformatted in a third, and, finally, transmitted in a fourth task priority.

5.2.7

Mnemonic: M1PDBR

Title: Data Block Processing Routine

Priority: Background

Invoked by: Activated by M1CMSC

Invokes: M2DBS1, M2DBS2, M2DBS3

Inputs: From Data Sets: MBDBPR, MNXXXX

<u>Parameter</u>	<u>Source</u>
Satellite data	(MCDATA) M1DBPR
Control Parameters	(MCDBFL) M1DBPR

Outputs: To Data Sets: MWDMSD

<u>Parameter</u>	<u>Destination</u>
Satellite Messages	(MWMSSG)

Processing

M1PDPR processes the data collected by M1DBPR. This data is processed by receiver number for any entry in MCDBFL which is nonzero. M2DBS1 is called for Subframe 1, and will handle both SV data or GT data. The ephemeris data in Subframes 2 and 3 is processed by calling M2DBS2. Almanac data in subframe 5 is processed by calling M2DBS3. The SV messages in subframe 4 are copied into the array MWMSSG by receiver number. The data collection status words in MCDBFL are reset to 0 and M1PDBR then suspends itself by calling X3WAIT.

5.2.8

Mnemonic: M2BLNK

Title: Blank Leading Zeroes on Control Display Unit

Priority: 640 ms

Invoked by: M1ADIS

Invokes: None

Inputs: From data set MNXXX, Argument list START, END, DISP

<u>Parameter</u>	<u>Source</u>
First character position to be blanked (START)	M1ADIS
Last character position to be blanked (END)	M1ADIS
Display (1=upper, 2=lower) (DISP)	M1ADIS
CDU display buffers (MNOCDU(79-98))	M1ADIS

Outputs:

<u>Parameter</u>	<u>Destination</u>
CDU display buffers (MNOCDU(79-98))	M1ADIS

Processing:

M2BLNK blanks leading zeroes in the CDU displays as determined by the inputs START, END, and DISP. START indicates the first character position to be blanked, END the last, and DISP the display to be blanked.

5.2.9

Mnemonic: M2BUS0

Title: Serial Bus (STDM) Message Output Routine

Priority: 160 Milliseconds

Invoked by: M1CRNC

*invokes: F\$RQMY

Inputs: From Data Sets: MMALRT, MAXXXX, MNXXXX, MSXXXX

<u>Parameter</u>		<u>Source</u>
Acknowledgement time-out constant	(MACKNL)	Block Data
Completion time-out constant	(MACMPL)	Block Data
Status of navigation sources	(MMSTUS(1-5))	N1MITK, N1SVPN, N2SVSL, N2FOTP
Message acknowledge flag	(MSMACK)	M2STIN
Message acknowledge time-out cntr	(MSMATC)	M2BUS0
Message acknowledge time-out flag	(MSMATO)	M2BUS0
Command acknowledge/complete flag	(MSMCMP)	M2STIN
Message complete time-out counter	(MSMCTC)	M2BUS0
Message complete time-out flag	(MSMCTO)	M2BUS0
MRC action command	(MSMRCA)	M1CRNC, M2BUS0
Pending MRC action command	(MSMRCP)	M1CRNC, M2STIN
Message transmit flag	(MSMXMT)	M2BUS0
Message retransmit flag	(MSRXMT)	M2BUS0
Copy of MMSVID	(MNCRCV(9-13))	M2STIN
BUS message delay counter	(MSBMDC)	M2BUS0
Soft cancel command flag	(MSCANC)	M1CRNC, M2STIN

Outputs: To Data Sets: MNXXXX, MSXXXX

<u>Parameter</u>		<u>Destination</u>
Track failure flag	(MNSTAT)	M1ADIS
Message acknowledge flag	(MSMACK)	M2STIN
Message acknowledge time-out cntr	(MSMATC)	M2BUS0
Message acknowledge time-out flag	(MSMATO)	M2BUS0
Command acknowledge/complete flag	(MSMCMP)	M2BUS0
Message complete time-out counter	(MSMCTC)	M2BUS0
Message complete time-out flag	(MSMCTO)	M2BUS0
MRC action command	(MSMRCA)	M1CRNC, M2STIN M2BUS0
Pending MRC action command	(MSMRCP)	M2BUS0, M2STIN,
Message transmit flag	(MSMXMT)	M2STIN, M2BUS0
Message retransmit flag	(MSRXMT)	M2BUS0
BUS message delay counter	(MSBMDC)	M2BUS0
Soft cancel command flag	(MSCANC)	M2BUS0
RCVR3B BUS message buffer	MNBSID(329-344))	(RCVR)

Processing

M2BUS0 controls the transmission of the STDM BUS output for the RCVR3B message. When a message is to be transmitted, this routine loads the appropriate data into the message block, sets the update flag, sends the message, and resets the update flag.

The processing flow for M2BUS0 is as follows: check for a soft cancel command pending. Reset semaphore and transmit cancel message. Set MRC action command to cancel. Check if message to be retransmitted. Check if message has been transmitted. Check for acknowledged. Check for completion of command. Message completed; check for message pending. Check for aiding available for concentrated search. If MRC message is pending, delay message by 320 MS. Set MRC action command to action pending. Set MRC pending action to command to no action. Copy the SV'S in MMSVID into BUS message. Copy MRC action command and set update flag. Reinitialize flags after message transmission set mode error flag returned by reciever to no error. Set message acknowledge time-out counter to zero. Set message complete time-out counter to zero. Set BUS message delay counter. (320 ms delay) set message retransmit flag to "don't retransmit". Set message flag to "message transmitted". Set MSMACK flag to "no acknowledgement". Set message completion flag to "not completed". Initialize acknowledgement and completion time-out counters message completion time-out logic. Message complete time-out; check

if second time. Message has timed out twice. Set track failure flag and transmit message again. Message acknowledgement time-out logic. Check for time out. Message acknowledge time-out. Check if second time. Message acknowledge has timed out twice. Set track fail flag and send message again. Send transmitted message again. (retransmit)

5.2.10

Mnemonic: M2CRUS

Title: Control Display Input/Output Interface

Priority: 100 ms

Invoked by: M1CCIO

Invokes: None

Inputs: From data set AMNXXX (ASM language version of MNXXX)

<u>Parameter</u>	<u>Source</u>
Previous kybd row number (MNOCDU(10))	CRIM interface card
Previous kybd column number (MNOCDU(12))	CRIM interface card
Key input enable (MNOCDU(16))	CRIM interface card
16-segment display characters (MNOCDU(39-58))	M1CCIO

Outputs:

<u>Parameter</u>	<u>Destination</u>
Mode switch position (MNOCDU(1))	CRIM interface card
Data switch position (MNOCDU(2))	CRIM interface card
Waypoint switch position (MNOCDU(3))	CRIM interface card
Format switch position (MNOCDU(4))	CRIM interface card
Last kybd row number (MNOCDU(10))	CRIM interface card
Last kybd column number (MNOCDU(12))	CRIM interface card
Keyboard row number (MNOCDU(14))	CRIM interface card
Keyboard column number (MNOCDU(15))	CRIM interface card
Caution light (MNOCDU(17))	CRIM interface card
"Processor active" flag (MNOCDU(18))	CRIM interface card

Processing:

First, M2CRUS checks the "processor active" flag and sets it, if it is not already set. The controller monitor is turned on, and then the mode, data, and waypoint switches are checked and their corresponding values updated. The format switch latch is checked. If key input is enabled (MNOCDU(16) = 0), the keyboard is scanned for a push. If none is detected, the last row and last column numbers are cleared. If a push is detected, the new row and new column

numbers are updated, if there is a change.

For output, the 16-segment characters are displayed, upper display first, left to right, top half of character first. The controller monitor is turned off and the caution light is checked to see if it should be on or off.

5.2.11

Mnemonic: M2CVNM

Title: Integer to ASCII conversion

Priority: 640 ms

Invoked by: MIADIS

Invokes: ISHFT

Inputs: From data set MNXXXX, argument list NUM, DISP, CHRNM,

DEC, POS, SIGN, HEX

<u>Parameter</u>	<u>Source</u>
Decimal number to be converted (NUM)	M1ADIS
Display number (1=upper, 2=lower) (DISP)	M1ADIS
Number of digits to display (CHRNM)	M1ADIS
Decimal/hex flag (0=dec, -1=hex) (DEC)	M1ADIS
First character position (POS)	M1ADIS
Hex number to be converted (HEX)	M1ADIS

Outputs:

<u>Parameter</u>	<u>Destination</u>
CDU displays (MNOCDU(79,98))	M1ADIS
Sign of converted number (SIGN)	M1ADIS

Processing:

M2CVNM converts a number into a form suitable for display on the CDU. It accepts, as input the number to be converted (either decimal or hex), the position it is to occupy on the CDU, the length, and which display it is to occupy, upper or lower. The converted number, expanded into its ASCII equivalent characters, is copied into the array MNOCDU(79-98), along with the sign of the number, if it is decimal.

5.2.12

Mnemonic: M2DBS1

Title: Move Data Block 1

Priority: Background

Invoked by: M1PDBR

Invokes: M2MOVE, X3REG, X3REL, LAND, IDR, ISHFT, EASHFT

Inputs: From Data Sets: MBDBPR, MMALRT

Arguments: N => Receiver Number

<u>Parameter</u>		<u>Source</u>
Processing Flag	(MCDBFL)	M1DBPR, M1PDBR
Data Block 1	(MCDATA)	M1DBPR, M2MOVE
NAV Satellite ID's	(MMSVID)	N2SVSL

Outputs: To Data Sets: MIEPHM, MTTCDs

<u>Parameter</u>		<u>Destination</u>
Ephemeris Data	(MIEPHM)	N2SVEC, N2SVSL N1SVPN
Ephemeris Validity Flag	(MIEVFL)	N1SVPN, N2SVEC
Clock Data	(MTTCDS)	N2SVEC, N2IONQ
Clock Validity Flag	(MTTVFL)	N2SVEC, N2IONQ

Processing

M2DBS1 performs the unpacking and reorganization of Data Block 1. N is the receiver number which is passed as the parameter since all data is organized by receiver. For GT'S, words 3 thru 7 are processed for X Y Z position and saved for ephemeris data. The eighth word is process for age of clock data and GPS reference time and saved for clock data. The ninth and tenth words are for the 3 polynomial coefficients for clock data. Special care for coefficient A(0) because for GT'S, it is a 24 bit number and for SV'S it is a 22 bit number.

X3REQ is used to obtain sole access of the satellite ID'S which are then copied to local data then sole access is dropped by calling X3REL. The ID'S are searched for a match with the satellite ID'S from the receiver and the generic ID ISV is obtained. For GT'S the ephemeris data collected above is moved to MIEPHM(J, ISV) J=1 to 8 after sole access is obtained of MIEPHM. The access for the data array is released and the ephemeris validity flag MIEVFL(ISV) is set true. Then sole access for clock data array is obtained and the clock data obtained above is moved to MTTDAT(J, ISV) J=1 thru 7. The sole access is then released and the clock data validity flag MTTVFL(ISV) is set to true. The data collection status word is reset to 255 signifying that 8 words of data have been collected.

5.2.13

Mnemonic: M2DBS2

Title: Move Data Block II

Priority: Background

Invoked by: M1PDBR

Invokes: M2MOVE, X3REQ, X3REL, IBCLR, ISHFT, EASHFT

Inputs: From Data Sets: MBDBPR, MMALRT

Arguments: N => Receiver Number

Parameter

Source

Data Block I	(MCDATA)	M1DBPR, M2MOVE
NAV Satellite ID's	(MMSVID)	N2SVSL

Outputs: To Data Sets: MIEPHM

Parameter

Destination

Ephemeris Data	(MIEPHM)	N2SVEC, N2SVSL N1SVPN
Ephemeris Validity Flag	(MIEVFL)	N1SVPN, N2SVEC

Processing

M2DBS2 is the unpacking routine for data block 2 subframes 2 and 3. The data is received as 24-bit words and is reorganized to fit either 16-bit or 32-bit word(s).

The following tables shows what 24-bit word the data comes from and the word number(s) it is placed in the ephemeris data array.

For subframe 2:

<u>DESCRIPTION</u>	<u>SOURCE</u> <u>WORD NO.</u>	<u>DESTINATION</u> <u>WORD NO.</u>
Age of data	3	21
Sine of orbit radius	3	17
Mean motion difference	4	18
Mean anomaly	5	1 and 2
Cosine of arg of latitude	6	19
Eccentricity	7	3 and 4
Sine of arg of latitude	8	20
Square root semi major axis	9	5 and 6
Time of ephemeris data	10	15 and 16

For subframe 3:

<u>DESCRIPTION</u>	<u>SOURCE</u> <u>WORD NO.</u>	<u>DESTINATION</u> <u>WORD NO.</u>
Cosine of inclination angle	3	22
Right ascension	4	7 and 8
Sine of inclination angle	5	23
Inclination angle	6	9 and 10
Cosine of orbit radius	7	24
Argument of perigee	8	11 and 12
Rate of right ascension	9	13 and 14

The ID'S are copied from the NAV list to local data and compared with the ID for receiver N to obtain the generic id such that $MMSVID(ISV) = MMSVAD(N)$. Then locking out interrupts to communications memory by calling X3REQ the ephemeris data collected above is copied to MIEPHM(J, ISV) J=1 to 24. The interrupts are then enabled by X3REL and the ephemeris availability flag is set to true. The data collection status for subframes 2 and 3 are set to 255.

5.2.14

Mnemonic: M2DBS3

Title: Process Almanac Data

Priority: Background

Invoked by: M1PDBR

Invokes: X3REQ, X3REL, LAND, ISHFT, EASHFT, IASHFT

Inputs: From Data Sets: MBDBPR

Arguments: N => Receiver Number

<u>Parameter</u>	<u>Source</u>
Data Block	(MCDATA) M1DBPR
ID by Receiver	(MCSVAD) M1DBPR

Outputs: To Data Sets: MPALMC

<u>Parameter</u>	<u>Destination</u>
Almanac Data	(MPALMC) N2SVEC, N2SVSL N1SVPN
Almanac Validity Flag	(MIEVFL) N1SVPN, N2SVEC

Processing

M2DBS3 performs the unpacking of almanac data by receiver number N. The data received in MCDATA are 24-bit numbers and has to be reorganized into standard TI formats of 16-bit or 32-bit numbers. The following table describes the data, the word number in the source array and the word number in the destination array in local memory.

<u>DESCRIPTION</u>	<u>SOURCE</u> <u>WORD NO.</u>	<u>DESTINATION</u> <u>WORD NO.</u>
Eccentricity	3	1 and 2
Time of data	4	11
Inclination angle	4	12
Health word	5	13
Rate of right ascension	5	14
Square root semi major axis	6	3 and 4
Right ascension	7	5 and 6
Argument of perigee	8	7 and 8
Mean anomaly	9	9 and 10
Time parameter 0	10	15
Time parameter 1	10	16

The interrupts into communications memory are locked out so the almanac data collected above can be copied into MPADAT(I,NSV) I=1 to 16 where NSV is the satellite id associated with receiver N (ie. NSV=MCSVAD(N)). The data collection status word MCDWPC(5,N) is set to 255.

5.2.15

Mnemonic: M2HPFP

Title: Hewlett-Packard Floating Point Conversion Routine

Priority: Background

Invoked by: M111UD

Invokes: (None)

Inputs: From Data sets: (None)

Argument list:

- (1) Source address of floating point number to be converted
- (2) Type of conversion required
- (3) Destination address of converted number

Outputs: To Data Set: (None)

Argument list:

Converted floating point number

Processing

M2HPFP converts IBM / Texas Instruments 9900 format, single or double precision floating point values (32 bit or 64 bit), to Hewlett-Packard (HP) format, single or double precision floating point values (32 or 48 bit). This conversion is required of certain data transmitted through the Instrumentation Interface Unit (IIU) to the Hewlett-Packard HP21MX data acquisition computer.

The fortran calling sequence is as follows:

```
CALL M2HPFP ( IBMFP, NPREC, HPFP )
```

where, IBMFP - Single or double precision IBM format

floating point input value

NPREC - Integer value representing precision desired:

1 = Single (32 bit), from double precision input

2 = Double (48 bit), from double precision input
3 = Single (32 bit), from single precision input
HPFP - Single or double precision HP format floating
point output value

(In the following detailed description, values will be stated as decimal numbers, unless stated otherwise.)

The IBM floating point format has a sign bit followed by 7 exponent bits and 24 or 56 mantissa bits, depending on the precision. The exponent value is the actual hexadecimal exponent plus 64 (i.e., it has an effective range from -64 to +63). Thus, the exponent value after adjustment is always positive, and likewise, the mantissa is always a positive number (not complemented). The sign bit, if set, means the represented number is negative.

The HP format has a sign bit followed by 23 or 39 mantissa bits, depending on the precision, followed by seven exponent bits, and then one exponent sign bit. Both the mantissa and exponent are complemented if they are negative, i.e., if their respective sign bits are set.

(Note that the IBM exponent is a power of 16; the HP exponent is a power of 2.)

After obtaining passed arguments, M2HPFP clears to zero the last two words of input if NPREC has a value of three, meaning the input value is single precision. The hexadecimal exponent is converted to a bit (binary)

exponent, and the mantissa is renormalized to eliminate leading zeroes. The renormalization logic is separated into positive and negative fraction branches, since negative values must be complemented first and tested for leading ones instead of zeroes.

The sign of the mantissa is then set, and the exponent and its sign are also set. The HP floating point value (34 or 48 bits) is moved to the desired destination address passed by the calling routine.

5.2.16

Mnemonic: M2IIUT

Title: Instrumentation Interface Unit (IIU) Data Transmitter

Priority: 160 ms

Invoked by: M1CRNC

Invokes: (none)

Inputs: From Data sets: MNXXXX, MTTCDs, MIEPHM, MWDMSD,
MPALMC, MMALRT, NINTRF, MBDBPR,
MAXXXX

	<u>Parameter</u>		<u>Source</u>
IIU block ID table		MAAUID(1-16)	Block Data
IIU block starting addr. table		MAAUAD(1-16)	Block Data
IIU block length table		MAAUBL(1-16)	Block Data
IIU block priority table		MAAUPR(1-16)	Block Data
BLOCK 002	Baseline Blk	MNZBLK(2- 65)	NAVP
BLOCK 006	Baseline Blk	MNZBLK(72-104)	MSCP
BLOCK 007	Baseline Blk	MNZBLK(108-264)	NAVP
BLOCK 010	Baseline Blk	MNZBLK(266-319)	NAVP
BLOCK 011	Baseline Blk	MNZBLK(326-379)	NAVP
BLOCK 018	Baseline Blk	Variable	MSCP
BLOCK 201	Filter Input	MNZBLK(388-544)	NAVP
BLOCK 202	SV/GT Motion	MNZBLK(546-644)	NAVP
BLOCK 206	Raw Meas	MNBSIO(1-106)	NAVP
BLOCK 207	Receiver Aid	MNBSIO(345-392)	MSCP
BLOCK 208	SV Control	MMSVID(1- 38)	NAVP
BLOCK 209	CDU Data	MNOCDU(1- 40)	MSCP
BLOCK 210	Receiver Cntl	MNCRCV(1-145)	MSCP
BLOCK 211	Status Vector	MNSTAT(1-112)	MSCP
BLOCK 212	Data Block 1	MTTCDS(1- 49)	MSCP
BLOCK 213	Data Block 2	MIEPHM(1-130)	MSCP

Outputs: To Data Set: MZXXXX

<u>Parameter</u>	<u>Destination</u>
IIU Blocks output array (MDZIIU)	IIU

Processing

M2IIUT determines which IIU data blocks are to be transmitted, and writes a block of data to the IIU when it is ready to accept it. There are 16 unique blocks, with lengths varying from 36 words to 160 words. The types of blocks are:

- (1) MSCP (fixed-address, direct copy);
- (2) NAVP (fixed-address, formatted);
- (3) Variable address.

For the variable address block, the MSCP blocks are merely copied at transmit time from various areas of Communications Memory. NAVP blocks are formed up and reformatted by the NAVP Subsystem into a buffer area in Communications Memory, and then copied at transmit time. The variable-address block (of which there is only one) is copied from a contiguous area, with the starting address relocatable by the GPS-HDUE operator at any place in Communications Memory.

The processing flow for M2IIUT is as follows.

- (1) Check for a 3840 millisecond boundary.
- (2) Find all MSCP and variable blocks and set ready-to-transmit flag.
- (3) Check "IIU - busy" flag; if not ready, branch to exit this routine; if ready, continue.
- (4) Check "ready-to-transmit" array, MNCIIU (1-16), for any IIU blocks ready, beginning with the highest priority; if none ready, branch to exit this routine; if ready, continue.
- (5) Copy FTF to IIU buffer.
- (6) Preset word number counter to one.
- (7) If NAVP block, FTF is in data, so advance word counter by two only for MSCP blocks; one exception is Block 206.
- (8) Check for variable type; if yes, copy the starting

address and advance word counter; if no, continue.

(9) Calculate end address for current block process.

(10) Copy the identification number and block length as a single packed word to the IIU buffer.

(11) Copy data for the remainder of the block to the IIU buffer.

(12) Reset the block "ready-to-transmit" flag to zero.

(13) Set the IIU transmit flag and exit this routine.

5.2.17

Mnemonic: M2MOVE

Title: Data Block Processing Utility

Priority: Background

Invoked by: M2DBS1, M2DBS2

Invokes: LAND, IOR, EASHFT

Inputs: From Data Sets: MBDBPR

Arguments: II => Double Word Number in MCDATA

N => Receiver Number

Parameter

Source

Data Block I

(MCDATA) M1DBPR

Outputs: To Data Sets: MBDBPR

Parameter

Destination

Source Data

(MCDATA) M2DBS1, M2DBS2

Processing

M2MOVE will combine the third byte of the double word II-1 with the first three bytes of double word II of source data for receiver N. The unit of storage from the source is a 24 bit word so a double precision number is split between two words as shown in the before and after image below.

```
BEFORE  MCDATA(II-1,N) = xxxxxxxxxxxxxxxx
          MCDATA(II,N)  = 1111111100000000
          MCDATA(II,N)  = 2222222233333333
          MCDATA(II,N)  = 4444444400000000

AFTER   MCDATA(II,N)   = 1111111122222222
          MCDATA(II,N)   = 3333333344444444
```

5.2.18

Mnemonic: M2STIN

Title: Status And Input Bus Message Routine

Priority: 160 ms

Invoked by: M1CRNC

Invokes: X3REQ, X3REL, IBTEST, IEOR, IABS

Inputs: From Data Sets: MSXXXX, MAXXXX, MNXXXX, MMALRT,
MIEPHM, MTTCD5

<u>Parameter</u>		<u>Source</u>
IIU status codes map	(MARMDE)	Block Data
Altitude-hold degraded-mode flag	(MMALHO)	N2FOTP
Sole access/release flag	(MMFLAG)	N1SVPN
Range bias degraded-mode flag	(MMRBHO)	N2FOTP
Navigation source status	(MMSTUS)	N2SVSL, N1SVPN, N2FOTP, N1MITK
SV identification numbers	(MMSVID)	N2SVSL
Update flag for array, MMSVID	(MMSVUD)	N2SVSL
Modulo-16 FTF time	(MNCRCV(166))	M1CMSC
Pseudo-range measurement	(MNBSIO(11))	M1CMSC
RCV2 bus message	(MNBSIO(145-298))	B2MSTR

Outputs: To Data Sets: MMALRT, MIEPHM, MNXXXX

<u>Parameter</u>		<u>Destination</u>
Altitude-hold degraded-mode flag	(MMALHO)	N2FOTP, N1MITK
Ephemeris data validity flag	(MIEFLG)	N2SVEC, N1SVPN
Number of valid nav. sources	(MMNVSV)	N1MITK
Range bias degraded-mode flag	(MMRBHO)	N1MITK, N2FOTP
Navigation source status	(MMSTUS)	N1SVPN, N1MITK, N2FOTP
Dual frequency flag	(MNCRCV(83-87))	N2IONO
Receiver control table	(MNCRCV(38-62))	M1ADIS
Receiver LRU status	MNSTAT(9)	M1ADIS
Range differences	(MNCRCV(88-92))	N2IONO

Processing

M2STIN processes the Serial-Time-Division-Multiplex (STDM) input bus messages (type RCVR2), generates status information, controls the receiver command sequences, and filters the L1/L2 frequency range difference data when in the precision mode.

The following describes the processing flow for M2STIN.

(1) Process the receiver command messages (type RCVR2B), and determine channel assignments.

A channel assignment is made for a given generic SV number between 1 and 5 provided the corresponding status word in the RCV2B bus message, words 8 to 12, is positive and between 1 and 5. This is recorded by storing the status in the proper place in MMRECN. For precision mode, the channel from which L2 measurements are taken is copied from word 13 in RCV2B into location MSL2RC.

(2) Process the receiver status/data messages (type RCVR2A), and replace generic SV assignments with actual transmitter numbers.

The RCV2A message is ordered by receiver channel number. Receiver control information is taken from the RCV2A bus message and recorded in MNCRCV(38) to MNCRCV(62). This includes SV ID number, tracking status, commanded mode, frequency and antenna. An SV ID number is assigned provided its receiver assignment as given by the RCV2B message is consistent with its position in RCV2A. The current generic SV number associated with the precision channel is determined and recorded in MSPRSV.

(3) Determine the number of sources for navigation and degraded-mode data.

A source is considered to be active for navigation provided it has a status of 4 or 5, is in a valid ranging

mode, and has a valid receiver channel assignment. The number of sources that the system is currently navigating with is stored in MMNVSV. Values of the degraded mode control flags, MMALHO and MMRBHO, are set according to the value of MMNVSV. This is fully discussed in Section 3.3.7 E.

(4) Determine status of receiver tracking data; calculate delay time in reacquisition.

If there is no soft cancel pending, the previous command has completed and there is no command pending, then the channels which are in reacquisition mode (13) are determined. If twenty seconds pass during which at least one channel was in reacquisition, then it is assumed that reacquisition has failed and a new concentrated search is commanded.

(5) Check receiver command acknowledge and set flag.

A command must be acknowledged before it is examined for completion. The message complete flag (MSMCMP) is set provided the message complete indication in the RCV2B message is positive and agrees with the command (MSMRCA).

(6) Process error report and memory read/write report.

The error report is saved from the RCV2F message in MNSTAT(66-80). The RCV2G message is copied into MNMEMQ(11-15).

(7) Process data for IIU Block 006 and CDU interface.

The SV ID numbers for CDU display are saved in MNCRCV(15-20) by receiver number. Data for the first four

SV's with a status of 3, 4 or 5 is saved for IIU Block 6. This includes SV ID, channel health, mode, frequency and antenna.

(8) Process command/status data for constellation change.

M2STIN examines the constellation update flag (MMSVUD), and copies MMSVID into MNCRCV(9-13). It then commands a soft cancel, and sets the command pending to whatever the current operator command (MSOPCB) is.

(9) Process precision mode command.

For a precision mode command to be issued the following conditions must be met. The soft cancel flag (MSCANC) must not be set. Precision mode must not be already commanded or pending. The previous command must be completed, and no other command can be pending. There must be five good receivers and we must be navigating with four sources. If we are in degraded mode, or there is less than five good receivers, then precision mode is cancelled by issuing a soft cancel command.

(10) Process precision measurements.

Precision mode measurements are processed provided a precision mode command has been issued and completed. This processing is described in Section 3.3.6 B.

5.3 NAVIGATION SOFTWARE MODULE DESCRIPTIONS

5.3.1

Mnemonic: N1INIT

Title: Navigation Subsystem Initialization Task

Priority: Executed at Power Up, Background

Invoked by: EXECUTIVE

Invokes: X3ACT, X3WAIT, N2NVIN, X3STOP

Inputs: Data Sets MNXXXX, XQPRUP

<u>Parameters</u>	<u>Sources</u>
First 20 ms update flag (XQPRUP)	Executive Subsystem
NAV "go/no go" flag (MNCNAV(2))	M1ADIS

Outputs: none

Processing:

N1INIT performs the initialization for the navigation subsystem. X3ACT is utilized to activate a task passed as its argument. The navigation - master state communications task X1COMN is the first task activated. N1INIT will then successively suspend itself by calling X3WAIT until the first 20 ms update flag XQPRUP is not zero. N1INIT will keep on calling X3WAIT until the NAV GO/NO,GO flag is set to go by master state. At that time, the navigation data sets are initialized by calling N2NVIN. Then the satellite coefficient task N1SVPN is activated. After the activation of N1SVPN, N1INIT will again call X3WAIT. This leaves only N1SVPN active, therefore, N1SVPN will execute and satellite coefficients will be ready for the other NAV tasks when they are activated. N1MITK, N1XFRM, N1MCNS and N1NFLT are activated and then N1INIT cancels itself by calling X3STOP.

5.3.2

Mnemonic: N1MCNS

Title: Satellite Clock and Doppler Corrections

Priority: 1280 ms

Invoked by: Activated by N1INIT

Invokes: X3REG, X3REL

Inputs: Data Sets MNXXXX, MMALRT, NINNER, MTTCD5, NINOUT,
NCONST, MPALMC

<u>Parameters</u>	<u>Sources</u>
Source ID (MMSVID)	N2SVSL
NAV Execution State (MNCNAV)	N2NVIN, M1ADIS
Clock Correction Data (MTTDAT)	M1DBS1
Almanac Data (MPADAT)	M1DBS3
Clock Corrections Flag (MTTVFL)	M2DBS1

Outputs: Data Sets NINNER, MNXXXX

<u>Parameters</u>	<u>Destination</u>
Clock Corrections (NGZCLK)	N2MCNI
Doppler Corrections (NGZDOP)	N2MCNI
NAV Initialization State (MNCNAV(1))	N2MCNI

Processing:

N1MCNS computes the clock and doppler corrections for either satellites or ground transmitters. The navigation state must be set to "go" and the initialization flag must not be -1. N1MCNS will process all active navigation sources. If the source is a ground transmitter then a clock offset is set to 0.07247077224 seconds. The offset for satellites is zero. If the clock correction ephemeris availability flag is not zero, then sole access of the clock correction data is obtained. The data is copied to local variables, and the following constants are computed.

$$AO = 1.164153218E-10 \text{ MTTDAT}(3 \text{ and } 4)$$

A1 = 1.136868377E-13 MTTDAT(7)

A2 = 2.775557562E-17 MTTDAT(6)

TOC = 16 MTTDAT(7 and 8)

For ground transmitters without ephemeris the clock correction is set to the negative of the clock offset from above, and the doppler correction is set to zero. For a satellite without ephemeris, sole access is obtained for the almanac data, and the time correction almanac is copied to local variables. The following constants are then computed:

A0 = 7.629394531E-6 MPADAT(5)

A1 = 2.910383046E-11 MPADAT(6)

A2 = 0

TOC = 4096 MPADAT(1)

The clock correction and the doppler correction are then computed as follows:

$$dt = FTF/50 - TOC.$$

FTF is the current user time in 20 ms units. If the absolute value of dt is less than 302400 (a half week in seconds), then it adds 608400 (a week in seconds) to dt to compensate for end of week crossovers. The quantity dt is the elapsed time between current time and the clock correction reference time TOC.

$$NGZCLK(ISV) = A0 + dt(A1 + A2 dt) - Offset$$

$$NGZDOP(ISV) = -(A1 + 2 A2 dt) C$$

Where ISV takes values 1,2,3,4,5 and are the generic SV or GT ID's, and C is the speed of light.

5.3.3

Mnemonic: N1MITK

Title: Inner Loop Control Task

Priority: 320 ms

Invoked by: Activated by N1INIT

Invokes: N2MCNI, N2NEGS, N2HMTR, N2NVIN, X3WAIT, X3TIMM

Inputs: Data Sets MNXXXX, MMALRT, NINNER, NINTRF, NINOUT,
NCONST, NOUTER, NTRANS

<u>Parameters</u>	<u>Sources</u>
Source ID (MMSVID)	N2SVSL
Source Status (MMSTUS)	N2SVSL, N2MCNI, N2SVPN, N2FOTP
NAV Execution State (MNCNAV)	N2NVIN, M1ADIS
Number of Active Sources (MMNVSV)	N1MCNS
Measured Range (NGRNGM)	N2MCNI
Computed Range (NGRNGC)	N2HMTR
Range Bias (NGRBIS)	N2MCNI, N2NEGS
Measured Range Rate (NGRTMR)	N2MCNI
Computed Range Rate (NGRTCM)	N2HMTR
Range Rate Bias (NGRTBS)	N2NVIN, N1MITK
User States (NSTATE)	N2NEGS, N1MITK
Range Gains (NGRGNS)	N2COVR
Range Rate Gains (NGRTGN)	N2COVR
Altitude Hold Gains (NALHGN)	N2COVR

Outputs: Data Sets NINNER, MNXXXX

<u>Parameters</u>	<u>Destination</u>
User States (NSTATE)	N2NEGS, N2MCNI, N2HMTR
States for Outer Loop (NGFSTE)	N2FOTP
Source Status (MMSTUS)	N2MCNI, N2FOTP, N2SVSL, N1SVPN
Satellite Position (NGSVPF)	N2IOND, N2FOTP
Satellite Velocity (NGSVVF)	N2IOND, N2FOTP

Processing:

N1MITK calls N2NEGS to propagate the state forward in time, and it incorporates range and range rate measurements for state estimation. It also provides the necessary logic for interface with the Outer Loop.

The first step for N1MITK is to compute the inner loop

cycle number by calling X3TIMM with an argument of 192, and then dividing the result by 16 and adding 1. The resulting number between 1 and 12 is stored in NGFNDX, and it represents the position of the current 320 ms period within the current 3.84 second outer loop. If navigation is not active, then N1MITK will call X3WAIT to suspend itself. Otherwise, measurements are computed by calling N2MCNI with the option equal to 0. If NAV initialization has been requested and the inner loop cycle number is 12 then N2NVIN is called. The propagation task N2NEQS is called to advance the states 320 ms forward in time.

The measurement incorporation loop will be executed if there are 2 or more sources available, and NAV initialization is complete. If not then N2HMTR is called with parameter ISV and 2 where ISV = 1 thru 5 to have range and range rate computed for aiding. The measurement incorporation loop is executed 5 times while ISV goes from 1 to 5. ISV is index for NAV'S generic source ID's. If there are no measurements or no gains for source ISV, then range and range rate for aiding is computed by calling N2HMTR with ISV and 2 as parameters.

The incorporation loop starts by calling N2HMTR with ISV and 0 as parameters to compute the range. Then the range residual is computed by the measured range minus the range bias plus the computed range (ie. $NGRNGM(ISV) - NGRBIS + NGRNGC(ISV)$). The rolling average of the normalized range residuals is updated.

$$\text{NGSUMS} = a \text{NGSUMS} + b \text{NGZMRS}(\text{ISV})^2 / \text{NGINVM}(\text{ISV})$$

where

a = weight ratio for previous average = 0.95.

b = weight for new normalized residuals = 0.05.

The range is then incorporated into the states by:

$$\text{NSTATE}(\text{J}) = \text{NSTATE}(\text{J}) + \text{NGRGNS}(\text{J}, \text{ISV}, \text{NGN}) \text{NGZMRS}(\text{ISV})$$

New state = the old state plus range gain times the residuals

where

J = 1 thru 11 for each state variable,

NGN = the gains buffer switch.

The range rate incorporation is similar to that of the range, first calling N2HMTR with ISV and 1 as parameter to compute the range rate. The range rate residual is figured by the measured range rate minus the range rate bias plus the computed range rate (ie. $\text{NGZMRS}(\text{ISV}) = \text{NGRTMR}(\text{ISV}) - \text{NGRTBS} + \text{NGRTCM}(\text{ISV})$). The rolling average of the normalized range rate residuals is:

$$\text{RDSUMS} = \text{RDSUMS} a + b \text{NGZMRS}(\text{ISV}+5)^2 / \text{NGINVM}(\text{ISV}+5)$$

where

a and b are the same as for range.

The range rate is then incorporated by:

$$\text{NSTATE}(\text{J}) = \text{NSTATE}(\text{J}) + \text{NGRTGN}(\text{J}, \text{isv}, \text{ngn}) \text{NGZMRS}(\text{ISV}+5)$$

New state = the old state plus range rate gains times the residuals.

Where

J and NGN are the same as above for range.

If range bias hold is active or being deactivated, then states 10 and 11 are not updated.

If altitude hold is active or being deactivated then the altitude hold and altitude rate hold must be incorporated. NXYZSV is the user position saved by N1MITK the last time when altitude hold was inactive. The following two variables are then computed for residuals:

ERRA = sum of (NGTRO1(3,I) (NXYZSV(I) - NGACPS(I)),
where I goes from 1 to 3.

ERRAD = sum of (-NGTRO1(3,I) NGACVL(I)),
where I goes from 1 to 3.

Then the states are updated by:

NSTATE(J) = NSTATE(J) + NALHGN(J,1,NIALGN) ERRA
new state = the old state plus the altitude hold gains time
the altitude hold residual,

where

J = 1 thru 11 for each state,
NIALGN is the buffer switch,

and

NSTATE(J) = NSTATE(J) + NALHGN(J,2,NIALGN) ERRAD,
new state = the old state plus altitude rate hold gains
times the residual.

If the inner loop cycle number is 1 then the satellite position and velocity is saved for the outer loop and the source status is set to 4 if it is 3. N1MITK will call N2MCNI with option equal to 1 to calculate the receiver aiding, and to save data for IIU data block retrieval. N1MITK then suspends itself by calling X3WAIT.

5.3.4

Mnemonic: N1NFLT

Title: Outer Loop Control Task

Priority: 3840 ms

Invoked by: Activated by N1INIT

Invokes: X3WAIT, N2IOND, N2FOTP, X3TIMM

Inputs: Data Sets MNXXXX

<u>Parameters</u>	<u>Sources</u>
NAV GO/NO GO Flag (MNCNAV(2))	M1ADIS
NAV Initialization Flag (MNCNAV(1))	M1ADIS

Outputs: none

Processing:

N1NFLT is the control task for the outer loop. It controls the gains determination, covariance update, covariance propagation, and Iono/Tropo correction computational procedures. If NAV is not active (ie. MNCNAV(2) is 0 or MNCNAV(1) is not 0), then N1NFLT is suspended by calling X3WAIT. Otherwise, N2IOND is called to calculate the Iono/Tropo and lever arm corrections, and N2FOTP is called to perform the outer loop filter related computations. Finally, X3TIMM is called with an argument of 192 to determine the peak amount of time consumed during the execution of N1NFLT.

5.3.5

Mnemonic: N1SVPN

Title: Satellite Coefficient Control Task

Priority: Background

Invoked by: Activated by N1INIT

Invokes: N2SVSL, N2SVEC, N2CGOG

Inputs: Data Sets MNXXXX, MMALRT, MIEPHM, MTTCD5, NSVPOS

<u>Parameters</u>	<u>Sources</u>
Source ID (MMSVID)	N2SVSL
Source Status (MMSTUS)	N1SVPN, N2SVSL, N2MCNI, N2FOTP
Coefficient Time (NGFITT)	N1SVEC
Current User Time (MNBSID)	M1CMSC
Ephemeris Availability (MIEVFL)	M2DBS2
Clock Corrections (MTTVFL)	M2DBS1
H Matrix Ready Flag (NGOGFL)	N2FOTP

Outputs: Data Sets MMALRT, NSVPOS

<u>Parameters</u>	<u>Destination</u>
Source Status (MMSTUS)	NAV software
Geometry Flag (NGOGFL)	N2FOTP
Coefficient Buffer (NGFITX)	N2SVEC
Time of Coefficients (NGFITT)	N2SVEC

Processing:

N1SVPN controls the satellite and/or ground transmitter position and velocity coefficients, and controls the computation of geometry figure of merit. The coefficient arrays are double buffered, so while new coefficients are being computed, old coefficients can still be used. N1SVPN controls which buffer can be used, and which is to be updated. The first function that N1SVPN services is to update the source status to 2 if it is equal to 1. This indicates that we are waiting for ephemeris. N2SVSL is called to update any constellation changes before any

changes to coefficients are made. The next step looks at each source and acts depending on the status. If the status is 1 then polynomials are computed in N2SVEC from the almanac, centered at 60 seconds past the current time. These coefficients are always put in buffer one and the buffer switch (NGFITX) is set for buffer two. For a status of 2, coefficients are computed again 60 seconds ahead, but only if ephemeris and clock corrections are available. Then the status is changed to 3, meaning polynomials were computed with ephemeris. The buffer switch is flipped again and coefficients are computed centered at 120 seconds ahead of the current reference time. This fills the second buffer. For a status of 3, 4 or 5, polynomial coefficients are to be computed every two minutes, always flipping the buffer switch 60 seconds after the current reference time.

Every two minutes N1SVPN will call N2CGDG to compute a geometry figure of merit. N1SVPN takes care to handle the end of week boundary in checking time, and will not call N2CGDG if there is less than 4 sources in use, indicated by NGDGFL. Ephemeris data validity Indicators are packed and stored in MNZBLK(72) for EDU indication of whether each source is using ephemeris or almanac.

5.3.6

Mnemonic: N1XFRM

Title: Relative Navigation Task

Priority: 640 ms

Invoked by: Activated by N1INIT

Invokes: N2OPMG, N2MGCP, N2MOLD, N2MSLH, N2WPCM, DSGRT, DATAN2

Inputs: Data Sets MNXXXX, NCONST, NINNER, NINTRF, NINOUT,
NOUTER, NTRANS, NSVPOS

<u>Parameters</u>	<u>Sources</u>
User position (NGXCPS)	N2MCNI
User velocity (NGXCVL)	N2MCNI
User acceleration (NGXCAC)	N2MCNI
Local datum (MNCWYP(13))	M1ADIS
Waypoints (MNIWYP(1-8))	M1ADIS
CDU switches (MNOCDU(2,3))	M2CRUS
Sum of normalized residuals(NGNRSD)	N2FOTP
Geometry figure of merit (NGDTRM)	N2CGDG
Range bias rate (NGXTBS)	N2MCNI
Iteration paramter (NGAGQQ)	N2NVIN, N1XFRM

Outputs: Data Sets MNXXXX, NTRANS

<u>Parameters</u>	<u>Destination</u>
Local datum position, velocity altitude, ground speed, heading goodness of geometry, sum of normalized residuals, and range bias rate (MNNDIS)	M1ADIS
Waypoints: WGS-72 (NWPLAT, NWPLON, NWPALT)	N2WPCM
Rotation matrix (NGTRO1)	N2IOND, N2NVIN
Block 002 data (MNZBLK(2-65))	M1IIUO
Block 206 data (MNBSIO)	M1IIUO

Processing:

N1XFRM performs the waypoint position initialization, the data collection and computation for IIU Block 002 and the data collection, computation and conversions for the CDU display.

For waypoint initialization, N1XFRM processes 1 of 9

positions and altitudes only when there has been a waypoint location user input. The position is either local datum latitude and longitude in degree minutes seconds, or local datum military grid coordinates. The latitude and longitude are converted from degrees minutes and seconds to radians. Military grid coordinates are first converted to local datum latitude and longitude in N2MGGP. Then N2MOLD is used to compute latitude and longitude corrections which is subtracted from the local datum values to get WGS-72 values. N2MSLH is used to compute an altitude correction to convert mean sea level altitude to WGS-72 altitude by adding the correction to the input value.

The data obtained for IIU Block 002 is user position, velocity, acceleration, time bias and time bias rate. This data is saved by N2MCNI for N1XFRM. The user EFEC coordinates x,y,z are used to compute WGS-72 latitude, longitude and altitude. Iteration is not necessary because the iteration parameter was saved from the previous 640 millisecond execution of N1XFRM. The following sequence of equations is used to compute latitude, longitude and altitude:

Let a = Semi Major Axis WGS-72
b = Semi Minor Axis WGS-72
EC2 = Earth eccentricity squared WGS-72
x = NGXCPS(1) EFEC Position
y = NGXCPS(2) " "
z = NGXCPS(3) " "
G = NGAGGG Iteration Parameter

$$R = (x^2 + y^2) / (1 + Q) + z^2$$

$$Q = (a^2 EC^2) / \text{sqrt}(R - EC^2 z^2)$$

$$\text{NGLATD} = (1 - Q b) \text{sqrt}(R)$$

$$\text{NGLNGD} = \text{arc tan}((1 + Q) z / \text{sqrt}(x^2 + y^2))$$

$$\text{NGALTD} = \text{arc tan}(y / x)$$

The 3x3 rotation matrix, NGTRO1, is computed from the EFEC position and applied to the geodetic velocity and acceleration to get east, north and up velocity and acceleration by the following sequence of equations.

Let LON = WGS-72 Longitude NGLNGD
 LAT = WGS-72 Latitude NGLATD

NGTRO1(i, j)	i=1,2,3; j=1,2,3	
-sin(LON)	cos(LON)	0.0
-cos(LON) sin(LAT)	-sin(LON) sin(LAT)	cos(LAT)
cos(LON) cos(LAT)	sin(LON) cos(LAT)	sin(LAT)

For local level velocity:
 LLCV = NGTRO1 NGXCVL

For local level acceleration:
 LLCA = NGTRO1 NGXCAC

These computations are performed every 640 milliseconds but are saved every 3.84 seconds for IIU block 002. Columns 2 and 3 of the rotation matrix, altitude, east north and up velocity and acceleration, range bias and range bias rate are all stored in MNZBLK for IIU Block 002. The WGS-72 latitude, longitude, east velocity and north velocity are saved every 640 milliseconds in MNBSID for IIU Block 206.

The data for the CDU is processed depending on the CDU switch setting found in MNOCDU(2). For the UTM and LAT switch positions, the latitude and longitude from the block

002 computations are used. N2MOLD is used to compute latitude and longitude correction that is added to the WGS-72 latitude and longitude to produce local datum values. Then for UTM, N2GPMG is called to convert the local datum latitude and longitude to the military grid coordinates and store them in MNNDIS(1-8). For LAT the local datum latitude and longitude is converted to degrees minutes and seconds and stored in MNNDIS(1-6). If the switch position is WIND then the sum of the normalized residuals and geometry figure of merit are stored in MNNDIS, words 18, 7 and 8. In the ALT switch setting, the altitude correction computed in N2MSLH is subtracted from the WGS-72 altitude and both the mean sea level altitude and WGS-72 altitude is stored in MNNDIS(9-12) and scaled to feet. The VEL selection will use the east north and up velocities computed for IIU Block 002, converting the east and north velocities to knots, the up velocity to feet per minute, and storing the values in MNNDIS(15-17). The ground speed and heading are computed and scaled to knots and degrees. N2WPCM, the waypoint computation routine, is called for switch settings RNG, XTRK and TIME to compute the range, bearing horizontal and vertical cross track errors and time to go.

5.3.7

Mnemonic: N2CGDG

Title: Compute Geometry Figure of Merit

Priority: Background

Invoked by: N1SVPN

Invokes: none

Inputs: Data Sets NSVPOS

Parameters
H-Matrix (NGXHMR)

Sources
N2FOTP

Outputs: Data Sets NSVPOS

Parameters
Determinant of H-Matrix (NGDTRM)

Destination
N1XFRM

Processing:

N2CGDG computes the geometry figure of merit from the H-Matrix saved by N2FOTP. We compute the determinant of the 4x4 H-Matrix, knowing that row 4 is always all ones and that N2CGDG is not called in degraded navigation. The 4x4 matrix is expanded by row 4. Each 3x3 minor is expanded by the first column except for the (4,1) cofactor where column 2 is used.

5.3.8

Mnemonic: N2COVR

Title: Update Covariance and Compute Gains

Priority: 3840 ms

Invoked by: N2FOTP

Invokes: none

Inputs: Data Sets MNXXXX, MMALRT, NOUTER, NCONST, NINOUT

Arguments ISV: the Generic Source ID

<u>Parameters</u>	<u>Sources</u>
Receiver Command (MNCRCV(3))	M1ADIS
Source H Vector (NGHMTR)	N2FOTP
Altitude Hold Flag (MMALHO)	N2FOTP, M1STIN
Range Bias Hold Flag (MMRBHO)	N2FOTP
Gains Buffer Switch (NIGAIN)	N2FOTP
Alt Hold Buffer Switch (NIALHG)	N2FOTP
Covariance Matrix (NGCOVP)	N2COVR, N2FCPF

Outputs: Data Sets NINOUT, NOUTER

<u>Parameters</u>	<u>Destination</u>
Covariance Matrix (NGCOVP)	N2COVR, N2FCPF, N2FOTP
Range Gains (NGRGNS)	N1MITK
Range Rate Gains (NGRGNS)	N1MITK
Altitude Hold Gains (NALHGN)	N1MITK
Innovation Vector (NGINVM)	N2MCNI, N2FOTP

Processing:

N2COVR updates the covariance matrix and computes new gains for a specified satellite or ground transmitter. The input parameter ISV is the generic ID with magnitude of 1 thru 5 for range and range rate updates, and 100 or -100 for altitude hold and altitude rate hold updates. If ISV is negative the update is for range measurements. Otherwise it is for range rate measurements. The covariance matrix is normally an 11x11 matrix unless the range bias hold is being activated or is active, in that case the range bias and

range bias rate states are not updated so the covariance matrix is then an 9x9 matrix. The measurement noise SIGMA is set to 25 for the range calculations, .25 for range rate, .1 for altitude hold, and .01 for altitude rate hold. For C/A ranging, the noise is scaled up by a factor of 25 since C/A introduces more noise.

The following equations are the basic formulas used by N2COVR for updating the covariance matrix.

$$K = PH' / (HPH' + SIGMA)$$

$$P = (I - KH)P$$

where the following description for K, P, H and I are for the formula above, and how they are used in the NAV software.

K is the 11x1 gains vector for source ISV.

P is the 11x11 covariance matrix being updated. For our application we define $P=UDU'$, which is incorporated into the algorithm used and U and D are 11x11 matrices of the following form.

$U = \begin{matrix} 1 & x & x & x & x & x & x & x & x & x & x \\ 0 & 1 & x & x & x & x & x & x & x & x & x \\ 0 & 0 & 1 & x & x & x & x & x & x & x & x \\ 0 & 0 & 0 & 1 & x & x & x & x & x & x & x \\ 0 & 0 & 0 & 0 & 1 & x & x & x & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 1 & x & x & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & x & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{matrix}$	$D = \begin{matrix} D1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & D2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & D4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & D6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & D8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D11 \end{matrix}$
---	--

To save space by not storing the diagonal of U, we store the U and D matrices together as one and defined as follows:

```

NGCOVP =  D1  x   x   x   x   x   x   x   x   x   x   x
           0   D2  x   x   x   x   x   x   x   x   x   x
           0   0   D3  x   x   x   x   x   x   x   x   x
           0   0   0   D4  x   x   x   x   x   x   x   x
           0   0   0   0   D5  x   x   x   x   x   x   x
           0   0   0   0   0   D6  x   x   x   x   x   x
           0   0   0   0   0   0   D7  x   x   x   x   x
           0   0   0   0   0   0   0   D8  x   x   x   x
           0   0   0   0   0   0   0   0   D9  x   x   x
           0   0   0   0   0   0   0   0   0   D10 x   x
           0   0   0   0   0   0   0   0   0   0   D11
    
```

H is a 1x11 vector set in N2FOTP and is defined from the array NGHMTR as:

H = (NGHMTR(1), NGHMTR(2), NGHMTR(3), 0, 0, 0, 0, 0, 0, 1, 0)
For range

H = (NGHMTR(1), NGHMTR(2), NGHMTR(3), NGHMTR(4), NGHMTR(5),
NGHMTR(6), 0, 0, 0, 0, 1) For range rate

H = (NGTRO1(3,1), NGTRO1(3,2), NGTRO1(3,3), 0, 0, 0, 0, 0,
0, 0, 0) for altitude or altitude rate hold.

I is the 11x11 identity matrix.

The gains buffer switch NIGAIN(ISV) is set to the gains in use so N is set to the other buffer, 3-NIGAIN(ISV). If ISV is negative then K is stored in NGRGNS(J,II,N) where II=-ISV. If ISV is positive then K is stored in NGRTGN(J,ISV,N). If ISV = -100 then K is stored in NALHGN(J,1,NN) where NN=3-NIALGN, the altitude hold buffer switch, and if ISV = 100 then K is stored in NALHGN(J,2,NN). J above goes from 1 to 11 for each filter state. The denominator of K, HPH'+SIGMA, is the innovation factor for source ISV and is stored in NGINVM(ISV) for range updates, and NGINVM(ISV) for range rate.

5.3.9

Mnemonic: N2FCPG

Title: Propagate the Covariance Matrix and Add Process Noise

Priority: 3840 ms

Invoked by: N2FOTP

Invokes: none

Inputs: Data Sets NOUTER, NCONST

<u>Parameters</u>	<u>Sources</u>
Covariance Matrix (NGCOVP)	N2COVR
Fudge Factor (NGFFCT)	N2FOTP, N2NVIN

Outputs: Data Sets NOUTER

<u>Parameters</u>	<u>Destination</u>
Covariance Matrix (NGCOVP)	N2COVR

Processing:

N2FCPG propagates the error covariance matrix forward 3.84 seconds in time, and applies the appropriate process noise. The formula used by N2FCPG is $P = \text{PHI } P \text{ PHI}' + Q$, where P is the covariance matrix, PHI is an 11x11 known deterministic matrix, and Q is the process noise. The matrix P is in UDU' form, where U and D are combined as described in the N2COVR documentation. PHI is defined in section 3.3.7 of the navigation system level description. PHI P PHI' is computed in the UDU' form but before Q can be added, UDU' must be evaluated.

Q is the following constant matrix:

Q =	60	0	0	16	0	0	1.9095	0	0	0	0
	0	60	0	0	16	0	0	1.9095	0	0	0
	0	0	60	0	0	16	0	0	1.9095	0	0
	16	0	0	5.3921	0	0	1.5715	0	0	0	0
	0	16	0	0	5.3921	0	0	1.5715	0	0	0
	0	0	16	0	0	5.3921	0	0	1.5715	0	0
	1.9095	0	0	1.5715	0	0	0.9227	0	0	0	0
	0	1.9095	0	0	1.5715	0	0	0.9227	0	0	0
	0	0	1.9095	0	0	1.5715	0	0	0.9227	0	0
	0	0	0	0	0	0	0	0	0	.24	.1
	0	0	0	0	0	0	0	0	0	.1	.02

Q is multiplied by the scalar NGFFCT computed in N2FOTP.

After Q is added to P then P is converted back to UDU' form
by the Cholesky Decomposition algorithm.

5.3.10

Mnemonic: N2FOTP

Title: Gains and Covariance Control Routine

Priority: 3840 ms

Invoked by: N1NFLT

Invokes: N2COVR, N2FCPG

Inputs: Data Sets MNXXXX, MMALRT, NOUTER, NTRANS, NINOUT,
NCONST, NSVPOS

<u>Parameters</u>	<u>Sources</u>
Source ID (MMSVID)	N2SVSL
Source Status (MMSTUS)	N1SVPN, N2MCNI, N2FOTP, N2SVSL
Altitude Hold Flag (MMALHO)	N2FOTP, M1STIN
Satellite Velocities (NGSVVF)	N1MITK
Gains Buffer Switch (NIGAIN)	N2FOTP
Alt Hold Buffer Switch (NIALHG)	N2FOTP
Satellite Position (NGSVPF)	N1MITK
User Position State (NGFSTE)	N1MITK
Altitude Hold Flag (MMALHO)	N2FOTP, M1STIN
Range Bias Hold Flag (MMRBHO)	N2FOTP, N1STIN
Rotation Matrix (NGTRO1)	N1XFRM
Residuals (NGZMRS)	N1MITK
Innovation Matrix (NGINVM)	N2NVIN, N2COVR

Outputs: Data Sets NINOUT, MNXXXX

<u>Parameters</u>	<u>Destination</u>
H Vector (NGHMTR)	N2COVR
H Matrix (NGXHMR)	N2CGOG
Altitude Hold Status (MMALHO)	M2STIN
Range Bias Hold Status (MMALHO)	M2STIN
Sum Normal Residuals (NGNRSD)	N2MCNI
Fudge Factor (NGFFCT)	N2MCNI

Processing:

N2FOTP computes the H vector for range and delta range and will call N2COVR to compute range gains, range rate gains, and update the covariance matrix for each source. If we let ISV denote the generic source ID, ISV = 1,2,3,4, or 5, then MMSVID(ISV), the source ID, must not be zero and

MMSTUS(ISV), the source status, must be greater than 3 before gains are computed. The H vector for range is computed as follows:

$$\begin{aligned} S1 &= \text{NGFSTE}(1) - \text{NGSVPF}(\text{ISV}, 1) \\ S2 &= \text{NGFSTE}(2) - \text{NGSVPF}(\text{ISV}, 2) \\ S3 &= \text{NGFSTE}(3) - \text{NGSVPF}(\text{ISV}, 3) \end{aligned}$$

Where NGFSTE are the filter state estimates of the geodetic position and NGSVPF are the satellite positions.

$$\begin{aligned} \text{SR} &= \text{sqrt}(S1^2 + S2^2 + S3^2) \\ &\text{Range from the user to the satellite} \\ \text{NGHMTR}(1) &= -S1/\text{SR} \\ \text{NGHMTR}(2) &= -S2/\text{SR} \\ \text{NGHMTR}(3) &= -S3/\text{SR} \end{aligned}$$

N2COVR is called with -ISV as the parameter to compute range gains and update the covariance matrix.

The H matrix is saved in NGXHMR for the goodness of geometry computation in N2CGOQ only if there are four sources available and if N2CGOQ is not currently executing.

The range rate H vector is computed by:

$$\begin{aligned} S1 &= \text{NGFSTE}(4) - \text{NGSVPF}(\text{ISV}, 1) \\ S2 &= \text{NGFSTE}(5) - \text{NGSVPF}(\text{ISV}, 2) \\ S3 &= \text{NGFSTE}(6) - \text{NGSVPF}(\text{ISV}, 3) \end{aligned}$$

Where NGFSTE(4,5,6) are the filter state estimate for velocity and NGSVPF is the same as above.

$$\text{SRR} = \text{NGHMTR}(1) S1 + \text{NGHMTR}(2) S2 + \text{NGHMTR}(3) S3$$

Elements 1 2 and 3 of NGHMTR are copied to elements 4 5 and 6.

$$\begin{aligned} \text{NGHMTR}(1) &= (\text{SRR} \text{ NGHMTR}(4) - \text{NGFSTE}(4) + \text{NGSVVF}(\text{ISV}, 1)) / \text{SR} \\ \text{NGHMTR}(2) &= (\text{SRR} \text{ NGHMTR}(5) - \text{NGFSTE}(5) + \text{NGSVVF}(\text{ISV}, 2)) / \text{SR} \end{aligned}$$

NGHMTR(3) = (SRR NGHMTR(6)-NGFSTE(8)+NGSVVF(ISV,3))/SR

Where SR is from above and NGSVVF is the satellite velocity. N2COVR is called with +ISV as the parameter for range rate gains and to update the covariance.

The gains buffer switch is flipped for the source and the status for the source, MMSTUS(ISV) is set to 5.

If altitude hold is active or being activated (ie. MMALHO = 1 or 2) then the third row of the rotation matrix (NGTRO1) will be the H vector for range and N2COVR is called using -100 as the parameter to compute altitude hold gains (NGALGN). The altitude hold rate H vector has zeros for the first 3 elements and again the third row of the rotation matrix for the last 3 elements. The altitude hold buffer switch is flipped and the altitude hold status is set to active.

If range bias hold is being activated then it is switched to active. If altitude hold is being deactivated then it is switched off because we just computed a full set of gains for four sources. If range bias hold is being deactivated then it is switched off.

N2FOTP will compute the sum of the normalized residuals squared from the residuals and the innovation matrix. NGZMRS is a 10 element array that contains both the range residuals (elements 1 thru 5) and the range rate residuals (elements 6 thru 10). NGINVM is structured similar to NGZMRS and contains the innovation matrix. First the running total RSD is initialized to zero and the count J is set to zero. By letting ISV go from 1 to 5 the following

equations are executed if the ID, MMSVID(ISV), is not zero and the status, MMSTUS(ISV), is 5.

$$RSD = RSD + NGZMRS(ISV)^2 / NGINVN(ISV)$$

$$ISV5 = ISV + 5$$

$$RSD = RSD + NGZMRS(ISV5)^2 / NGINVN(ISV5)$$

$$J = J + 2$$

After all 5 sources have been processed and J is not zero then $NGNRSD = RSD / J$.

The fudge factor is initialized to 1 in N2NVIN and here is adjusted by factors of 4 and limited to the values 1, 4, 16 and 64. This is a scaler to our process noise to help keep the sum of the normalized residuals bounded. If NAV is being initialized the the fudge factor is not adjusted. If $NGNRSD$ is greater than 1 and the $NGFFCT$ is less than 64 then $NGFFCT$ is scaled up by a factor of 4. If $NGNRSD$ is less than 0.1 and $NGFFCT$ is greater than 1 then $NGFFCT$ is scaled down by a factor of 4.

5.3.11

Mnemonic: N2QPMG

Title: Convert Latitude Longitude to Military Grid

Priority: 640 ms

Invoked by: N1XFRM

Invokes: N2OSMG, DSQRT, DSIN, DCOS

Inputs: Data Sets NCONST, NELIPS, NTRANS

Arguments: CNV - Compute Convergence or Military Grid

LAT - Local Level Latitude

LON - Local Level Longitude

<u>Parameters</u>	<u>Sources</u>
Local Datum (NGAIDC)	N1XFRM
Delta Flattening (NELDLF)	Block Data (NELIPS)
Change in Semi Major Axis (NELDLA)	Block Data (NELIPS)
WGS-72 Longitude (NGLNGD)	N1XFRM

Outputs: Arguments: MG - Zone number, Zone letter,
Column letter, Row letter
EST - Easting
NTH - Northing

<u>Parameters</u>	<u>Destination</u>
Convergence angle (NGCNVG)	N2WPCM

Processing:

N2QPMG is called whenever necessary to convert a latitude and longitude to military grid coordinates or compute the convergence angle between true and magnetic north. The latitude and longitude inputs for military grid computations must be a local datum latitude and longitude. The earth's radius and flattening are computed by adding the delta a (NELDLA) and delta f (NELDLF) to the WGS-72 earth's radius and flattening. The input value for convergence angle is the WGS-72 longitude NGLNGD. The following

sequence of computations will give us the UTM coordinates:

$$f = f' + \text{NELDLF}(\text{NGAISP})$$

where

f' is the WGS-72 earths flattening
 $\text{NGAISP} = \text{NELISP}(\text{NGAIDC})$ = The local spheroid associated with the local datum.

f is the Local Datum Earths Flattening

$$\text{EC2} = 2f - f^2$$

EC2 is the Local Datum Eccentricity Squared

(1) $\text{MG}(1) = 31 + \text{LAT} (\text{degrees}) / 6$
 $\text{MG}(1)$ is the UTM Zone Number

$$\text{DL} = \text{LON} - 6 \text{MG}(1) + 183$$

DL is the longitude offset from center

$$\text{ETA2} = \cos^2 (\text{LAT}) / (\text{EC2} - 1)$$

If CNV is equal to 1 then the following sequence of code is used:

(2) $\text{MG}(2) = \text{LTR}(13 + \text{LON} (\text{degrees}) / 8)$
 $\text{MG}(2)$ is the UTM Zone Letter

where

LTR is a one to one mapping of the numbers 1 to 24 to the letters A to Z less I and O.

$$n = (1 - \sqrt{1 - \text{EC2}})^2 / \text{EC2}$$

$$a = a' + \text{NELDLA}(\text{NGAISP})$$

a is the Semi Major Axis for the Local Level Spheroid

$$S = a (1-n) \left(\frac{\text{LAT}^2 + 5/4 \sin^2 (\text{LAT})}{2} + 3n \sin(\text{LAT}) \cos(\text{LAT}) \right) + 5/4 n (\cos (\text{LAT}) - \sin (\text{LAT})) - 1$$

S is the Meridional arc from the equator to Latitude

$$R = a / \sqrt{1 - \text{EC2} \sin^2 (\text{LAT})}$$

R is the Radius of Curvature Normal to the Meridian

$$Z = R \cos(\text{LAT})$$

$$X = Z R \left(1 - \tan^2(\text{LAT}) + \text{ETA}^2 + Z \left(\frac{1}{4} + \tan^2(\text{LAT}) \right) \right. \\ \left. (\tan(\text{LAT}) - 18) / 20 \right)$$

$$Y = S + R Z \tan^2(\text{LAT}) \left(\frac{1}{2} + \frac{5}{12} Z \left(5 - \tan^2(\text{LAT}) \right) \right. \\ \left. + 9 \text{ETA}^2 \right)$$

X and Y are the cartesean coordinates relative to the equator and the central meridian and are converted to the UTM Easting and Northing by:

$$\text{Easting} = X * .9996$$

$$\text{Northing} = Y * .9996$$

If Northing is less than zero then add 10000000 to it.

The Military Grid Coordinates are now computed from the UTM Easting and Northing.

- (3) $\text{MG}(3) = \text{LTR} \left(\left(\text{Easting} / 100000 \right) + B \left(\left(\text{MG}(1) - 1 \right) \bmod 3 \right) \right)$
MG(3) is the Column Letter for Military Grid
- (4) $\text{EST} = \text{Easting} \bmod 100000$
EST is the Military Grid Easting

N is the offset row identifier computed by N2OSMG
- (5) $\text{MG}(4) = \text{LTR} \left(\left(\left(N + \text{Northing} \right) \bmod 100000 \right) + 1 \right)$
MG(4) is the Military Grid Row Letter
- (6) $\text{NTH} = \text{Northing} \bmod 100000$
NTH is the Military Grid Northing

The convergence angle is the angle between true and grid north.

NGCNVG is computed only when CNV input is 2:

$$\text{DLC} = \text{DL} \cos^2(\text{LAT})$$

$$\text{TNC} = 2 - \tan^2(\text{LAT})$$

$$\text{ETC} = \text{ETA}^2 (3 + 2 \text{ETA}^2) + 1$$

$$(7) \text{NGCNVG} = \left(\left(\text{DLC} \text{TNC} + \text{ETC} \right) \text{DLC} + 1 \right) \text{DL} \sin(\text{LAT})$$

5.3.12

Mnemonic: N2HMTR

Title: Computed Range and Range Rate

Priority: 320 ms

Invoked by: N1MITK

Invokes: DSGRT

Inputs: Data Sets NSVPOS, MMALRT, NINNER, NCONST

Arguments ISV = Generic Source ID

IR = Execution Mode

<u>Parameters</u>	<u>Sources</u>
Source ID (MMSVID)	N2SVSL
Coefficient Buffer Switch (NGFITX)	N1SVPN
User Time (NGTIME(2))	N2MCNI
Measured Range (NGRNGM)	N2MCNI
Fit 0 Coefficient (NGFIT0)	N2SVEC
Fit 1 Coefficient (NGFIT1)	N2SVEC
Fit 2 Coefficient (NGFIT2)	N2SVEC
User Position (NGACPS)	N2NEQS, N1MITK
User Velocity (NGACVL)	N2NEQS, N1MITK

Outputs: Data Sets NINNER

<u>Parameters</u>	<u>Destination</u>
Computed Range (NGRNGC)	N2MCNI, N1MITK, N2IONO, N2SVSL
Computed Range Rate (NGRTCM)	N2MCNI, N1MITK
Satellite Position (NGSVPS)	N1MITK
Satellite Velocity (NGSVP1)	N1MITK

Processing:

N2HMTR computes the satellite position corrected for the earths rotation, the satellite velocity corrected for the earths rotation, the computed range and the computed range rate. ISV from the parameter list is the generic source ID and IR is the execution mode. If IR is 0 then the computed range and satellite position is computed. If IR is 1 then the computed range, computed range rate and satellite

velocity is computed. If IR is 2 then everything is computed. The computed range is computed for IR = 0 or 1 since the user position has been updated after the previous call to N2HMTR. IR is 2 when range and range rate are needed for aiding and measurements are not being incorporated.

If source ISV is not valid (ie. MMSVID(ISV) = 0) then N2HMTR returns to N1MITK. First the reference time is computed as follows:

$$\text{NGTSTP}(\text{ISV}) = \text{NGTIME}(2) + \text{NGRNGM}(\text{ISV}) / c$$

where

c is the speed of light

Let I = NGFITX(ISV), the coefficient buffer switch

$ER = 2.432387791 \times 10^{-13}$, Earth Rotation Rate/Speed of Light

SS = NGTSTP(ISV) - NGFIT1(ISV, I)

Delta time from coefficient computation to Transmission

End of the week crossover is checked and SS is adjusted if needed.

The satellite position is:

$$\text{NGSVPS}(\text{ISV}, \text{J}) = \text{SS} (\text{NGFIT1}(\text{J}, \text{ISV}, \text{I}) + \text{SS} \text{NGFIT2}(\text{J}, \text{ISV}, \text{I})) + \text{NGFITO}(\text{J}, \text{ISV}, \text{I})$$

where

J = 1 thru 3 for the 3 coordinates of the earth fixed earth centered coordinate system.

The satellite position is corrected for the earths rotation by the following sequence of equations.

CCR = NGRNGC(ISV) ER

S1 = NGSVPS(ISV, 1)

S2 = NGSVPS(ISV, 2)

NGSVPS(ISV, 1) = S1 + CCR S2
NGSVPS(ISV, 2) = S2 - CCR S1

The computed range is calculated from the user and satellite position as follows.

DX = NGACPS(1) - NGSVPS(ISV, 1)
DY = NGACPS(2) - NGSVPS(ISV, 2)
DZ = NGACPS(3) - NGSVPS(ISV, 3)

$$\text{NGRNGC}(\text{ISV}) = \sqrt{\text{DX}^2 + \text{DY}^2 + \text{DZ}^2}$$

The computed range from the user to the satellite.

Satellite velocity:

NGSVP1(ISV) = NGFIT1(J, ISV, I) + 2 NGFIT2(J, ISV, I) SS

where

J and I are as above in the satellite position equations.

Corrected for earths rotation:

CCRT = ER NGRTCM(ISV)
S3 = NGSVP1(ISV, 1)
S4 = NGSVP1(ISV, 2)
NGSVP1(ISV, 1) = S3 + CCR S4 + CCRT S2
NGSVP1(ISV, 1) = S4 + CCR S3 - CCRT S1

where

ER, S1, S2 and CCR are from the position above.

The computed range rate:

DDX = NGACVL(1) - NGSVP1(ISV, 1)
DDY = NGACVL(2) - NGSVP1(ISV, 2)
DDZ = NGACVL(3) - NGSVP1(ISV, 3)

SS = DX DDX + DY DDY + DZ DDZ
NGRTCM(ISV) = SS / NGRNGC(ISV)

where

DX, DY and DZ are from computed range above.

5.3.13

Mnemonic: N2IONO

Title: Compute the Iono/Tropo and Lever Arm Corrections

Priority: 3840 ms

Invoked by: N1NFLT

Invokes: DEXP

Inputs: Data Sets MNXXXX, MMALRT, NINNER, NTRANS, NINOUT,
NCONST

<u>Parameters</u>	<u>Sources</u>
Source ID (MMSVID)	N2SVSL
Source Status (MMSTUS)	N1SVPN, N2MCNI, N2FOTP, N2SVSL
Range Diff Avial Flag (MNCRCV)	M2STIN
Range Difference (MNCRCV)	M2STIN
Range to Satellite (NGRNGC)	N2HMTR
Altitude (NGALTD)	N1XFRM
Satellite Position (NGSVPF)	N1MITK
User Position State (NGFSTE)	N1MITK
Distance Between Antennas (NGATDS)	Block Data
Rotation Matrix (NGTRO1)	N1XFRM

Outputs: Data Sets NINOUT, MNXXXX

<u>Parameters</u>	<u>Destination</u>
Iono/Tropo Corrections (NGDELA)	N2MCNI
Iono Delay (MNCRCV(93-102))	M1IIUO
Tropo Delay (MNCRCV(103-112))	M1IIUO
Lever Arm Corrections (NGLEVR)	N2MCNI

Processing:

N2IONO computes the ionospheric and tropospheric delays, and the antenna lever arm corrections for each source. If the source ID is zero or the status is less than 3 then this source is skipped and the next source is processed. The tropospheric corrections and the lever arm corrections are computed for both satellites and ground transmitter whereas the iono corrections are only computed for satellites. If the source is a satellite then work

values for Iono/Tropo corrections are computed as follows:

$$\begin{aligned} \text{TIOND} &= 5.5550564 \times 10^{-10} \text{ L1L2(ISV)} && \text{If L1/L2 are valid} \\ \text{TIOND} &= 0 && \text{Otherwise} \end{aligned}$$

where

L1L2(ISV) is the L1/L2 difference in MNCRCV(87+ISV)
ISV = the generic SV ID

$$\text{SEL} = (\text{RANGE}^2 - 6.648 \times 10^{14}) / (-1.276 \times 10^7 \text{ RANGE})$$

If SEL is less than .0871557427 then SEL = .0871557427

$$\text{TTROPO} = 2.175 \text{ e}^{-3} (-0.1439 \times 10^3 \text{ NGALTD}) / \text{SEL}$$

where

RANGE = NGRNGC(ISV) Range to the source
NGALTD is the user altitude

If the source is a ground transmitter then the work values are computed as follows:

$$\text{TIOND} = 0$$

$$\text{TTROPO} = \text{RANGE} \text{ e}^{-3} (-.6936 \times 10^{-4} \text{ NGALTD})$$

The Iono delays and the Tropo delays are then computed as follows and the delays are stored in MNCRCV(93 - 113):

$$\text{IOND(ISV)} = \text{TIOND} \text{ c}$$

$$\text{TROPO(ISV)} = \text{TTROPO}$$

where

IOND is a real*8 equivalence of MNCRCV(93-103)
TROPO is a real*8 equivalence of MNCRCV(104-113)
c is the speed of light

The IONO/TROPO corrections are computed as follows:

$$NGDELA(ISV) = TIONO + TTROPO / c$$

Satellite position and the user position were saved by NIMITK in NGSVPF and NGFSTE for use in computing the lever arm corrections. The calculation is as follows:

$$\begin{array}{ll} DX = NGSVPF(ISV,1) - NGFSTE(1) & \text{Delta X} \\ DY = NGSVPF(ISV,2) - NGFSTE(2) & \text{Delta Y} \\ DZ = NGSVPF(ISV,3) - NGFSTE(3) & \text{Delta Z} \end{array}$$

$$NGLEVR(ISV) = NGATAS (NGTRO1(3,1) DX + NGTRO1(3,2) DY + NGTRO1(3,3) DZ) / NGRNGC(ISV)$$

where

NGATAS is the distance between antennas
NGTRO1 is the rotation matrix
ISV is the generic source ID.

5.3.14

Mnemonic: N2MCNITitle: Master Control Interface RoutinePriority: 320 msInvoked by: N1MITKInvokes: X3TIMMInputs: Data Sets MNXXXX, MMALRT, NOUTER, NTRANS, NINOUT,
NCONST, NINNER, NINTRF, MTTCD5

Arguments OPTION: Execution mode

<u>Parameters</u>	<u>Sources</u>
Source ID (MMSVID)	N2SVSL
Source Status (MMSTUS)	N1SVPN, N2MCNI, N2SVSL, N2FOTP
Range Measurements (MNBSIO)	B2MSTR
Range Rate Measurements (MNBSIO)	B2MSTR
Iono/Tropo Corrections (NGDELA)	N2IONO
Clock Corrections (NGZCLK)	N1MCNS
Lever Arm Corrections (NGLEVR)	N2IONO
Doppler Corrections (NGZDOP)	N1MCNS
Covariance Matrix (NGCOVR)	N2COVR, N2FCPG
Residuals (NGZMRS)	N1MITK
Filter States (NGFSTE)	N1MITK, N2NEQS
Computed Range (NGRNGC)	N2HMTR
Computed Range Rate (NGRTCM)	N2HMTR
Range Gains (NGRGNS)	N2COVR
Range Rate Gains (NGRGNS)	N2COVR

Outputs: Data Sets NINNER, MNXXXX

<u>Parameters</u>	<u>Destination</u>
Measured Range (NGRNGM)	N1MITK
Measured Range Rate (NGRTMR)	N1MITK
Receiver Aiding (MNBSIO)	Receiver subsystem
IIU Blocks (MNZBLK)	M11IUD

Processing:

N2MCNI has the responsibility for computing the measured pseudo range and pseudo range rate, initializing range bias, computing receiver aiding, and gathering data for IIU blocks 7, 10, 11, 201 and 202. The data that N2MCNI receives and transmits in communications memory is double

buffered and the buffer used depends on the time. XBTIMM is called to determine the buffer switch settings.

If the argument OPTION is 0, then N2MCNI performs the measurement recovery logic. The pseudo range and range rate measurements are computed for each source in use if it has a valid ID, a receiver number assigned, a NAV status of greater than 3, a receiver tracking status of 9 or 11, L1 frequency measurements, and positive measured values.

The measured pseudo range is:

$$\text{NGRNGM}(\text{ISV}) = (\text{C1 PR}(\text{NR}) + \text{NGDELA}(\text{ISV}) - \text{NGZCLK}(\text{ISV})) \text{ C}$$

where

$\text{C1} = 3.593812892 \times 10^{10}$
PR(NR) is the measured range from receiver NR
ISV is the generic source ID
C is the speed of light

The measured pseudo range rate is:

$$\text{NGRTMR}(\text{ISV}) = \text{C1} (\text{C2} - \text{DR}(\text{NR})) - \text{NGZDOP}(\text{ISV})$$

where

$\text{C1} = 0.1040668523$
 $\text{C2} = 1232000$
DR(NR) is the measured range rate for receiver number NR

If NAV initialization is complete, then the predicted epoch is computed and the range bias is updated for week boundary if necessary. If the Navigation Subsystem is not initialized, and a measurement is available from a source for which ephemeris has been gathered, then range bias is set to $\text{NGRNGM}(\text{ISV}) + \text{NGRNGC}(\text{ISV})$, measured range plus computed range. Also if current time is in the last 320 ms

period before a 3.84 second boundary, then NAV initialization is marked complete.

The receiver aiding is computed when the OPTION is equal to 1.

The range aiding is:

$$RA = (NGRNGC(ISV) + .64 NGRTCM + (NGDELA(ISV) - NGZCLK(ISV)) C) C1$$

where

C is the speed of light

$$C1 = 3.335646952 \times 10^{-2}$$

To avoid integer overflow if the magnitude of RA is greater than 2140000000 then RA = 0. This is computed for each active source and RA is stored in MNBSIO words 379 thru 388 depending on ISV.

Range rate aiding is:

$$RRA = (NGRTBS - NGRTCM(ISV) - NGZDOP(ISV)) / 1.0416666666$$

If the magnitude of RRA is greater than 32700 then RRA is set to zero. RRA is stored in MNBSIO words 347 thru 352 depending on ISV.

Time bias aiding is:

$$TB = (NGRBIS + .64 NGRTBS) 3.335640952 \times 10^{-2}$$

If the magnitude of TB is greater than 2140000000 then TB = 0. TB is then stored in MNBSIO words 354 and 355 or 370 and 371.

The following table shows what data is gathered for IIU
 Blocks and word numbers in MNZBLK where they are stored.

<u>NAME</u>	<u>DESCRIPTION</u>	<u>BLOCK</u>	<u>WORDS</u>
NGCOVP	Filter Covariance D,U Matrix	007	108-239
NGZMRS	Pseudo Range Residuals	010	268-287
NGZMRS	Pseudo Range Rate Residuals	011	328-347
MNBSIO(5)	Week Number	201	388
MNBSIO(3)	Epoch Number	201	390
NQRNGM	Pseudo Range	201	391-410
NGRTMR	Pseudo Range Rates	201	411-420
NGSVPS	Satellite Positions	201	423-482
NGSVVF	Satellite Velocities	201	483-512
NSTATE	Filter States	201	515-544
NGDELA	Iono/Tropo Corrections	202	549-558
NGLEVR	Lever Arm Corrections	202	559-568
NGZCLK	Clock Corrections	202	569-578
NGFFCT	Fudge Factor	202	579
NGINVM	Innovation Vector	202	580-599
NGTIME	Time of Captured Data	202	601-602
NCRNQC	Computed Range	202	603-622
NGRTCM	Computed Range Rate	202	623-632
NGNRSD	Sum of Normalized Residuals	202	633
NGRONS	Range Gains	202	549-658
NGRTON	Range Rate Gains	202	549-658

5.3.15

Mnemonic: N2MQGP

Title: Convert Military Grid to Latitude and Longitude

Priority: 640 ms, 320 ms or Background

Invoked by: N1XFRM, N2NVIN

Invokes: N2OSMG, DSQRT, DSIN, DCOS

Inputs: Data Sets NCONST, NELIPS, NTRANS

Arguments: MG(4) - Zone Number, Zone Letter,
Column Letter, Row Letter

EST - Easting

NTH - Northing

Parameters

Local Datum (NGAIDC)

Delta Flattening (NELDLF)

Change in Semi Major Axis (NELDLA)

Sources

N1XFRM

Block Data

Block Data

Outputs: Arguments: LAT - Local Level Latitude
LON - Local Level Longitude

Processing:

N2MQGP is called from N2NVIN for initialization if inputs were in military grid coordinates or from N1XFRM when waypoints have been entered using military grid coordinates. The latitude and longitude outputs are considered to be local datum latitude and longitude. The military grid Column letter, Row letter, Easting and Northing are first converted to UTM Easting and Northing and then to Latitude and Longitude.

A local eccentricity squared is first computed by the following equations:

$$f = f' + NELDLF(NGAISP)$$

where

f' = WGS-72 earths flattening
NGAISP is the local spheriod id
NELDLF is difference between local flattening
and WGS-72 flattening
 $EC2 = 2f - f^2$ The local eccentricity squared

The following is used to compute the UTM Easting:

$$(1) \text{ Easting} = (\text{NBR}(\text{MG}(3)) - 8((\text{MG}(1)-1) \bmod 3)) 100000 + \text{EST}$$

where

NBR is a statement function that maps the letters
A to Z less I and O to the numbers 1 to 24.
MG(1) = Military Grid Zone Number
MG(3) = Military Grid Column Letter
EST = Military Grid Easting

The UTM Northing is computed by the following sequence of
calculations:

N = Row offset identifier computed in N2OSMG

IBOT = (NBR(MG(2)) - 13) 8.8956
If IBOT is < zero then add 99 to IBOT

IROW = NBR(MG(4)) - 1 - N + ((IBOT + N) mod 20)
If IROW is < than IBOT then add 20 to IROW

$$(2) \text{ Northing} = \text{IROW} 100000 + \text{NTH}$$

where

MG(4) = Military Grid Row Letter
MG(2) = Military Grid Zone Letter

NTH = Military Grid Northing

The UTM Easting and Northing are first converted to
cartesian coordinates relative to the equator by:

$$X = (\text{Easting} - 500000) / .9996$$

$$Y = (\text{Northing} - 10000000) / .9996 \quad \text{if } \text{NBR}(\text{MG}(2)) < 78$$

$$Y = \text{Northing} / .9996 \quad \text{otherwise}$$

The Latitude and Longitude in radians are computed by the

following sequence of computations:

$$S = (1 - \sqrt{1 - EC^2}) / (1 + \sqrt{1 + EC^2})$$

$$a = a' + NELDLA(NGAISP)$$

where

a' is the WGS-72 Semi Major Axis

NELDLA is the difference between WGS-72 and local level

a is the local level Semi Major Axis

$$w = Y / (a (1 - S)(1 + 5/4S)^2)$$

w is the Rectifying Latitude

$$PHI = w + S(3/2 \sin(2w) + 21/16 S \sin(4w))$$

PHI is the Footprint Latitude

$$R = a / \sqrt{1 - EC^2 \sin^2(PHI)}$$

R is the Radius of Curvature Normal to the Meridian

$$ETA^2 = EC^2 \cos^2(PHI) / (1 - EC^2)$$

$$Z = X / R$$

$$LAMBDA = 6 MG(1) + 183$$

LAMBDA is the Longitude of Central Meridian

$$t = \tan(PHI) \quad \text{the tangent of PHI}$$

$$c = \cos(PHI) \quad \text{the cosine of PHI}$$

$$(3) \quad LAT = PHI + (1+ETA^2) t Z (-1+Z (t^2/4 + 5/12))$$

$$(4) \quad LON = LAMBDA + Z/c (1 - Z^2/6 (1 + 2 t^2 + ETA^2) + Z (1/4 + t (7/5 + 6/5 t)))$$

5.3.16

Mnemonic: N2MOLD

Title: Molodensky Latitude and Longitude corrections

Priority: 640 ms, 320 ms or Background

Invoked by: N1XFRM, N2NVIN

Invokes: none

Inputs: Data Sets NCONST, NELIPS, NTRANS

Arguments: LAT - Latitude
LON - Longitude

<u>Parameters</u>	<u>Sources</u>
Local datum (NGAIDC)	N1XFRM
Delta Flattening (NELDLF)	Block Data
Delta Major Axis (NELDLA)	Block Data
Delta X (NELTDX)	Block Data
Delta Y (NELTDX)	Block Data
Delta Z (NELTDX)	Block Data

Outputs: Arguments: DELAT - Delta latitude
DELON - Delta longitude

Processing:

N2MOLD computes the Molodensky adjustments to convert WGS-72 Latitude and Longitude to a local datum Latitude and Longitude or vice versa. Each local datum has a different center of the earth location, and the 46 different local datums are mapped to 11 different spheroids. Each spheroid has a different eccentricity, semi-major axis, and a flattening. Applying the difference between WGS-72 spheroids and the local datum spheroid, the delta latitude and longitude are computed to be added to the local datum latitude and longitude or subtracted from the WGS-72 latitude and longitude to get the other.

The following symbols are used in the equations below:

NGAISP = Local Spheroid number = NELISP(NGAIDC)
 DX = NELDLX(NGAIDC)
 DY = NELDLY(NGAIDC)
 DZ = NELDLZ(NGAIDC)
 DX, DY, DZ are offsets of x, y, z of the center
 of the earth.
 DF = Change in Earths Flattening = NELDLF(NGAISP)
 DA = Change in Semi Major Axis = NELDLA(NGAISP)
 EC2 = Eccentricity Squared
 a = Semi Major Axis WGS 72
 f = Earth Flattening WGS 72

The equations are as follows:

$$t1 = -DX \sin(LAT) \cos(LON) - \\ DY \sin(LAT) \sin(LON) + \\ DZ \cos(LAT)$$

$$t2 = 2 (a DF + f DA) \cos(LAT) \sin(LAT)$$

$$RM = a (1 - EC2) / (\sqrt{1 - EC2 \sin^2(LAT)})^3$$

RM is the Radius of Curvature in the Meridian

$$(1) \quad DELAT = (t1 + t2) / RM$$

$$t1 = -DX \sin(LON) + DY \cos(LON)$$

$$RN = a / \sqrt{1 - EC2 \sin^2(LAT)}$$

RN is the Radius of the Prime Vertical

$$(2) \quad DELON = t1 / (RN \cos(LAT))$$

5.3.17

Mnemonic: N2MSLH

Title: Mean Sea Level Altitude Correction

Priority: 640 ms, 320 ms or Background

Invoked by: N1XFRM, N2NVIN

Invokes: none

Inputs: Data Sets NCONST, NELIPS, NTRANS

Arguments: LAT - Local level latitude
LON - Local level longitude

<u>Parameters</u>	<u>Sources</u>
Table of Corrections (NELITN)	Block Data

Outputs: Arguments: HCDR - Altitude Corrections

Processing:

N2MSLH computes an Altitude correction that is the difference between Mean Sea Level and WGS-72 geodetic altitude. The correction is computed in meters from a gridded database which contains the corrections for every 10 degrees of latitude and longitude. Given a latitude and longitude, N2MSLH will interpolate the four points surrounding them. The first step is to compute the indices in the table. Let RLON and RLAT be longitude and latitude in 10 degree units and let ILON and ILAT be the truncated values of RLON and RLAT.

For longitude indices:

I1 = ILON + 19
 I2 = I1 + 1 If longitude is positive
 I2 = I1 - 1 If longitude is negative
 If I2 is equal to 37 then I2 = 1

For latitude indices:

$$J1 = ILAT + 10$$

$$J2 = J1 + 1 \quad \text{If latitude is positive}$$

$$J2 = J1 - 1 \quad \text{If latitude is negative}$$

Interpolation offset is computed by:

$$X = \text{abs}(RLON - ILON)$$

$$Y = \text{abs}(RLAT - ILAT)$$

where

abs is the absolute value.

WWW is a statement function that is used to apply a weight to the four points in interpolation where

$$WWW(X, Y) = X^2 Y^2 (9 - 6(X + Y) + 4XY)$$

Then the correction is computed as follows:

$$\begin{aligned} \text{HCOR} = & WWW(X, Y) \text{NELITN}(I2, J2) \\ & + WWW(1-X, Y) \text{NELITN}(I1, J2) \\ & + WWW(X, 1-Y) \text{NELITN}(I2, J1) \\ & + WWW(1-X, 1-Y) \text{NELITN}(I1, J1) \end{aligned}$$

5.3.18

Mnemonic: N2NEQS

Title: Propagation of User State

Priority: 320 ms

Invoked by: N1MITK

Invokes: none

Inputs: Data Sets NCONST, NINNER

<u>Parameters</u>	<u>Sources</u>
User Position (NGACPS)	N2NVIN, N2NEQS, N1MITK
User Velocity (NGACVL)	N2NVIN, N2NEQS, N1MITK
User Acceleration (NGACAC)	N2NVIN, N2NEQS, N1MITK
Range Bias (NGRBIS)	N2NVIN, N2NEQS, N1MITK, N2MCNI
Range Rate Bias (NGRTBS)	N2NVIN, N2NEQS, N1MITK, N2MCNI

Outputs: Data Sets NINNER

<u>Parameters</u>	<u>Destination</u>
User Position (NGACPS)	N2MCNI, N2HMTR, N1MITK
User Velocity (NGACVL)	N2MCNI, N2HMTR, N1MITK
User Acceleration (NGACAC)	N2MCNI, N2HMTR, N1MITK
Range Bias (NGRBIS)	N2MCNI, N2HMTR, N1MITK

Processing:

N2NEQS integrates the navigation equations of motion to propagate user position, velocity, acceleration and range bias forward 320 ms in time. The equation that N2NEQS uses is $\dot{X} = \text{PHI } X$ where X is the vector with 11 entries of filter states to be propagated and PHI is the constant state transition matrix.

$\underline{X} =$

NGACPS(1)	x	position
NGACPS(2)	y	position
NGACPS(3)	z	position
NGACVL(1)	x'	velocity
NGACVL(2)	y'	velocity
NGACVL(3)	z'	velocity
NGACAC(1)	x''	acceleration
NGACAC(2)	y''	acceleration
NGACAC(3)	z''	acceleration
NGRBIS	RB	range bias
NGRTBS	RB'	range rate bias

	1	0	0	.32	0	0	RA	0	0	0	0
	0	1	0	0	.32	0	0	RA	0	0	0
	0	0	1	0	0	.32	0	0	RA	0	0
	0	0	0	1	0	0	VA	0	0	0	0
	0	0	0	0	1	0	0	VA	0	0	0
PHI =	0	0	0	0	0	1	0	0	VA	0	0
	0	0	0	0	0	0	AA	0	0	0	0
	0	0	0	0	0	0	0	AA	0	0	0
	0	0	0	0	0	0	0	0	AA	0	0
	0	0	0	0	0	0	0	0	0	1	.32
	0	0	0	0	0	0	0	0	0	0	0

where

RA = .04942709324
VA = .3035243056
AA = .8988252315

N2NEGS does not store all of PHI, since only 4 constants are used therefore the following equations are breakdowns of the above formula.

Position:

$$NGACPS(I) = NGACPS(I) + .32 NGACVL(I) + NGACAC(I) RA$$

Velocity:

$$NGACVL(I) = NGACVL(I) + NGACAC(I) VA$$

Acceleration:

$$NGACAC(I) = NGACAC(I) AA$$

Range Bias:

$$NGRBIS = NGRBIS + .32 NGRTBS$$

I goes from 1 to 3 for the 3 coordinates x, y, z of the earth fixed, earth centered coordinate system.

5.3.19

Mnemonic: N2NVINTitle: NAV InitializationPriority: At power up or when update button pushedInvoked by: N1INIT, N1MITKInvokes: N2MGGP, N2MOLD, N2MSLH, DCOS, DSINInputs: Data Sets MNXXXX, NCONST

<u>Parameters</u>	<u>Sources</u>
Waypoint Number (MNCWYP(16))	M1ADIS
User Heading (MNCWYP(11))	M1ADIS
User Speed (MNCWYP(12))	M1ADIS
User Altitude (MNIWYP(9,10))	M1ADIS
User Position (MNIWYP(1-8))	M1ADIS
Local Datum ID (MNCWYP(13))	M1ADIS

Outputs: Data Sets NTRANS, NOUTER, NINOUT, NINNER

<u>Parameters</u>	<u>Destination</u>
NAV Initialization Flag (MNCNAV(1))	N2MCNI
Computed Range (NGRNGC)	N1MITK, N2MCNI
Computed Range Rate (NGRTCM)	N1MITK
Range Bias Rate (NGRTBS)	N2MCNI
Altitude (NGBALT)	N1XFRM
Speed (NGVTAS)	N1XFRM, N2WPCM
Heading (NGPSIT)	N1XFRM, N2WPCM
WGS-72 Latitude (NGLATD)	N1XFRM, N2IOND, N2WPCM
WGS-72 Longitude (NGLNGD)	N1XFRM, N2IOND, N2WPCM
Mean Sea Level Altitude (NGALTD)	N1XFRM, N2IOND, N2WPCM
Waypoint Latitude (NWPLAT)	N2WPCM
Waypoint Longitude (NWPLON)	N2WPCM
Waypoint Altitude (NWPALT)	N2WPCM
Rotation Matrix (NGTRO1)	N1XFRM, N2NVIN, N2FOTP
User Position Earth-Fixed (NGACPS)	N2MCNI, N2HMTR, N1XFRM
Iteration Parameter (NGAGQQ)	N1XFRM
Altitude Hold Position (NXYZSV)	N1MITK
Acceleration Earth-Fixed (NGACAC)	N2MCNI, N1MITK, N2NEQS
Velocity Earth-Fixed (NGACVL)	N2MCNI, N2HMTR
Covariance Matrix (NGCOVP)	N2MCNI, N2COVR, N2FCPG
Average Range Residual (NGSUMS)	N1MITK
Average Range Rate Resid. (RDSUMS)	N1MITK
Fudge Factor (NGFFCT)	N2MCNI, N2FOTP
Iono/Tropo Corrections (NGDELA)	N2MCNI
Sv Clock Corrections (NGZCLK)	N2MCNI
Lever Arm Corrections (NGLEVR)	N2MCNI
Doppler Corrections (NGZDOP)	N2MCNI
Innovation Factor (NGINVM)	N2MCNI

Range Gains (NGRGNS)	N1MITK, N2MCNI
Range Rate Gains (NGRTGN)	N1MITK, N2MCNI
Altitude Hold Gains (NALHGN)	N1MITK, N2MCNI

Processing:

N2NVIN is the initialization routine for the navigation data sets. The initialization state flag is set to 3 for the first step in initialization. The computed ranges and range rates are set to 22000000 and 0. The initial user states are set according to the operator inputs. Altitude, speed and heading are taken from operator inputs and scaled to meters, meters per second and radians. Position is either in military grid or latitude and longitude, both of which are relative to a local datum. N2MGGP is called to convert the military grid to local datum latitude and longitude. The latitude and longitude inputs are converted from degrees minutes seconds to radians. Then N2MOLD will compute corrections to convert the local datum latitude and longitude to WGS-72 latitude and longitude. N2MSLH is called for the altitude correction which is added to the mean sea level altitude to get the WGS-72 altitude. If the waypoint switch is not in position zero then the WGS-72 latitude, longitude and altitude are saved in the waypoint position array.

The rotation matrix is computed as follows:

```

NGTRO1(1,1) = -sin(NGLNGD)
NGTRO1(1,2) =  cos(NGLNGD)
NGTRO1(1,3) =  0
NGTRO1(2,1) = -cos(NGLNGD) sin(NGLATD)
NGTRO1(2,2) = -sin(NGLNGD) sin(NGLATD)
NGTRO1(2,3) =  cos(NGLATD)

```

```

NGTRO1(3,1) = cos(NGLNGD) cos(NGLATD)
NGTRO1(3,2) = sin(NGLNGD) cos(NGLATD)
NGTRO1(3,3) = sin(NGLATD)

```

The earth fixed user coordinates are computed by:

```

R = a / sqrt(1 - e2 sin2(NGLATD))
Q = R NGALTD cos(NGLATD)
NGACPS(1) = Q cos(NGLNGD)
NGACPS(2) = Q sin(NGLNGD)
NGACPS(3) = ((1 - e2) R NGALTD) sin(NGLATD)

```

where

```

a = Earths Equatorial Radius
e = Earths Eccentricity

```

The iteration parameter used to convert earth fixed coordinates to latitude longitude and altitude is set to eccentricity squared. The east and north velocities are computed from the input speed and heading as follows:

```

East Velocity = NGVTAS sin(NGPSIT)
North Velocity = NGVTAS cos(NGPSIT)

```

Up velocity and the earth fixed accelerations are set to 0. The altitude hold position is copied from NGACPS. The earth fixed velocities are computed from the east and north velocities and the rotation matrix:

```

NGACVL(i) = NGTRO1(1,i) NGAVES + NGTRO1(2,i) NGAVNR
i = 1,2,3 => (x',y',z')

```

The off diagonal elements of the covariance matrix are set to 0, while the diagonal elements are set as follows. The (1,1) (2,2) and (3,3) elements are set to 1.0 E10. The (4,4) (5,5) and (6,6) elements are set to 10000. The (7,7)

(8,8) and (9,9) elements are set to 100. The (10,10) element is set to $1.0 \text{ E}10$, and the (11,11) element is set to zero.

The rolling averages of normalized range and range rate residuals squared are set to 50. The fudge factor is 1, and all the correction, Iono/Tropo, SV Clock, Antenna Lever Arm and Doppler, are set to 0. The innovations for ranges are set to 25 and the innovations for range rates are set to 0.25. All the range gains, range rate gains and altitude hold gains are set to 0.

5.3.20

Mnemonic: N2OSMG

Title: Row Offset Identifier

Priority: 640 ms, 320 ms or background

Invoked by: N2GPMG, N2MGGP

Invokes: none

Inputs: Data Sets NELIPS

Arguments: MG - Military Grid Zone Number and
Zone Letter

<u>Parameters</u>	<u>Sources</u>
Local Spheriod ID (NGAISP)	NIXFRM

Outputs: Arguments: NNN - Row Offset

Processing:

N2OSMG computes the Row Offset Identifier for military grid conversion. NNN is computed depending on Zone Number, Zone Letter and Spheriod ID. NNN is computed as follows:

For the following situations.

- (i) NGAISP = 2 and $47 > MG(1) > 51$
- (ii) NGAISP = 3 and $50 \leq MG(1) \leq 52$
- (iii) NGAISP = 4 and $81 \leq MG(2)$ and $MG(1) \leq 51$
- (iv) NGAISP = 5 and $MG(1) \geq 47$
- (v) NGAISP = 7

Then $NNN = 0$ If $MG(1)$ is odd
 $NNN = 5$ If $MG(2)$ is even

For NGAISP = 6

Then $NNN = 10$ if $MG(1)$ is even
 $NNN = 15$ if $MG(1)$ is odd

For all other cases

$NNN = 10$ if $MG(1)$ is odd
 $NNN = 15$ if $MG(1)$ is even

5.3.21

Mnemonic: N2SVEC

Title: Satellite Fit Coefficients or Ranges

Priority: Background

Invoked by: N1SVPN, N2SVSL

Invokes: DSGRT, DSIN, DCOS, X3REQ, X3REL

Inputs: Data Sets NSVPOS, NCONST, MNXXXX, MMALRT, MIEPHM
MPALMC, NINTRF, NINNER

Arguments: ISV - Generic SV ID or SV ID
TIM - Time of Coefficients or Ranges
MDE - Mode for Coefficients or Ranges

<u>Parameters</u>	<u>Sources</u>
Satellite ID (MMSVID)	N2SVSL
Coefficient Buffer Switch (NGFITX)	N1SVPN
Ephemeris Avail Flag (MIEVFL)	M2DBS2, M2DBS1
Ephemeris Data (MIEDAT(1-24))	M2DBS2, M2DBS1
Almanac Data (MPADAT(1-16))	M2DBPR, M2DBS3
Source Status (MMSTUS)	N1SVPN, N2FOTP, N2SVSL, N2MCNI
GT Positions (NGCGTX, NGCGTY, NGCGTZ)	Block Data
User position (NGACPS)	N1MITK, N2NEQS, N2NVIN

Outputs: Data Sets NSVPOS

<u>Parameters</u>	<u>Destination</u>
SV Positions (NGXYZP)	N2SVEC
SV Fit 0 Coefficient (NGFIT0)	N2HMTR
SV Fit 1 Coefficient (NGFIT1)	N2HMTR
SV Fit 2 Coefficient (NGFIT2)	N2HMTR
Time of Coefficients (NGFITT)	N2HMTR
SV Ranges from Almanac (NGSVRN)	N2SVSL
SV Rising/Setting Flag (NGSVST)	N2SVSL

Processing:

N2SVEC will compute satellite fit coefficients if the input MDE is 1, or it will compute satellite ranges from almanac if MDE is 2. The choice of MDE = 2 is used for automatic satellite selection only.

When MDE is 1 the input ISV is the generic ID having a

value of 1 to 5. The actual ID is obtained from MMSVID(ISV). The satellite positions are computed at time TIM and 60 seconds before TIM and 60 seconds after TIM.

If MMSVID(ISV) is greater than 32, and there is no ephemeris available, then the source is a ground transmitter whose position is gotten from NGCGTX, NGCGTY, NGCGTZ. otherwise, the ground transmitter coordinates x,y,z are found in array MIEDAT, and must be converted to meters by applying the scale factor $7.8125E-3$. For GT's the fit coefficients NGFIT0 are the x,y,z components of the GT, and all NGFIT1 and NGFIT2 entries are 0.

For satellites N2SVEC checks MIEVFL for ephemeris validity. If there is no ephemeris, then almanac is used in data set MPALMC. When ephemeris is used, sole access is requested for ephemeris data by calling X3REQ. Then the data is copied from MIEDAT to a local 24 word array. The ephemeris data is either 16 or 32 bit numbers, where one word is 16 bits. The following table shows what the data is and how it is scaled.

	<u>NAME</u>	<u>WORD NUMBER</u>	<u>SCALE FACTOR</u>	<u>DESCRIPTION</u>
1.	MMO	1 and 2	2 PI -31	Mean Anomaly
2.	ECNT	3 and 4	2 -33	Eccentricity
3.	SRA	5 and 6	2 -18	Square Root of A
4.	OMO	7 and 8	2 PI -31	Right Ascension
5.	IIO	9 and 10	2 PI -31	Inclination Angle
6.	OMG	11 and 12	2 PI -43	Argument of Perigee

7.	QMD	13 and 14	2	PI	Rate of Right Ascension
			4		
8.	TDE	15 and 16	2		Ephemeris Reference Time
			-5		
9.	CRS	17	2		Sine of Harmonic Correction to Orbit Radius
			-43		
10.	DLN	18	2	PI	Mean Motion Correction
			-29		
11.	CUC	19	2		Cosine of Harmonic Correction to Argument of Latitude
			-29		
12.	CUS	20	2		Sin of Harmonic Correction to Argument of Latitude
			-29		
13.	CIC	22	2		Cosine of Harmonic Correction to Inclination
			-29		
14.	CIS	23	2		Sine of Harmonic Correction to Inclination
			-29		
15.	CRC	24	2		Cosine of Harmonic Correction to Orbit Radius

where

PI is 3.141592654.

X3REL is used to release the sole access request of ephemeris data. If ephemeris is not available then almanac is used. Again X3REQ is used to obtain sole access of almanac data which is copied from MPADAT to a local 14 word array. The scale factors are listed below.

	<u>NAME</u>	<u>WORD NUMBER</u>	<u>SCALE FACTOR</u>	<u>DESCRIPTION</u>
			-21	
1.	ECNT	1 and 2	2	Eccentricity
			-11	
2.	SRA	3 and 4	2	Square Root of A
			-23	
3.	DMO	5 and 6	2	PI
				Right Ascension

			-23	
4.	DMG	7 and 8	2 PI	Argument of Perigee
			-23	
5.	MMO	9 and 10	2 PI	Mean Anomaly
			12	
6.	TOE	11	2	Time of Almanac Data
			-19	
7.	IIO	12	2 PI	Actual Inclination Angle
			-38	
8.	OMD	14	2 PI	Rate of Right Ascension

and DLN, CUC, CUS, CIC, CIS, CRC, CRS are all zeros.

The positions at time TTK are computed as follows:

$$NNN = \sqrt[3]{\mu} / SRA + DLN$$

Corrected Mean Motion

$$\mu = \text{WGS-72 Earths Universal Gravitational Constant}$$

$$= 3.986008E14 \text{ meters}^3/\text{seconds}^2$$

$$TTK = TIM - 60 - TOE$$

Time from Epoch

$$MMK = NNN TTK + MMO$$

Mean Anomaly

EEK is set initially to MMK
Iterate four times using the equation:

$$EEK = \sin(EEK) ECNT + MMK$$

Eccentric Anomaly

$$RRK = -(\cos(EEK) ECNT - 1)$$

Corrected Radius

$$COSV = (\cos(EEK) - ECNT) / RRK$$

$$SINV = \sqrt[2]{-(ECNT - 1)} \sin(EEK) / RRK$$

Cosine and Sine of True Anomaly

$$SINF = SINV \cos(OMG) + COSV \sin(OMG)$$

$$COSF = COSV \cos(OMG) - SINV \sin(OMG)$$

Sine and Cosine of Argument of Latitude

$$SINH = 2 COSF SINF$$

$$COSH = 2 COSF COSF - 1$$

Sine and Cosine of Second Harmonic for Perturbations

$$SINDU = SINH CUS + COSH CUC$$

$$COSDU = -(SINDU SINDU) / 2 - 1$$

$$SINU = COSDU SINF + SINDU COSF$$

$$COSU = COSDU COSF - SINDU SINF$$

Sine and Cosine of Corrected Argument of Latitude

$$XORB = (RRK SRA^2 + (CRC COSH) + (CRS SINH)) COSU$$

$$YORB = (RRK SRA^2 + (CRC COSH) + (CRS SINH)) SINU$$

X and Y Positions in Orbital Plane

$$SINDI = CIC COSH + CIS SINH$$

$$COSDI = -(SINDI SINDI) / 2 - 1$$

$$SINI = \cos(IIO) SINDI + \sin(IIO) COSDI$$

$$COSI = \cos(IIO) COSDI - \sin(IIO) SINDI$$

Sine and Cosine of Corrected Inclination

$$OMK = (OMD - OME) TTK + OMO - TOE OME$$

Corrected Longitude of Ascending Node at SV
Transmit Time
OME = Earths Rotation Rate

$$NGXYZP(1, j) = XORB \cos(OMK) - YORB COSI \sin(OMK)$$

$$NGXYZP(2, j) = XORB \sin(OMK) + YORB COSI \cos(OMK)$$

$$NGXYZP(3, j) = YORB SINI$$

where

j = 1, 2, 3 for the 3 sets of coordinates
 For j = 1 coordinates at time TIM - 60 seconds
 For j = 2 coordinates at time TIM
 For j = 3 coordinates at time TIM + 60 seconds

Increment time TTK by 60 seconds and perform the above steps again.
 Fit coefficient are then ready to be computed after 3 sets of
 satellite positions have been computed.

$$NGFITO(i) = NGXYZP(i, 2)$$

$$NGFIT1(i) = (NGXYZP(i, 3) - NGXYZP(i, 1)) / 120$$

$$NGFIT2(i) = (NGXYZP(i, 3) - 2 NGXYZP(i, 2) + NGXYZP(i, 1)) / 7200$$

where

i = 1, 2, 3 for the 3 coordinates x, y, z

When MDE is 2 then only 2 sets of satellite positions are computed. The range to both positions are computed by the sum of the squares of the differences between user and satellite coordinates. The first range is stored in NGSVRN for automatic satellite selection. The second range is subtracted from the first to determine rising or setting.

If the difference is negative then
set NGSVST to 1 for setting

If the difference is positive then
set NGSVST to 2 for rising

If MDE is 2 and the almanac does not exist then both
NGSVRN and NGSVST are set to zero.

5. 3. 22

Mnemonic: N2SVSL

Title: Satellite Selection Routine

Priority: Background

Invoked by: N1SVPN

Invokes: N2SVEC, X3TIMM, X3REG, X3REL

Inputs: Data Sets MNXXXX, MMALRT, MIEPHM, MTTCDs, NSVPOS,
MPALMC, NINNER

<u>Parameters</u>	<u>Sources</u>
Active Sources (MMSVID)	N2SVSL
Source Status (MMSTUS)	N1SVPN, N2SVSL, N1MITK, N2FOTP
Operator Sources (MNCRCV)	M1ADIS
Range to Source (NGRNGC)	N1MITK
Ranges from Almanac (NGSVRN)	N2SVEC
Source Movement (NGSVST)	N2SVEC
Receiver Number (MMRECN)	N2FOTP

Outputs: Data Sets MMALRT

<u>Parameters</u>	<u>Destination</u>
Source Status (MMSTUS)	NAV routines
Active Sources (MMSVID)	N1SVPN, N2SVEC, N2FOTP
Ephemeris Avail Flag (MIEVFL)	N2SVEC
Clock Corrections Avail (MTTVFL)	N2SVEC
Source ID (MNCRCV)	M1ADIS

Processing:

N2SVSL will construct the list of active sources, either automatically or from operator requests, and keep it updated. The sources can be either ground transmitters or satellites.

The automatic mode will only select satellites, and will select four or less depending on the elevation and availability of the source. The steps taken for automatic satellite selection are as follows:

1. Obtain a source ID already in use from the active list MMSVID.
2. If the ID is zero and it is the first ID that is zero then look at the next source.
3. If the ID is zero, and is not the first zero ID, then a search is made of all 24 satellites to determine an addition.
4. If the ID is not zero, and the range is greater than 25233032 meters (less than 5 degrees elevation), then the source is marked as not visible and a search is made for a replacement.
5. If the ID is not zero and the range is less than 25233032, the source is placed in the commanded array MMSVCM.

The steps for searching the satellites for the best source to add is as follows:

1. N2SVEC is called, if needed, every two minutes to have ranges computed from the almanacs. The rising/setting flag NGSVST is set to zero if no almanac is available, to 1 if the source is setting and to 2 if the source is rising.
2. A search is made of all 24 ranges computed and the following tests are performed:
 - A. Is the source available? If not look at the next source.
 - B. Are we already using the source? If so look at the next source.
 - C. Is the source higher than 5 degrees elevation? If not look at the next source.
 - D. On the first search of the ranges we choose the source that is rising and is the closest satellite to 5 degrees elevation.
 - E. A second search is made only if the first search did not select a source. The second time through, we look at setting sources and choose the source with the smallest range (ie. is highest in the sky).

When a source is selected then the rising/setting status is

set to zero to indicate the satellite is now unavailable for further selections.

If the satellite selection is manual, then the sources desired are checked for validity. If the source has a valid ID and an almanac exists, then it is copied to the commanded SV array. If the source has no almanac or the ID is invalid, then MNCRCV(20) is set for the CDU to respond.

The next step N2SVSL does is to construct add and delete queues from the commanded sources array. If the active sources in MMSVID are not equal to zero, and the ID is not in the commanded sources, then the source is added to the delete queue. If the commanded source is not equal to zero, and is not in the active array, then the source is put in the add queue. All additions that can be made are done before any deletions.

To add a source to the active list, we find a slot in MMSVID where the current ID is zero, and place the ID from the top of the add queue there. The status is set to 1, the receiver number is set to 0 and the update flag is set to -1. N2SVSL will add as many sources as possible, until either the add queue is empty or the active list has 5 sources.

To delete a source we first locate the source in the active list that matches the top of the delete queue. The source will be deleted if the status is less than 4 or the receiver number is zero. The source will not be deleted if there are less than 5 sources with status of 5 (ie. MMNVSU <

5), and the source to be deleted has a status of 4 or 5. If the source can be deleted, then MMNVSV is set to 4, if it is 5. The ID is set to zero. The status is set to zero. The ephemeris and clock corrections availability flags are set to zero, and the receiver number is set to zero. The delete queue is updated to remove the deleted source. Then, if the delete queue is not empty, it is reordered, and the next source is processed for deletion.

5.3.23

Mnemonic: N2WPCM

Title: Waypoint Calculations Routine

Priority: 640 ms

Invoked by: N1XFRM

Invokes: N2GPMG

Inputs: Data Sets: MNXXXX, NTRANS, NCONST, NINTRF

Arguments: NB = Buffer switch for MNOWYP.

<u>Parameter</u>		<u>Source</u>
Convergence angle for waypoint calc.	(NGCNVG)	N2GPMG
Present user earth-fixed position	(NGXCPS)	N2MCNI
Present user altitude	(NGALTD)	N2MCNI
Direction cosine matrix	(NGTRO1)	N1XFRM, N2NVIN
Waypoint switch number	(MNOCDU)	M1ADIS
Waypoint latitude (or northing)	(MNIWYP)	M1ADIS
Waypoint longitude (or easting)	(MNIWYP)	M1ADIS
Waypoint altitude	(MNIWYP)	M1ADIS
User ground speed	(NGACVG)	M1ADIS
Earth eccentricity-squared	(NGCNST)	Block Data
Earth radius	(NGCNST)	Block Data
Horizontal approach angle	(MNIWYP)	M1ADIS
Vertical approach angle	(MNIWYP)	M1ADIS

Outputs: Data Sets: MNXXXX

<u>Parameter</u>		<u>Destination</u>
Range to waypoint	(MNOWYP)	M1ADIS
Bearing to waypoint	(MNOWYP)	M1ADIS
Time-to-go to waypoint	(MNOWYP)	M1ADIS
Horizontal crnss-track error	(MNOWYP)	M1ADIS
Vertical cross-track error	(MNOWYP)	M1ADIS
Waypoint-behind flag	(MNSTAT)	M1ADIS

Processing

N2WPCM calculates several waypoint parameters: range, bearing, time-to-go, horizontal cross-track (steering) error, and vertical cross-track (steering) error for use by the GPS user/operator via the Control/Display Unit (CDU) for local-level, real time navigation.

AD-A076 481

TEXAS INSTRUMENTS INC DALLAS EQUIPMENT GROUP

F/G 17/7

GLOBAL POSITIONING SYSTEMS (GPS) HIGH DYNAMIC USER EQUIPMENT. (--ETC(U)

F04701-75-C-0180

AUG 79

UNCLASSIFIED

SAMS0-SD-TR-79-12-VOL-3

NL

3 OF 4

AD
A076481



The processing flow for N2WPCM is as follows: (1) Determine if earth-fixed coordinates have already been calculated from a previous call; if not, do this conversion; (2) Convert earth-fixed coordinates to local-level (east, north, vertical) with origin at present user position; (3) Calculate range from the users position to waypoint; (4) Call subroutine, N2GPMG, with arguments: "2, latitude, longitude" to get convergence angle correction for bearing; (5) Calculate bearing to waypoint; (6) Determine if waypoint is "before" or "behind" user; (7) Determine time-to-go to waypoint; (8) Determine horizontal and vertical cross-track errors.

5.4 EXECUTIVE SOFTWARE MODULE DESCRIPTIONS

5.4.1

Mnemonic: B1SLIO

Title: Slave bus I/O service task

Priority: 10 ms

Invoked by: R1MRC

Invokes: X3ERRA, X3WAIT, F\$RGMV

Inputs: From data set XCOUNT, BCRVRT, BUSDTA; bus message
pointed to by BUSDTA

<u>Parameter</u>	<u>Source</u>
10 ms count, mod 2 (XC0010)	XIIT10
Message destination address (BCRVRT)	BCRVRT
Message source address (BUSDTA)	BUSDTA

Outputs: Bus message (pointed to by BCRVRT)

Processing:

B1SLIO provides service to the slave serial bus interface module (SBIM) in the receiver subsystem. It may either receive a 14-word message or transmit a 15-word message clocked serially into/out from memory.

Upon power-up, the bus is emptied (the contents are all clocked out). In normal operation, the task will execute only in the latter half of a 20 ms period. If this condition is satisfied, the hardware status register is checked to see if a message is present. If so, it is taken off of the bus and placed into memory starting at the location pointed to by BCRVRT. The correct offset into BCRVRT is determined from the routing indicator passed as

the first word of each message. If no message is present, the bus message beginning at the location pointed to by BUSDTA is put onto the bus. The correct offset into BUSDTA is computed from the FTF count, mod 16, passed as the first word of the message.

B1SLIO will log an error on input if the bus does not reply when expected, if a timing error was detected in the hardware, or if a parity error was detected. These errors are detected by reading the hardware status register.

5.4.2

Mnemonic: B2MSTR

Title: Master STD M BUS Interface Routine

Priority: 20 Milliseconds

Invoked by: M1CMSC

Invokes: (none)

Inputs: From Data sets: AMNXXX, MKXXXX

<u>Parameter</u>	<u>Source</u>
Hardware FIFO status register,	at address F9FA (hex)
Hardware shift register, bidirectional,	
serial to/from parallel, 16-bit,	at address F9FA (hex)
Hardware FIFO, 16-word,	at address F9FE (hex)
Vector of BUS I/O flags	(MNBSCN(1-10)) M1CMSC
Vector of BUS routing indicators	MNBSCN(11-20) M1CMSC
RCV3A - output BUS msg	(MNBSIO(313-328)) M1CRNC, M1CMSC
RCV3B - output BUS msg	(MNBSIO(329-344)) M2BUSO, M1ADIS
RCV3C - output BUS msg, up bfr	(MNBSIO(345-360)) M1CRNC
RCV3D - output BUS msg, lo bfr	(MNBSIO(361-376)) M1CRNC
RCV3E - output BUS msg, up bfr	(MNBSIO(377-392)) M2STIN
RCV3F - output BUS msg, lo bfr	(MNBSIO(393-408)) M2STIN

Outputs: To Data Sets: MNXXXX

<u>Parameter</u>	<u>Destination</u>
Hardware shift register, bidirectional,	
serial to/from parallel, 16-bit,	at address F9FA (hex)
Hardware FIFO, 16-word,	at address F9FE (hex)
RCV2A1 - input BUS msg	(MNBSIO(145-158)) M1CMSC, M2STIN, M1DBPR
RCV2A2 - input BUS msg	(MNBSIO(159-172)) M1CMSC, M2STIN, M1DBPR
RCV2A3 - input BUS msg	(MNBSIO(173-186)) M1CMSC, M2STIN, M1DBPR
RCV2A4 - input BUS msg	(MNBSIO(187-200)) M1CMSC, M2STIN, M1DBPR
RCV2A5 - input BUS msg	(MNBSIO(201-214)) M1CMSC, M2STIN, M1DBPR
RCV2B - input BUS msg	(MNBSIO(215-228)) M2STIN
RCV2C - input BUS msg	(MNBSIO(229-242)) M2STIN
RCV2D - input BUS msg	(MNBSIO(243-256)) M2STIN
RCV2E - input BUS msg	(MNBSIO(257-270)) M2STIN
RCV2F - input BUS msg	(MNBSIO(271-284)) M2STIN
RCV2G - input BUS msg	(MNBSIO(285-298)) M2STIN

Processing

B2MSTR is the Serial Time Division Multiplex (STDM) master bus controller, residing in the MSCP. It implements the following functions: (1) supervises the STDM bus input/output service; (2) provides STDM bus hardware status checks; (3) provides modulo-16 Fundamental Time Frame (FTF) count synchronization checks between the RCVF and MSCP subsystems.

The processing flow for STDM bus service is as follows: Array, MNBSCN (of Common, MNXXXX), contains two parts - I/O flags and select words. I/O flags are scanned and interpreted to find the bus message to be handled during current FTF. For input message processing, the First-In-First-Out shift register (FIFO) is set up to receive. When it is empty, the select word is retrieved, and data is moved from FIFO to array, MNBSIO. For output message processing, the FIFO is set up to transmit. When it is empty, the routing indicator is retrieved, the message source buffer address in array, MNBSIO, is calculated, the select word and length. message is retrieved, the FIFO is loaded repeatedly, one 16-bit word each time, until the message transmission is completed.

5.4.3

Mnemonic: DATAN2

Title: Double precision arctangent

Priority: Reentrant utility

Invoked by: N1XFRM, N2WPCM

Invokes: X3ERR, F\$RGM2, F\$RITP, F\$XRER

Inputs: X, Y from argument list

<u>Parameter</u>	<u>Source</u>
Argument passed (X)	Calling subprogram
Argument passed (Y)	Calling subprogram

Outputs: DATAN2

<u>Parameter</u>	<u>Destination</u>
Arctangent of X/Y in radians (DATAN2)	Calling subprogram

Processing:

DATAN2 computes the arctangent of the value X/Y and returns it as the value DATAN2. It requires the floating point arithmetic unit. X, Y, and DATAN2 must all be declared REAL*8. DATAN2 will log an error if X and Y are both equal to zero. The range of returned values is -PI to PI.

5.4.4

Mnemonic: DCOS

Title: Double precision cosine function

Priority: Reentrant utility

Invoked by: N2GPMG, N2MGGP, N2MOLD, N2NVIN, N2SVEC, N2WPCM

Invokes: X3ERR, F\$RQMY, F\$RITP, F\$XRER

Inputs: X from argument list

Parameter

Input angle in radians (X)

Source

Calling subprogram

Outputs: DCOS

Parameter

Cosine of specified angle

Destination

Calling subprogram

Processing:

DCOS computes the cosine of a given angle X in radians. DCOS and X must be declared REAL*8. The range of returned values is -1 to 1. DCOS will log an error if X is outside the range $-4*PI$ to $4*PI$.

5.4.5

Mnemonic: DEXP

Title: Double precision e**(X)

Priority: Reentrant utility

Invoked by: N2IONO

Invokes: X3ERR, F\$RQMY, F\$RITP, F\$XRER

Inputs: X from argument list

Parameter

Input value (X)

Source

Calling subprogram

Outputs: DEXP

Parameter

Returned value of e**(X)

Destination

Calling subprogram

Processing:

DEXP computes an approximation to the value of $e^{**}(X)$. DEXP and the input value X must be declared REAL*8. If X exceeds 174.673, DEXP will log an error. The returned value is always positive.

5.4.6

Mnemonic: DSIN

Title: Double precision sine function

Priority: Reentrant utility

Invoked by: N2QPMG, N2MGGP, N2MULD, N2NVIN, N2SVEC, N2WPCM

Invokes: X3ERR, F\$RGMY, F\$RITP, F\$XRER

Inputs: X from argument list

Parameter

Source

Input value (X)

Calling subprogram

Outputs: DSIN

Parameter

Destination

Returned value of sin(X) (DSIN)

Calling subprogram

Processing:

DSIN computes an approximation to sin (X). DSIN and X are declared REAL*8. Output values for DSIN are in the range -1 to 1. DSIN will log an error if the input value is outside the range $-4*PI$ to $4*PI$.

5.4.7

Mnemonic: DSGRT

Title: Double precision square root function

Priority: Reentrant utility

Invoked by: N1XFRM, N2FOTP, N2GPMG, N2HMTR, N2MGGP, N2MOLD,
N2NVIN, N2SVEC, N2WPCM

Invokes: X3ERR, F\$RQMY, F\$RITP, F\$XRER

Inputs: X from argument list

<u>Parameter</u>	<u>Source</u>
Input value (X)	Calling subprogram

Outputs:

<u>Parameter</u>	<u>Destination</u>
Returned square root value (DSGRT)	Calling subprogram

Processing:

DSGRT computes a double precision approximation to the square root of input value X. DSGRT and X are declared REAL*8. DSGRT logs an error if X is less than zero.

5.4.8

Mnemonic: EASHFT

Title: Extended Integer Arithmetic Shift

Priority: None (Executive service routine)

Invoked by: M2DBS1, M2DBS2, M2DBS3, M2MOVE

Invokes: X3ERRA

Inputs: From data set XIMSK, from argument list

<u>Parameter</u>	<u>Source</u>
Input number (NUMBER)	Calling subprogram
Shift count (ISHIFT)	Calling subprogram
Interrupt mask (XIMSKS)	XIMSK

Outputs:

<u>Parameter</u>	<u>Destination</u>
Shifted number	Calling subprogram

Processing:

EASHFT is a utility that performs an arithmetic shift on an extended integer argument. Passed as inputs are the number to be shifted and the shift count. Interrupts of level 6 and below are masked off. If the shift count is greater than 0, the number is shifted to the left and vice versa. An error is logged if the absolute value of the shift count is > 15 or if the sign bit changes during shifting. The shifted value is returned in registers 0 and 1 of the calling program.

5.4.9

Mnemonic: IASHFT

Title: Integer Arithmetic Shift

Priority: None (Executive service routine)

Invoked by: M2DBS3

Invokes: X3ERRA

Inputs: From data set XIMSK; NUMBER, ISHIFT from argument list

<u>Parameter</u>	<u>Source</u>
Input number (NUMBER)	Calling subprogram
Shift count (ISHIFT)	Calling subprogram
Interrupt mask (XIMSKS)	XIMSK

Outputs:

<u>Parameter</u>	<u>Destination</u>
Shifted number	Calling subprogram

Processing:

IASHFT is a utility that performs an arithmetic shift on an integer argument. Passed as inputs are the number to be shifted and the shift count. If the shift count is > 0 , the number is shifted to the left and vice versa. An error is logged if the absolute value of the shift count is > 15 or if the sign bit changes during shifting. The shifted value is returned in register 0 of the calling program.

5.4.10

Mnemonic: X1COMM

Title: MSC-NAV Executive communications

Priority: 20 ms

Invoked by: X1IT20 (20 ms interrupt handler)

Invokes: X3WAIT

Inputs: From data sets XJOINT, XZ, AXMNAV

<u>Parameter</u>	<u>Source</u>
X1COMM status (XJFINI)	X1COMM
FPAU status (XJFPAU)	X1COMM
Update command (XZUPDA)	X1IPOW, X1IT20

Outputs:

<u>Parameter</u>	<u>Destination</u>
Update command copy (XJUPDA)	X9JOIN

Processing:

X1COMM is responsible for the executive communications between the master state and navigation processors. On its initial execution, it initializes the FPAU status and exclusive access flags, along with the MSC/NAV communication flag. It then checks to see if X1COMM has run, checking the flag XJFINI for 1 millisecond.

The update command information is checked to see if the command needs to be moved to common memory. The command is copied unless it is also a restart command.

5.4.11

Mnemonic: X1COMN

Title: Navigation/master control executive communications

Priority: 20 ms

Invoked by: X1IT20 (Nav 20 ms interrupt handler)

Invokes: X3WAIT

Inputs: From data set XJDINT

<u>Parameter</u>	<u>Source</u>
FPAU status (XJFPAU)	F\$RITP

Outputs:

<u>Parameter</u>	<u>Destination</u>
X1COMN status (XJFINI)	X1COMM

Processing:

X1COMN opens the FPAU for contention if the Nav processor was using it in the last frame. Upon completion, it signals X1COMM by setting XJFINI to -1.

5.4.12

Mnemonic: X3ACT

Title: Process activate utility

Priority: None (executive service routine)

Invoked by: X1IPOW, M1CMSC, N1INIT, R1AID, R1MRC, R1PCK,

R1RNG, R1RRM, R1SRC

Invokes: X3ERRA

Inputs: From data sets XOPRC, XOLEV, XOTSK, XACRU, XIMSK,

XWORK, XDATA, XINT, XCOUNT; from argument list

ZOYYYY, RNUM

<u>Parameter</u>	<u>Source</u>
X3ACT workspace (XWORKS)	XWORK
Non-error interrupt mask (XIMSKE)	XIMSK
Pointer to XLEVEL block (OTSKPL)	XOTSK
Pointer to XPROC block (OTSKPR)	XOTSK
Maximum number of rcvrs (OTSKMC)	XOTSK
Length of proc block (PRCBLK)	XOPRC
Pointer to next process in same priority (OPRCNP)	XOPRC
Pointer to previous process in same priority (OPRCPP)	XOPRC
Pointer to first process in this priority (OLEVPF)	XOLEV
Entry point to task (OTSKEP)	XOTSK
FTF unit from which priority is formed (OLEVFU)	XOLEV
Process status (OPRCST)	XOPRC
CRU address for interrupt clear (XACRUC)	XACRU
CRU address for interrupt enable/disable (XACRUE)	XACRU
Pointer to task block (ZOYYYY)	Calling program
Receiver number (if any)	Calling program

Outputs: To data sets XOPRC, XCOUNT

<u>Parameter</u>	<u>Destination</u>
Pointer to next process in this priority (OPRCNP)	XOPRC
Pointer to previous process in this priority (OPRCPP)	XOPRC
Process program counter	XOPRC

Process workspace pointer
1 millisecond epoch

XOPRC
XCOUNT

Processing:

X3ACT is designed to place a designated task into the process table corresponding to its assigned priority. From information contained in the task block (priority level pointer, process pointer, max number of concurrent processes, and task entry point), X3ACT inserts the task into the process block for its priority. The process block is a doubly linked list which is pointed to by the variable OLEVPF. The process block contains, for each entry, a pointer to the preceding and succeeding tasks in the priority and the tasks workspace pointer, program counter and status register.

Receiver tasks are handled differently. First, the process block is of variable size and must be computed. Also, the receiver number is passed as an extra argument. If this receiver number exceeds the maximum, an error is logged. If the task to be activated is the 1 ms epoch task, the 1 ms interrupt is cleared, enabled, and the 1 ms epoch count is cleared.

In addition to the case cited above, errors are also logged if the requested task is already active, or if more than two arguments are passed to X3ACT.

5.4.13

Mnemonic: X3CANC

Title: Receiver process deactivation

Priority: None (executive service routine)

Invoked by: R1AID, R1MRC, R1PCK, R1RNG, R1RRM, R1SRC

Invokes: X3ERRA

Inputs: From data sets XOPRC, XOLEV, XOTSK, XACRU, XIMSK, XINT,
XWORK; from argument list ROYYYY, RNUM

<u>Parameter</u>	<u>Source</u>
Pointer to task block (ROYYYY)	Calling subprogram
Receiver number (RNUM)	Calling subprogram
Interrupt mask (XIMSK)	XIMSK
X3CANC workspace (XWORKS)	XWORK
Pointer to level block (OTSKPL)	XOTSK
Pointer to process block (OTSKPR)	XOTSK
FTF on which process was constructed (OLEVFU)	XOLEV
Pointer to next process in priority (OPRCNP)	XOPRC
Pointer to previous process in priority (OPRCPP)	XOPRC
CRU address to clear interrupts (XACRUC)	XACRU
CRU address to enable/disable interrupts (XACRUE)	XACRU

Outputs:

<u>Parameter</u>	<u>Destination</u>
Bit map of enabled interrupts (XIENAB)	XINT
Pointer to next process in priority (OPRCNP)	XOPRC
Pointer to previous process in priority (OPRCPP)	XOPRC

Processing:

X3CANC provides the means for process deactivation for the receiver subsystem. It removes the cancelled subprogram's process block from the priority level linked list. If the cancelled task is a 1-ms task, the 1-ms

interrupt is disabled , cleared, and removed from the bit map.

X3CANC will log an error if more than two arguments are passed, or if the receiver number passed is greater than that allowed.

5.4.14

Mnemonic: X3ERR, X3ERRA

Title: Error processing

Priority: None (executive service routine)

Invoked by: Several

Invokes: X3ERR invokes X3ERRA

Inputs: From data sets XIMSK, XHOME

<u>Parameter</u>	<u>Source</u>
Interrupt mask for all interrupts (XIMSKO)	XIMSK
Subsystem ID (XHOMSU)	XHOME
20 ms time mark -- second word (XCO022)	XCOUNT
Next location in error buffer to be filled (XEPTRF)	XERROR
Location following last word retrieved (XEPTRE)	XERROR
First word in error buffer (XEBUFS)	XERROR
Location following last word in error buffer (XEBUFE)	XERROR
Number of errors dropped (XEDROP)	XERROR

Outputs:

<u>Parameter</u>	<u>Destination</u>
Next word in error buffer to be filled (XEPTRF)	XERROR
Number of errors dropped (XEDROP)	XERROR

Processing:

X3ERR and X3ERRA perform the error processing function for the three subsystems. X3ERR collects the arguments from a calling FORTRAN program and passes them to X3ERRA which does the actual work. First, the processor ID is added into the first two bits of the error code word (1=MSC, 2=NAV, 3=RCV). Next, X3ERRA checks to see if there is enough room in the error buffer to insert the error message. If not, it

reports an error dropped. It then stuffs the error message into the buffer and returns control to the calling task.

Note: Error messages can be up to 11 words long. If an error message is passed to X3ERR that exceeds this limit, an error so noting is logged. If one is passed directly to X3ERRA, the message is truncated to 11 words.

5.4.15

Mnemonic: X3REL

Title: Relinquish sole access

Priority: None (Executive service routine)

Invoked by: M1DBPR, M2DBS1, M2DBS2, M2DBS3, M2STIN, N1MCNS,
N2SVEC, N2SVSL

Invokes: X3ERRA

Inputs: From data sets XIMSK, XWORK; from argument list SEMA4

<u>Parameter</u>	<u>Source</u>
X3REL workspace (XWORK)	XWORK
Mask to disable 5 ms interrupt (XIMSKP)	XIMSK
Mask to restore 20 ms interrupt (XIMSK7)	XIMSK
Address of exclusive access semaphore (SEMA4)	Calling subprogram

Outputs:

<u>Parameter</u>	<u>Destination</u>
Address of exclusive access semaphore (SEMA4)	Calling subprogram

Processing:

X3REL releases exclusive access to the data set passed as an argument. The address of the exclusive access semaphore is passed through the workspace in register 14. The semaphore is set to -1 and the 20 ms interrupt is re-enabled. An error is logged if the number of arguments passed to X3REL is not one.

5.4.16

Mnemonic: X3REG

Title: Request sole access utility

Priority: None (executive service routine)

Invoked by: M1DBPR, M2DBS1, M2DBS2, M2DBS3, M2STIN, N1MCNS,
N2SVEC, N2SVSL

Invokes: X3ERRA

Inputs: From data sets XIMSK, XWORK; from argument list SEMA4

<u>Parameter</u>	<u>Source</u>
X3REG workspace (XWORK)	XWORK
Mask to disable 20 ms interrupt in X3REG (XIMSKP)	XIMSK XIMSK
Mask to disable 20 ms interrupt in calling task (XIMSKS)	XIMSK
Exclusive access semaphore (SEMA4)	Calling task

Outputs:

<u>Parameter</u>	<u>Destination</u>
Exclusive access semaphore (SEMA4)	Calling task

Processing:

X3REG is designed to grant to the calling program exclusive access to a specified data set. A semaphore associated with the data set is set to 1 when access is granted. The 20 ms interrupt is disabled to ensure exclusivity. An error is logged if access cannot be granted within 1 millisecond of the request. Also, if the number of arguments passed to X3REG is not one, an error is logged.

5.4.17

Mnemonic: X3STOP

Title: Process stop server

Priority: None (executive service routine)

Invoked by: N1INIT, Receiver subsystem

Invokes: X1DISP

Inputs: From data sets XOPRC, XOLEV, XIMSK

<u>Parameter</u>	<u>Source</u>
Number of FTF's since start of process (OLEVFA)	XOLEV
Pointer to next process in priority (OPRCNP)	XOPRC
Pointer to previous process in priority (OPRCPP)	XOPRC
Non-error interrupt mask (XIMSKE)	XIMSK

Outputs:

<u>Parameter</u>	<u>Destination</u>
Workspace pointer of preempted process (OPCRWP)	XOPRC
Program counter of preempted process (OPRCPC)	XOPRC
Status register of preempted process (OPRCST)	XOPRC
FTF count at process end (OPRCEC)	XOPRC
Pointer to process now running (OLEVFN)	XOLEV
Pointer to next process in priority (OPRCNP)	XOPRC
Pointer to previous process in priority (OPRCPP)	XOPRC

Processing:

X3STOP is designed to deactivate the calling task while requesting the dispatcher to execute the next process in the priority. The workspace pointer, program counter and status register of the calling task are saved in the process block along with the current FTF count. The pointer to the next process to be run is set to point to the next process. The

calling process is removed from the process chain, and finally the dispatcher is called to transfer control to the next process.

5.4.18

Mnemonic: X3STOR

Title: Stopping service for 1 ms task

Priority: None (executive service routine)

Invoked by: R1AID, R1BSN, R1MRC, R1PCK, R1RNG, R1RRM

Invokes: None

Inputs: From data set XACRU

<u>Parameter</u>	<u>Source</u>
CRU address to clear interrupts (XACRUC)	XACRU
CRU address to enable/disable interrupts (XACRUE)	XACRU

Outputs: To data sets XOPRC, XOLEV, XINT

<u>Parameter</u>	<u>Destination</u>
Process workspace pointer (OPRCWP)	XOPRC
Process program counter (OPRCPC)	XOPRC
Bit map of enabled interrupts (XIENAB)	XINT
Pointer to previous process in priority (OPRCPP)	XOPRC
Pointer to first process in priority (OLEVPF)	XOLEV

Processing:

X3STOR deactivates the receiver's 1 ms interrupt task and returns control to the task interrupted. The 1 ms interrupt is cleared, disabled, and removed from the bit map of active interrupts. Also, the process chain is cleared since there is only the one task in it.

5.4.19

Mnemonic: X3TIME

Title: 20 ms count fetcher

Priority: None (executive service routine)

Invoked by: M1CMSC

Invokes: X3ERRA

Inputs: None

Outputs: From data set XCOUNT

<u>Parameter</u>	<u>Destination</u>
20 ms count, word 1 (XC0020)	R0 of calling task
20 ms count, word 2 (XC0022)	R1 of calling task

Processing:

X3TIME transfers the current time, expressed in 20 ms units, to registers zero and one of the calling task.

5.4.20

Mnemonic: X3TIMM

Title: Modulo(X) 20 ms time fetcher

Priority: None (executive service routines)

Invoked by: X1IT20, M1ADIS, M1CCID, M1CMSC, M1DBPR, N1CMNS

N1MITK, N1NFLT, N1SVPN, N1XFRM, N2MCNI

N2SVSL

Invokes: X3ERRA

Inputs: From data set XCOUNT; from argument list MOD

<u>Parameter</u>	<u>Source</u>
20 ms count - word 1 (XC0020)	XCOUNT
20 ms count - word 2 (XC0022)	XCOUNT
Modulus for returned value (MOD)	Calling task

Outputs: To calling task

<u>Parameter</u>	<u>Destination</u>
Modulo time	RO of calling task

Processing:

X3TIMM returns the 20 ms count in the modulo of the passed parameter. The result is placed into the task's RO. An error is logged if more than one argument is passed or if the argument is negative.

5.4.21

Mnemonic: X3WAIR

Title: Wait service for the 1 ms epoch task

Priority: None (executive service routine)

Invoked by: R1AID, R1BSN, R1MRC, R1PCK, R1RNG, R1RRM

Invokes: None

Inputs: None

Outputs: From data set XOPRC

<u>Parameter</u>	<u>Destination</u>
Process workspace pointer (DPRCWP)	XOPRC
Process program counter (DPRCPC)	XOPRC

Processing:

X3WAIR stores away the workspace pointer and program counter of the 1 ms interrupt handler to indicate the place to return to when the next interrupt occurs. Control is returned to the interrupted task upon completion of X3WAIR.

5.4.22

Mnemonic: X3WAIT

Title: Process wait service

Priority: None (executive service routine)

Invoked by: X1RPRT, X1COMM, X1COMN, X1RPRT, M1ADIS, M1CCIO,
M1CMSC, M1CRNC, M1DBPR, M1IIUO, M1PDBR, N1INIT, N1MCNS,
N1MITK, N1NFLT, N1SVPN, N1XFRM, B1SLIO, R1AID, R1CAL,
R1CC, R1DDT, R1FMT, R1MRC, R1MTM, R1NSE, R1PCK, R1PIN,
R1RNG, R1RRM, R1SCH, R1SET, R1SRC

Invokes: X1DISP

Inputs: From data set XOLEV

<u>Parameter</u>	<u>Source</u>
Number of FTF's since period readied (OLEVFA)	XOLEV

Outputs: To data set XOPRC

<u>Parameter</u>	<u>Destination</u>
Process workspace pointer (OPRCWP)	XOPRC
Process program counter (OPRCPC)	XOPRC
Process status register (OPRCST)	XOPRC

Processing:

X3WAIT saves the calling task's WP, PC, and ST, selects the next process in the priority, and summons the dispatcher to grant control to the next process.

5.4.23

Mnemonic: X9JOIN

Title: Allocation of data set XJOINT

Priority: None

Invoked by: None

Invokes: None

Inputs: None

Outputs: None

Processing:

X9JOIN allocates the global data set XJOINT, which contains the flags responsible for executive communications between the MSC and NAV subsystems. XJOINT contains the following data:

<u>Variable</u>	<u>Value</u>	<u>Significance</u>
XJFINI	-1 0	X1COMN complete X1COMM noted that X1COMN complete
XJUPDA word 1	-1 0	Update cmd sent Command received
word 2	-1 0	20-ms count update Restart
word 3,4	0	20-ms count
XJFPAU	-1 -2 1 2	FPAU last used by MSC FPAU last used by NAV FPAU now used by MSC FPAU now used by NAV
XJFPCN	not 0 0	Priority running in NAV cannot gain exclusive access to FPAU Priority running in NAV can gain exclusive access to FPAU
XJFPEX	0 -1	NAV priority has gotten exclusive FPAU access NAV priority does not have exclusive FPAU access

5.4.24

Mnemonic: X9MCSP

Title: Allocation of global data sets in MSC subsystem

Priority: None

Invoked by: None

Invokes: None

Inputs: None

Outputs: None

Processing:

X9MCSP allocates all global data sets residing on the local PMM's and DMM's in the master control subsystem as follows:

XGROUP -- Scheduling categories; contains number of priorities per category (20-ms and background) and pointer to the highest priority.

XSTART -- Power-on table; includes interrupts to be enabled, number of priorities, number of processes, number of tasks to activate, and which ones to activate.

XDATA -- Length of utility storage space for RCVF

XNUMBR -- Lengths of tables and subblocks

XHOME -- Subsystem identity; contains data elements identifying set as MSC subsystem.

XARR -- Allocation of ROM and RAM; contains ROM and RAM boundaries along with allocations for ROM checksums.

XZ -- Pointers to commands and reports; contains pointers to 20-ms update command, memory read/write command, memory contents report, error report, and first update

command received flag.

XTASK -- Task definition table; contains the following
for each task:

- word 1) Pointer to information block
- word 2) Priority level pointer
- word 3) Process pointer
- word 4) Max number of concurrent tasks
- word 5) Task entry point

XFTF -- Fundamental time frame table; contains for each
priority a block formed as following:

- word 1) FTF unit from which priority is formed
- word 2) Priority period in FTF's
- word 3) Pointer to variable storage space for
utilities in the priority
- word 4) Pointer to workspace for FPAU use

AXGPRU -- 20-ms update command received flag

AXMNAV -- NAV processor error flag

XUTIL -- Reentrant utility variable storage space

XPROC -- Process status table; contains the following
information for each process:

- word 1) Pointer to next process in priority
- word 2) Pointer to preceding process in priority
- word 3) Workspace pointer (WP)
- word 4) Program counter (PC)
- word 5) Status register (ST)
- word 6) FTF count when last given control
- word 7) Ending FTF count

XLEVEL -- Priority level table; contains following
information for each priority:

- word 1) Pointer to process now running
- word 2) Pointer to first process in priority
- word 3) 0
- word 4) FTF unit from which priority was formed
- word 5) Period in FTF's
- word 6) Number of FTF's since period was readied
- word 7) Number of FTF's priority took to run last
period
- word 8) Pointer to variable storage space for

utilities in priority
word 9) Number of utilities called - number of
utilities returned
word 10) Pointer to workspace for FPAU use
word 11) Flag indicating if the priority is using the
FPAU

XXWORK -- workspaces for FPAU use

5.4.25

Mnemonic: X9MPMB

Title: Allocation of global data sets on MPM board

Priority: None

Invoked by: None

Invokes: None

Inputs: None

Outputs: None

Processing:

X9MPMB allocates the 240-word RAM residing on the MPM board in each of the three processors. The following data sets are included:

XWORK -- Common workspaces used by executive programs

XCOUNT -- Interrupt counters; includes 20-ms and 1-ms counts along with 20-ms mod(16), 5-ms mod(4), and 10-ms mod(2).

XINT -- Interrupt status table; bit map of enabled interrupts, processor interrupt mask.

XERROR -- Error recording buffer; used by X3ERRA to log program errors

XPSTER -- Processor self-test buffer; contains information on four possible processor self-test errors.

XACRU -- CRU addresses; CRU bases to clear and disable/enable interrupts.

XIMSK -- Interrupt masks used by the executive.

5.4.26

Mnemonic: X9NAVP

Title: Allocation of global data sets in NAV subsystem

Priority: None

Invoked by: None

Invokes: None

Inputs: None

Outputs: None

Processing:

X9NAVP allocates all global data sets residing on the local PMM's and DMM's in the navigation subsystem as follows:

XGROUP -- Scheduling categories; contains number of priorities per category (20-ms and background) and pointer to the highest priority.

XSTART -- Power-on table; includes interrupts to be enabled, number of priorities, number of processes, number of tasks to activate, and which ones to activate.

XDATA -- Length of utility storage space for RCVP

XNUMBR -- Lengths of tables and subblocks

XHOME -- Subsystem identity; contains data elements identifying set as NAV subsystem.

XARR -- Allocation of ROM and RAM; contains ROM and RAM boundaries along with allocations for ROM checksums.

XZ -- Pointers to commands and reports; contains pointers to 20-ms update command, memory read/write command, memory contents report, error report, and first update

command received flag

XTASK -- Task definition table; contains the following
for each task:

- word 1) Pointer to information block
- word 2) Priority level pointer
- word 3) Process pointer
- word 4) Max number of concurrent tasks
- word 5) Task entry point

XFTF -- Fundamental time frame table; contains for each
priority a block formed as following:

- word 1) FTF unit from which priority is formed
- word 2) Priority period in FTF's
- word 3) Pointer to variable storage space for
utilities in the priority
- word 4) Pointer to workspace for FPAU use
=0 for NAV priorities which can have
exclusive FPAU use.

AXGPRU -- 20-ms update command received flag

XUTIL -- Reentrant utility variable storage space

XPROC -- Process status table; contains the following
information for each process:

- word 1) Pointer to next process in priority
- word 2) Pointer to preceding process in priority
- word 3) Workspace pointer (WP)
- word 4) Program counter (PC)
- word 5) Status register (ST)
- word 6) FTF count when last given control
- word 7) Ending FTF count

XLEVEL -- Priority level table; contains following
information for each priority:

- word 1) Pointer to process now running
- word 2) Pointer to first process in priority
- word 3) 0
- word 4) FTF unit from which priority was formed
- word 5) Period in FTF's
- word 6) Number of FTF's since period was readied
- word 7) Number of FTF's priority took to run last
period
- word 8) Pointer to variable storage space for

utilities in priority
word 9) Number of utilities called - number of
utilities returned
word 10) Pointer to workspace for FPAU use
word 11) Flag indicating if the priority is using the
FPAU

XXWORK -- workspaces for FPAU use

5. 4. 27

• Mnemonic: X9RCVP

Title: Allocation of global data sets in RCV subsystem

Priority: None

Invoked by: None

Invokes: None

Inputs: None

Outputs: None

Processing:

X9RCVP allocates all global data sets residing on the local PMM's and DMM's in the receiver subsystem as follows:

XGROUP -- Scheduling categories; contains number of priorities per category (1-ms, 5-ms, 20-ms and background) and pointer to highest priority.

XSTART -- Power-on table; includes interrupts to be enabled, number of priorities, number of processes, number of tasks to activate, and which ones to activate.

XDATA -- Length of utility storage space for RCVP

XNUMBR -- Lengths of tables and subblocks

XHOME -- Subsystem identity; contains data elements identifying set as RCV subsystem.

XARR -- Allocation of ROM and RAM; contains ROM and RAM boundaries along with allocations for ROM checksums.

XZ -- Pointers to commands and reports; contains pointers to 20-ms update command, memory read/write command, memory contents report, error report, and first update command received.

XTASK -- Task definition table; contains the following
for each task:

- word 1) Pointer to information block
- word 2) Priority level pointer
- word 3) Process pointer
- word 4) Max number of concurrent tasks
- word 5) Task entry point

XFTF -- Fundamental time frame table; contains for each
priority a block formed as following:

- word 1) FTF unit from which priority is formed
- word 2) Priority period in FTF's
- word 3) Pointer to variable storage space for
utilities in the priority
- word 4) Pointer to workspace for FPAU use

XQPRUP -- 20-ms update command received flag

XPROC -- Process status table; contains the following
information for each process:

- word 1) Pointer to next process in priority
- word 2) Pointer to preceding process in priority
- word 3) Workspace pointer (WP)
- word 4) Program counter (PC)
- word 5) Status register (ST)
- word 6) FTF count when last given control
- word 7) Ending FTF count

XLEVEL -- Priority level table; contains following
information for each priority:

- word 1) Pointer to process now running
- word 2) Pointer to first process in priority
- word 3) 0
- word 4) FTF unit from which priority was formed
- word 5) Period in FTF's
- word 6) Number of FTF's since period was readied
- word 7) Number of FTF's priority took to run last
period
- word 8) Pointer to variable storage space for
utilities in priority
- word 9) Number of utilities called - number of
utilities returned
- word 10) Pointer to workspace for FPAU use
- word 11) Flag indicating if the priority is using the
FPAU

5.4.28

Mnemonic: X9TRAP

Title: Allocation and initialization of interrupt transfer vectors

Priority: None

Invoked by: None

Invokes: None

Inputs: None

Outputs: None

Processing:

X9TRAP allocates the interrupt transfer vectors (WP, PC) for the seven levels of interrupts in the processors as follows:

- Level 1 -- Memory parity error
- Level 2 -- Unwanted
- Level 3 -- I-bus time-out
- Level 4 -- FPAU increment or 1-ms epoch
- Level 5 -- FPAU status or 5-ms time mark
- Level 6 -- 20-ms time mark
- Level 7 -- Unwanted

6.0 HARDWARE MODULE DESCRIPTIONS

6.1 RECEIVER LRU MODULES

6.1.1 WIDEBAND MODULE

6.1.1.1 GENERAL DESCRIPTION

The primary functions performed by the Wideband Module (WBM) are those of RF signal amplifying, mixing, pre-filtering, and automatic gain control. The incoming RF is mixed down to IF for output to the NBM's. A pre-filtering function allows the processor to select between an L1 or L2 path within the WBM which limits interference from sources outside the stopband. The module is equipped with automatic gain control which is responsive to both signal power and noise power.

6.1.1.2 FUNCTIONAL DESCRIPTION (refer to Figure 6.1.1-1)

As shown in Figure 6.1.1-1 the WBM is a series of mixers, amplifiers and filters which process the incoming signal. The module accepts L1 and L2 ($154-F_0$ and $120-F_0$ respectively) from an external source. The signal is routed to an amplifier of 15 dB gain. The input amplifier forwards its signal to two filters, one of which is for L1 and the other for L2. Each filter has a bandpass of approximately 25 MHz. The output of each filter is fed to a SPDT RF switch which allows the processor to select either an L1 or L2 path.

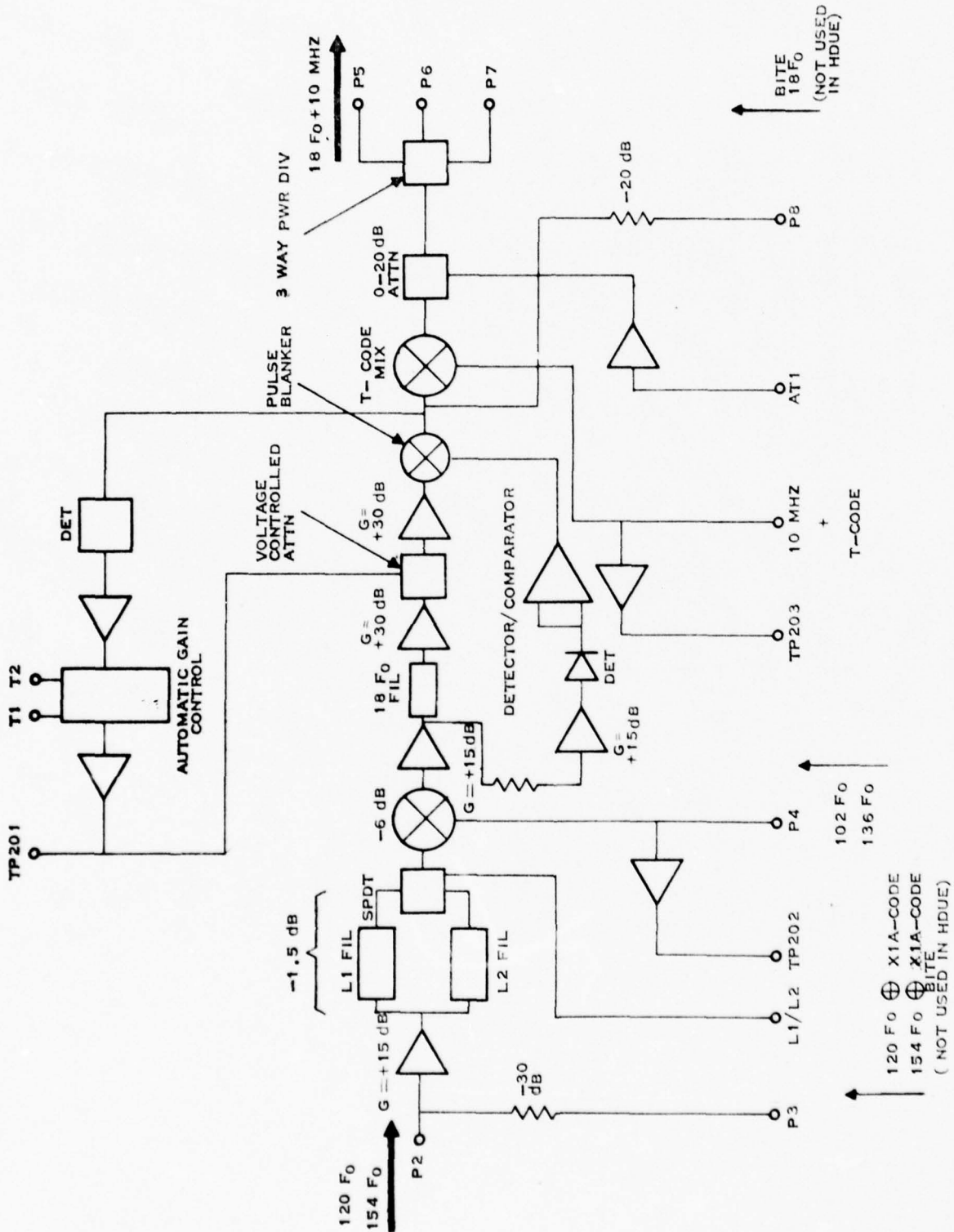


Figure 6.1.1-1. Functional Block Diagram Wideband Module

The RF switch output is forwarded to the first down-conversion mixer which accepts L.O. input of either $136-F_0$ or $102-F_0$. This L.O. frequency is determined by the desired mode of operation (L1 or L2) and is controlled external to the module. The mixer produces an IF signal of $18-F_0$ which is routed to a second 15 dB gain amplifier. The output of this amplifier is fed into an $18-F_0$ Surface Wave Filter (SWF) which has a bandwidth of 15 MHz. The output from the SWF is forwarded to a 30-dB gain amplifier prior to input to a Voltage Controlled Attenuator. This attenuator may be varied between 1.5 and 40 dB as a function of its input control voltage (AGC Signal). The output of the attenuator is inputted to another 30-dB gain amplifier which feeds a Pulse Blanker mixer. This mixer performs the function of attenuating the $18-F_0$ signal if it exceeds an abnormally high power threshold as detected by the Detector/Comparator section which derives its input from a point prior to the SWF.

The output of the Pulse Blanker mixer is forwarded to the Automatic Gain Control (AGC) section and a T-code mixer. In the AGC section the output of the blanker mixer is detected and is routed through a time constant control circuit which allows the system processor to invoke either a 2-second or a 1-msec time constant on the detected signal. The AGC signal is then forwarded to the Voltage Controlled Attenuator. The T-Code Mixer mixes the $18-F_0$ IF with a 10-MHz signal which has been mixed with T-Code. T-Code is a

pseudo-random code created by sampling the replica P-code in the system at a 10-KHz rate.

The output of the T-Code Mixer is an $18-F_0 + 10\text{-MHz}$ IF signal which is routed to a processor-controlled 20-dB attenuator. The attenuation options are either 0 dB or 20 dB, with the 20-dB option utilized with the BITE signal which is normally 20 dB above the nominal SV signal at this point in the system. The output of the attenuator is forwarded to a 3-Way Power Splitter which provides $18-F_0 + 10\text{-MHz}$ to the NBM's.

6.1.1.3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE
RF	SV signal	L1 and L2 carriers modulated with P-code, C/A-code, and data (-95 dBm nominal)	external preamp
RF	L1,L2 BITE	Built-in test signal (-50 dBm)	BITM not used in HDUE
RF	BITE 18 Fo	Built-in test signal (-30 dBm)	BITM not used in HDUE
RF	L01	136 Fo or 102 Fo (17 dBm)	FSM
RF	10MHz + T-code	mixer signal	FSM
PWR	5.2 VDC	40 ma	pwr sup
PWR	5.0 VDC	70 ma	pwr sup
PWR	-5.0 VDC	10 ma	pwr sup
PWR	12.0 VDC	100 ma	pwr sup
PWR	-12.0 VDC	60 ma	pwr sup
Logic	CLL1L2	TTL (to select either the L1 or L2)	DM
Logic	CLATT	TTL (to invoke 20-dB attenuator)	DM
Logic	CLTAU1	TTL (to select the 2-sec time constant for AGC)	DM
Logic	CLTAU2	TTL (to select the 1-msec time constant for AGC)	DM

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTIN.
RF	VHF Signal	18 Fo, 10 MHz, P (or C/A) code, data, + T-code (-62 to -25 dBm))	NBM's

6.1.2 NARROWBAND MODULE

6.1.2.1 GENERAL DESCRIPTION

The Narrow-Band Module (NBM) contains the code correlation circuitry followed by a down converter to baseband to obtain the loop tracking error signals data demodulation.

6.1.2.2 FUNCTIONAL DESCRIPTION (refer to Figure 6.1.2-1)

I. Code correlation and down conversion

The NBM module receives a signal of $18F_o + 10$ MHz (F_o is nominally 10.23 MHz) modulated with P code (and/or C/A code), SV data, and T code from the WBM. The 10 MHz is used to offset the IF from harmonics of F_o and the T code is used to aid in CW jamming immunity. The signal is mixed in the correlator with the replica P and T codes (or replica C/A and T codes), amplified, then fed through a narrow band surface wave filter to give the carrier ($18 F_o + 10$ MHz) with data. This signal is mixed with $17 F_o$ and fed through a crystal filter to give $F_o + 10$ MHz with data. The signal is then fed through an AGC controlled amplifier, split and fed to two separate channels: 1) an error channel and 2) a data channel.

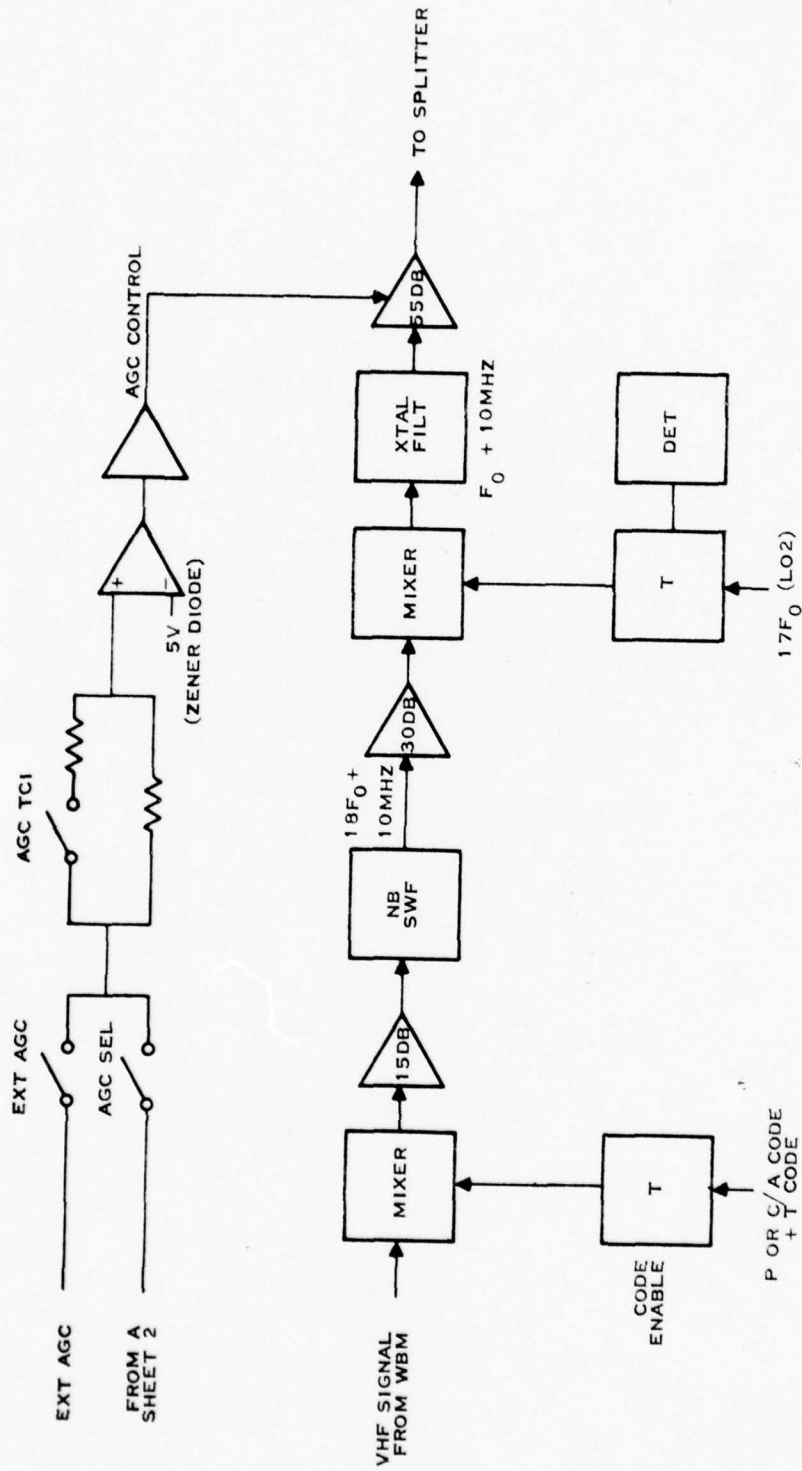


Figure 6.1.2-1. Functional Block Diagram of the NBM (Sheet 1)

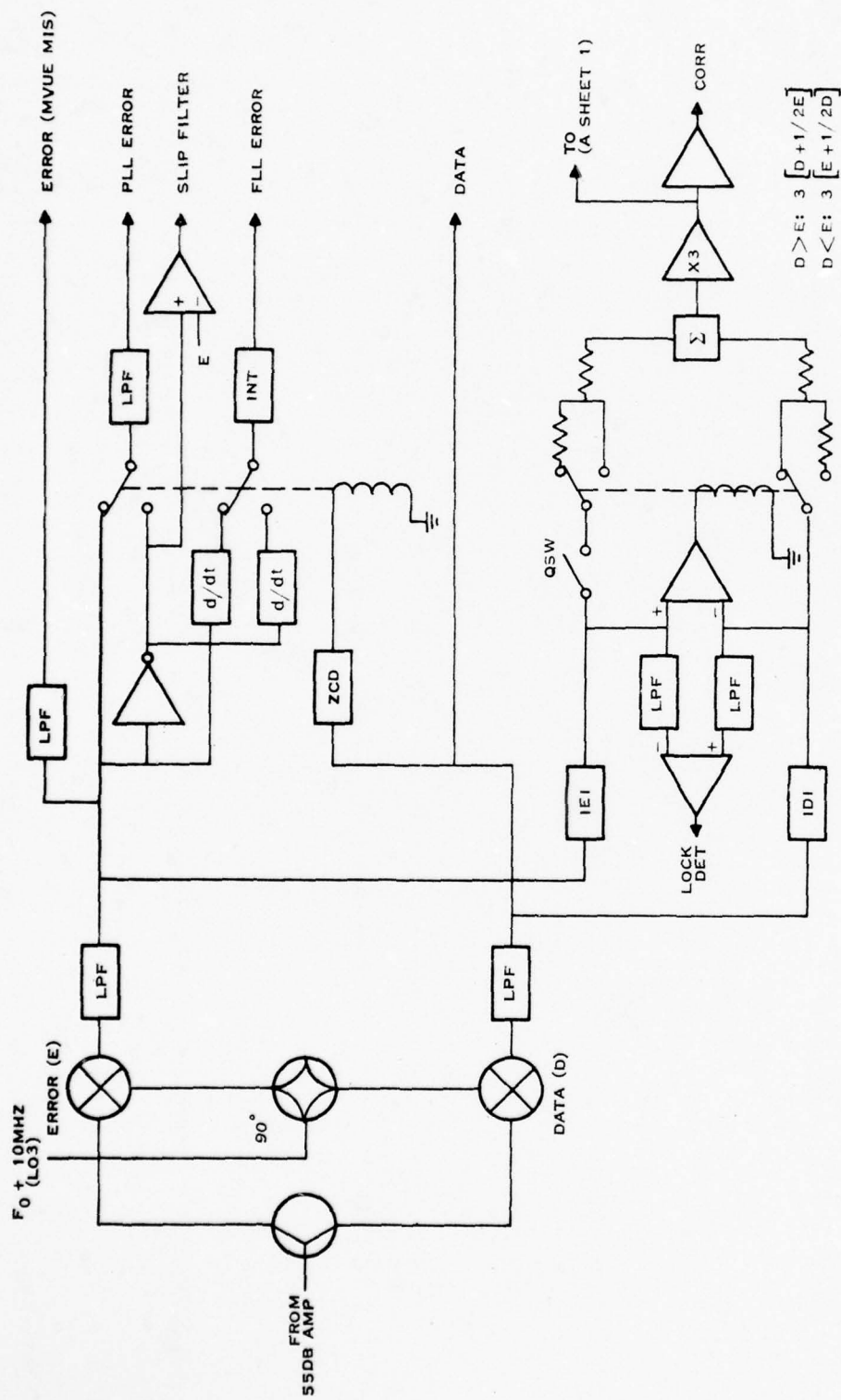


Figure 6.1.2-1. Functional Block Diagram of the NBM (Sheet 2)

$F_0 + 10 \text{ MHz}$ from the frequency synthesizer module is fed through a quadrature hybrid to mixers in both of these channels, with the data channel having a 90 degree phase shift with respect to the error channel. The signals from both mixers are fed through Low Pass Filters (LPF) leaving only data and error signals. Data is removed from the error signal by selecting an inverted or noninverted signal with a zero crossing detector fed from the data signal. The filtered output is the Phase Lock Loop (PLL) error voltage.

The above method is used to remove data from differentiated error signals. The resultant signal is passed through an integrating circuit for the Frequency Lock Loop (FLL) error voltage.

II. Determination of existence and quality of code lock

Both error and data signals are passed through absolute value detectors and compared. When the absolute value of the data signal is 0.3V greater than that of the error, lock is declared on the DOLOCK output.

The degree of correlation is determined by feeding the Error (E) and Data (D) signals through a circuit that is an approximation of an envelope detector. The absolute values of the E and D signals are fed through a switch, a resistor divider network and a summing

circuit to an amplifier set up for a gain of 3X. The switch is controlled by a comparator monitoring the absolute values causing an output of $3(|D| + 1/2 |E|)$ for $|D| > |E|$ or $3(|E| + 1/2 |D|)$ for $|E| > |D|$. This value (correlation voltage) is sent externally to the output module and internally to an amplifier referenced to 5 VDC to develop AGC.

Because most of the useful information is in the data channel when the receiver is in solid phase lock, a Q-switch was implemented to remove the error channel input to the correlation voltage signal to noise ratio, but subsequently discarded because of the inherent complexity under marginal phase lock conditions.

To enable the processor to perform a quality check of the carrier track precision the error signal is applied to a comparator whose output (HISLIP) goes to the output module for an ultimate cycle slip rate accumulation in the processor.

6.1.2.3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE
RF	VHF Signal	18 Fo, 10 MHz, P (or C/A) code, data, + T code (-62 to -25 dBm)	WBM
RF	L02	173.91 MHz nominal (9dBm)	FSM
RF	L03	Fo and 10 MHz (9dBm)	FSM
Power	5 VDC RF	25 ma	pwr sup
Power	-12 VDC	26 ma	pwr sup
Power	12 VDC	36 ma	pwr sup
Power	5V logic	35 ma	pwr sup
Analog	EXT AGC	External AGC (not used)	OM
Code	Code	TTL (P (or C/A) code and T code)	CGM
Logic	CLAGCTC	TTL (AGC time constant control bit)	OM
Logic	CLAGSEL1	TTL (internal/ external AGC control bit)	OM
Logic	CLGSW	TTL (Q switch control bit)	OM
Logic	Code Enable	TTL (replica correlation code enable bit)	BITM

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTINATION
Analog	EPLL	Phase Lock Loop Error voltage	FSM
Analog	EFLL	Frequency Lock Loop Error voltage	FSM
Analog	CHxCOR	Correlation voltage	OM
Analog	Data	Data signal	OM
Logic	HISLIPA	TTL (slip detection bit)	OM
Logic	LOSLIPA	not connected in module	OM
Logic	DOLOCK	TTL (lock detection bit)	OM

6.1.3 FREQUENCY SYNTHESIZER MODULE

6.1.3.1 GENERAL DESCRIPTION

The Frequency Synthesizer Module (FSM) comprises the circuitry necessary to provide local oscillator signals to various mixers in the GPS receiver. These local oscillator signals are phase locked to the received signals.

6.1.3.2 FUNCTIONAL DESCRIPTION (refer to Figure 6.1.3-1)

I. Voltage Controlled Oscillator (VCO) and its internal control loop.

The VCO is a push-pull oscillator with a tank circuit, whose frequency is controlled both by the voltage applied and by the L1/L2 digital input from the digital control circuitry. The L1/L2 digital input causes a section of the tank circuit to be bypassed in L1 to produce $136 F_o$. In L2 the full tank circuit is in for an output of $102 F_o$. One half of the output frequency ($68 F_o$ or $51 F_o$) is tapped off of one side of the push-pull oscillator and fed through a divide-by-4 (or 3) circuit to produce $17 F_o$. For L1 its a divide-by-4 and L2 a divide-by-3.

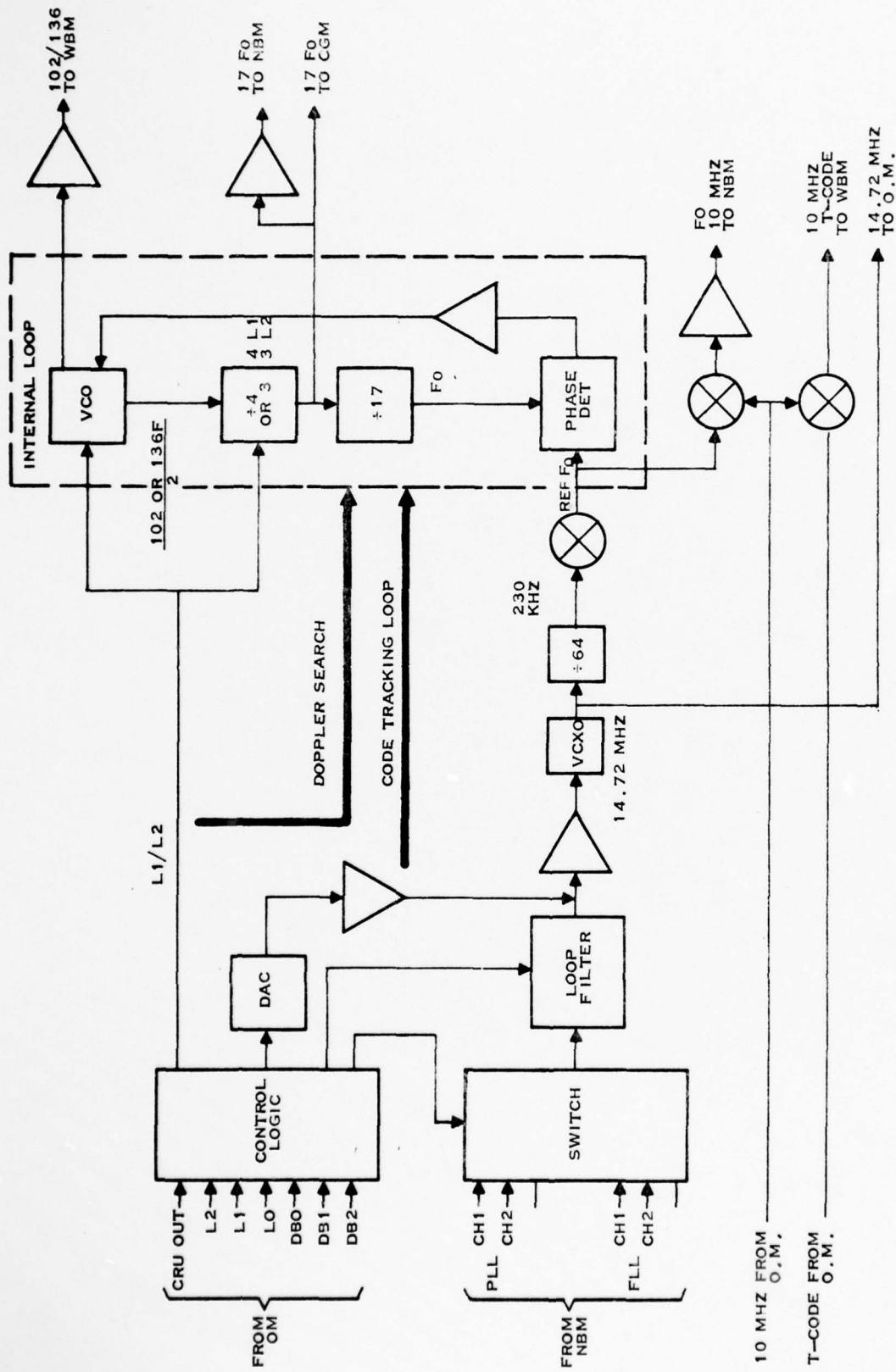


Figure 6.1.3-1. Functional Block Diagram Frequency Synthesizer Module

The output of this divider ($17 F_o$) is fed through a divide-by-17 circuit to a phase detector. The phase detector is referenced to a F_o with a frequency controlled by the SV signal to allow for doppler shift and for reference oscillator drift, or by a logic-generated calibration/search signal. The voltage developed by the detector in comparing the phase difference is fed through an active filter to control the VCO frequency.

The outputs of the internal loop are the $102/136 F_o$ and the $17 F_o$. The $102/136 F_o$ (the first L.O.) is sent to the wide band module to be mixed with the RF. The $17 F_o$ (the second L.O.) is sent to the narrow band module to be mixed with the IF for the second conversion and to the code generator module to be mixed with the P or C/A code.

II. Frequency synthesizer portions of the carrier tracking loop.

The Phase Lock Loop (PLL), Frequency Lock Loop (FLL) Narrowband Module NBM channel 1 or 2 error voltage inputs are applied to a switch, the positioning of which is determined by the control logic. The error selected is applied to a loop filter with a response determined by the control logic. The output of the loop filter controls the frequency of the Voltage Controlled Crystal Oscillator (VCXO).

The output of the VCXO (nominally 14.72 MHz) is divided by 64 giving a nominal 230 KHz. This mixed with 10 MHz from the Signal Distribution Module (SDM) (developed by the reference oscillator) produces the reference F_0 (nominally 10.23 MHz) for the aforementioned internal loop.

The external outputs of the section just described are 14.72 MHz and F_0 . The 14.72 MHz is sent to the output module. F_0 is mixed with the 10 MHz and sent to the last mixer stage in the NBM's

III. Doppler search

Doppler search is accomplished by altering the logic inputs to the control logic, in turn altering the digital input to the Digital-to-Analog Converter (DAC). The output of the DAC, representing the digital input, is fed through two amplifiers to the VCXO in order to control the frequency. The frequency is systematically altered to cover all practical doppler frequencies until lock is established and the carrier tracking loop takes control.

6.1.3.3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE
RF	10 MHz	TTL (reference signal)	SDM
Power	12 VDC	10 ma	pwr sup.
Power	-15 VDC	55 ma	pwr sup.
Power	-5 VDC	6 ma	pwr sup.
Power	5.2 V RF	570 ma	pwr sup.
Power	5 V Logic	102 ma	pwr sup.
Power	12 VDC Standby	40 ma (nominal)	pwr sup.
Logic	CPDB0	TTL (decode address line)	OM
Logic	CPDB1	TTL (decode address line)	OM
Logic	CPDB2	TTL (decode address line)	OM
Logic	CPDACLOAD	TTL (DAC load pulse)	OM
Logic	500CRUOUT	TTL (CRU data)	OM
Logic	500L0	TTL (L field decode bit)	OM
Logic	500L1	TTL (L field decode bit)	OM
Logic	500L2	TTL (L field decode bit)	OM
Code	T code	TTL (5 KHz code for CW jamming immunity aid)	CGM
Analog	EPLL A	Phase Lock Loop Error voltage	NBM A
Analog	EPLL B	Phase Lock Loop Error voltage	NBM B
Analog	EFLL A	Freq. Lock Loop Error voltage	NBM A
Analog	EFLL B	Freq. Lock Loop Error voltage	NBM B

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTIN.
RF	10 MHz + T code	TTL	WBM
RF	14.72 MHz	TTL	OM
RF	L02	2nd local oscillator--17 Fo (6 dBm)	NBM A
RF	L02	2nd local oscillator--17 Fo (6 dBm)	NBM B
RF	L03	3rd local oscillator--Fo + 10 MHz (9 dBm)	NBM A
RF	L03	3rd local oscillator--Fo + 10 MHz (9 dBm)	NBM B
RF	L01	1st local oscillator-- 102/136 Fo (7 dBm)	WBM
RF	17Fo	6 dBm	

6.1.4 CODE GENERATOR MODULE

6.1.4.1 GENERAL DESCRIPTION

The Code Generator Module (CGM) is capable of generating a P-code and C/A-code replica under the control of the system processor. Codes are generated in various versions (early, late and prompt) and distributed to the Narrowband Module(s) to perform the correlation function. The CGM is primarily a CRU device which interfaces with the system processor via the RMBL address structure. All code control is exercised via the CRU interface and all responses by the various functions resident in the module are read by the processor via this interface.

6.1.4.2 FUNCTIONAL DESCRIPTION

The CGM can best be described by the functions that are performed while referring to its functional block diagram in Figure 6.1.4-1. The functions include: 1) code selection, 2) code initialization, 3) code starting, 4) code output/control, and 5) range data reading. A key item included in the CGM implementation is the Transfer Register (TR) utilized throughout various functions. This device is a 8-bit holding register (CRU latches) which is preset via processor control to route data to various sections of CGM hardware. This is accomplished by executing load commands associated with destination functions.

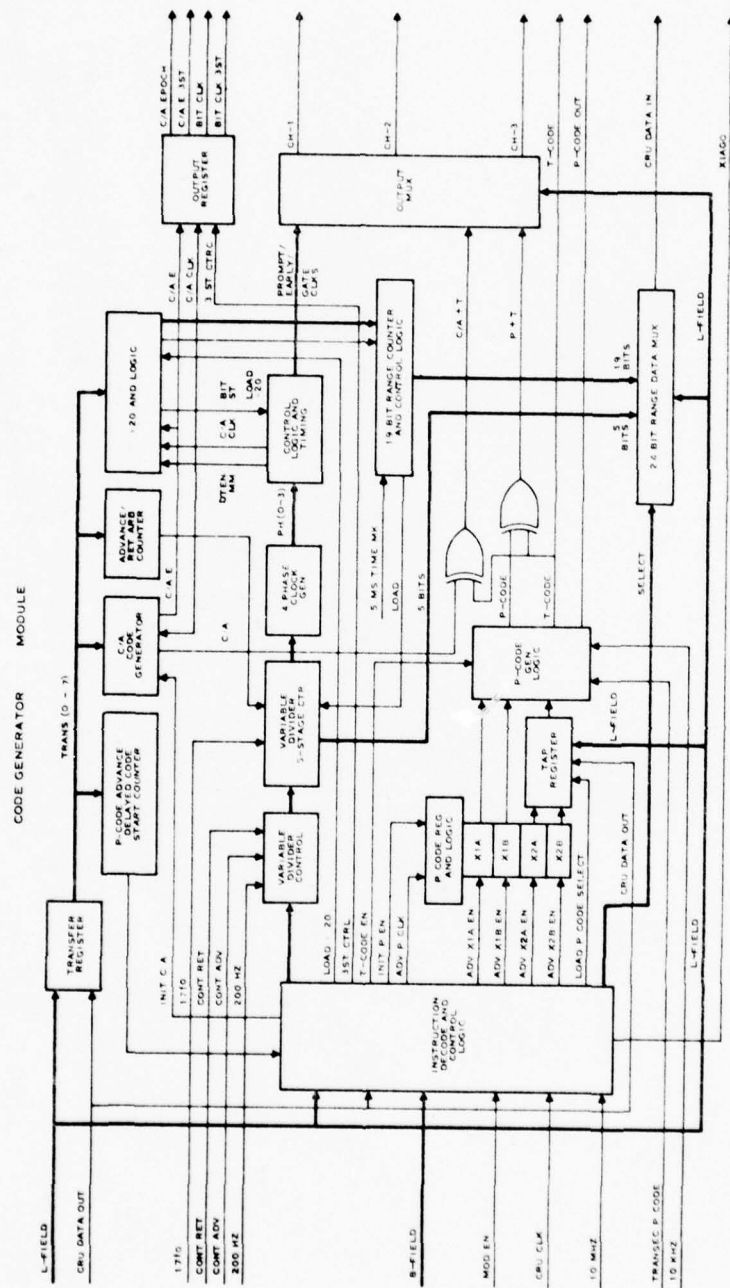


Figure 6.1.4-1. Functional Block Diagram

I. Code selection

Code selection of C/A-code is accomplished by loading the appropriate bit pattern into the TR. The processor then executes a unique C/A-code load command via the CRU structure to load the bit pattern into the C/A-code generator. P-code selection is accomplished by again performing a CRU operation in the CGM's CRU space. The six bits are serially loaded into Tap Registers to define the phase relationship between the X1 and X2 registers of the P-code generator.

II. Code initialization

Code initialization is also a CRU operation. For C/A-code the processor executes a singular initiate command which forces the chip to the C/A EPOCH state. The generator remains in this state until other commands are issued.

P-code is initialized by first initializing the generator to the state associated with the beginning of the week via CRU operations. The four registers that form the P-code generator must each be advanced to appropriate states associated with the desired P-code state. The advance operation is accomplished by sequentially selecting one of the four registers to be advanced. The TR is then utilized to define the number of clocks to be advanced by presetting a P-Code Advance/Delayed Code Start Counter. The processor then enables the advance function which runs to completion automatically in 0.8 msec maximum. The next register is

then selected to repeat the operation.

III. Code starting

Code starting for the C/A-code is accomplished by executing a single CRU instruction. P-code can be started with two methods. Both methods require that C/A-code be initialized prior to P-code. The first method requires merely executing a CRU instruction to start the code. The second method involves a programmed delay start. The programmed delay is executed by loading delay information into the P-code Advance/Delayed Code Start Counter via the TR. The processor then arms the P-code generator via other CRU commands to start after the delay period has expired.

IV. Code output/control

Code Output/Control deals with code alignment control and output buffering and selection. Code alignment control is intended to provide control over the code phase with respect to the received code. The control exercised is to either advance or retard the replica code phase in increments of $1/17$ of a P-code chip. The task is accomplished by presetting the ADVANCE/RETARD Counter with the number of increments desired via the TR. The processor then executes the appropriate command to arm the Variable Divide 5-Stage Counter to the desired modulus (16 for advance or 18 for retard). The loading of the MSB's into the ADVANCE/RETARD Counter also starts the advance or retard

operation.

The C/A-code circuitry generates an output pulse (C/A epoch) at the end of each cycle. These pulses drive the Divide-By-20 Counter which generates a pulse output each time the counter modulus. The processor must align the occurrence of the resulting 50-Hz pulse with the data bit changes. It accomplishes this alignment by changing the contents of the Divide-by-20 Counter, thus changing the time at which the next modulo occurs. The 5-bit counter value is output to the TR and then to a CRU output to load the Divide-By-20 Counter.

Code output selection is accomplished via the Output Multiplexer section which allows the processor to select various code versions with CRU controls. The code generator creates three versions of C/A-code and P-code. These code versions are early, late, and prompt. The early and late C/A versions are 38/170 of a C/A-code chip before and after prompt while the P versions are 4/17 of a P-code chip before and after prompt. The processor may select one of these six versions to be outputted to each of three outputs from the Output Multiplexer section. The processor may also enable a "code flop" function which involves arming the Output Multiplexer with two sets of select commands to control output code. The processor may then enable the code flop command which forces output control to be alternated on 20-msec time marks between the two sets of commands.

The CGM provides a P-code output that can be enabled to

a "scrambler" for use in a secure environment. The scrambled P-code, called TRANSEC, can be enabled to the output selection logic by performing the appropriate CRU operation.

During built-in-testing, the BITM will modulate an X1-code onto the test signals. The processor commands the CGM to output an X1-code instead of the full P-code.

To overcome mixer leakage problems in the presence of CW jamming, the processor may enable the T-code by performing a CRU output. The T-code will modulate the code stream (and a 10-MHz offset signal) in a pseudo random fashion.

V. Range data reading

Range data reading is accomplished by performing a CRU operation to obtain the contents of two counters resident in the module. The counters include the Variable Divider 5-Stage Counter and the 19-bit Range Counter. The counters count the number of 17-Fo cycles between the data clock and a 20-msec timing pulse provided by the system master oscillator. This information is the pseudo-range.

6.1.4.3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY TYPE	SIGNAL NAME	CHARACTERISTIC	SOURCE
RF	17Fo	ECL	FSM
Power	5.0 VDC	1.2a	pwr sup
Clock	20ms Time Mark	TTL (Fundamental Time Frame (FTF))	CM
Clock	10KHZ	TTL (100-ms Time Marks)	CM
Logic	B0-B2	TTL (Bit field of processor address bus)	MPM
Logic	L0-L2	TTL (Latch field of processor address bus)	MPM
Logic	MODEN	TTL (Module Enable)	DM
Logic	CRU CLK	TTL (CRU Clock)	MPM
Logic	CRU DATA OUT	TTL (CRU serial Data Out)	MPM
Clock	10 MHz	TTL (10-MHz clock)	CM
Logic	T/S Pcode	Transformed Secure P-Code	not used
Logic	ADVANCE	TTL (P-code Advance command)	not used
Logic	RETARD	TTL (P-code Retard command)	not used
Logic	Ext DHO Syn	TTL (External Direct Handover Synchronization pulse)	EIOM (MVUE)

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTIN.
Clock	T-code	TTL (10-KHz PRN sq wave)	FSM
Clock	Data Clock	TTL (50-Hz pulse stream, in sync with incoming signal)	OM
Data	CRUIN	TTL (CRU serial data)	MPM
Logic	C/A EPOCH	TTL (C/A-CODE epoch)	OM, FSM
Logic	X1AGD	TTL (X1A register epoch)	BITM
Data	Code 1,2,3	TTL (output multiplexer code output)	NBM

6.1.5 OUTPUT MODULE

6.1.5.1 GENERAL DESCRIPTION

The receiver Output Module (OM) contains the circuitry that modifies the signals between the processor and the receiver RF modules, allowing transfer of information in both directions.

6.1.5.2 FUNCTIONAL DESCRIPTION (refer to Figures 6.1.5-1 and 6.1.5-2)

The OM circuitry may be divided into four main signal processing functions. These functions are: (1) processor control interface (2) range rate count, (3) data conversion, and (4) correlation voltage conversion.

I. Processor control interface

The OM deciphers information sent from the processor through the CRIM and sends the results as control signals to respective receiver modules.

RXSEL determines the receiver channel, M, B and L fields determines the address, CRUOUT contains the data, and CRUCLK contains the processing clock.

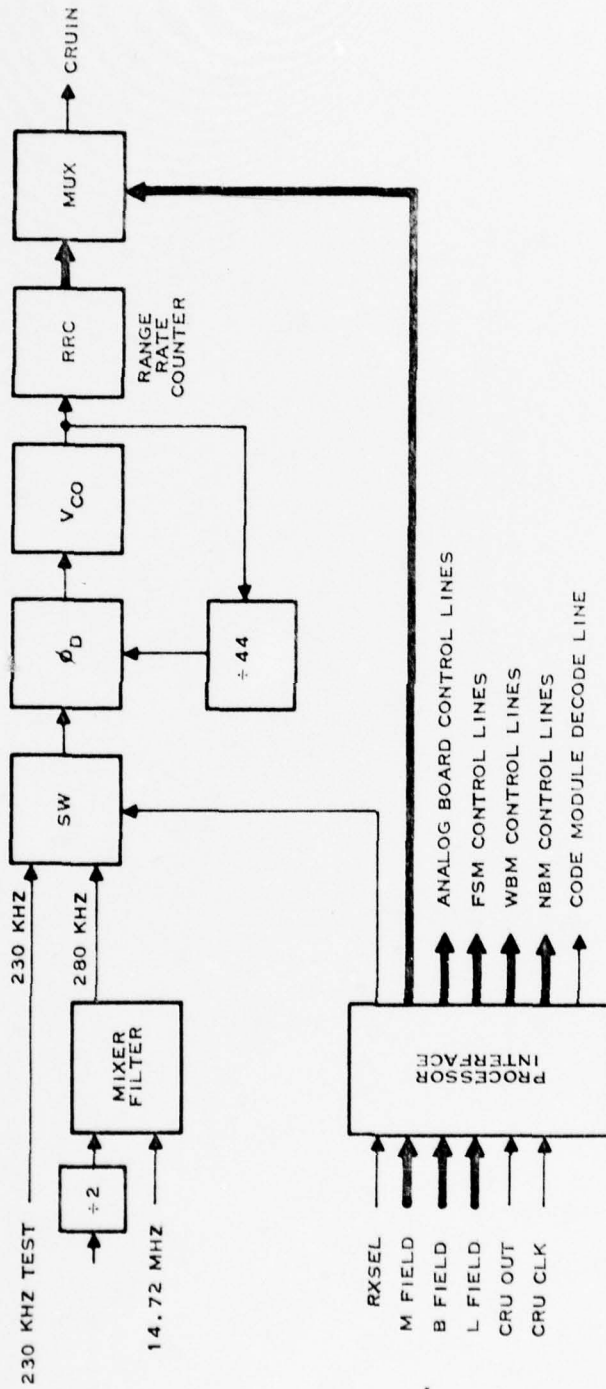


Figure 6.1.5-1. Functional Block Diagram Output Module Digital Board

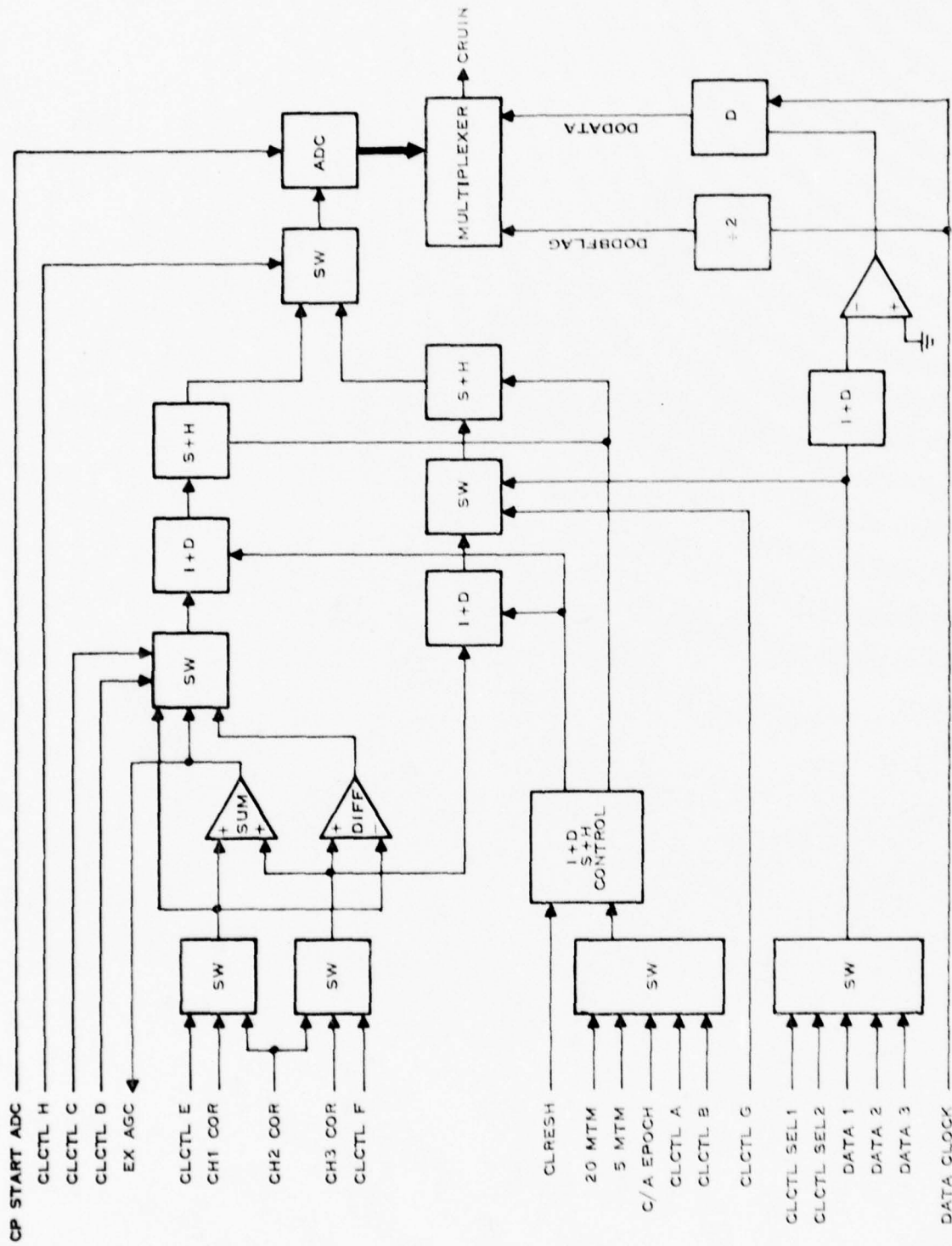


Figure 6.1.5-2. Function Block Diagram, Output Module, Analog Board

II. Range Rate Count (RRC)

The RRC contains a mixer, phase lock loop, and counter. 10 MHz from the reference oscillator is divided by 2 and fed to a mixer along with 14.72 MHz from the frequency synthesizer module. This 14.72 MHz has a frequency shift on it caused by doppler and clock bias. A low pass filter selects the difference between the second harmonic of 5 MHz and the 14.72 MHz to give a nominal 280 KHz. This 280 KHz is fed through a switch to the signal input of a phase detector.

The VCO output (a nominal 12.32 MHz) is fed through a divide by 44 circuit to the compare input of the phase detector. The phase detector output controls the frequency of the VCO.

The VCO output is fed to a series of ripple through counters which are enabled and preset at predetermined intervals. The output of these counters is applied to a series of decoders whose outputs, representative of the range rate, go to CRUIN.

For a counter test, 230 KHz from the built in test module is switched into the input of the phase detector, thus controlling the VCO at a nominal 10.12 MHz.

III. Data conversion

Data enters the OM from the NBMs and progresses through a switch to an Integrate-and-Dump (I+D)

circuit, which is controlled by the Data Clock. The I+D output is applied to a comparator which converts the signal to a TTL level and applies it to the D input of a flip-flop. This data is clocked through the flip-flop with the same DATA CLOCK to a multiplexer, which sends this data to CRUIN when selected.

For a histogram of the data the analog signal is also passed through a switch to a Sample and Hold (S+H) circuit. Then it is fed through another switch to an Analog-to-Digital Converter (ADC). The resulting digital output, representing a histogram of the SV data, is fed to the multiplexer where it is sent out on CRUIN.

IV. Correlation voltage conversion

The correlation voltages from the NBMs is fed through switches to the input of a sum circuit, difference circuit, and a switch. The processor selects one of these signals and applies it to an I+D circuit. The selected integrated correlation signal is applied to a S+H circuit. The signal is clocked through this S+H circuit with a selected timing signal and applied through a switch to the ADC. The digital output of the converter is applied to the multiplexer and is clocked out on the CRUIN line by the logic signals from the processor.

6. 1. 5. 3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE
Power	5VDC	160 ma	pwr sup
Power	-5VDC	27 ma	pwr sup
Power	12VDC	25 ma	pwr sup
Power	-12VDC	2 ma	pwr sup
RF	14.72MHz	TTL	FSM
RF	10MHz	TTL	CM
Timing	C/A epoch	TTL (1 msec epoch)	CGM
Timing	Data Clock	TTL (20 msec epoch)	CGM
Timing	1 MHz	TTL	CM
Timing	5 m TM	TTL (5 msec Timing Mark)	CM
Timing	100 KHz	TTL	CM
Timing	20 m TM	TTL (20 msec Timing Mark)	CM
Logic	M0	TTL (CRU address)	CRIM
Logic	M1	TTL (CRU address)	CRIM
Logic	M2	TTL (CRU address)	CRIM
Logic	B0	TTL (CRU address)	CRIM
Logic	B1	TTL (CRU address)	CRIM
Logic	B2	TTL (CRU address)	CRIM
Logic	L0	TTL (CRU address)	CRIM
Logic	L1	TTL (CRU address)	CRIM
Logic	L2	TTL (CRU address)	CRIM

Logic	CRUOUT	TTL (CRU data from processor)	CRIM
Logic	CRUCLK	TTL (CRU clock from processor)	CRIM
Logic	RXSEL	TTL (receiver select)	CRIM
Logic	DOLOCKA	TTL (lock detection bit)	NBM A
Logic	DOLOCKB	TTL (lock detection bit)	NBM B
Logic	HISLIPA	TTL (slip detection bit)	NBM A
Logic	HISLIPB	TTL (slip detection bit)	NBM B
Logic	LOSLIPA	TTL (not connected in NBM)	NBM A
Logic	LOSLIPB	TTL (not connected in NBM)	NBM B
Signal	CH1COR	Analog (correlation voltage)	NBM A
Signal	CH2COR	Analog (correlation voltage)	NBM B
Signal	DATA1	Analog (SV data)	NBM A
Signal	DATA2	Analog (SV data)	NBM B

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTIN.
Logic	CPDB0	TTL (decode address lines)	FSM
Logic	CPDB1	TTL (decode address lines)	FSM
Logic	CPDB2	TTL (decode address lines)	FSM
Logic	CPDACLOAD	TTL (D/A converter load pulse)	FSM
Logic	M700	TTL (decode bit)	CGM
Logic	CRUIN	TTL (data out of DM)	CM
Logic	CLTAU1	TTL (time constant control bit)	WBM
Logic	CLTAU2	TTL (time constant control bit)	WBM
Logic	CLTAU3	TTL (time constant control bit)	WBM

Logic	CLL1L2	TTL (bandpass filter control bit)	WBM
Logic	CLATT	TTL (attenuator control bit)	WBM
Logic	CLAGCTC11	TTL (AGC time constant control bit)	NBM A
Logic	CLAGCTC21	TTL (AGC time constant control bit)	NBM A
Logic	CLAGCSEL1	TTL (AGC control bit)	NBM A
Logic	CLAGCHOLD1	TTL (not used)	NBM A
Logic	CLGSWA	TTL (G switch control bit)	NBM A
Logic	CLAGCTC12	TTL (AGC time constant control bit)	NBM B
Logic	CLAGCTC22	TTL (AGC time constant control bit)	NBM B
Logic	CLAGCSEL2	TTL (AGC control bit)	NBM B
Logic	CLAGCHOLD2	TTL (not used)	NBM B
Logic	CLGSWB	TTL (G switch control bit)	NBM B
Logic	CLANTCTL	TTL (antenna control bit)	ANT SW
Logic	500CRUOUT	TTL (CRU data)	FSM
Logic	500L0	TTL (L field decode bit)	FSM
Logic	500L1	TTL (L field decode bit)	FSM
Logic	500L2	TTL (L field decode bit)	FSM

6.1.6 CLOCK MODULE

6.1.6.1 GENERAL DESCRIPTION

The Clock Module (CM) processes and generates timing signals to be used in the GPS receiver modules. The CM also contains a 6-to-1 data multiplexer for processor input data (CRUIN) from the five receiver channels.

6.1.6.2 FUNCTIONAL DESCRIPTION (refer to Figure 6.1.6-1)

I. Processing of timing signals

The CM processes 10MHz to produce 10MHz, 1MHz, 100kHz, 10kHz, 200Hz, and 50Hz TTL timing signals, synchronized with the incoming signal from the reference oscillator.

10 MHz from the reference oscillator is converted to a TTL level, fed out directly, and divided down in a series of 3 divide-by-10 synchronous counters to produce the 10MHz, 1MHz, 100kHz and 10kHz. The 10kHz is fed through a divide-by-10 and a divide-by-5 synchronous counter for 200Hz, and finally through a divide-by-4 synchronous counter for 50Hz.

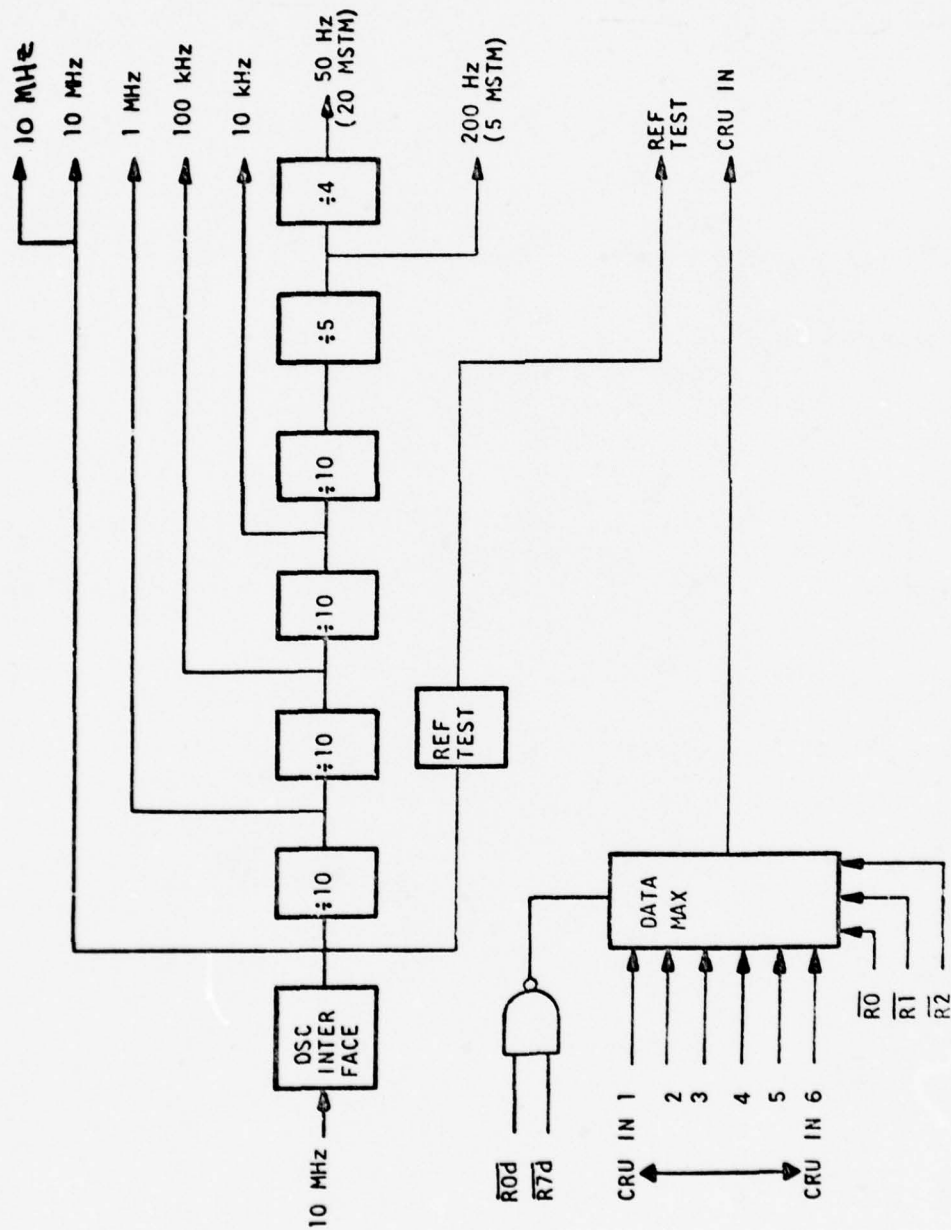


Figure 6.1.6-1. Functional Block Diagram Clock Module

II. CRUIN multiplexer

The CRUIN multiplexer is a TTL tri-state multiplexer that selects one of six CRUIN data lines. The multiplexer data selector is controlled by a three bit R field and the multiplexer output is enabled by the R0d and R7d signals. The five CRUIN outputs from the receiver channels, using five of the six inputs, are thus reduced to one for interface to the processor.

6.1.6.3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE
RF	10MHz Ref.	1 VRMS (1Kohm Load) Sinewave	Ref. Osc.
Power	5.2VDC	135 ma	pwr sup
Logic	CRUIN 1	TTL (channel 1 CRU data)	OM, CGM
Logic	CRUIN 2	TTL (channel 2 CRU data)	OM, CGM
Logic	CRUIN 3	TTL (channel 3 CRU data)	OM, CGM
Logic	CRUIN 4	TTL (channel 4 CRU data)	OM, CGM
Logic	CRUIN 5	TTL (channel 5 CRU data)	OM, CGM
Logic	/R0	TTL	MPM
Logic	/R1	TTL	MPM
Logic	/R2	TTL	MPM
Logic	/R0d	TTL	MPM
Logic	/R7d	TTL	MPM

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTIN.
Timing	10MHz Clock	TTL	FSM, OM, BITM
Timing	1MHz	TTL	OM
Timing	100KHz	TTL	not used
Timing	10KHz	TTL	CGM
Timing	1KHz	TTL	not used
Timing	200Hz	TTL	OM, MPM
Timing	50Hz	TTL	MPM
Logic	CRUIN	TTL	MPM

6.1.7 BUILT-IN-TEST MODULE

6.1.7.1 GENERAL DESCRIPTION

The Built-In-Test Module (BITM) generates seven test signals which are distributed to other parts of the system for calibration and fault isolation functions. The module is capable of generating L1 and L2 pseudo-replicas along with other lower frequency signals which are compatible with other inputs within the system to allow fault isolation. The term pseudo-replicas is used above since a complete L1 and L2 is not generated. The only code available is the X1-code from the P-code registers associated with P-code generation. Data for the RF carrier is only the Barker Code associated with the 50-Hz subframes.

The BITM is a CRU device which interfaces with the system processor via the RMBL address structure. The module can be controlled via this interface and is capable of accepting TTL status information to be read by the system processor for fault isolation.

6.1.7 FUNCTIONAL DESCRIPTION

As shown in Figure 6.1.7-1, the BITM is subdivided into three major areas: 1) RF generation, 2) code generation, and 3) processor interface.

I. RF generation

The RF generation area generates a series of RF signals for system distribution. The signals include $18-F_0$ for the Narrowband Module input, BITE L1 and L2 for Wideband Module input, and 230 KHz for Output Module input. All BITE RF signals are sourced by an F_0 generator which is derived by mixing an external 10-MHz signal (from the Master Oscillator in the system) with a signal from a 230-KHz crystal oscillator. The sum signal is filtered in a 10.23-MHz crystal filter and amplified in the F_0 bandpass filter. The F_0 signal is then routed to an impulse generator which is a squaring amplifier providing a signal rich in harmonics.

The squared signal is routed to three filter sections. The first section is the $18-F_0$ section which amplifies and filters the eighteenth harmonic. The filtered signal is then mixed with $18-F_0$ code to form the $18-F_0$ BIT signal for the Narrowband Modules. The second and third sections are closely intertwined to generate the BIT L1 and L2 signals. The second section is the $17-F_0$ section which filters and amplifies the seventeenth harmonic. The filtered $17-F_0$ signal version is routed to a power splitter. One output from the power splitter is forwarded to a mixer to be mixed with L-Band code. The third section is the $120-F_0$ section which involves filtering and amplifying the 24th harmonic ($24-F_0$) and multiplying by a factor of 5

to obtain $120-F_0$, which is in turn filtered and amplified again. This $120-F_0$ signal is forwarded to a mixer to be mixed with the second output of the $17-F_0$ power splitter. The summation signal ($137-F_0$) is filtered and amplified in the $137-F_0$ band pass filter. Two key signals have been established. They are the $17-F_0$ signal with L-Band code and the $137-F_0$ signal. Each signal is routed to an RF switch. The control signals to these switches can either enable or inhibit the two outputs from each switch section. The outputs from each switch section are routed to two mixers which generate sum and difference signals ($120-F_0$ or L2, and $154-F_0$ or L1). The output of one mixer is routed out of the module to the Wideband Module, while the output of the other mixer is forwarded to a power splitter that provides the Antenna BIT signals. The latter signals are used for calibration that encompasses any hardware prior to the Wideband Module, such as preamplifiers and cables.

II. Code generation

The code generation area is primarily digital. The X1-CODE is generated under control of the system processor. The processor exercises initialization control over the code generator to reset and arm the generator. The X1-code generator is driven by an F_0 version clock once the X1 enable signal is activated to

allow generator operation. The Barker Code Generator is also reset and armed via the processor. This generator is driven by an external 50-Hz data clock. The outputs of the X1-Code Generator and the Barker Code Generator are exclusive OR'ed to form the L-Band Code version that is mixed with the 17-Fo signal. The L-Band Code and the Barker Code are routed to a multiplexer. The output of the multiplexer (18-Fo Code) can be either L-Band Code or Barker Code depending on the selection made via the processor.

III. Processor interface

The hardware in the processor interface is primarily digital to interface with the processor. The RMBL Address Decode hardware is processor CRU output logic designed to control the various functions within the module. RF signals are enabled or disabled at various points, and code generation control and initializations are performed via this interface. The CRU input logic accepts status inputs from external sources to be input to the processor via CRU operations.

6.1.7.3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY TYPE	SIGNAL NAME	CHARACTERISTIC	SOURCE
Power	5.2 V RF	125 ma	pwr sup
Power	- 5.0 VDC	10 ma	pwr sup
Power	12.0 VDC	145 ma	pwr sup
Logic	X1A START	TTL (X1-Code Generator enable)	CGM
Data	CRU OUT	TTL (serial CRU data)	MPM
Data	/L0-/L2	TTL (Latch field of processor Address Bus)	MPM
Data	/B0-/B2	TTL (Bit field of processor Address Bus)	MPM
Logic	RCVR gen	TTL (RMBL address decode enable)	MPM
Logic	/CRUCL	TTL (CRU Clock strobe)	MPM
Logic	DATACLK	TTL (50-Hz Data Clock)	CGM
RF	10MHZ	TTL (10-MHZ reference)	ref osc

II. Outputs

CATEGORY TYPE	SIGNAL NAME	CHARACTERISTIC	DESTIN.
Logic	230KHZ	TTL (buffered)	NBM
Logic	Code	TTL (Narrowband code enable)	NBM
RF	18Fo	-30 dBm	WBM
RF	BITE 154Fo	-50 dBm	WBM
RF	Ant1 + 2	1575.42 MHZ and 1227.6 MHZ at -15 dBm	ext preamp

6.1.8 SIGNAL DISTRIBUTION MODULE

6.1.8.1 GENERAL DESCRIPTION

The Signal Distribution Module (SDM) receives signals from both the clock module and the built in test module, amplifies them and distributes them to the five receiver channels.

6.1.8.2 FUNCTIONAL DESCRIPTION (refer to Figure 6.1.8-1)

The SDM receives L1 and L2 (154 Fo and 120 Fo) signals, modulated with data and the X1 epoch of the P code, from the built in test module. These signals are amplified and split five ways for use in the wideband modules of each receiver channel. These signals are not presently used.

The SDM also receives 18 Fo from the built in test module and handles it the same as L1 and L2. These signals also are not presently used.

The SDM also receives 10 MHz from the clock module and feeds it through a five-way splitter for use in the frequency synthesizer and the output module of each receiver channel.

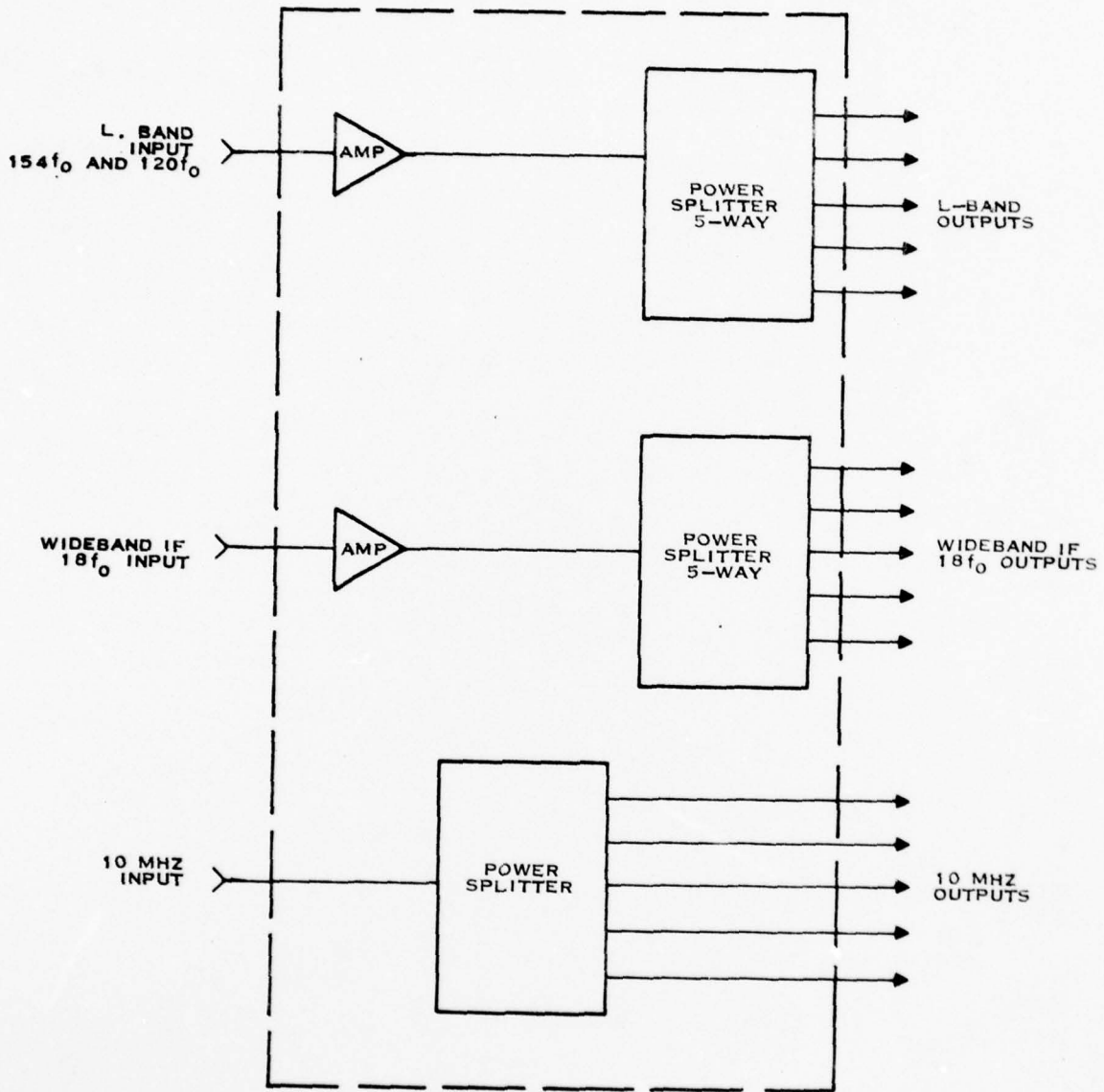


Figure 6.1.8-1. Functional Block Diagram Signal Distribution Module

6.1.8.3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE
RF	L-Band	not presently used (test signal)	BITM
RF	18 Fo	not presently used (test signal)	BITM
RF	10 MHz	TTL (reference signal)	CM
Power	5.2 V RF	120 ma	pwr sup
Power	15 VDC	27 ma	pwr sup

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTIN.
RF	L-Band	not presently used (test signals)	WBM(5)
RF	18 Fo	not presently used (test signals)	WBM(5)
RF	10 MHz	TTL (reference signals)	FSM(5), OM(5)

6.1.9 REFERENCE OSCILLATOR

6.1.9.1 GENERAL DESCRIPTION

The reference oscillator contains the a stable 10-MHz source and associated circuitry.

6.1.9.2 FUNCTIONAL DESCRIPTION (refer to Figure 6.1.9-1)

I. 10MHz Oscillator

The oscillator is a HP10544A with a voltage regulator on it's input voltage. The output signal is a minimum of 1 VRMS sinewave with an output impedance of 1 K ohm from an AC coupled (0.01 uf) emitter follower.

II. Voltage Regulator

15VDC is reduced to 12VDC with two regulator circuits. The output of one of these supplies voltage for the reference oscillator. The second supplies the voltage for the heater controller within the oscillator case.

Heater voltage for the reference oscillator is from 28VDC fed through a line filter.

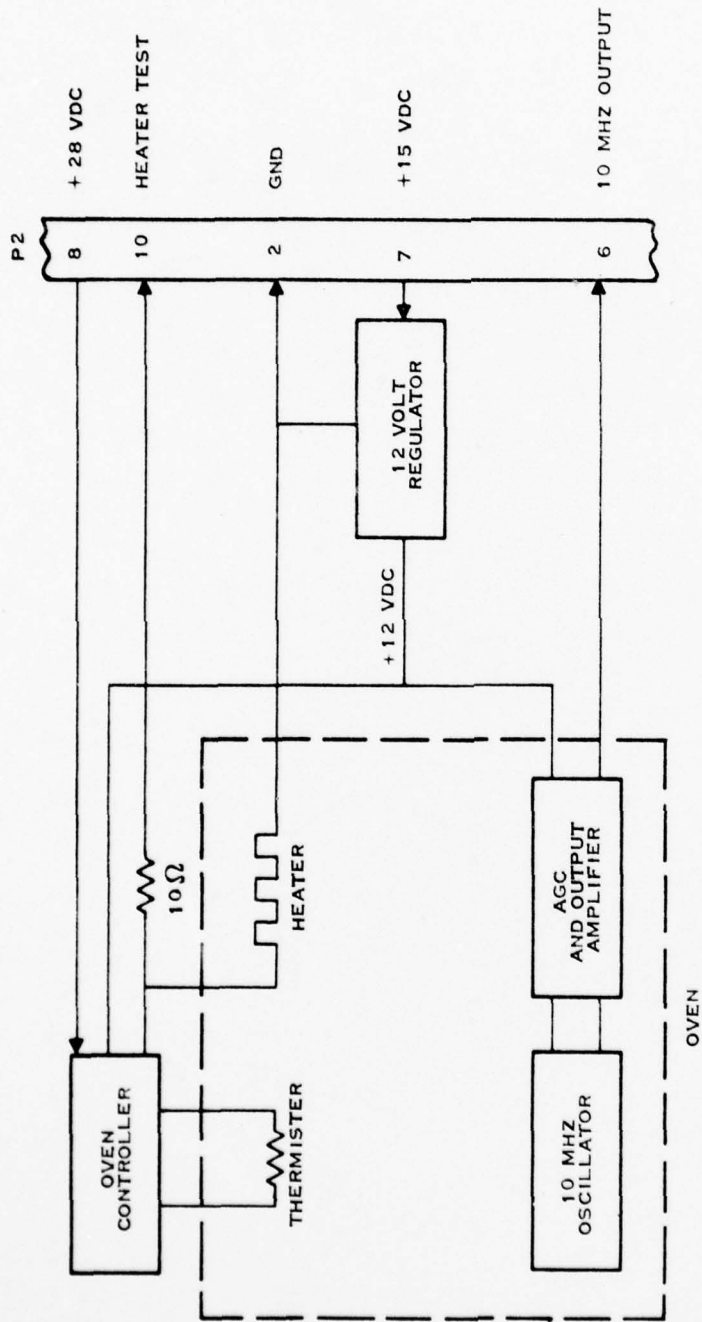


Figure 6.1.9.1. Functional Block Diagram Reference Oscillator

6. 1. 9. 3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE
Power	15VDC	0.33 w	Pwr sup
Power	28VDC	15.4 w during warm-up 2.94 w after warm-up	AC/DC Conv

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTIN.
RF	10MHz	1 VRMS sinewave	CM

6. 1. 10 LRU PREAMPLIFIER MODULE

6. 1. 10. 1 GENERAL DESCRIPTION

The LRU preamplifier sets the basic receiver noise figure and provides the required gain at L-Band (L1 and L2 frequencies)

6. 1. 10. 2 FUNCTIONAL DESCRIPTION

A LRU preamplifier is located on each RF input line coming from the GFE antenna and Preamp/PPDS front end. These LRU preamplifiers each have 27 dB gain, 4.7 dB noise figure, and a VSWR of less than 1.5:1 at both L1 and L2 frequencies.

6. 1. 10. 3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE
RF	SV signal	L1 and L2 carriers modulated with P-code, C/A-code, and data	Ant. input connectors
Power	12 VDC	83 ma max	power sup.

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTIN.
RF	SV signal	Input signal amplified by 27 dB	Ant-switch

6.1.11 ANTENNA SWITCH

6.1.11.1 GENERAL

The antenna switch contains the circuitry that allows selection of either antenna by the five receiver channels.

6.1.11.2 FUNCTIONAL DESCRIPTION (refer to Figure 6.1.11-1)

The antenna switch module is a 2 X 5 matrix switch that directs received signals from two antennas to combinations of five receiver channels. This is accomplished by dividing the RF signals at each of the antenna inputs into five equal signals presenting 1/5 of antenna A and 1/5 of antenna B to two selectable ports of five SPDT RF switches. These switches are controlled by TTL logic from the output module and they direct the selected signals in any combination to the five receiver channels.

AD-A076 481

TEXAS INSTRUMENTS INC DALLAS EQUIPMENT GROUP
GLOBAL POSITIONING SYSTEMS (GPS) HIGH DYNAMIC USER EQUIPMENT. (--ETC(U)
AUG 79

F/G 17/7

F04701-75-C-0180

UNCLASSIFIED

SAMSO-SD-TR-79-12-VOL-3

NL

4 OF 4

AD
A076481



END
DATE
FILMED
12-79
DDC

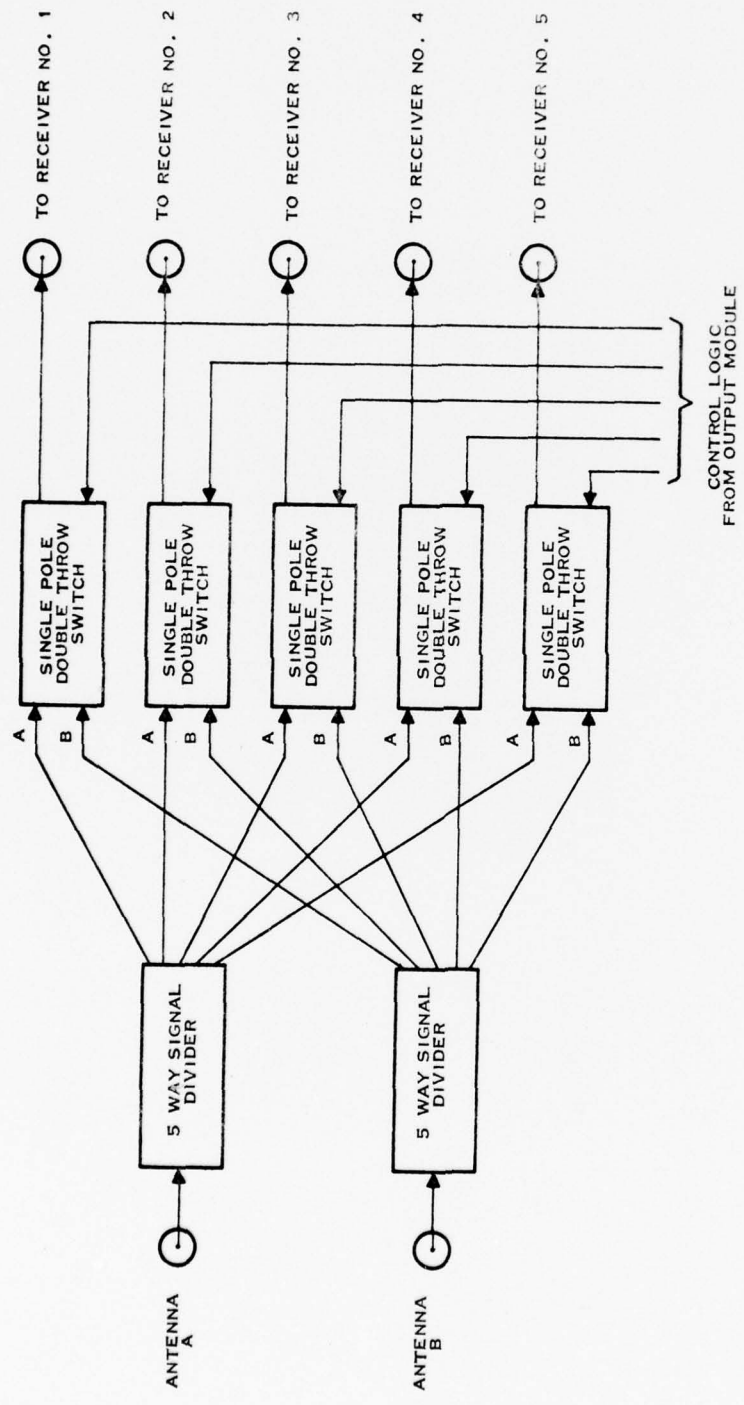


Figure 6.1.11-1. Functional Block Diagram Antenna Switch

6. 1. 11. 3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE
RF	SV signal	L1 and L2 carriers modulated with P-code, C/A-code, and data	LRU preamps
Power	5VDC	3 w max.	Pwr sup
Power	-5VDC	3 w max.	Pwr sup
Logic	Rx. 1 thru 5 Control	TTL (Hi for antenna A for resp. receiver channel)	OM

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTIN.
RF	SV signal	Source RF attenuated by 9 dB	WBM for resp RCVR chan.

6. 1. 12 RECEIVER POWER SUPPLY

6. 1. 12. 1 GENERAL DESCRIPTION

A DC-to-DC converter provides the power requirements for the RCVR-LRU. The power supply also provides internal monitoring of all supply voltages for operation within specified tolerances.

6. 1. 12. 2 FUNCTIONAL DESCRIPTION

The RCVR-LRU power supply (Figure 6. 1. 12-1) consists of seven sections: two switching regulators, two DC-to-DC converters, a precision reference, an over/under voltage (OV/UV) monitor, and an over temperature monitor.

I. Switching regulators

28 VDC primary power for the RCVR-LRU is regulated by one of two switching regulators operating at 40KHz.

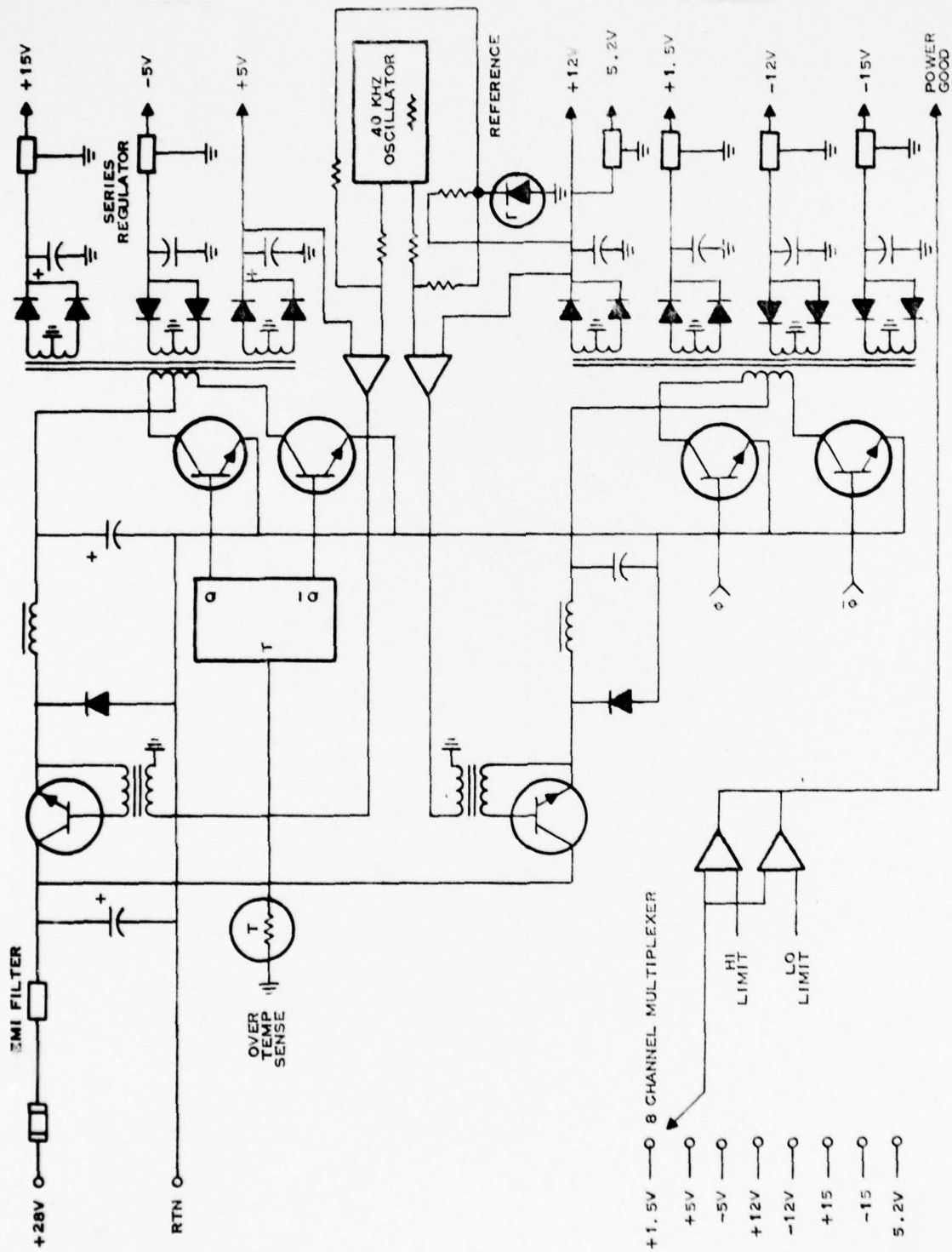


Figure 6.1.12-1. Functional Diagram Receiver Power Supply

II. DC-to-DC converters

The output of each regulator supplies the primary of a DC-to-DC converter switching at a 20-KHz rate. These transformers provide multiple secondaries which are rectified and filtered to provide the voltages required by the receiver. The two highest current loads occur on the +5-volt and +12-volt outputs. One of these voltages is assigned to each converter. These outputs are sensed, compared to a reference, and feedback generated for the switching regulator which supplies the respective converter primary, thus providing closed loop control.

All other voltages roughly track the primary regulation as a result of the transformer winding ratios. These windings are designed to supply approximately 3 volts above the desired output voltage at the rectifier/filter node. A regulator then drops each voltage to the desired output level while providing increased regulation and ripple filtering at relatively high efficiency due to the small regulator drop and lower current requirements.

III. Precision reference

The +12-volt supply provides power to a precision reference supply which establishes the desired level of the output voltages and the OV/UV sense levels. This section contains a 40-KHz sawtooth oscillator which modulates a +5-volt reference point. Comparators sensing the +5-volt output and the +12-volt output (divided down to +5 volts)

produce duty cycle modulated pulse trains with ON times proportional to the error between the reference and the output. These waveforms are coupled back to control the duty cycle of the switching regulators. By this means the nominal 15-volt transformer primary voltage is adjusted to correct for input voltage and output load variations.

IV. Over/under voltage monitor

The OV/UV detection is provided by a window detector whose inputs are multiplexed to sample the output voltages in sequence. The level of each input is shifted to +5 volts by either a resistive divider or an inverting amplifier (negative voltages). These are sampled by an eight-channel analog multiplexer which is driven by a counter clocking at the sample rate of the window detector. Any output voltage which deviates more than ± 5.3 percent from its specified value causes the window detector to generate an out-of-limit condition by causing the power good signal to go to a logical "0". Scanning is performed on a continuous basis.

V. Over-temperature monitor

In the event the internal temperature of the RCVR-LRU exceeds a preset threshold, power is removed from the switching regulators. This condition remains until the temperature returns to the safe operating limits of the RCVR-LRU components.

ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE
Power	28 VDC	6.3 a	AC/DC Conv

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTIN.
Power	1.5 VDC	500 ma	RCVR-LRU
Power	5 VDC	10.6 a	RCVR-LRU
Power	5.2 VDC	3.6 a	RCVR-LRU
Power	-5 VDC	560 ma	RCVR-LRU
Power	12 VDC	2.5 a	RCVR-LRU
Power	-12 VDC	720 ma	RCVR-LRU
Power	15 VDC	20 ma	RCVR-LRU
Power	-15 VDC	180 ma	RCVR-LRU
Logic	PSPG	TTL (power supply good)	CRIM

6.1.13 MICROPROCESSOR MODULE

6.1.13.1 GENERAL DESCRIPTION

The Microprocessor Module (MPM) provides computational and functional control capability for the HDUE. Information transfers between the microprocessor and memory for instruction fetch operations and data storage/retrieval operations is accomplished by a bi-directional memory bus under microprocessor control. Input/output functions for the microprocessor are accomplished by the direct instruction driven interface designated as a Communication Register Unit (CRU) interface. A direct memory access (DMA) capability is implemented such that the microprocessor releases control and/or data signals to affect the transfer between the external device and memory.

6.1.13.2 FUNCTIONAL DESCRIPTION (refer to Figure 6.1.13-1)

From the block diagram shown in Figure 6.1.13-1, the MPM consists of a microprocessor, address decode logic, read-only memory (ROM), ROM power switching circuitry, read/write random access memory (RAM), clock generation logic, output buffer logic, and input buffer logic.

A complete list of the microprocessor instruction set is shown in Table 6.1.13-1. Data and control interfaces are described as follows:

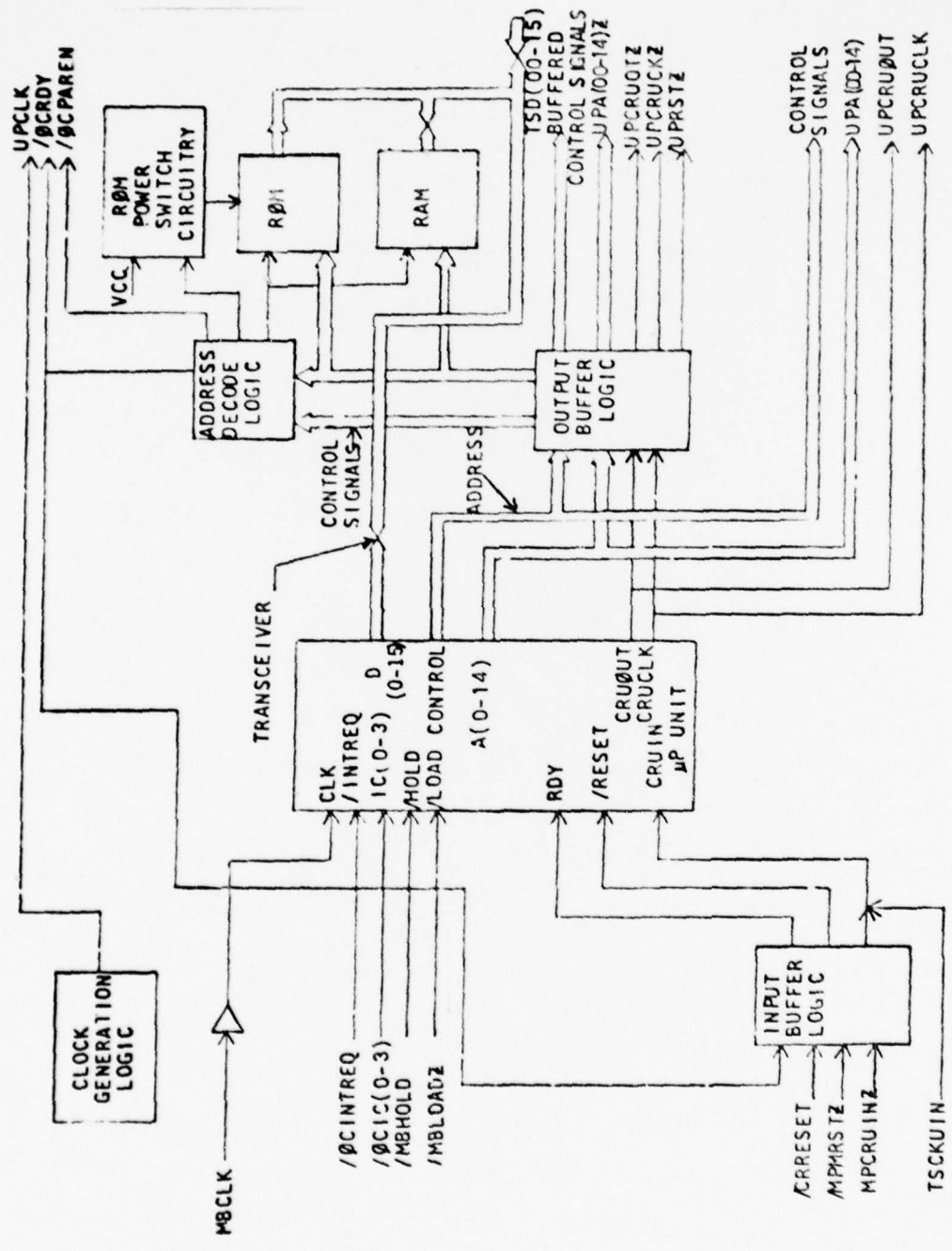


Figure 6.1.13-1. Function Block Diagram Microprocessor Module

Table 6.1.13-1. Instruction Set

ALPHABETIC LIST OF ASSEMBLY LANGUAGE INSTRUCTIONS

Mnemonic	Operands	OP Code	Instruction
A	G,G<=	A000	ADD (WORD)
AB	G,G<=	B000	ADD (BYTE)
ABS	G, -	0740	ABSOLUTE VALUE
AI	WR<=,I	0220	ADD IMMEDIATE
ANDI	WR<=,I	0240	AND IMMEDIATE
B	G	0440	BRANCH
BL	G	0680	BRANCH AND LINK (W11)
BLWP	G	0400	BRANCH, LOAD WORKSPACE POINTER
C	G,G	8000	COMPARE (WORD)
CB	G,G	9000	COMPARE (BYTE)
CI	WR,I	0280	COMPARE IMMEDIATE
CKOF	----	03C0	CLOCK OFF (NOTES 1,3)
CKON	----	03A0	CLOCK ON (NOTES 1,3)
CLR	G	04C0	CLEAR OPERAND
COC	G,WR	2000	COMPARE ONES CORRESPONDING
CZC	G,WR	2400	COMPARE ZEROES CORRESPONDING
DEC	G	0600	DECREMENT BY ONE
DECT	G	0640	DECREMENT BY TWO
DIV	G,WR<=	3C00	DIVIDE
IDLE	----	0340	COMPUTER IDLE (NOTE 1,3)
INC	G	0580	INCREMENT BY ONE
INCT	G	05C0	INCREMENT BY TWO
INV	G	0540	INVERT
JEQ	PC	1300	JUMP ON EQUAL (ST BIT 2=1)
JGT	PC	1500	JUMP GREATER THAN (ST BIT 1=1)
JH	PC	1800	JUMP HIGH (ST BITS 0=1 AND 2=0)
JHE	PC	1400	JUMP HIGH OR EQUAL (ST BIT 0=1 OR 2=1)
JL	PC	1A00	JUMP LOW (ST BITS 0=0 AND 2=0)
JLE	PC	1200	JUMP LOW OR EQUAL (ST BIT 0=0 OR 2=1)
JLT	PC	1100	JUMP LESS THAN (ST BITS 1=0 AND 2=0)
JMP	PC	1000	JUMP UNCONDITIONAL
JNC	PC	1700	JUMP NO CARRY (ST BIT 3=0)
JNE	PC	1600	JUMP NOT EQUAL (ST BIT 2=0)
JNO	PC	1900	JUMP NO OVERFLOW (ST BIT 4=0)
JOC	PC	1800	JUMP ON CARRY (ST BIT 3=1)
JOP	PC	1C00	JUMP ODD PARITY (ST BIT 5=1)
LDCR	G,NOTE 4	3000	LOAD CRU (NOTE NINE)
LDD	G	07C0	LONG DISTANCE DESTINATION (NOTE 1,2)
LDS	G	0780	LONG DISTANCE SOURCE (NOTE 1,2)
LI	WR<=,I	0200	LOAD IMMEDIATE
LIMI	I	0300	LOAD INTERRUPT MASK IMMEDIATE (NOTE 1)
LMF	WR,NOTE 5	0320	LOAD MAP FILE (NOTE 1,2)
LREX	----	03E0	LOAD ROM AND EXECUTE (NOTE 1,3)
LWPI	I	02E0	LOAD IMMEDIATE TO WORKSPACE POINTER
MOV	G,G<=	C000	MOVE WORD
MOVB	G,G<=	D000	MOVE BYTE
MPY	G,WR<=	3800	MULTIPLY
NFG	G	0500	NEGATE (TWO'S COMPLEMENT)
ORI	WR<=,I	0260	OR IMMEDIATE
RSET	----	0360	RESET AU (NOTE 1,3)
RTWP	----	0380	RETURN FROM SUBROUTINE (NOTE 8)
S	G,G<=	6000	SUBTRACT WORD
SB	G,G<=	7000	SUBTRACT BYTE
SBO	CRU	1D00	SET CRU BIT TO ONE (NOTE 9)
SBZ	CRU	1E00	SET CRU BIT TO ZERO (NOTE 9)
SETO	G	0700	SET ONES
SLA	WR<=,NOTE 6	0A00	SHIFT LEFT ARITHMETIC
SOC	G,G<=	E000	SET ONES CORRESPONDING (WORD)
SOCB	G,G<=	F000	SET ONES CORRESPONDING (BYTE)
SRA	WR<=,NOTE 6	0800	SHIFT RIGHT (MSB EXTENDED)

ALPHABETIC LIST OF ASSEMBLY LANGUAGE INSTRUCTIONS (Continued)

Mnemonic	Operands	OP Code	Instruction
SRC	WR<=,NOTE 6	0B00	SHIFT RIGHT CIRCULAR
SRL	WR<=,NOTE 6	0900	SHIFT RIGHT LOGICAL
STCR	G<=,NOTE 4	3400	STORE FROM CRU (NOTE 9)
STST	WR	02C0	STORE STATUS REGISTER
STWP	WR	02A0	STORE WORKSPACE POINTER
SWPB	G	06C0	SWAP BYTES
SZC	G,G<=	4000	SET ZEROES CORRESPONDING (WORD)
SZCB	G,G<=	5000	SET ZEROES CORRESPONDING (BYTE)
TB	CRU	1F00	TEST CRU BIT (NOTE 9)
X	G	0480	EXCLUDE
XOP	G,NOTE 7	2C00	EXTENDED OPERATION
XOR	G,WR<=	2800	EXCLUSIVE OR

<= INDICATES THE ADDRESS INTO WHICH THE RESULTS ARE PLACED WHEN 2 OPERANDS ARE SPECIFIED.

NOTES

1. PRIVILEGED INSTRUCTION - MODEL 990/10
2. MODEL 990/10 WITH MAP OPTION ONLY
3. NOT IMPLEMENTED IN THE TMS 9900
4. THE SECOND OPERAND IS THE NUMBER OF BITS TO BE TRANSFERRED, 0-15 (0=16)
5. THE SECOND OPERAND SPECIFIES A MEMORY MAP FILE #0 OR 1.
6. THE SECOND OPERAND IS THE SHIFT COUNT, 0-15, 0 MEANS THE COUNT IS IN BITS 12-15 OF WORKSPACE REGISTER 0. WHEN COUNT =0 AND BITS 12-15 OF WORKSPACE REGISTER 0 = 0, SHIFT COUNT IS 16.
7. SECOND OPERAND SPECIFIES THE EXTENDED OPERATION, 0-15 DISPOSITION OF THE RESULT MAY OR MAY NOT BE IN OPERAND ONE, AS DETERMINED BY THE USER.
8. WHEN PRIVILEGED INSTRUCTION BIT (7) OF THE STATUS REGISTER IS SET, ONLY BITS 0-6 ARE RETURNED TO THE STATUS REGISTER BY RTWP.
9. WHEN PRIVILEGED INSTRUCTION BIT (7) OF THE STATUS REGISTER IS SET, CRU ADDRESSES GREATER THAN >E00 (INCLUDING EXPANSION CHASSIS 7) ARE ILLEGAL.

I. Memory interface

To accomplish transfers between the microprocessor and memory, the microprocessor provides a 15-bit address on the memory address bus along with the necessary memory control signals. Memory data to or from the microprocessor is transferred via a 16-bit bi-directional data bus. The direction of the data flow on the data bus is controlled by the microprocessor. The addressed memory is either the memory contained on the MPM or memory external to the MPM. The memory interface also provides the capability for the microprocessor to relinquish control of the memory interface to an external device. Upon request by the external device, the microprocessor permits the external device to control the memory bus such that it transfers directly between the external device and memory. During such operations, the external device provides the necessary memory address and control signals. Upon completion of its transfers, the external device removes its request, and control of the memory bus is reverted to the microprocessor.

II. CRU interface

The microprocessor has the capability of serially transferring data to or from an external device by utilizing its CRU interface. The transfers are performed under microprocessor instruction control.

The instruction type determines the number of bits to be transferred (from one bit to 16 bits) and the direction of the data transfer. To perform transfers, the microprocessor provides a unique 12-bit address on the address bus for each bit to be transferred. If the data bit(s) is to be an output from the microprocessor, the microprocessor provides the output data on the CRU output signal line and a clock pulse on the CRU clock signal line. If the data is an input to the microprocessor, the addressed external device responds by placing input data on the CRU input signal line. No CRU clocks are provided by the microprocessor during CRU input operations.

III. Maintenance panel interface

The microprocessor interfaces to the maintenance panel (MP) to permit MP user to input data into the microprocessor or memory, or perform a breakpoint operation. Input data is entered via switches on the MP and read into the microprocessor or memory by executing a CRU input instruction with the appropriate address. Data to be displayed on the MP is outputted to the MP by executing a CRU output command with the appropriate CRU address. The breakpoint function permits the user to select, via switches on the MP, a breakpoint memory address or mode and upon coincidence of these with the microprocessor memory address and

mode, the microprocessor stops execution and reverts to the execution of MP firmware.

IV. Interrupt interface

The microprocessor has the capability to accept up to 15 external interrupts. Upon receipt of an interrupt request signal and a 4-bit encoded interrupt code, the microprocessor performs a context switch, provided the received interrupt code is less than or equal to the interrupt mask in the microprocessor. If not, the interrupt request is ignored until the above condition is satisfied.

6.1.13.3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTICS	SOURCE
Power	5 VDC	1.5 a	pwr sup
Power	1.5 VDC	750 ma	
Logic	/MBHOLD	TTL (to halt MPM for DMA)	DMM
Logic	/DCRDY	TTL (memory ready status)	DMM
Data	MPCRUIZ	Tri-state (MP serial input data)	MP
Logic	MPLOADZ	TTL (non-maskable interrupt to MP load routine)	MP
Logic	MPMRSTZ	TTL (zero level non-maskable interrupt to reset MPM)	MP
Logic	/DCINTREQ	TTL (interrupt request)	CRIM
Logic	/DCIC(0-3)	TTL (interrupt code)	CRIM
Logic	MBCLK	TTL (MPM instruction cycle clock)	MPM

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTICS	DESTIN.
Logic	UPA(00-14)B	Tri-state (memory address bus)	DMM
Logic	/UPMEMENB	Tri-state (valid memory address)	DMM
Data	TSCRUIIN	Tri-state (CRU serial input data)	MPM
Logic	/UPDBINB	Tri-state (to disable output buffers)	DMM
Logic	/UPWE	Open-collector (write enable)	DMM
Logic	UPIAGB	Tri-state (memory instruction fetch)	DMM
Logic	/UPENDCY	Open-collector (end of present memory operation)	DMM
Logic	UPSEOC	TTL (synchronized end-of-cycle)	DMM
Logic	/OCPAREN	TTL-DR (parity enable for RAM)	DMM
Logic	UPHOLDA	Open-collector (MPM hold acknowledge)	DMM
Logic	/UPWAIT	Open-collector (MPM in WAIT state)	DMM
Data	UPCRUOUT	TTL (CRU serial output data)	OM, CGM
Logic	UPCRUCLK	TTL (CRU data clock)	OM, CGM
Data	UPA(00-14)Z	TTL (MP address bus)	MP
Data	UPCRUOUTZ	TTL (MP serial data out)	MP
Logic	UPCRUCKZ	TTL (MP serial data clock)	MP
Logic	/UPMEMENZ	TTL (MP valid memory address)	MP
Logic	UPDBINZ	TTL (MP disable output buffer)	MP
Logic	UPIAGZ	TTL (MP instruction fetch)	MP

Logic	UPENDCYZ	TTL (MP end-of-cycle)	MP
Logic	UPRSTZ	TTL (MP reset)	MP
Analog	UPVCCSAM	5 VDC (status)	MP
Logic	UPCLK	TTL-DR (clock output source)	MP
Logic	UPB.25 mh	TTL-DR (cycle clock output)	MPM
Logic	UPBDBIN	TTL (output buffer disable)	MP
Logic	UPRWRDY	TTL (read/write ready)	MPM
Logic	UPRWOE	TTL (enable RAM output)	DMM
Logic	UPRWWE	TTL (write operation)	DMM

III. Bi-directional

CATEGORY	SIGNAL NAME	CHARACTERISTICS	SOURCE/ DESTIN.
Data	TSD(00-15)	Tri-state (data bus)	DMM

6.1.14 DATA MEMORY MODULE

6.1.14.1 GENERAL DESCRIPTION

The Data Memory Module (DMM) is a Random Access Memory module with capabilities to both read and write 4096 17-bit words. The module can be programmed to respond to any 4K memory space within a 20-bit address space.

6.1.14.2 FUNCTIONAL DESCRIPTION (refer to Figure 6.1.14-1)

As shown in Figure 6.1.14-1 the DMM is designed to interface onto a processor Data Bus and Address Bus. The Data Bus is a 17-bit bus which either is connected directly to the 4096 Memory Array for write operations, or is connected to the Output Data Buffer section. These output buffers are operable during a read operation from the Memory Array under the control of the Data Bus IN/READ command from the system. The Address Bus is a 20-bit address bus which is subdivided into three sections. The five MSB's of the address bus are utilized to define which of 32K memory sectors this module is to be located in when system requirements are greater than 32K. The next 4 MSB's are utilized to define where within a 32K memory space the 4K module will reside. The last 11 LSB's are utilized to address the Memory Array itself. The Module Select section generates all module enable commands derived from the 9 MSB'S of the address bus, while the Address Buffer section drives the Memory Array.

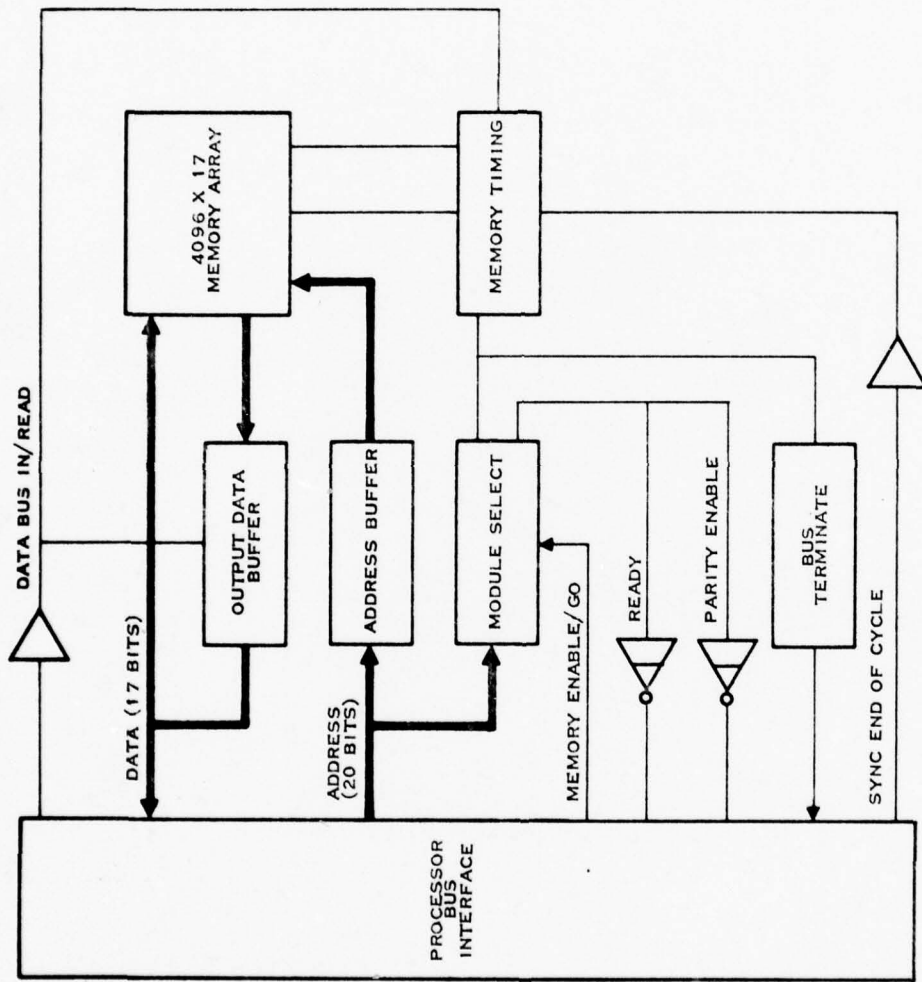


Figure 6.1.14-1. Functional Block Diagram Data Memory Module

The READY and PARITY enable signals generated by the Module Select section serve to interface with the system to satisfy the proper handshake requirements. The Bus Terminate section serves to allow interfacing with the I-BUS. This section generates a terminate status when either a read or write operation has been concluded on the DMM. The Memory Timing section is primarily a combinational logic block which controls read or write timing to the Memory Array.

ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE
Power	5 VDC	300 ma	pwr sup
Power	-5 VDC	50 ma	pwr sup
Power	12 VDC	300 ma	pwr sup
Data	TLAD(0-14)	TTL (system Address bits utilized to address within a 32K boundary)	MPM
Data	MSS(0-3)	TTL (Memory Sector Select inputs to define which 32K boundary a module is to occupy.)	system selects
Data	TLDAT 16	TTL (parity Data bit)	CRIM
Logic	MEMEN or TLGO	TTL (indicates that a memory operation is in progress)	MPM
Logic	TLREAD or UPDBIN	TTL (defines a Read operation when a memory function is in progress)	MPM
Logic	UPSEOC	TTL (enables data onto output data bus. Signal is active during last cycle of memory operation.)	MPM
Logic	DMLWR	TTL (enables data to be written into the memory array)	MPM

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTIN.
Logic	OCDY	TTL (indicates that the addressed memory is ready to read or write)	MPM
Logic	OCPAREN	TTL (indicates that parity needs to be checked or generated on the Parity Data Line)	CRIM
Logic	TLTM	TTL (denotes when the memory device has completed a read or write operation)	I-BUS

III. Bi-directional

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE/ DESTIN.
Data	TLDAT(0-15)	TTL (Data bus bits)	MPM

6.1.15 COMMUNICATION REGISTER INTERFACE MODULE

6.1.15.1 GENERAL DESCRIPTION

The Communication Register Interface Module (CRIM) performs a number of processor related functions. The functions include interrupt encoding, processor CRU decoding, address bus buffering, system reset control, and parity bit generation and verification for the system.

6.1.15.2 FUNCTIONAL DESCRIPTION

The CRIM is functionally shown in Figure 6.1.15-1.

I. Interrupt encoding

The Interrupt Control Logic is tasked with accepting a maximum of 8 external interrupt stimuli from the system and priority encoding the information for the system processor. Via the CRU function the processor may selectively mask off discrete interrupt inputs when the system warrants this feature.

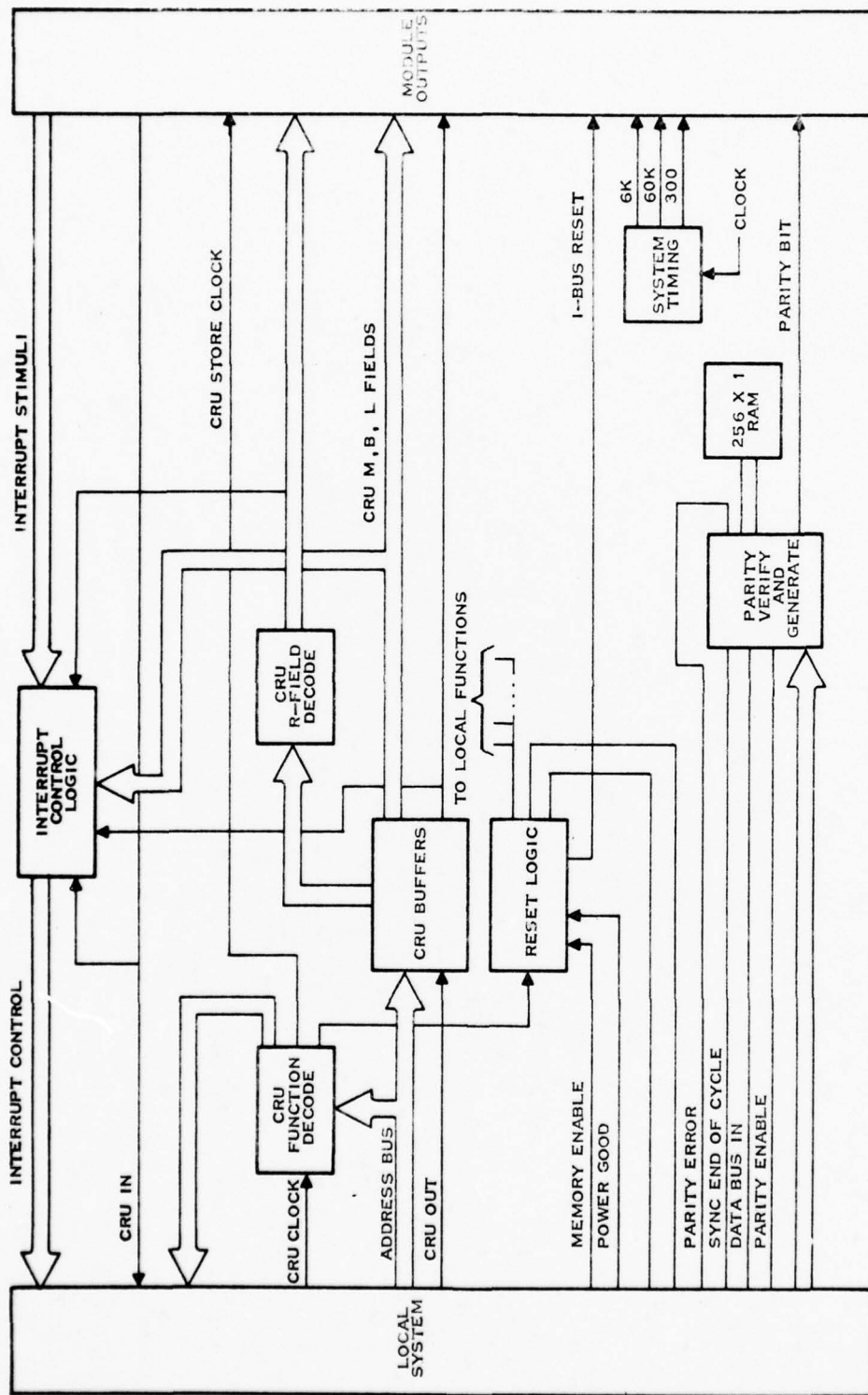


Figure 6.1.15-1. Functional Block Diagram Communication Register Interface Module

II. Processor CRU decoding

The CRU R-Field Decode section is tasked with decoding bits 03 thru 05 (R-Field of Address Bus) to define one of 8 CRU receiver spaces to be addressed for CRU control. These decoded signals are utilized in systems having more than one receiver channel. The CRU Function Decode section provides a set of decodes for special system functions. These special functions include IDLE, CLKON, and RESET which are special states that the processor could assume.

III. Address bus buffering

The CRU Buffers buffer the system address to route the M, B, and L Fields of the Address Bus to the system.

IV. System reset control

The Reset Logic section generates reset commands for various applications. The Power Good input to this section results in a level 0 (unmaskable) interrupt to the system processor to initialize the entire system (software and hardware). Software may also generate reset commands via instructions dedicated for this task.

V. Parity bit generation and verification

The Parity verify and generate section generates an odd parity bit as a function of bits on the Data Bus. The resultant Parity Bit is routed off the CRIM module to be stored in system memory. During a read operation when parity is verified a parity error may be generated to be routed to the system for disposition. The CRIM unit is also equipped with RAM storage for parity bits generated for 256 addresses in high memory space.

6.1.15.3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY TYPE	SIGNAL NAME	CHARACTERISTIC	SOURCE
Power	+5VDC	900 ma	pwr sup
Logic	/DCPAREN	Open-collector (indicates that parity needs to be checked or generated)	system memory
Logic	UPSEDC	TTL (Memory End of Cycle. Active during last cycle of any memory operation)	MPM
Logic	UPDBIN	TTL (denotes when processor is executing a read operation)	MPM
Data	/I(1-7)	TTL (Interrupt stimulus input)	system
Logic	/ICAS (8)	TTL (highest priority Interrupt)	system
Logic	UPWAIT	TTL (denotes when processor is in wait state because of a not ready condition from memory)	MPM
Logic	PSPG	TTL (Power Supply Power Good)	pwr sup
Data	UPA(0-14)	TTL (Processor Address Bus)	MPM
Logic	UPCRUCLK	TTL (Processor CRU Clock)	MPM
Data	UPCRUDOUT	TTL (Processor CRU serial output data)	MPM
Logic	/UPMEMEN	TTL (Denotes when processor is executing a memory operation (read or write))	MPM

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTIN.
Logic	/CRPARERR	TTL (denotes that a read parity error has occurred)	MPM
Logic	/DCINTREG	Open-collector (denotes that an interrupt is pending)	MPM
Data	/DCIC(1-3)	Open-collector (encoded interrupt code indicates code associated with interrupt pending)	MPM
Logic	/OCDY	Open-collector (delayed wait from processor)	system
Data	TSCRUIIN	Tri-State (serial CRU data)	MPM
Data	/CRM(0-2)	TTL (Module field of address bus)	system
Data	/CRL(0-2)	TTL (Latch field of address bus)	system
Data	/CRB(0-2)	TTL (Bit field of address bus)	system
Data	/CRR(0-7)	TTL (decoded Register field of address bus)	system
Data	/CRA(3-5)	TTL (Register field of address bus)	system
Logic	/CRSTORCK	TTL (gated Store Clock from MPM)	MPM

III. Bi-directional

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE/ DESTIN.
Data	TSD(00-15)	TTL (System Data Bus)	MPM
Logic	TSPAR	Tri-state (Parity bit generated by the system as well as the bit generated by the CRIM unit during write operation)	system memory

6. 1. 16 SERIAL BUS INTERFACE MODULE

6. 1. 16. 1 GENERAL DESCRIPTION

The Serial Bus Interface Module (SBIM) comprised of two modules, SBIM 1 and SBIM 2, provides the interface between the Microprocessor Model (MPM) local bus and the Serial Time Division Multiplex (STDM) data bus. The SBIM has the capability to serve as either a master or a slave interface to the STDM data bus. The interface is in accordance with McDonnell Aircraft Corporation F-15 Digital Interface Design Specification, Report H009C, Contract No. F33657-69-C-0657.

6. 1. 16. 2 FUNCTIONAL DESCRIPTION (refer to Figure 6. 1. 16-1)

As a STDM bus master interface, the SBIM receives data and control from the local MPM, issues polling commands or data to the remote slaves, and receives status and data from the remote slaves for transfer to the MPM.

As a STDM bus slave interface, the SBIM receives polling commands from the bus master and responds with status and data. In the receive mode, it buffers up to sixteen words of data and sets a flag indicating to the local MPM when there is a word for transfer from the buffer. In the transmit mode, it receives up to fifteen words of data from the local MPM and transmits this data bit-serially to the STDM bus master. When the transmit buffer is empty, the SBIM sets a flag requesting additional data from the MPM. The SBIM, when a master, contains a

clock generator that is used for the STDM 1-MHz clock.

The circuitry to perform the afore mentioned functions is contained in two modules, SBIM 1 and SBIM 2. Basically the SBIM 1 module contains the control and SBIM 2 contains the analog interface circuitry. As shown in the block diagram of Figure 6.1.16-1, the SBIM consists of the following:

- (1) bi-phase Input/Output (I/O) bus communications
- (2) clock circuitry
- (3) control shift register
- (4) error detect circuitry
- (5) STDM bus parity circuitry
- (6) select word control bit latches
- (7) word count latch
- (8) controller with programmable read only memory (PROM)
- (9) First-In-First-Out (FIFO) buffer
- (10) status register
- (11) memory bus address decode
- (12) slave address comparator
- (13) microprocessor interface buffers.

I. Bi-phase I/O bus communications

The bi-phase bus communications performs the basic bus driving and receiving functions for the STDIM bus.

II. Clock circuitry

The clock circuitry, when the SBIM is used as a bus master, utilizes an internally generated 4-MHz oscillator signal to create a 1-MHz clock to be used as the basic timing unit within the SBIM and to be used by the bi-phase I/O bus communications as the bus clock. When the SBIM is a slave, the clock circuitry buffers the clock received by the bi-phase I/O bus communications for timing within the SBIM.

III. Shift register

The shift register may be parallel loaded by the MPM or the controller and PROM within the SBIM, or it may be loaded serially from the STDIM bus. The MPM writes into the shift register to load the select word only when the SBIM is used as a master. This action automatically causes the controller and PROM within the SBIM to begin placing the select word onto the STDIM bus. The MPM does not return to read the shift register as no bit pattern is guaranteed except following an operation terminated interrupt for a command select word. After a command word transfer is completed the original select word exists in the shift

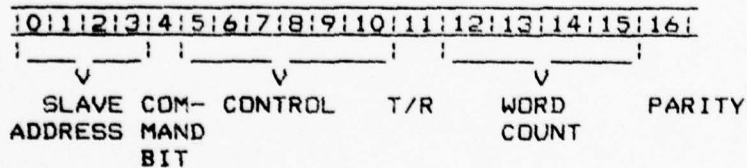
register if transfer was completed with no errors. The shift register is parallel loaded by the controller and PROM from the FIFO when the select word indicates a data transfer to a slave.

The shift register can also be loaded from the STDN bus. If the shift register is being loaded from the bus, it is under control of the PROM on the SBIM. If the MPM should address the shift register while it is being serially loaded, the MPM completes its memory cycles and the SBIM does not change.

IV. Error detect circuitry

The error detect circuitry monitors the input signals from the data bus and sets status bits 1 and 2. The bus timing error indicates when the no-data time period between data words (8 usec) is not properly maintained.

The select word is as follows:



V. Bus parity circuit

The bus parity circuit establishes parity for data that is transmitted onto the bus and checks parity of words that are received from the bus.

VI. Control bit latch

The control bit latch stores bit 4 and 11 from the select word. These bits are used along with the controller and PROM for proper data movement. When the SBIM is a master, only bit 11 (T/R) is used. This determines whether the master is in either the transmit or receive mode after the select word is transmitted. During power-up, this bit must be logical zero to indicate that a master is transmitting or a slave is receiving. When the SBIM is a slave, the control latch stores both bits. The T/R bit functions the same as in the master. If a one is in the command bit and a zero in the status bit 11, the slave loads the select word into the FIFO and then transmits the select word identically to the master as required by the F-15 bus specification. Control bits 5-10 of a command word are monitored by the MPM for proper action. Use of the bits is determined by the software.

VII. Word count latch

Bits 12-15 of the select word are loaded into this latch for use by the PROM controller. The PROM controller uses this latch as a counter for data movement on the bus. The output of the latch is placed in the status register bits 12-15 so that the MPM can monitor the progress of the SBIM in the present data transfer operation.

VIII. Controller and PROM

The controller and PROM contains the microprogram for all the data movement within the SBIM and the data timing and the data transfer on the STDM bus.

IX. First-in-first-out buffer

The FIFO is used as a data holding register. This register is capable of holding sixteen data words. If more words are attempted to be placed in this register they will be ignored. Status indicators of FIFO empty (SBBUFEMT) and FIFO full (SBBUFULL) are a part of the status word. The FIFO may be loaded by either the MPM or the shift register. In order to control which is loading the FIFO, status bit 11 is used from the status register.

X. Status register

All status register bits (see table 6.1.16-1) can be read by the MPM but only bit 11 may be written into by the MPM. When writing into the status register, the MPM must write with bits placed in the bit location in which they will be read. A write operation erases all previous data placed in bit 11.

XI. Address decode

The address decode circuitry decodes the address lines from the MPM in order to enable access to the FIFO, shift register, and status register.

XII. Address comparator

The address comparator circuit is used in a slave SBIM. This circuitry looks at bits 0-3 of the select word to determine if this is the slave that is being addressed. If address agreement exists, then the SBIM takes the action dictated by the select word control bits.

XIII. Microprocessor interface buffers

These buffers are used to guarantee that the SBIM will only put one load on the lines that interface to the MPM, and that the SBIM is capable of driving all interface lines going to the MPM.

Table 6.1.16-1 Status Bit Description

<u>BIT NO.</u>	<u>DESCRIPTION</u>	<u>COMMENT</u>
0	Logic 0 indicates SBIM is active Logic 1 indicates SBIM is not active, and the shift register may be read properly or written into.	Set by SBIM
1	Logic 1 indicates no reply to a select word.	Set by SBIM
2	Logic 1 indicates bus timing error, no data was on the bus when it should have been. Logic 0 otherwise.	Set by SBIM
3	Logic 1 when operation is terminated. Logic 1 otherwise.	Set by SBIM
4	Logic 1 when there was a parity error on the data received by the SBIM. Logic 0 otherwise.	Set by SBIM
5	Logic 1 indicates that the FIFO is empty. Logic 0 otherwise.	Set by SBIM
6	Logic 1 indicates the FIFO is full. Logic 0 otherwise.	Set by SBIM
7-10	Hardwired to a logic 1.	
11	Logic 1 indicates FIFO can be loaded by MPM and read by SBIM. Logic 0 indicates FIFO can be loaded by SBIM and read MPM.	Set by SBIM
12-15	Binary word count remaining in present operation.	Set by SBIM

6.1.15.3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE
Power	5VDC	1.6 a	pwr sup
Power	-5VDC	160 ma	pwr sup
Power	12VDC	70 ma	pwr sup
Power	-12VDC	80 ma	pwr sup
Logic	SBSENS	TTL (to enable SLAVE-only functions and disable MASTER-only functions)	MPM
Data	MBAD(0-3)	TTL (SLAVE address lines)	STDM
Logic	UPSEDC	TTL (synchronized end of memory cycle)	MPM
Data	UPA(00-14)	TTL (15-bit address bus)	MPM
Logic	UPWE	TTL (to indicate when memory write data is available to be written into memory)	MPM
Logic	UPDBIN	TTL (to indicate output buffers of MPM have been disabled and data bus input of read data is allowed)	MPM

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTIN.
Logic	OCOPTERM	Open Collector (STDM bus operation terminated interrupt)	MPM
Logic	QCRDY	TTL (to indicate memory ready to read or write during next clock cycle)	MPM

III. Bi-directional

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE/ DESTIN.
Data	TSD	Tri-state (16-bit data bus for movement of data to and from memory locations)	MPM
Logic	SBCKKZ	*TR DR (clock for transferring data between RCVR and NAV LRU's)	STDM
Data	SBDATAZ	TR DR (communication data between RCVR and NAV LRU's)	STDM

*TR DR = Transformer Drive

6.2 NAVIGATION LRU MODULES

6.2.1 MICROPROCESSOR MODULE

Refer to paragraph 6.1.13 for a description of the Microprocessor module.

6.2.2 DATA MEMORY MODULE

Refer to paragraph 6.1.14 for a description of the Data Memory module.

6.2.3 COMMUNICATION REGISTER INTERFACE MODULE

Refer to paragraph 6.1.15 for a description of the Communication Memory module.

6.2.4 SERIAL BUS INTERFACE MODULE

Refer to paragraph 6.1.16 for a description of the Serial Bus Interface module.

6.2.5 DRIVER/RECEIVER INTERFACE MODULE

6.2.5.1 GENERAL DESCRIPTION

The Driver/Receiver Interface Module (DRIM) serves as an interface to extend the Communication Register Unit (CRU) of the Communication Register Interface Module (CRIM) to a remotely located CRU device*.

*The CRU device in HDUE is the Control Display Unit (CDU)

6.2.5.2 FUNCTIONAL DESCRIPTION (refer to Figure 6.2.5-1)

In order to perform the above inface function the DRIM contains three catagories of circuitry: (1) drivers, (2) receivers, and (3) stop-the-clock circuitry.

I. Drivers

The DRIM contains 14 differential signal drivers that drive the CRU address lines /CRM(0-2), /CRB(0-2) and /CRL(0-2) (which correspond to address lines UPA(6-14), respectively), register enable /CRR(6), CRU data line /CRCRUOTB, CRU strobe /CRSTORCK, and two spares. The design of the DRIM allows the omission of devices if the respective signal is not to be used.

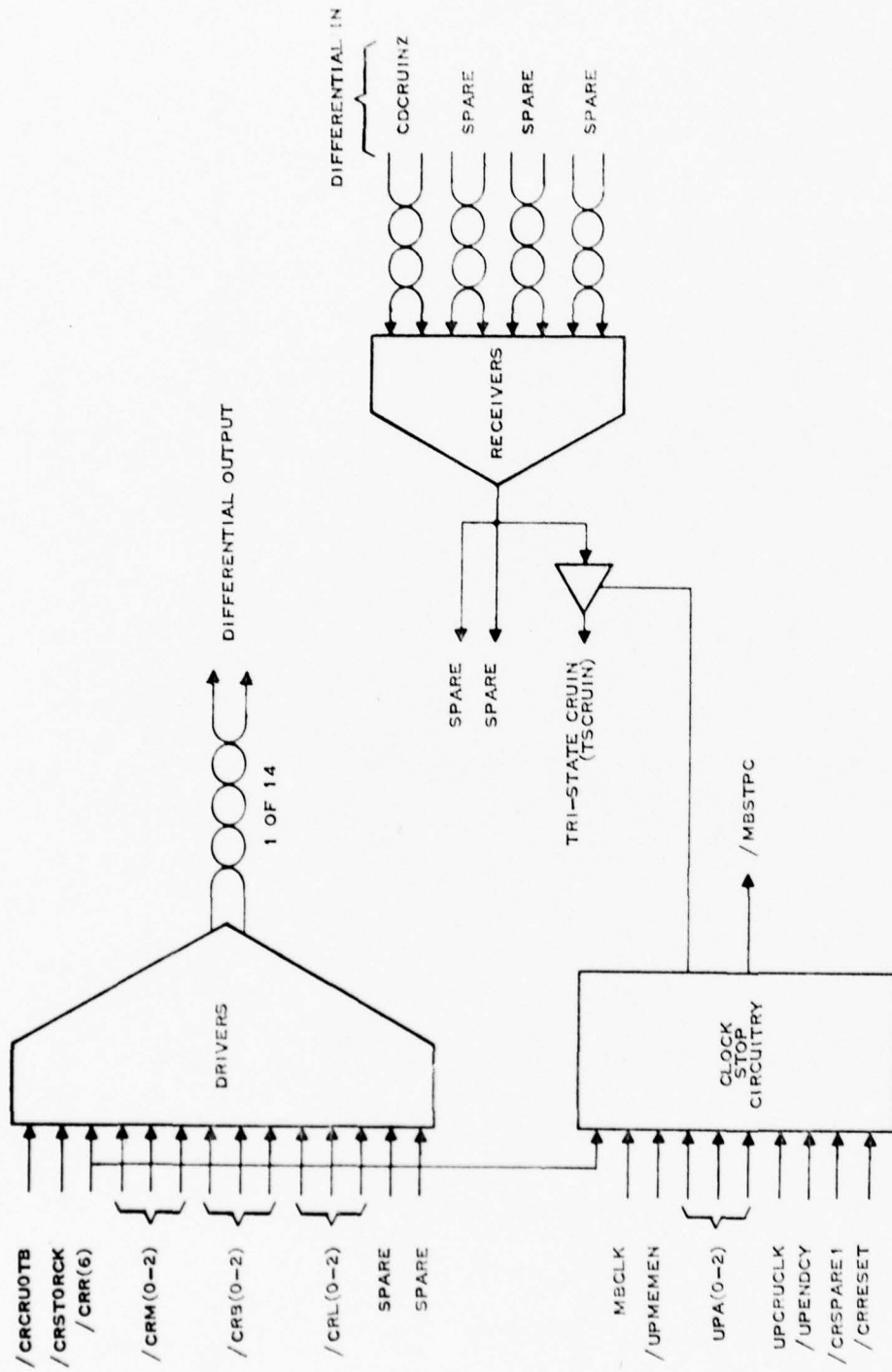


Figure 6.2.5-1. Functional Block Diagram Driver/Receiver Interface Module

II. Receivers

The DRIM contains 4 differential signal receivers to receive CDCRUINZ and /CDCRUINZ (CRU data from the CRU device) to drive TSCRUIIN when enabled by register enable /CRR(6) or /CRSPARE1. The DRIM also has the capability of differentially receiving three spare differential pairs.

III. Stop Clock Circuitry

The DRIM provides the capability to stop the system clock (MBCLK) by activating /MBSTOPC during CRU input instructions which use register enable /CRR(6) or /CRSPARE1. that is, /CRR(6) is assigned to HDUE CDU functions. The clock remains stopped for sufficient time to allow for transmission line propagation delays. Although /MBSTOPC is activated by both CRU input and CRU output functions, UPCRCLK is used to reset /MBSTOPC on CRU output functions causing no stoppage of MBCLK. When /MBSTOPC is set by CRU input functions, no UPCRCLK is present to reset /MBSTOPC. In this instance, resetting of /MBSTOPC is accomplished by a timer circuit which will time out after a predetermined period of time. The timer circuit time-out period is sufficient to allow for transmission line propagation delay. During the time /MBSTOPC is active for CRU input operations, /MBSTOPC is used to stop system clock.

6.2.5.3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE
Power	5VDC	400 ma	pw'r sup
Logic	/CRCRUOTB	TTL (CRU output data buffered)	CRIM
Logic	/CRSTORCK	TTL (CRU output data strobe buffered)	CRIM
Logic	/CRR(6)	TTL (Register enable line representing CRU addresses hex 1800 through hex 1BEF)	CRIM
Logic	/CRM(0-2)	TTL (Module address code UPA (6-8))	CRIM
Logic	/CRB(0-2)	TTL (Byte address code UPA (9-11))	CRIM
Logic	/CRL(0-2)	TTL (Latch address code UPA (12-14))	CRIM
Logic	/CRSPARE1	TTL (Spare enable line which is OR'd with /CRR(6))	CRIM
Logic	SPARD1 and SPARD2	TTL (Spare differential driver inputs)	CRIM
Logic	/CRRESET	TTL (CRIM reset signal)	CRIM
Logic	/UPENDCY	TTL (End of cycle from processor which indicates second half of CRU operation)	CRIM
Logic	/UPMEMEN	TTL (Memory enable indicating memory is being addressed)	CRIM
Logic	UPA(00-02)	TTL (Most significant top three bits of address bus which indicate CRU function when all three are low)	CRIM

Logic	MBCLK	TTL (System clock provided on motherboard)	CRIM
Logic	UPCRUCLK	TTL (CRU output data strobe)	CRIM
Data	CDUCRUINZ and /CDUCRUINZ	*TTL DR (CRU serial input data)	CDU-LRU
Data	SPARIN1Z thru SPARIN3Z and /SPARIN1Z thru /SPARIN3Z	*TTL DR (Spare)	CDU-LRU

*TTL DR = From a dual line driver type SN55114 or equivalent.

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTIN.
Logic	/MBSTOPC	TTL (Stop clock signal which may be used to stop MBCLK active)	CDU-LRU
Data	TSCRUIIN	TTL (CRU serial input data)	CRIM
Data	DRSPREC (1-3)	TTL (SPARE receiver outputs)	CRIM
Logic	/DRCRUOTZ and DRCRUOTZ	*TTL DR (CRU data)	CDU-LRU
Logic	/DRSTRCKZ and DRSTRCKZ	*TTL DR (CRU output strobe)	CDU-LRU
Logic	/DRMZ(0-2) and DRMZ(0-2)	*TTL DR (Module address code UPA(6-8))	CDU-LRU
Logic	/DRBZ(0-2) and DRBZ(0-2)	*TTL DR (Byte address code UPA(9-11))	CDU-LRU
Logic	/DRLZ(0-2) and DRLZ(0-2)	*TTL DR (Latch address code UPA(12-14))	CDU-LRU
Logic	/DRRZ(6) and DRRZ(6)	*TTL DR (REGISTER enable line representing CRU addresses hex 1800 through 1BFE)	CDU-LRU
Logic	DRPAR1Z and DRPAR2Z and /DRPAR1Z and /DRPAR2Z	*TTL DR (SPARES)	CDU-LRU

6.2.6 INTERNAL BUS INTERFACE MODULE

6.2.6.1 GENERAL DESCRIPTION

The Internal Bus Interface Module (IBIM) provides the required interface between a Microprocessor Module (MPM) and a parallel data internal bus (I-Bus). The IBIM and the MPM are the two elements used to make up an I-Bus MASTER device. An example of an I-Bus SLAVE device is the Data Memory Module (DMM). The I-Bus is an asynchronous high speed 16-bit data transfer bus and associated control lines which serve to transfer data between these high speed system elements.

6.2.6.2 FUNCTIONAL DESCRIPTION (refer to Figure 6.2.6-1)

The IBIM is used to transfer data, prioritize and acquire I-Bus control, and perform I-Bus HOLD and RESET functions. The functional block diagram for the IBIM is shown in Figure 6.2.6-1.

I. I-Bus MASTER-to-SLAVE write cycle

When a MASTER device has access to the I-Bus, it may accomplish a memory (SLAVE) write cycle through the following action. The IBIM asserts I-Bus GO (TLGO). At the same time, it must assert a write command with I-Bus READ (TLREAD). The MASTER also at this time generates valid WRITE DATA (TLDATA), valid parity (TLPAR), and a valid 15-bit ADDRESS (TLADR).

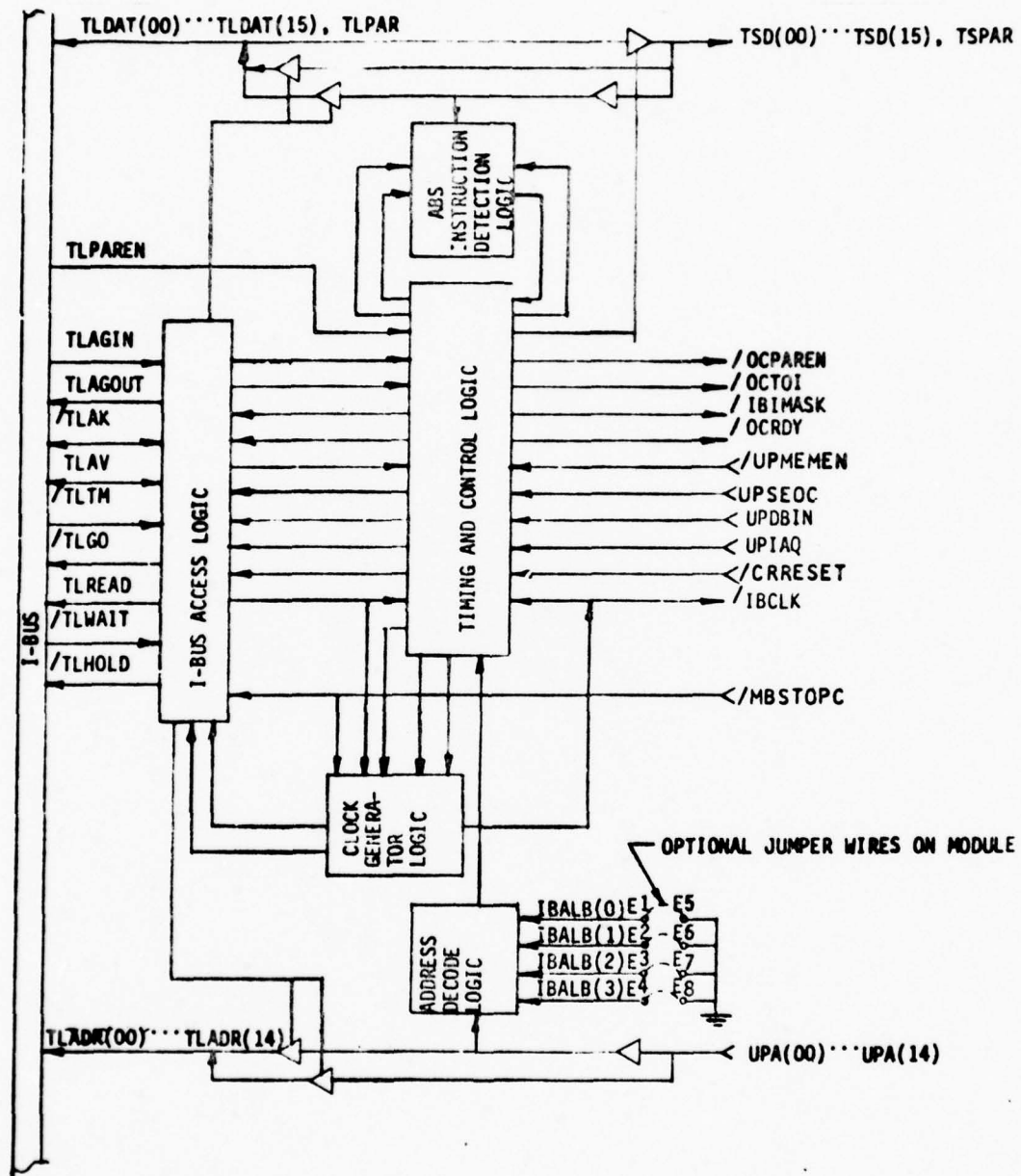


Figure 6.2.6-1. Functional Block Diagram 1-Bus Interface Module

All SLAVE devices interfaced to the I-Bus will receive the GO transmitted from the MASTER. The SLAVE devices must decode the address to determine which SLAVE is being addressed. When a SLAVE device decodes the address as valid, it must perform a write cycle and then assert I-Bus TERMINATE (TLTM).

When the IBIM receives the release TERMINATE, it may begin a new cycle or it may relinquish the I-Bus to another MASTER device.

II. I-Bus MASTER-to-SLAVE read cycle

When a MASTER device has access to the I-Bus, it may accomplish a memory read cycle through the following action. The IBIM asserts I-Bus GO (TLGO) and a valid ADDRESS (TLADR). All SLAVE devices will receive the GO transmitted by the MASTER. When this is done and the address is decoded as valid, the SLAVE device will begin to generate read data. In the case of a memory module, this means starting a read cycle. When READ DATA is valid, the SLAVE device must assert TERMINATE. If the address that is decoded by the SLAVE during a read cycle is that of RAM memory, then the PARITY ENABLE signal (TLPAREN) must be asserted by the SLAVE. This signal must have the same timing that READ DATA would have.

When the IBIM receives the release TERMINATE, it may begin a new cycle or it may relinquish the I-Bus to another MASTER device.

III. Priority and acquisition of I-Bus control

The three signals, I-Bus ACCESS GRANTED (TLAG), I-Bus ACKNOWLEDGED (TLAK), and I-Bus AVAILABLE (TLAV), are utilized by the IBIM. Their purpose is to schedule the next I-Bus MASTER during the data transfer operation of the present I-Bus MASTER. All IBIM's are connected cascade in order of priority. Every IBIM has an identical ACCESS CONTROLLER.

IV. I-Bus special functions

In addition to the signals associated with data transfers and I-Bus control, there are three signals with special functions. These signals are: RESET, I-Bus WAIT, and I-Bus HOLD. The IBIM does not implement the I-Bus WAIT.

RESET (TLRES) is generated by the IBIM either during a power-up sequence or during the execution of a MPM RESET instruction. It is part of the I-Bus and thus is available to all devices interfacing to the I-Bus.

The functions of TLRES are to allow an I-Bus MASTER to: (1) reset all the I-Bus devices, except itself, in response to a MPM RESET instruction; or, (2)

reset all I-Bus devices, including itself, during power turn-on. During power turn-on, TLRES will remain at ground until after all DC power voltages are stable and within regulation.

I-Bus HOLD is asserted on an IBIM when its associated MPM is executing an ABS instruction for an I-Bus operand. ABS is intended to be used as a software "interlock". ABS reads a memory word, tests it, and if negative, subtracts it from zero and stores it back in memory where it was originally read. In the use of ABS as a software interlock in multiprocessor systems, it is possible for one processor to modify a memory word while another processor is performing ABS on that same word. This interference prevents the use of ABS as a software interlock. I-Bus HOLD prevents the interference by holding I-Bus ACCESS from the MPM performing ABS.

6.2.6.3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTICS	SOURCE
Power	5VDC	350 ma	pwr sup
Logic	/TLPAREN	TTL (parity check enable)	I-bus
Logic	/TLTM	TTL (terminate command)	I-Bus
Logic	/UPMEMEN	Tri-state (valid address status)	MPM
Logic	UPSEOC	TTL (end of instruction cycle)	MPM
Logic	UPDBIN	TTL (read/write control)	MPM
Logic	/MBSTOPC	TTL (used to freeze clock)	MPM
Data	UPA(00-15)	Tri-state (address bus)	MPM

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTICS	DESTIN.
Logic	TLREAD	TTL DR (read control)	I-Bus
Logic	TLAG	TTL (I-Bus access granted)	I-Bus
Logic	/TLAK	TTL (I-Bus available)	I-Bus
Logic	/TLRES	TTL DR (RESET)	I-Bus
Logic	TLHOLD	TTL DR (I-Bus hold signal)	I-Bus
Logic	TLGO	TTL (to initiate data transfer)	I-Bus
Data	TLADR(00-15)	TTL (address lines)	I-Bus
Logic	/IBCLK	TTL (clock signal)	I-Bus
Logic	/OCPAREN	Open collector (parity enable)	I-Bus
Logic	/OCTOI	TTL DR (non-existent memory)	I-Bus
Logic	/DCRDY	Open-collector (ready signal)	I-Bus

III. Bi-directional

CATEGORY	SIGNAL NAME	CHARACTERISTICS	SOURCE/ DESTIN.
Data	TLDAT(00-15)	Tri-state (I-Bus data lines)	I-Bus
Logic	TLPAR	Tri-state (I-Bus parity)	I-Bus
Logic	TSD(00-15)	Tri-state (data Bus)	MPM
Logic	TSPAR	Tri-state (parity)	MPM

6.2.7 FLOATING POINT ARITHMETIC UNIT

6.2.7.1 GENERAL DESCRIPTION

The Floating Point Arithmetic Unit (FPAU) is a local bus device which performs IBM 360/370 formatted single precision and double precision floating point arithmetic and conversion operations. The FPAU consists of 1) a local bus interface, 2) a 60-bit mantissa Arithmetic and Logic Unit (ALU), 3) a 16-bit operand monitor and exponent ALU, and 4) a control unit.

6.2.7.2 FUNCTIONAL DESCRIPTION (refer to Figures 6.2.7-1 and 2)

The FPAU is a local bus device that is capable of processing floating point instructions fetched by the processor controlling the local bus. The FPAU is activated by storing the workspace pointer at a fixed memory address (hex FBFE). After successful completion of this instruction the FPAU monitors the local bus control signals. When an Instruction Acquisition (IAQ) cycle occurs the FPAU copies the address bus value into the FPAU Program Counter (PC) and the data bus value in the FPAU Instruction Register (IR). The occurrence of the first floating point operation code in the IR causes the FPAU to begin operation. The FPAU then exerts HOLD, stopping the microprocessor operation, and the FPAU reads the accumulator value which is stored in Workspace 0 through 3 (WS0-WS3).

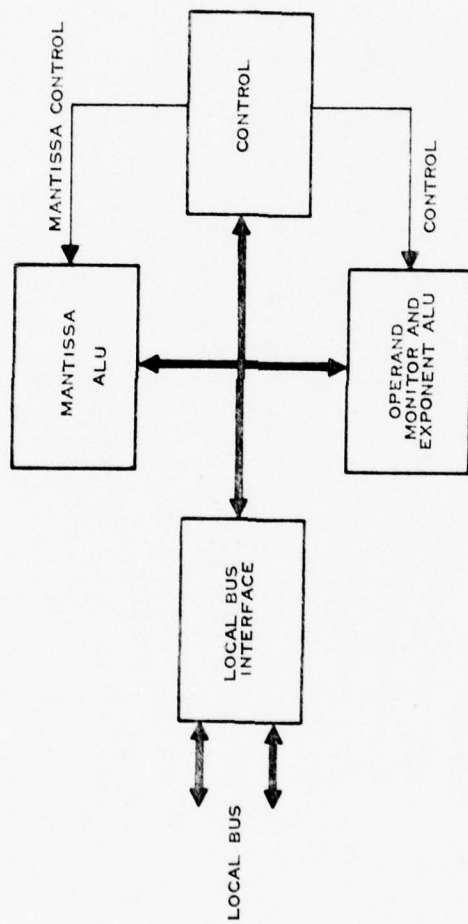


Figure 6.2.7-1. Simplified Block Diagram Floating Point Arithmetic Unit

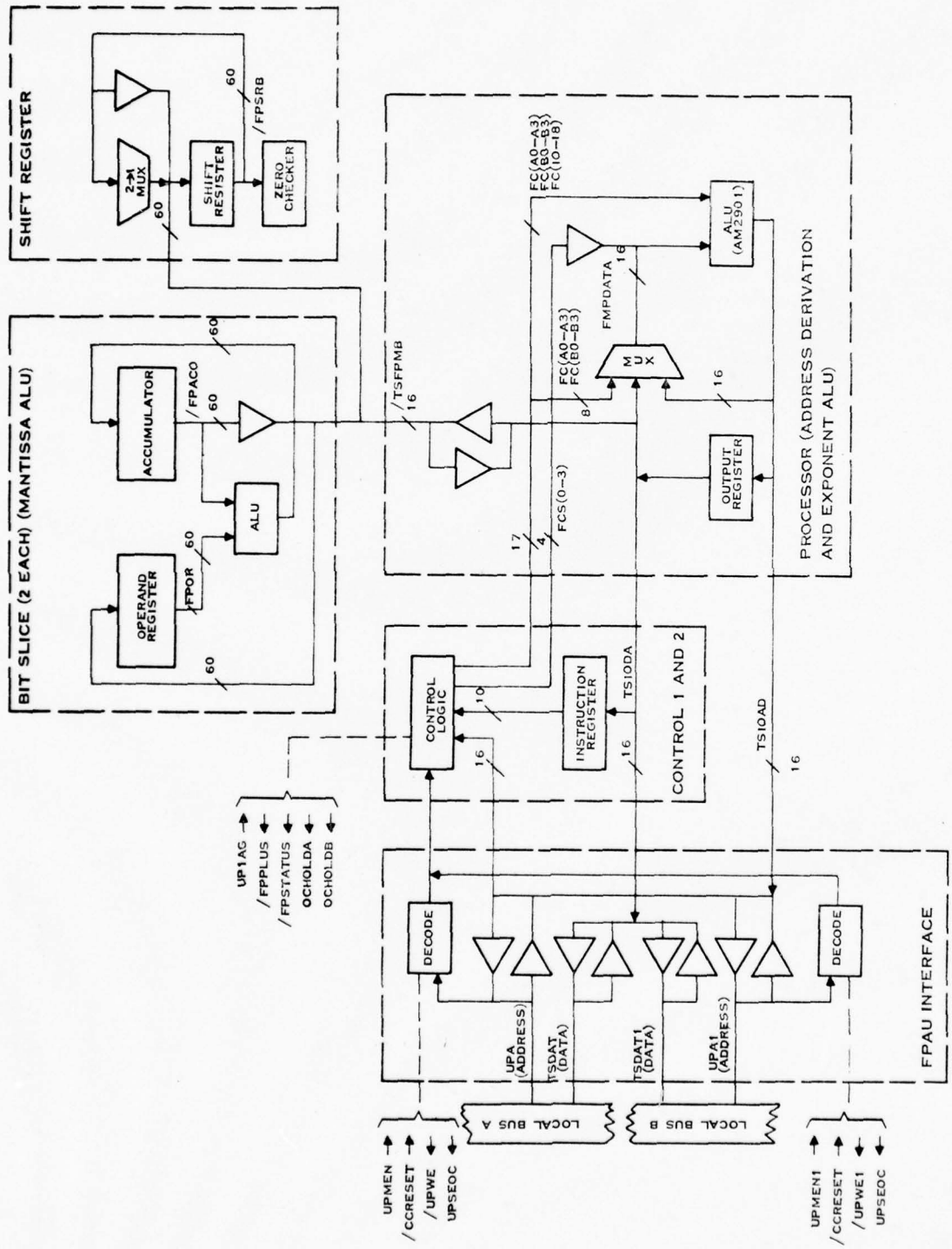


Figure 6.2.7-2. Function Block Diagram Floating Point Arithmetic Unit

The FPAU then interprets the address mode bits of the IR and performs the operand derivation for the source operand. The source operand is read and the floating point instruction is performed. The HOLD signal is released and the microprocessor processes the instruction as a NO-OP. The microprocessor then fetches the next instruction. Succeeding floating point instructions perform similar operand derivation and instruction execution steps. The occurrence of the first non-floating point instruction causes the FPAU to exert the HOLD signal and restore the accumulator value to W50-W53. The execution of an XIT instruction causes the FPAU to restore the accumulator and to stop monitoring the local bus.

The status of the floating point unit is available to be read by the microprocessor at the conclusion of each instruction. The format of the FPAU status word is shown in Table 6.2.7-1. The word is accessed by performing a read at memory address hex FBFE. A floating point status interrupt is generated each time one of these bits is set. (By altering the motherboard wiring, the FPAU will not generate this interrupt when an XIT instruction occurs.) On an overflow condition, a floating point status interrupt is generated. On an underflow condition, the FPAU will also set the accumulator to true zero. Table 6.2.7-2 illustrates the conditions that cause overflow and underflow.

TABLE 6.2.7-1 Floating Point Status Format

Bit 0 - Completion Bit

If 1 indicates that an XIT instruction has been performed.

Bit 1 - Overflow

Arithmetic Operation - If 1 indicates that the result of the operation cannot be represented as a valid floating point quantity.

Fix Operation - If 1 indicates that the result cannot be represented as a valid fixed point number.

Bit 2 - Underflow

If 1 indicates that the result of the operation cannot be represented as a valid floating point number.

Bit 3 - Illegal Op-Code

If 1 indicates that an illegal floating point Op-Code was encountered.

TABLE 6.2.7-2 Overflow and Underflow Conditions

Operation	Overflow Conditions	Underflow Conditions
Addition/ Subtraction	If Characteristic of Result > 127	If Characteristic of Result is Negative
Fix	For 16 Bit Result Input > 32767 32 Bit Result Input > 2147483647	None
Multiply	If Characteristic of Result > 127	If Characteristic of Result is Negative
Divide	Division by 0 if Result Characteristic > 127	If Characteristic of Result is Negative
Float	None	None

The FPAU performs the following functions:

I. Single precision (32-bit) functions

Add (Accumulator + Source Operand)
Subtract (Accumulator - Source Operand)
Multiply (Accumulator X Source Operand)
Divide (Accumulator / Source Operand)
Negate (-Accumulator)

II. Double precision (64-bit) functions

Add (Accumulator + Source Operand)
Subtract (Accumulator - Source Operand)
Multiply (Accumulator X Source Operand)
Divide (Accumulator / Source Operand)
Negate (-Accumulator)

III. All fixed point to floating point conversions

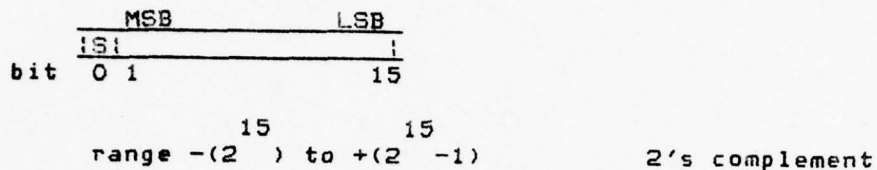
Integer to single precision
Integer to double precision
Extended integer to single precision
Extended integer to double precision

IV. All floating point to fixed point conversions

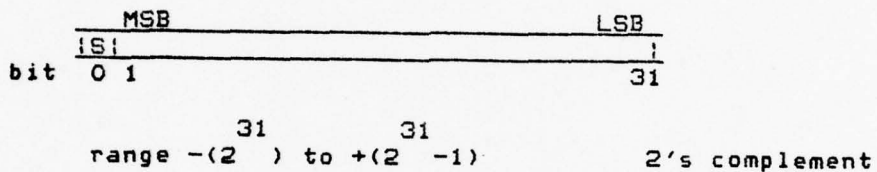
Single precision to integer
Single precision to extended integer
Double precision to integer
Double precision to extended integer

The FPAU fixed and floating point formats are:

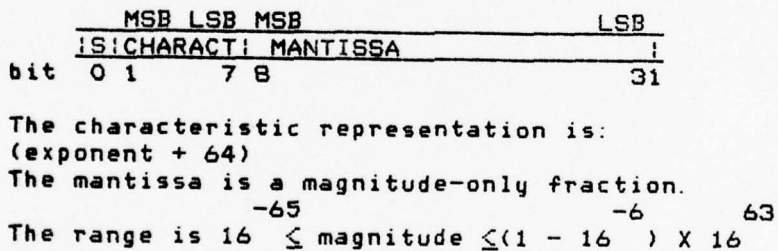
I. 16-bit Fixed Point Format



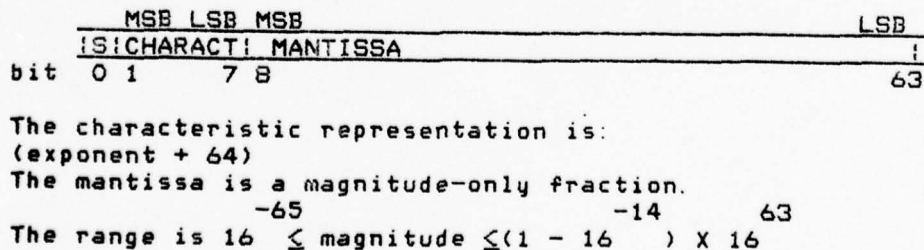
II. 32-bit Fixed Point Format



III. 32-bit Floating Point Format



IV. 64-bit Floating Point Format



The circuitry to perform the afore described operations is contained on 7 printed circuit boards: 1) FPAU Interface, 2) Control 1, 3) Control 2, 4) Processor, 5), 6) two Bit slice and 7) Shift Register. (Refer to Figure 6.2.7-1 and 2.)

I. FPAU Interface

The FPAU Interface board (FI) contains the circuitry to communicate with two external processors, one at a time. The communications circuitry consists of tri-state buffers used for transmitting and receiving the local bus data and address. Logic for decoding the address information is also contained on the FI board. Figure 4.2-1 shows the FPAU in a two processor arrangement.

II. Control 1 and Control 2

The FPAU Control boards (FC) contain the logic circuitry (mostly PROMS) to interpret FPAU operations and to exert HOLD on the external processor while a FPAU operation is in progress.

III. Processor

The Processor board (FM) has the circuitry for address derivation and exponent ALU. In order to perform this function the board contains: 1) the circuitry for a multiplex selection of the address,

data, and process control information, 2) an AM2901 ALU, and 3) an output register. The Processor also contains bi-directional drivers for communicating with the BIT Slice boards and enabling data onto the tri-state data bus.

IV. BIT Slice

The two BIT Slice boards combined perform the arithmetic operations on the mantissa. Each board has a 32-bit capability in order to form one 60-bit word with minimal round off error. For performing these operations each BIT Slice board contains an ALU, an operand register, and an accumulator. The ALU uses the information in both of these registers to perform the floating point operation with the accumulated results being stored back in the accumulator. The tri-state drivers transmit the results to both the Processor and the Shift Register boards.

V. Shift Register

The Shift Register board performs shift operations. The circuitry to perform these operations consists of a bi-directional shift register feeding a zero checker, a tri-state driver, and a 2-to-1 multiplexer. The zero checker checks the data word for zero. The 2-to-1 multiplexer left or right shifts the data in four-bit increments for normalization or

positional adjustments for arithmetic operations. The tri-state output of the multiplexer also feeds the data bus. The output the the tri-state driver feeds the data bus when strobed on.

6.2.7.3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE
Power	5VDC	5.5 a	pwr sup
Logic	/HOLDA	TTL (to indicate hold acknowledge)	MPM
Logic	/CRRESET	TTL (50-nsec min low reset pulse)	CRIM
Logic	UPIAQ	Tri-State (instruction acquisition)	MPM
Logic	/UPENDCY	TTL (to indicate end of cycle)	MPM

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTIN.
Logic	/OCRDY	Open-collector (to terminate transfer)	MPM
Logic	/OCHOLD	Open-collector (to suspend processor operation)	MPM
Logic	/UPWE	Open-collector (write strobe)	MPM
Logic	UPSEOC	Tri-state (to indicate end of cycle)	MPM
Logic	/FPPLUS	Tri-State (PC increment interrupt (Interrupt level 4))	CRIM
Logic	/FPSTATUS	Tri-State (FP status interrupt (Interrupt level 5))	CRIM

III. Bi-directional

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE/ DESTIN.
Data	UPA(00-14)	Tri-State (address)	MPM
Data	TSDAT(00-15)	Tri-State (data)	MPM
Logic	/UPMEMEN	Tri-State (to initiate a memory transfer)	MPM
Logic	UPDBIN	Tri-State (memory read control)	MPM

6.2.8 DATA BUS EXTENDER MODULE

6.2.8.1 GENERAL DESCRIPTION

The Data Bus Extender Module (DBEM) contains the circuitry necessary to interface with another DBEM, via an interconnecting cable, in order to perform a functional coupling of one Internal Data Bus (I-Bus) to a second I-Bus. The DBEM shall also be capable of interfacing to a TILINE* coupler instead of a second DBEM, using the same interconnecting cable interface as for the second DBEM, in order to perform a functional coupling of an I-Bus to a TILINE. In order to effect either coupling, the DBEM shall function as both an I-Bus MASTER device and an I-Bus SLAVE device on the I-Bus to which it is connected.

* TILINE is a registered trademark of Texas Instruments Incorporated.

6.2.8.2 FUNCTIONAL DESCRIPTION (refer to Figure 6.2.8-1)

A typical operation utilizing two DBEMS would be communication between a MPM and memory on an external bus or TILINE. (Refer to Figures 6.2.8-2, 3, and 4). In order to communicate with external memory the operation transverses through DBEM 1 and DBEM 2. In this case address decode logic of DBEM 1 determines that the memory address is within its bounds, sets itself up as a SLAVE device, and sends the control logic to DBEM 2 for it to become a MASTER device.

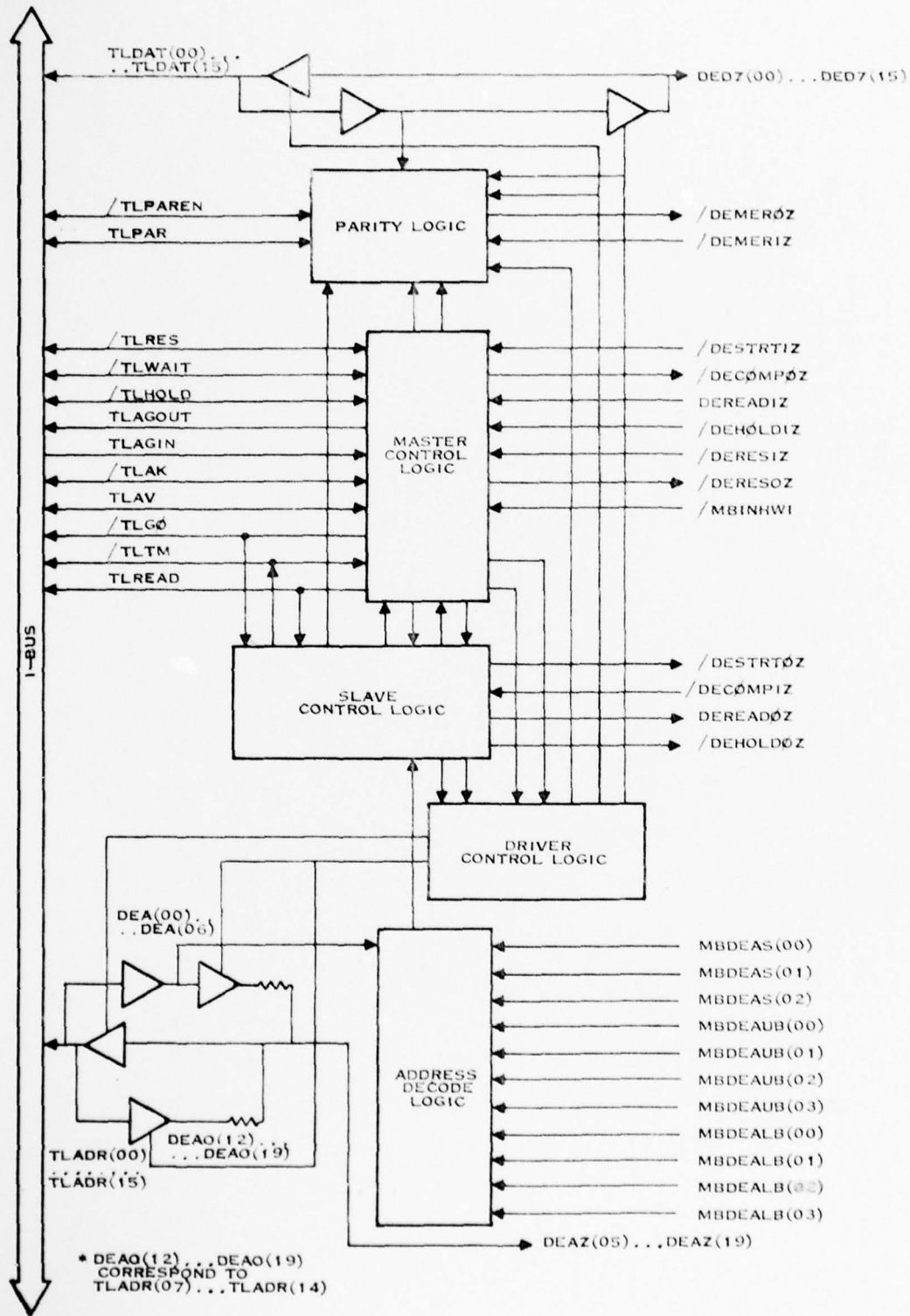


Figure 6.2.8-1. Functional Block Diagram Data Bus Extender Module

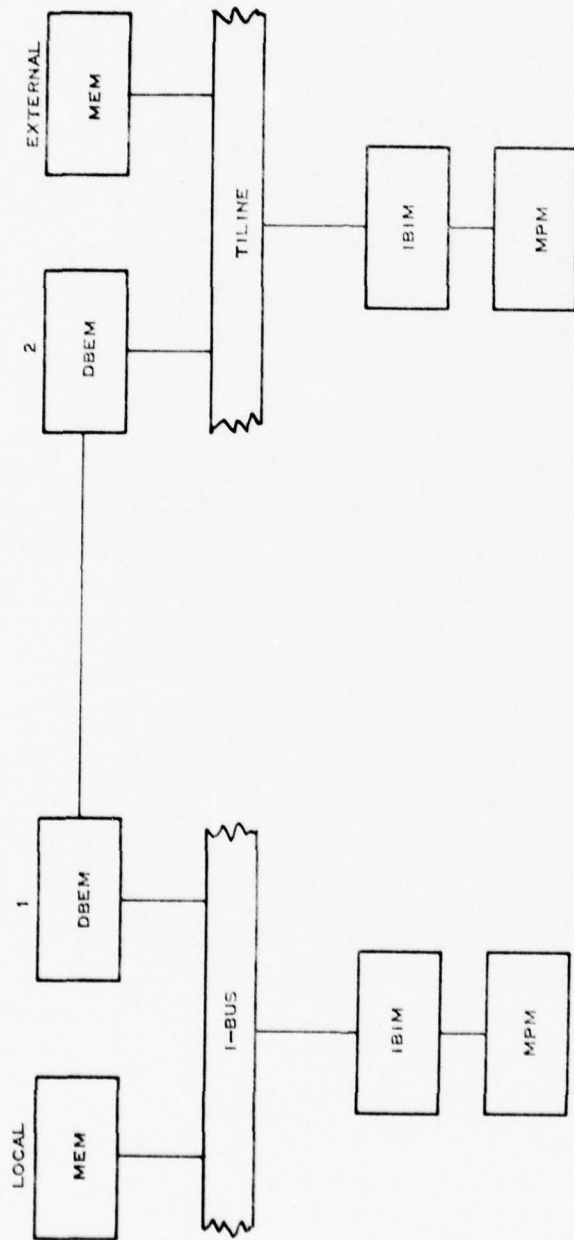


Figure 6.2.8-2. Simplified Block Diagram of Typical Data Bus Extender Module Application

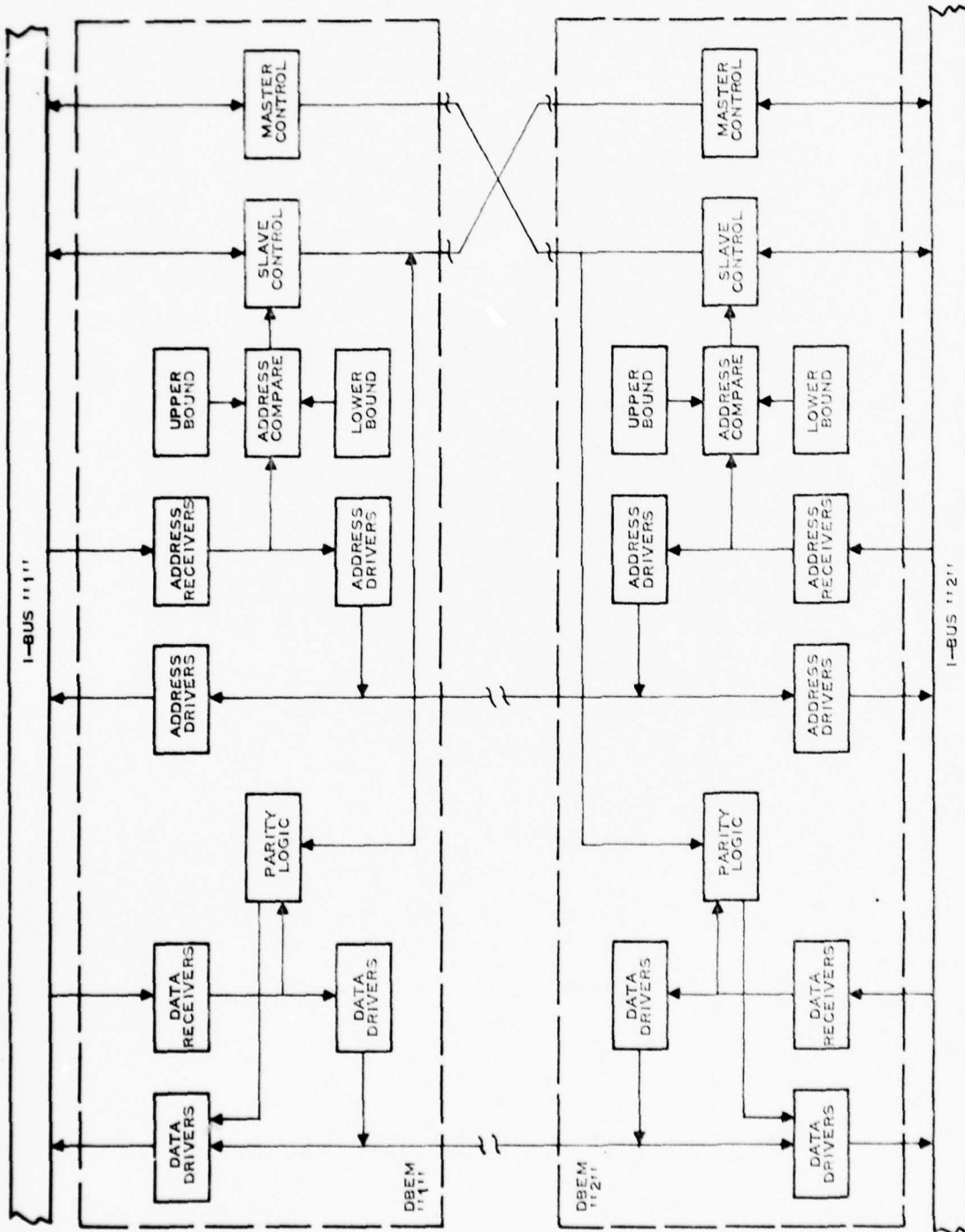


Figure 6.2.8-3. DBEM to DBEM Configuration

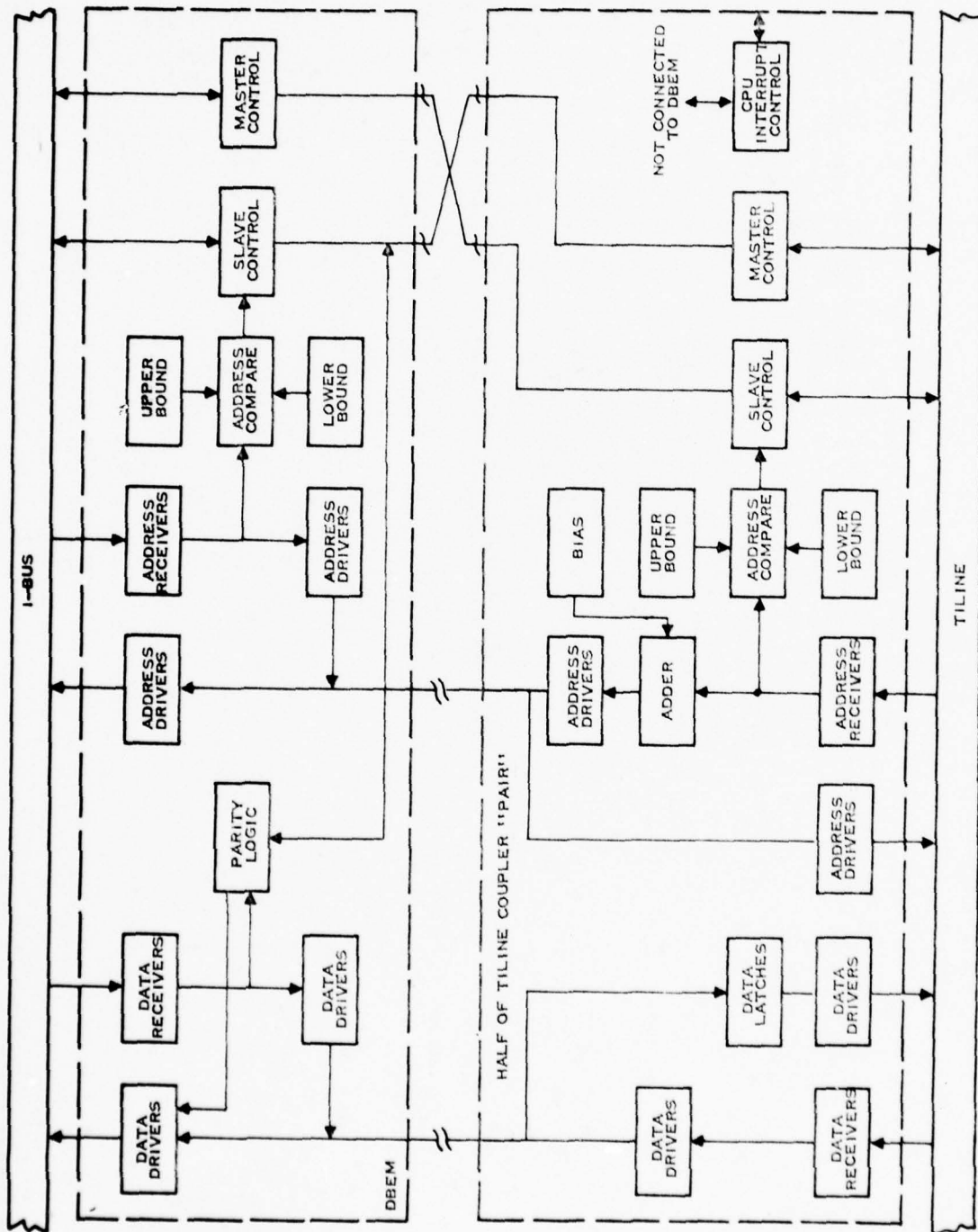


Figure 6.2.8-4. DBEM to TILINE Configuration

DBEM 2 vies with the external MPM for time to read or write to the external memory. For an operation in the opposite direction the DBEM's would reverse roles. A DBEM may assume the role of either a MASTER or a SLAVE whichever the situation dictates.

The circuitry to perform the aforementioned functions is divided into 6 logical sections for description: 1) address decode logic, 2) SLAVE control logic, 3) MASTER control logic, 4) driver control logic, 5) parity logic, and 6) address and data signal buffers.

I. Address Decode Logic

This circuitry decodes I-Bus addresses to determine whether the DBEM should initiate a cycle through the remote device for that address. MBDEAS(00) through MBDEAS(02) determines which segment of the I-Bus address space applies to the DBEM. MBDEAUB(00) through MBDEAUB(03) determines the upper bound within the selected segment of the I-Bus address space for which the DBEM is a SLAVE device. MBDEALB(00) through MBDEALB(03) determines the lower bound within the selected segment of the I-Bus address space for which the DBEM is a SLAVE device. These lines are set (i. e. hardwired) by the mother board.

(Refer to tables 6.2.8-1 through 3 for decode logic.)

TABLE 6.2-8-1. REMOVE CYCLE ADDRESS UPPER BOUND DETERMINATION

CONNECTIONS				Largest (Highest) Address That Will Cause the DBEM to Initiate A Cycle Through the Remote Device					
MBDEAUB(00)	MBDEAUB(01)	MBDEAUB(02)	MBDEAUB(03)	TLADR(03)	TLADR(04)	TLADR(05)	TLADR(06)		
GND	GND	GND	GND	0	0	0	0		
GND	GND	GND	NC	0	0	0	1		
GND	NC	NC	NC	0	0	1	0		
GND	NC	GND	GND	0	0	1	1		
GND	NC	GND	NC	0	1	0	0		
GND	NC	NC	GND	0	1	0	1		
GND	NC	NC	NC	0	1	1	0		
NC	GND	GND	GND	0	1	1	1		
NC	GND	GND	NC	1	0	0	0		
NC	GND	NC	GND	1	0	0	1		
NC	GND	NC	NC	1	0	1	0		
NC	NC	GND	GND	1	0	1	1		
NC	NC	GND	NC	1	1	0	0		
NC	NC	NC	GND	1	1	0	1		
NC	NC	NC	NC	1	1	1	0		
NC	NC	NC	NC	1	1	1	1		
NC	NC	NC	NC	1	1	1	0		

NOTES

- (1) GND means signal is ground on motherboard.
- (2) NC means No-Connection.
- (3) When MBDEAUB(00) thru MBDEAUB(03) are all grounded, the DBEM will not initiate cycles through the remote device.

TABLE 6.2.8-2. REMOTE CYCLE ADDRESS LOWER ROUND DETERMINATION

CONNECTIONS				Smallest (Lowest) Address That Will Cause the DBEM to Initiate A Cycle Through the Remote Device					
MBDEAUB (00)	MBDEAUB (01)	MBDEAUB (02)	MBDEAUB (03)	TLADR (03)	TLADR (04)	TLADR (05)	TLADR (06)	TLADR (06)	
GND	GND	GND	GND	0	0	0	0	0	
GND	GND	GND	NC	0	0	0	0	1	
GND	GND	NC	GND	0	0	1	1	0	
GND	GND	NC	NC	0	0	1	1	1	
GND	NC	GND	GND	0	1	0	0	0	
GND	NC	GND	NC	0	1	0	1	1	
GND	NC	NC	GND	0	1	1	1	0	
GND	NC	NC	NC	0	1	1	1	1	
NC	GND	GND	GND	1	0	0	0	0	
NC	GND	GND	NC	1	0	0	0	1	
NC	GND	NC	GND	1	0	1	1	0	
NC	GND	NC	NC	1	0	1	1	1	
NC	NC	GND	GND	1	1	0	0	0	
NC	NC	GND	NC	1	1	0	0	1	
NC	NC	NC	GND	1	1	1	1	0	
NC	NC	NC	NC	1	1	1	1	1	

NOTES

- (1) GND means signal is grounded on motherboard
- (2) NC means No-Connection

TABLE 6.2.8-3. REMOTE CYCLE ADDRESS SEGMENT DETERMINATION

CONNECTIONS			Segment of I-Bus Address Space for which Upper and Lower Bounds Apply		
MBDEAUB (00)	MBDEAUB (01)	MBDEAUB (02)	TLADR (00)	TLADR (01)	TLADR (02)
GND	GND	GND	0	0	0
GND	GND	NC	0	0	1
GND	NC	GND	0	1	0
GND	NC	NC	0	1	1
NC	GND	GND	1	0	0
NC	GND	NC	1	0	1
NC	NC	GND	1	1	0
NC	NC	NC	1	1	1

NOTES

- (1) GND means signal is grounded on motherboard
- (2) NC means No-Connection

II. SLAVE Control Logic

This circuitry performs the functions of an I-Bus SLAVE device as required to initiate cycles through the remote device for addresses designated by the address decode logic.

III. MASTER Control Logic

This circuitry performs the functions of an I-Bus MASTER device as required to perform cycles with a remote device. This circuitry also uses /MBINHVI to prevent simultaneous bilateral cycles between the DBEM and its remote device. /MBINHVI is grounded at only one device (either the "local" DBEM or the remote device) and is left not connected at the other device. The DBEM (or TILINE Coupler, in similar manner) for which /MBINHVI is left not-connected prevents simultaneous bilateral cycles.

IV. Driver Control Logic

This circuitry controls the address and data bus drivers in response to the SLAVE control logic and the MASTER control logic.

V. Parity Logic

This circuitry: (1) generates an odd-ones parity bit when the DBEM is performing a write cycle as a remote device, (2) checks the odd-ones parity bit on

the I-Bus, and generates /DEMERGZ if the parity bit is incorrect, if /TLPAREN is active when the DBEM is performing a write cycle as a remote device, and (3) generate an odd-ones parity (valid parity when /DEMERIZ is not active, invalid parity when /DEMERIZ is active) when the DBEM is initiating a read cycle through the remote device.

VI. Address and Data Signal Buffers

This circuitry buffers and drives all data, parity, and address lines.

6.2.8.3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE
Power	MBP5V	5 VDC 540 ma	pwir sup
Logic	/DEMERIZ	TTL (memory error for data read)	remote device
Logic	/DESTRTIZ	TTL (to indicate that the DBEM is to initiate a memory cycle on the local I-Bus)	remote device
Logic	DEREADIZ	TTL (read/write)	remote device
Logic	/DEHOLDIZ	TTL (to indicate that the I-Bus access is to be retained once acquired)	remote device
Logic	/DERESIZ	TTL (250nsec(min.) asynchronous low reset pulse)	remote device
Logic	/DECOMPIZ	TTL (to indicate that the memory cycle initiated by DBEM is complete)	remote device
Logic	/MBINHUI	Level (left open or grounded to differentiate DBEMS)	mother brd
Logic	/MBDEAS	Level (left open or grounded to determine which segment of I-Bus address applies to DBEM)	mother brd
Logic	/MBDEAUB	Level (left open or grounded to determine code upper bound)	mother brd
Logic	/MBDEALB	Level (left open or grounded to determine code lower bound)	mother brd
Logic	TLAGIN	TTL (MASTER priority)	TILINE

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTINATION
Logic	/DEMOROZ	TTL (to indicate parity error was detected for the data read from the local I-Bus)	remote device
Logic	/DECOMPOZ	TTL (to indicate that the memory cycle initiated by the remote device is complete)	remote device
Logic	/DERESOZ	TTL (250 nsec(min) asynchronous low reset pulse)	remote device
Logic	/DESTRTOZ	TTL (to indicate that the remote device is to initiate a memory cycle on the remote bus)	remote device
Logic	/DEREADOZ	TTL (read/write control)	remote device
Logic	/DEHOLDOZ	TTL (to indicate that the remote bus is to be retained once acquired)	remote device
Logic	/TLGO	*DR (to initiate a data transfer from MASTER to SLAVE)	I-Bus

*DR = greater than 3.2 volts high / less than 1.5 volts low

III. Bi-directional

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE/ DESTINATION
Data	DEDZ(00) through DEDZ(15)	Tri-state TTL (data)	remote device
Data	DEAZ(05) through DEAZ(19)	Tri-state TTL (address)	remote device
Data	TLDAT(00) through TLDAT(15)	Tri-state TTL (data)	I-Bus
Data	TLADR(00) through TLADR(15)	Tri-state TTL (address)	I-Bus
Logic	/TLTM	DR (SLAVE to master, to terminate data transfer)	I-Bus
Logic	TLPAR	Tri-state TTL (parity)	I-Bus
Logic	/TLPAREN	Tri-state TTL (SLAVE to MASTER, parity enable)	I-Bus
Logic	TLREAD	DR (MASTER to SLAVE read/write control)	I-Bus
Logic	/TLAK	DR (MASTER to MASTER, to acknowledge access granted)	I-Bus
Logic	TLAV	DR (MASTER to MASTER, to show TILINE available)	I-Bus
Logic	/TLRES	Tri-state TTL (reset from MASTER)	I-Bus
Logic	/TLWAIT	DR (TILINE wait from TILINE couplers)	I-Bus
Logic	/TLHOLD	DR (TILINE hold from a MASTER to all other MASTERS)	I-Bus

6.2.9 NAVIGATION LRU POWER SUPPLY

6.2.9.1 GENERAL DESCRIPTION

A DC-to-DC converter provides the power requirements for the NAV-LRU. The power supply also provides internal monitoring of all supply voltages for operation within specified tolerances.

6.2.9.2 FUNCTIONAL DESCRIPTION

The NAV-LRU power supply (Figure 6.2.9-1) consists of seven sections: two switching regulators, two DC-to-DC converters, a precision reference, an over/under voltage (OV/UV) monitor, and an over temperature monitor.

I. Switching regulators

28 VDC primary power for the NAV-LRU is regulated by one of two switching regulators operating at 40KHz.

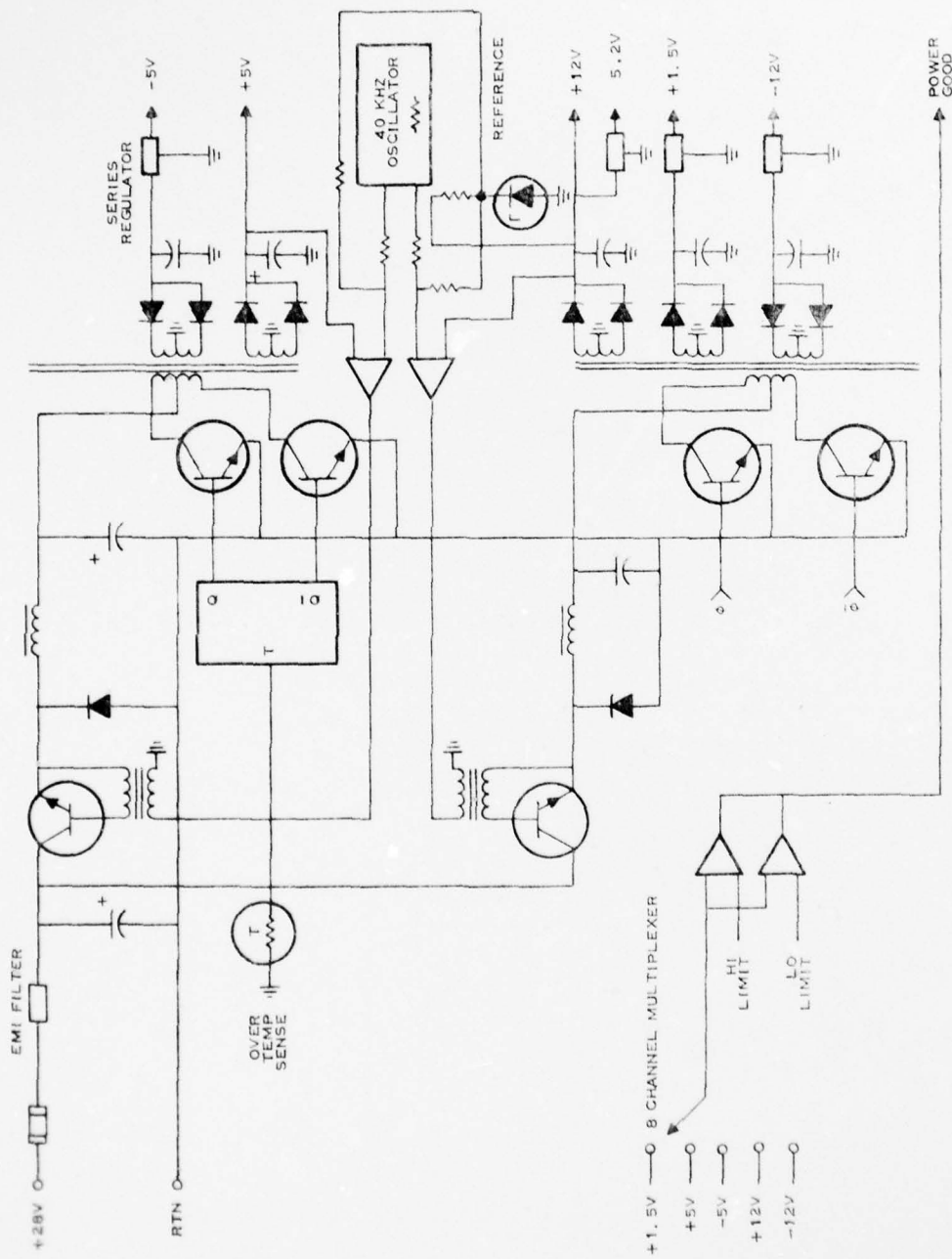


Figure 6.2.9-1. Functional Diagram Navigation LRU Power Supply

II. DC-to-DC converters

The output of each regulator supplies the primary of a DC-to-DC converter switching at a 20-KHz rate. These transformers provide multiple secondaries which are rectified and filtered to provide the voltages required by the NAV-LRU. The two highest current loads occur on the +5-volt and +12-volt outputs. One of these voltages is assigned to each converter. These outputs are sensed, compared to a reference, and feedback generated for the switching regulator which supplies the respective converter primary, thus providing closed loop control.

All other voltages roughly track the primary regulation as a result of the transformer winding ratios. These windings are designed to supply approximately 3 volts above the desired output voltage at the rectifier/filter node. A regulator then drops each voltage to the desired output level while providing increased regulation and ripple filtering at relatively high efficiency due to the small regulator drop and lower current requirements.

III. Precision reference

The +12-volt supply provides power to a precision reference supply which establishes the desired level of the output voltages and the OV/UV sense levels. This section contains a 40-KHz sawtooth oscillator which

modulates a +5-volt reference point. Comparators sensing the +5-volt output and the +12-volt output (divided down to +5 volts) produce duty cycle modulated pulse trains with ON times proportional to the error between the reference and the output. These waveforms are coupled back to control the duty cycle of the switching regulators. By this means the nominal 15-volt transformer primary voltage is adjusted to correct for input voltage and output load variations.

IV. Over/under voltage monitor

The OV/UV detection is provided by a window detector whose inputs are multiplexed to sample the output voltages in sequence. The level of each input is shifted to +5 volts by either a resistive divider or an inverting amplifier (negative voltages). These are sampled by an eight-channel analog multiplexer which is driven by a counter clocking at the sample rate of the window detector. Any output voltage which deviates more than ± 5.3 percent from its specified value causes the window detector to generate an out-of-limit condition by causing the power good signal to go to a logical "0". Scanning is performed on a continuous basis.

V. Over-temperature monitor

In the event the internal temperature of the

NAV-LRU exceeds a preset threshold, power is removed from the switching regulators. This condition remains until the temperature returns to the safe operating limits of the NAV-LRU components.

6.2.9.3 ELECTRICAL INTERFACE

I. Inputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	SOURCE
Power	28 VDC	5.8 a	AC/DC Conv

II. Outputs

CATEGORY	SIGNAL NAME	CHARACTERISTIC	DESTINATION
Power	1.5 VDC	1.0 a	NAV-LRU
Power	5 VDC	16 a	NAV-LRU
Power	-5 VDC	750 ma	NAV-LRU
Power	12 VDC	1.0 a	NAV-LRU
Power	-12 VDC	160 ma	NAV-LRU
Logic	PSPG	TTI (power supply good)	CRIM