ACROAMATICS DOCUMENT HISTORY

The following table indicates the revision level of *Technical Manual - 500V PCM/PAM Format Simulator*, Acroamatics Document Number 6000240, released on June 5, 2000, and contains a record of all revisions made since that date.

DN6000240 CHANGE HISTORY			
Rev	**Date**	**Action**	**Name**
	6-5-00	Initial Issue	DJM

TECHNICAL MANUAL
MODEL 500V
• PCM/PAM FORMAT SIMULATOR •

TABLE OF CONTENTS

TECHNICAL MANUAL
MODEL 500V
• PCM/PAM FORMAT SIMULATOR •

SECTION 1
INTRODUCTION

1.1 DESCRIPTION

The Model 500V PCM/PAM Format Simulator is a stand alone, VME compatible, 6U board that simulates PCM and PAM telemetry data streams. The simulator can generate complex PCM/PAM formats which contain dynamically varying data, as well as multiple subframe structures. The simulator is software configured from a VME Host computer, and allows you to define your formats with a minimal amount of description. Default entries are loaded into the data registers and program memory when certain values or functions are not specified. The unit provides numerous static and dynamic data sources for a flexible format generation.

PCM/PAM simulation formats are assembled by specifying data words and their output sequence. Format descriptions are written in a proprietary setup language, and are compiled from either data files or keyboard entry. The compiler uses ASCII input text to create the format image in the simulator's program memory, where it is executed when the system is placed in run mode. A description of the format is displayed for verification after the format has been compiled. The compiler is written in C programming language for portability. The programming language is compatible with the existing setup software used in the Acroamatics 2110 Series Telemetry Data Processors.

1.1.1 Format Parameters

The Model 500V provides 1024 static and eight dynamic data sources. The static data sources are programmed to fixed values. The dynamic data sources include two up/down modulo counters, two user-defined 2K data memories, two external inputs, a pseudo-random generator, and a program counter. You can use the modulo counters for ramp data or for ID subframe synchronization. The counters are presettable with programmable up and down limits, and increment or decrement upon each output or frame recycle. Dynamic data memories can be initialized to either built-in ramp, sine, triangle, or square wave functions, or to user-defined data. You can select the simulator to update dynamic data at each output or at a specified rate. Other data sources include two external 16-bit inputs. The external inputs allow remote data to be merged into the output stream. The remote input may be from a switch panel or a computer interface. Finally, you can select the program counter as a data source to generate unique values for all words in the format.

Each format word is programmed with individual attributes. These attributes include word length, parity generation, and word orientation. Static data words can range from 1 to 32 bits in length, and dynamic data words from 1 to 16 bits. Each word can have leading, trailing, or no parity status attached to itself. You can output all words in LSB or MSB orientation.

You can define the following parameters in a simulation program:

- Data Rate
- Output Data Code Type
- Parity
- 1024 user defined Static Data Words
- Two user controllable Counters
- Two Dynamic Data Memories
- Dynamic Data generation Rate
- Pseudo Random data
- Two External input words
- Program counter

You can specify data rates from 1Hz to 25MHz, and the following sixteen selectable PCM output codes are available:

- Miller Squared-S
- Miller Squared-M
- Delay Modulation-S
- Delay Modulation-M
- Bi-Phase-S
- Bi-Phase-M
- Bi-Phase-L
- Differential Bi-Phase-S
- Differential Bi-Phase-M
- Randomized NRZ-23
- Randomized NRZ-17
- Randomized NRZ-15
- Randomized NRZ-11
- NRZ-S
- NRZ-M
- NRZ-L

1.1.2 Mainframe & Subframe Format Description

Mainframe and subframe format descriptions define the order in which the card generates your simulated data. The simulator contains storage for two complete formats, enabling you to insert or switch from one format to the other during the simulation. You specify which program memory (A or B) is being loaded.

The mainframe format is constructed from data sources which are referenced to word positions in the frame. You can specify the length of the mainframe to a maximum of 4096 words. If the length is not specified, it is assumed to be the last word referenced in the frame description. Pseudo-random data or programmable fill data is automatically inserted in word positions not defined by your format.

The simulator supports up to three subframes. Subframes are declared within the mainframe or subframe format description. The subframes are constructed from static or dynamic data sources, and are referenced in the same manner as other data sources in either the mainframe or subframe. You can assign a different data source for each reference to a subframe word. There are two methods in which to describe a subframe format. The first method defines the number of words in the subframe and loads a continuous stream of data sources corresponding to the content of the frame. The second method describes the subframe in a matrix fashion defined by the number references made to the subframe and the depth of subframe.

1.1.3 Modes of Operation

The simulator has several modes of operation which you can specify from either compiler setup files or keyboard entry. The modes of operation are: Stop, Run A, Run B, and Ins. In Stop mode, the simulator output is disabled, awaiting a Run command. For Run mode, you specify which format you want generated (A or B). In Run A (B) mode the simulator executes the machine code loaded into program memory A (B), and generates the simulated output for format A (B). The Ins mode enables you to insert a frame of data from the alternate format into the output stream.

1.2 DOCUMENT CONVENTIONS

In this document register addresses and address offsets are hexadecimal numbers. Where it is necessary to refer to a hexadecimal number in the text, we use the C programming convention *0xNN* to refer to hexadecimal number *NN*. Bits in a register are numbered in decimal. The term *Device n* refers to an address destination on the TDP system A-Bus. The A-bus is the output data bus, and has eight possible destination devices. Although we do not use all eight, those destinations we do use are dedicated to specific functions in the TDP system. The term *DIT* refers to a data message in the TDP system. It stands for *Data, ID, and Time* , the three components of a TDP data message. We can label a DIT by the value of its ID tag, for example “DIT 0xFFF1,” the DIT that conveys the once per millisecond value of a time message. Frequently we use a functional label instead, for example, “the MILLISECOND DIT.”

MODEL 500V PCM/PAM FORMAT SIMULATOR SPECIFICATIONS

FUNCTION	CHARACTERISTICS
Bit Rate	1Hz to 25MHz, Tunable to 0.1% of programmed rate
Data Sources	1024 Static Registers Two User-Defined Dynamic Data Memories Two 16-bit Modulo Up/Down Counters Two 16-bit External Inputs 16-bit Pseudo-Random Generator 16-bit Program Counter
Word Lengths	Programmable for each data source Static data words range from 1 to 32 bits All other data sources range from 1 to 16 bits
Word Orientation	Programmable MSB/LSB for each data word
Parity Generation	Selectable odd/even, leading, trailing or no parity for each data word
Dynamic Data Memory	2K RAM, Presettable to ramp, sine, triangle or square wave functions, or user-defined input. Selectable data type: 1's Comp., 2's Comp., Sign Mag. and Off. Binary. Programmable time base.
CAPABILITY	
Format Storage	Stores two complete, selectable PCM formats Performs asynchronous frame insertion and format switching
Subframe Capability	Generates up to three subframes within the mainframe Subframe can be generated within a subframe
Frame Length	Maximum of 4096 words for each mainframe and subframe
OUTPUT	
Clock	0° Clock, TTL Compatible Two outputs are available from front panel Six outputs are available from P2 Connector
Data	NRZ-L Data, TTL Compatible Two outputs are available from front panel Six outputs are available from P2 Connector
PAM Output	±2.5 Volts balanced output, 10mA drive current
PCM Output	±2 Volts balanced output, 40mA drive current Two outputs are available from front panel Four outputs are available from P2 Connector
Output Code Type	Sixteen selectable output codes: NRZ-L/M/S, Biϕ-L/M/S, DBiϕ-M/S, DM-M/S, MDM-M/S, RNRZ 11, 15, 17 and 23
REQUIREMENTS	
Power	+5VDC at 1 Amp, +12VDC at 300 mA, –12VDC at 100 mA
Temperature	Operating: 0 to +40°C, Non-operating: –40 to +86°C
Relative Humidity	Up to 90% non-condensing
Air Flow	30 Linear FPM
Shock	Operating 6G, Non-operating 50G
Vibration	Operating 0.5G, 5 to 2000 Hz, Non-op 1.2G, 5 to 500 Hz

Specifications are subject to change without notice.

SECTION 2
INSTALLATION

2.1 GENERAL

This section contains installation information for the Acroamatics Model 500V PCM/PAM Format Simulator (FSIM). Your card part number may be either 6011500 or 6011500-10.

2.2 UNPACKING

Using proper ESD-protection procedures, open the cardboard shipping container and remove the card from the anti-static bag. Retain the container, anti-static bag, and foam packaging material for use if you must return the card.

2.3 FACTORY RETURN

When you return a card to the factory for repair or modification, include as much information as possible describing the failure mode or the modification/update you want.

Pack the card for shipment by wrapping it in the anti-static bag. Place the card into the shipping container, protecting it with the foam packing, and secure the container with reinforced tape. Provide the name and phone number of a technical contact we can talk to regarding the card.

> *Call Acroamatics at (805) 967-9909 to get a RMA number before returning any equipment to the factory, and include the RMA number in any correspondence or shipments to Acroamatics.*

2.4 INSTALLING

The FSIM card mounts in a standard VMEbus chassis. Mounting dimensions are shown in the assembly Drawing in Section 6 of this manual. Slide the card into one of your system VME chassis slots and seat the card firmly by pressing against the ears. Make the front panel cable connections appropriate to your system. Remove the board by pulling firmly on the outside of the ears. For connector locations see Figure 2-1.

2.5 CONNECTORS

The following pages contain tables of information on all the connections into and out of the FSIM card.

TABLE 2-1. MATING CONNECTOR LIST FOR MODEL 500VA

CONN.	FUNCTION	MATING CONNECTOR
P01	VMEbus	603-2-IEC-C096
P02	VMEbus	603-2-IEC-C096
J01	A BUS	3334-6660
J02	CLOCK OUTPUT 1	SMB 903-285P-51S
J03	DATA OUTPUT 1	SMB 903-285P-51S
J04	AUXILIARY	SMB 903-285P-51S
J05	AUXILIARY	SMB 903-285P-51S
J06	PCM CODE OUTPUT 1	SMB 903-285P-51S
J07	AUXILIARY	SMB 903-285P-51S
J08	AUXILIARY	SMB 903-285P-51S

TABLE 2-2. CONNECTOR LIST MODEL 500VA BACKPLANE CONNECTOR P01-ROW-A		
NOTE: ALL ($) SIGNALS UNUSED ON THIS CARD		
PIN	**SIGNAL**	**FUNCTION**
01	4VMED00	Data Bus 00
02	4VMED01	Data Bus 01
03	4VMED02	Data Bus 02
04	4VMED03	Data Bus 03
05	4VMED04	Data Bus 04
06	4VMED05	Data Bus 05
07	4VMED06	Data Bus 06
08	4VMED07	Data Bus 07
09	GND	Ground
10	SYSCLK ($)	System Clock
11	GND	Ground
12	9VMEDS1	Data Strobe 1
13	9VMEDS0	Data Strobe 0
14	9VMEWRT	Write
15	GND	Ground
16	9VMDACK	Data Transfer Acknowledge
17	GND	Ground
18	9VMASTB ($)	Address Strobe
19	GND	Ground
20	9VMIACK	Interrupt Acknowledge
21	9VMIAIN	Interrupt Acknowledge IN
22	9VMIAOT	Interrupt Acknowledge OUT
23	4VMAM04	Address Modifier 4
24	4VMEA07	Address Bus 07
25	4VMEA06	Address Bus 06
26	4VMEA05	Address Bus 05
27	4VMEA04	Address Bus 04
28	4VMEA03	Address Bus 03
29	4VMEA02	Address Bus 02
30	4VMEA01	Address Bus 01
31	-12 VDC	-12 Volts DC
32	+5 VDC	+5 Volts DC

TABLE 2-3. CONNECTOR LIST
MODEL 500VA BACKPLANE CONNECTOR P01-ROW-B

NOTE: ALL ($) SIGNALS UNUSED ON THIS CARD

PIN	SIGNAL	FUNCTION
01	9VMBSBY ($)	Bus Busy
02	9VMBCLR ($)	Bus Clear
03	ACFAIL ($)	AC Power Fail
04	9VMBGI0	Bus Grant 0 IN
05	9VMBGO0	Bus Grant 0 OUT
06	9VMBGI1	Bus Grant 1 IN
07	9VMBGO1	Bus Grant 1 OUT
08	9VMBGI2	Bus Grant 2 IN
09	9VMBGO2	Bus Grant 2 OUT
10	9VMBGI3	Bus Grant 3 IN
11	9VMBGO3	Bus Grant 3 OUT
12	9VMBRQ0 ($)	Bus Request 0
13	9VMBRQ1 ($)	Bus Request 1
14	9VMBRQ2 ($)	Bus Request 2
15	9VMBRQ3 ($)	Bus Request 3
16	4VMAM00	Address Modifier 0
17	4VMAM01	Address Modifier 1
18	4VMAM02	Address Modifier 2
19	4VMAM03	Address Modifier 3
20	GND	Ground
21	SERCLK ($)	Serial Clock
22	SERDAT ($)	Serial Data
23	GND	Ground
24	9VMIRQ7 ($)	Interrupt Request 7
25	9VMIRQ6 ($)	Interrupt Request 6
26	9VMIRQ5 ($)	Interrupt Request 5
27	9VMIRQ4 ($)	Interrupt Request 4
28	9VMIRQ3 ($)	Interrupt Request 3
29	9VMIRQ2 ($)	Interrupt Request 2
30	9VMIRQ1 ($)	Interrupt Request 1
31	+5 VSTDBY ($)	Stand-by +5 Volts DC
32	+5 VDC	+5 Volts DC

TABLE 2-4. CONNECTOR LIST
MODEL 500VA BACKPLANE CONNECTOR P01-ROW-C

NOTE: ALL ($) SIGNALS UNUSED ON THIS CARD

PIN	SIGNAL	FUNCTION
01	4VMED08	Data Bus 08
02	4VMED09	Data Bus 09
03	4VMED10	Data Bus 10
04	4VMED11	Data Bus 11
05	4VMED12	Data Bus 12
06	4VMED13	Data Bus 13
07	4VMED14	Data Bus 14
08	4VMED15	Data Bus 15
09	GND	Ground
10	SYSFAIL ($)	System Failure
11	9VMBERR ($)	Bus Error
12	9VMPRST	System Reset
13	9VMELWD	Long Word
14	4VMAM05	Address Modifier 5
15	4VMEA23	Address Bus 23
16	4VMEA22	Address Bus 22
17	4VMEA21	Address Bus 21
18	4VMEA20	Address Bus 20
19	4VMEA19	Address Bus 19
20	4VMEA18	Address Bus 18
21	4VMEA17	Address Bus 17
22	4VMEA16	Address Bus 16
23	4VMEA15	Address Bus 15
24	4VMEA14	Address Bus 14
25	4VMEA13	Address Bus 13
26	4VMEA12	Address Bus 12
27	4VMEA11	Address Bus 11
28	4VMEA10	Address Bus 10
29	4VMEA09	Address Bus 09
30	4VMEA08	Address Bus 08
31	+12 VDC	+12 Volts DC
32	+5 VDC	+5 Volts DC

TABLE 2-5. CONNECTOR LIST
MODEL 500VA BACKPLANE CONNECTOR P02-ROW-A

PIN	SIGNAL	FUNCTION
01	4EXW101	ACRO External Word 1 Bit 01
02	4EXW103	ACRO External Word 1 Bit 03
03	4EXW105	ACRO External Word 1 Bit 05
04	4EXW107	ACRO External Word 1 Bit 07
05	4EXW109	ACRO External Word 1 Bit 09
06	4EXW111	ACRO External Word 1 Bit 11
07	4EXW113	ACRO External Word 1 Bit 13
08	4EXW115	ACRO External Word 1 Bit 15
09	GND	ACRO Ground
10	4EXW201	ACRO External Word 2 Bit 01
11	4EXW203	ACRO External Word 2 Bit 03
12	4EXW205	ACRO External Word 2 Bit 05
13	4EXW207	ACRO External Word 2 Bit 07
14	4EXW209	ACRO External Word 2 Bit 09
15	4EXW211	ACRO External Word 2 Bit 11
16	4EXW213	ACRO External Word 2 Bit 13
17	4EXW215	ACRO External Word 2 Bit 15
18	GND	ACRO Ground
19	GND	ACRO Ground
20	N/C	
21	N/C	
22	N/C	
23	N/C	
24	4CLOCK2	ACRO SIM Clock 2
25	4DATA02	ACRO SIM Data 2
26	4CLOCK4	ACRO SIM Clock 4
27	4DATA04	ACRO SIM Data 4
28	4SIMIN1	ACRO SIM Code Out 1
29	4SIMIN2	ACRO SIM Code Out 2
30	4SIMIN3	ACRO SIM Code Out 3
31	4SIMIN4	ACRO SIM Code Out 4
32	9SMSTOP	ACRO Simulator Stop

TABLE 2-6. CONNECTOR LIST
MODEL 500VA BACKPLANE CONNECTOR P02-ROW-B

NOTE: ALL ($) SIGNALS UNUSED ON THIS CARD

PIN	SIGNAL	FUNCTION
01	+5 VDC	+5 Volts DC
02	GND	Ground
03	RESERVED ($)	
04	4VMEA24	Address Bus 24
05	4VMEA25	Address Bus 25
06	4VMEA26	Address Bus 26
07	4VMEA27	Address Bus 27
08	4VMEA28	Address Bus 28
09	4VMEA29	Address Bus 29
10	4VMEA30	Address Bus 30
11	4VMEA31	Address Bus 31
12	GND	Ground
13	+5 VDC	+5 Volts DC
14	4VMED16 ($)	Data Bus 16
15	4VMED17 ($)	Data Bus 17
16	4VMED18 ($)	Data Bus 18
17	4VMED19 ($)	Data Bus 19
18	4VMED20 ($)	Data Bus 20
19	4VMED21 ($)	Data Bus 21
20	4VMED22 ($)	Data Bus 22
21	4VMED23 ($)	Data Bus 23
22	GND	Ground
23	4VMED24 ($)	Data Bus 24
24	4VMED25 ($)	Data Bus 25
25	4VMED26 ($)	Data Bus 26
26	4VMED27 ($)	Data Bus 27
27	4VMED28 ($)	Data Bus 28
28	4VMED29 ($)	Data Bus 29
29	4VMED30 ($)	Data Bus 30
30	4VMED31 ($)	Data Bus 31
31	GND	Ground
32	+5 VDC	+5 Volts DC

TABLE 2-7. CONNECTOR LIST
MODEL 500VA BACKPLANE CONNECTOR P02-ROW-C

PIN	SIGNAL	FUNCTION
01	4EXW100	ACRO External Word 1 Bit 00
02	4EXW102	ACRO External Word 1 Bit 02
03	4EXW104	ACRO External Word 1 Bit 04
04	4EXW106	ACRO External Word 1 Bit 06
05	4EXW108	ACRO External Word 1 Bit 08
06	4EXW110	ACRO External Word 1 Bit 10
07	4EXW112	ACRO External Word 1 Bit 12
08	4EXW114	ACRO External Word 1 Bit 14
09	9EX1RDY	ACRO External Word 1 Ready
10	4EXW200	ACRO External Word 2 Bit 00
11	4EXW202	ACRO External Word 2 Bit 02
12	4EXW204	ACRO External Word 2 Bit 04
13	4EXW206	ACRO External Word 2 Bit 06
14	4EXW208	ACRO External Word 2 Bit 08
15	4EXW210	ACRO External Word 2 Bit 10
16	4EXW212	ACRO External Word 2 Bit 12
17	4EXW214	ACRO External Word 2 Bit 14
18	9EX2RDY	ACRO External Word 2 Ready
19	GND	GND
20	N/C	
21	N/C	
22	N/C	
23	N/C	
24	4CLOCK1	ACRO SIM Clock 1
25	4DATA01	ACRO SIM Data 1
26	4CLOCK3	ACRO SIM Clock 3
27	4DATA03	ACRO SIM Data 3
28	GND	ACRO Ground
29	GND	ACRO Ground
30	GND	ACRO Ground
31	GND	ACRO Ground
32	GND	ACRO Ground

TABLE 2-8. CONNECTOR LIST
MODEL 500VA FRONT PANEL CONNECTOR J01

NOTE: ALL ($) SIGNALS UNUSED ON THIS CARD

PIN	SIGNAL	FUNCTION
01		Not Used
02	9AWORD1	A Bus Last Word Flag
03	4SFLAG1	A Bus Last Transfer Flag
04	4APORT1 ($)	A Bus Port Select 1
05	GND	Ground
06	4APORT0 ($)	A Bus Port Select 0
07	4ADEST2	A Bus Destination Select 2
08	4ADEST1	A Bus Destination Select 1
09	4ADEST0	A Bus Destination Select 0
10	4AOUT15	A Bus 15
11	4AOUT14	A Bus 14
12	GND	Ground
13	4AOUT13	A Bus 13
14	4AOUT12	A Bus 12
15	4AOUT11	A Bus 11
16	4AOUT10	A Bus 10
17	GND	Ground
18	4AOUT09	A Bus 09
19	4AOUT08	A Bus 08
20	4AOUT07	A Bus 07
21	4AOUT06	A Bus 06
22	4AOUT05	A Bus 05
23	4AOUT04	A Bus 04
24	GND	Ground
25	4AOUT03	A Bus 03
26	4AOUT02	A Bus 02
27	4AOUT01	A Bus 01
28	4AOUT00	A Bus 00
29	GND	Ground
30	9AOACK0 ($)	A Bus Acknowledge 0
31	9AOREQ0 ($)	A Bus Request 0
32	GND	Ground
33	9AOACK1 ($)	A Bus Acknowledge 1
34	9AOREQ1 ($)	A Bus Request 1
35	GND	Ground
36	9AOACK2 ($)	A Bus Acknowledge 2
37	9AOREQ2 ($)	A Bus Request 2
38	GND	Ground
39	9AOACK3 ($)	A Bus Acknowledge 3
40	9AOREQ3 ($)	A Bus Request 3

TABLE 2-8. (continued) CONNECTOR LIST MODEL 500VA FRONT PANEL CONNECTOR J01		
NOTE: ALL ($) SIGNALS UNUSED ON THIS CARD		
PIN	**SIGNAL**	**FUNCTION**
41	GND	Ground
42	9AOWAIT	A Bus Wait
43	9AOREST ($)	A Bus Reset
44	GND ($)	Ground
45		Not Used
46		Not Used
47		Not Used
48		Not Used
49		Not Used
50		Not Used
51		Not Used
52		Not Used
53		Not Used
54		Not Used
55		Not Used
56		Not Used
57		Not Used
58		Not Used
59	GND	Ground
60	9AOSTRB	A Bus Strobe

TABLE 2-9. CONNECTOR LIST MODEL 500VA FRONT PANEL CONNECTOR J02		
PIN	**SIGNAL**	**FUNCTION**
01	4OUTCK1	PCM Clock Output 1
02	GND	Ground

TABLE 2-10. CONNECTOR LIST MODEL 500VA FRONT PANEL CONNECTOR J03		
PIN	**SIGNAL**	**FUNCTION**
01	4OUTDT1	PCM Data Output 1
02	GND	Ground

TABLE 2-11. CONNECTOR LIST MODEL 500VA FRONT PANEL CONNECTOR J04			
PIN	**JP2**	**SIGNAL**	**FUNCTION**
01	**1-2**	4OUTCK2	PCM Clock Output 2
01	**2-3**	4FRSTBT	First Bit Flag
02		GND	Ground

TABLE 2-12. CONNECTOR LIST MODEL 500VA FRONT PANEL CONNECTOR J05			
PIN	**JP3**	**SIGNAL**	**FUNCTION**
01	**1-2**	4OUTDT2	PCM Data Output 2
01	**2-3**	4MFRCYF	Minor Frame Flag
02		GND	Ground

TABLE 2-13. CONNECTOR LIST MODEL 500VA FRONT PANEL CONNECTOR J06		
PIN	**SIGNAL**	**FUNCTION**
01	4OUTCD1	PCM Code Output 1
02	GND	Ground

TABLE 2-14. CONNECTOR LIST MODEL 500VA FRONT PANEL CONNECTOR J07			
PIN	**JP4**	**SIGNAL**	**FUNCTION**
01	**1-2**	4OUTCD2	PCM Code Output 2
01	**2-3**	4EXTCLK	External Clock Input
02		GND	Ground

TABLE 2-15. CONNECTOR LIST MODEL 500VA FRONT PANEL CONNECTOR J08			
PIN	**JP5**	**SIGNAL**	**FUNCTION**
01	**1-2**	4PAMDAT	PAM Output
01	**2-3**	4SFRCYF	Major Frame Flag
02		GND	Ground

SECTION 3
OPERATION

3.1 HARDWARE STRUCTURE

The Model 500V is configured from the VME Host computer. The VME Host computer interfaces to eight hardware registers and 96K bytes of program memory. The registers contain Command and Status data on the simulator. The location of the registers is memory-mapped in the A16 Utility Address Space on a switch selectable 32-byte boundary. You access the registers through nonprivileged or supervisory A16/D16 memory instructions. The allocation of the A16 Utility Space is shown in TABLE 3-1 below.

3.2 SIMULATOR CONTROL REGISTERS

A16 Utility Registers		
Addr	**Description**	**Mode**
E	Frequency Synthesizer Setup	Read/Write
C	Command/Status Register 2	Read/Write
A	Command/Status Register 1	Read/Write
8	Counter Timer CT2 Setup	Write
6	Counter Timer CT1 Setup	Write
4	Counter Timer Prescalar	Write
2	A32 Base Address	Read/Write
0	Simulator Signature	Read

TABLE 3-1. A16 Utility Registers

3.2.1 Signature Register

The Signature Register contains the card signature, which identifies the card for system configuration purposes. The reading of this register returns a value 500 hexadecimal.

3.2.2 A32 Base Address Register

The Base Address Register establishes the A32 block address for the Simulator Program Memory. It stores the 15 MSBs of the A32 base address utilized when accessing the memory.

3.2.3 Counter Timer Registers

The Counter Timer Registers are used for the Dynamic memory rate function. The Counter Timer Prescalar is used to feed CT1 and CT2, which are used for Dynamic memory 1 and Dynamic memory 2 respectively. The three counter timer registers are 16 bit dividers with n+1 as the divisor, i.e., n=0 divides by 1.

3.2.4 Command/Status Register 1

Command/Status Register 1 provides control for running the simulator. TABLE 3-2 lists the command and status bits of the register.

Command/Status Register 1		
Bit	**Description**	**Mode**
15	Mainframe Recycle flag	R/W
14	Dynamic Data 2 Count Mode	R/W
13	SF1 Recycle flag	R/W
12	Dynamic 2 Memory Recycle	R/W
11	Dynamic 1 Memory Recycle	R/W
10	Dynamic Data 1 Count Mode	R/W
9-8	Pseudo-Random Mode	R/W
7	Run Program B	R/W
6-4	Counter 2 Mode	R/W
3	Run Program A	R/W
2-0	Counter 1 Mode	R/W

TABLE 3-2. Command/Status Register 1.

Command/Status Register 1 powers up with all bits set to zero.

The following is a description of the bits in Command/Status Register 1.

Bits 2-0, Counter 1 Mode, and bits 6-4, Counter 2 Mode, are decoded as follows.

```
000 = Decrement counter on each recycle of the mainframe
      instruction file.
001 = Decrement counter on each recycle of subframe 1
      instruction file.
010 = Decrement counter on each recycle of subframe 2
      instruction file.
011 = Decrement counter on access by any instruction file.
100 = Increment counter on each recycle of the mainframe
      instruction file.
101 = Increment counter on each recycle of subframe 1
      instruction file.
110 = Increment counter on each recycle of subframe 2
      instruction file.
111 = Increment counter on access by any instruction file.
```

Bits 9-8, Pseudo-Random Mode, are decoded as follows.

```
00 = Never reset the Pseudo Random data source.
01 = Reset the Pseudo Random source on the recycle of
     the mainframe instruction file.
10 = Reset the Pseudo Random source on each recycle of
     subframe 1 instruction file.
11 = Reset the Pseudo Random source on each recycle of
     subframe 2 instruction file.
```

Bits 10 and 14, the Dynamic Data Count Mode bits, determine when the Dynamic Data Memory Address counter is incremented. When the bit is high, the address counter is incremented with each use. When the bit is low, the address counter is incremented by the Counter Timer accessed by registers at addresses 4, 6, and 8.

Bits 11 & 12 indicate when the dynamic memories have recycled. When the bits are set to 1, the respective dynamic memory has reached the end of it's defined length. The status will stay on until a 1 is written to that bit, which will set it to 0.

Bits 13 & 15 are functionally similar to bits 11 & 12. A flag will be set to 1 to indicate when the mainframe instruction memory reaches it's last word. The same holds true for subframe 1 instruction memory. Writing a 1 to each bit will cause the latched state to clear.

The Run bits 3 and 7 are both set high to cause a frame from the alternate format to be inserted into the output stream at the time of the current formats mainframe recycle. When both of the bits are set low, the simulator will stop upon the next mainframe recycle.

3.2.5 Command/Status Register 2

Command/Status Register 2 controls the output code, the bit rate limiter, and the use of the external clock. This register powers up with all bits set to zero. TABLE 3-3 details the bits of this register.

Command/Status Register 2	
Value	**Description**
800	External Clock used
400	Limits Bit Rate for Single Stepping PCM
200	Enable PCM Output
100	Parity Odd
F	Miller Squared Space
E	Miller Squared Mark
D	Delay Modulation Space
C	Delay Modulation Mark
B	Biphase Space
A	Biphase Mark
9	Biphase Level
8	Differential Biphase Space
7	Randomized NRZ-23
6	Randomized NRZ-17
5	Randomized NRZ-15
4	Randomized NRZ-11
3	Differential Biphase Mark
2	NRZ Space
1	NRZ Mark
0	NRZ Level

TABLE 3-3. Command/Status Register 2

3.2.6 Frequency Synthesizer Setup Register

The Frequency Synthesizer Setup Register contains 16 bits, with the upper 5 bits, 15-11, controlling the prescaling, and the lower 11 bits, 10-0, controlling the VCO frequency.

3.3 SIMULATOR MEMORY

The 48K by 16-bits of program memory is initialized by the simulator compiler. This memory is divided into two partitions for storing formats A and B. Each partition is segmented into 4K blocks, and contains a programmed description of a simulated format. The simulator executes the code stored in this memory to generate the output data stream. The program memory is mapped in the VME A32 Memory Address Space on a programmable 128K byte boundary. You access the memory through nonprivileged or supervisory A32/D16 memory instructions. The allocation of the A32 Address Space is shown in TABLE 3-4.

A32 Memory Allocation	
Byte Addr	**Description**
17FFF → 17000	2k by 16-bit-word Dynamic Data 2 shadow file
16FFF → 16000	2k by 16-bit-word Dynamic Data 1 shadow file
15FFF → 15000	2k by 16-bit-word Dynamic Data 2 file
14FFF → 14000	2k by 16-bit-word Dynamic Data 1 file
13FFF → 12000	4k by 16-bit-word Parameter file
11FFF → 10000	4k by 16-bit-word Fixed Data file
FFFF → E000	4k by 16-bit-word Subframe 3 program B
DFFF → C000	4k by 16-bit-word Subframe 2 program B
BFFF → A000	4k by 16-bit-word Subframe 1 program B
9FFF → 8000	4k by 16-bit-word Mainframe program B
7FFF → 6000	4k by 16-bit-word Subframe 3 program A
5FFF → 4000	4k by 16-bit-word Subframe 2 program A
3FFF → 2000	4k by 16-bit-word Subframe 1 program A
1FFF → 0000	4k by 16-bit-word Mainframe program A

TABLE 3-4. A32 Memory Allocation

The following table details that section of A32 space containing the program format.

Program Word Format		
Bit	**Value**	**Description**
15-12		Resource Code
	0-3	Fixed Data Block
	6	Counter 1
	7	Counter 2
	8	Program Counter
	9	Pseudo Random
	A	Dynamic Data
	B	External Data 1
	C	External Data 2
	D	Jump to Subframe 1
	E	Jump to Subframe 2
	F	Jump to Subframe 3
11	1	Frame Recycle
10	1	ComplementData
9-8		Parameter Block
7-0		Fixed Data Words

TABLE 3-5. Program Instruction Word Format

The following table describes the values in the Parameter file section of A32 memory.

Parameter Word Format		
Bit	**Value**	**Description**
8-15		Unused
7 7	0 1	Serialize LSB First Serialize MSB First
6 6	0 1	Trailing Parity Leading Parity
5 5	0 1	No Parity Parity
4-0	0 ↓ 31	1-bit Word Length ↓ 32-bit Word Length

TABLE 3-6. Parameter Word Format

3.4 SOFTWARE DESCRIPTION

3.4.1 Data Source Setup

The Simulator translates Setup commands into information which formats the PAM/PCM data for simulation. A typical Simulator setup has the schematic

```
SIM PCM
CLR
   .
   .
   .
Data Source setup instructions
   .
   .
   .
FORMAT A
   .
   .
   .
Mainframe description
   .
   .
   .
SFn subframe description
   .
   .
   .
END SFn
other subframes
   .
   .
   .
END FORMAT A
RUN A
END SIM
```

3.4.2 Simulator Setup

The Simulator setup is divided into two sections, the data source setup and the format setup of the program. To invoke the correct Simulator setup function, use the command

SIM *function*

where *function* may be **PCM** or **EDIT.**

The **EDIT** function allows you to modify the data source values and properties once a setup has already been entered. No data source defaults are used in **SIM EDIT,** so you may redefine only the data source parameter you are interested in changing. You may not **EDIT** the format.

3.4.2.1 Clearing Memory

When the Simulator Setup program is invoked, you can reset all the programmable parameters to their initial values by entering the command

CLR

This command clears the Simulator of any previous setup. It sets the data sources to the default values and clears the program formats. When a program format memory is cleared, all the words are set to the pseudo-random word, so that any word which is left undefined in your program will be the pseudo-random word. The Simulator has two format memories which may be programmed; however, they share the data resources. None of the below mentioned source setup commands are mandatory for Simulator setup because the Simulator has initial values for all of the data sources. Defaults for missing parameters are supplied when a source setup command does not specify all of the parameters for that source. When a parameter for a source has been specified on one line, it does not need to be specified on following lines setting up that source. The order in which you enter the optional parameters on the command line does not matter. Improperly set values result in an error message and no action. To change a parameter without having defaults set, use **SIM EDIT.**

The source setup operations may be executed in any order. These operations include

SYMBOL	SETUP FUNCTION
RATE	Data rate in Hertz
CODE	Data transmit code.
OPA/EPA	Output parity odd or even
DATA*nn*	1024 user defined data words
CTR*n*	2 User defined counters
DYN*n*	2 Dynamic data memories
RND	Pseudo random data
EXT*n*	2 External input ports
PC	Program counter

The following paragraphs describe these source setup commands.

3.4.2.2 Data Rate

The PCM data rate is programmable. To set the data rate, use the command

RATE=*i.f*

where *i.f* is a number between 1 Hz and 25 MHz. The data rate will be set to within .1% of the desired rate. For example, the command

`RATE=100000`

will set the data rate to 100.0087 kHz. The **RATE** command accepts the characters **H, K,** and **M** which represent Hertz, Kilohertz and Megahertz, respectively. Therefore,

`RATE=2K` is the same as `RATE=2000`
`RATE=3M` is the same as `RATE=3000000`

You may use an external clock by entering the command

RATE=EXT

INITIAL VALUE: RATE=100K

3.4.2.3 Data Code

The PCM transmit code is selected from sixteen different codes. These codes are

NRZL	Non-Return to Zero Level
NRZM	Non-Return to Zero Mark
NRZS	Non-Return to Zero Space
RNRZ-11	Randomized Non-Return to Zero 11-Stage
RNRZ-15	Randomized Non-Return to Zero 15-Stage
RNRZ-17	Randomized Non-Return to Zero 17-Stage
RNRZ-23	Randomized Non-Return to Zero 23-Stage
BIPL	Biphase Level
BIPM	Biphase Mark
BIPS	Biphase Space
DBIPM	Differential Biphase Mark
DBIPS	Differential Biphase Space
DMM	Delay Modulation Mark
DMS	Delay Modulation Space
M2M	Miller Squared Mark
M2S	Miller Squared Space

The code may be set with the command

CODE *type*

The parameter *type* is one of the above mnemonics. For example, to set the data transmit code to be Biphase Mark, enter

CODE BIPM

DEFAULT: CODE NRZL

3.4.2.4 Word Properties

The optional word property parameters *Length, Orientation, & Parity* of all sources are user defined. Up to four different sets of word properties settings may be selected for each of the counters, one external source port, the pseudo-random word and the program address register. Each dynamic data word can have two orientation and parity settings. The word property parameters define how a source is to be output. They allow you to use the same data source for more than one type of word. By defining more than one set of length, orientation, and parity to a source, you may use the same data source in several different ways. For instance, you may need 8 bits for a filler to make a 24-bit word. If a counter is already set up for 16 bits, you may use a second set of property parameters for the counter to define the 8 bits you need.

These sets of word property selections are noted below by *n*, where *n* is 1 to 4 for set *n* of the word properties of the resource. If *n* is omitted, *n*=1 is assumed. For a single source, the word property settings for different sets may be specified on one line or multiple lines.

For all sources except fixed data words, the maximum length is 16 bits; fixed

data words maximum length is 32 bits. The length of a word is specified by **LEN***n*=*nn*.
DEFAULT: LENn=16

To specify which bit is justified (serialized first out of the Simulator), the Orientation parameter may be **LSB***n* (Least Significant Bit) or **MSB***n* (Most Significant Bit).
DEFAULT: MSBn

Parity may be **LPA***n* (Leading Parity), **TPA***n* (Trailing Parity) or **NPA***n* (No Parity).
DEFAULT: NPAn

3.4.2.5 Parity Control

All of the Simulator data sources can optionally have parity bits. To set the type of parity (Odd or Even) enter one of the following. All of the output when specified as **LPA** or **TPA** will be controlled by this option.

OPA	Odd parity
EPA	Even parity

INITIAL VALUE: OPA

3.4.2.6 Fixed Data Words

The Simulator allows you to define up to 1024 data words. To define a data word use the command

DATA*nn*[=*val*] [**LEN=***nn*] [*orien*] [*parity*]

where the **DATA***nn* word number *nn* is an integer from 0 to 1023, and the parameter *val* may be from 0 to 4294967295 (/FFFFFFFF). **DATA***nn* with no value will leave the current value for the data word. The maximum length is 32 bits. **DATA***nn* words have only one set of word properties. Defaults are used for any word property not specified on a single **DATA***nn* line. For example, to set data word number 3 to 574, a 16-bit word with Least Significant Bit orientation and trailing parity, enter the command

```
DATA3=574 LSB TPA
```

To set data word number 23 to a 31-bit data source with the value of hexadecimal 574326AA, enter the command

```
DATA23=/574326AA LEN=31
```

MSB and **NPA** will be set for **DATA23** by default. The initial value for all of the data words is zero.

DEFAULT: DATAnn LEN=16 MSB NPA

3.4.2.7 Counters

There are two programmable 16-bit counters that can keep track of the mainframe or subframe cycle where they are currently positioned, or can be used as a uniform increasing or decreasing data source. A counter will *increment* if the initial value (Preset) is less than the terminal value (Limit); a counter will *decrement* if the Preset value is greater than the Limit value. To set up the counters as a data source, enter the command

CTR*n* [**PRE=***nn*] [**LIM=***nn*] [*freq*] [**LEN***n*=*nn*] [*orien*] [*parity*]

n must be either 1 or 2, specifying the counter number.
PRE is the preset value of the counter; *nn* may be a number from 0 to 65535.
LIM is the limit of the counter; *nn* may be a number from 0 to 65535.
freq determines when the counter is incremented or decremented and can be one of the following.

REF	modified every reference
MFR	modified every Mainframe recycle
SF1	modified every Subframe One recycle
SF2	modified every Subframe Two recycle

Note there is no option for modification on Subframe Three.

DEFAULT:

CTR1	PRE=1	LIM=64	REF	LEN1=16	MSB1	NPA1
CTR1				LEN2=16	MSB2	NPA2
CTR1				LEN3=16	MSB3	NPA3
CTR1				LEN4=16	MSB4	NPA4
CTR2	PRE=4095	LIM=0	REF	LEN1=16	MSB1	NPA1
CTR2				LEN2=16	MSB2	NPA2
CTR2				LEN3=16	MSB3	NPA3
CTR2				LEN4=16	MSB4	NPA4

3.4.2.8 Dynamic Data

The Simulator provides two Dynamic Data memories, each with 2048 16-bit words, with several user selectable functions. To set up this data source use the command

DYN*n* [*function*] [*type*] [**RATE=***i.f*] [**LEN=***nn*] [*orien*] [*parity*]

n must be either 1 or 2, specifying the dynamic memory number. There are two sets of Orientation and Parity word properties for each dynamic data word.

The optional parameter *function* must be one of the following.

RMP	A Ramp function
SIN	A Sine wave
SQR	A Square wave
TRI	A Triangle wave
LOAD	A User defined function

The optional parameter *type* specifies the data type the Simulator will generate.

type must be one of the following.

1CM	Ones Complement
2CM	Twos Complement
OBN	Offset Binary
SMG	Sign Magnitude

The parameter **RATE** specifies the frequency (in cycles per second) of the function. The rate range is .000003 through 12203 for all functions except **LOAD.** If the rate parameter is omitted or is set equal to zero, the data read counter increments at every reference, making the wave period a function of the reference rate. For example, to set up Dynamic Data 1 to be a sine wave with a period of 3.5 Hz, expressed as one's complement data, enter

```
DYN1 SIN 1CM RATE=3.5
```

The optional parameter **LOAD** specifies a user defined function. When **LOAD** is present the data points of the function must be entered immediately following the **DYN***n* command line. The entered data is used exactly as entered by the user, there is no conversion for the generated data; the *type* parameter is ignored. The command schematic is

DYN*n* **LOAD** [**RATE=***i.f*] [**LEN=***nn*] [*orien*] [*parity*]
data points
[**RPT** *nn* data points]
END

where data points are the user-defined values of the function. The maximum number of data points that can be loaded is 2048.

The optional command **RPT** (Repeat) can be used only with the LOAD function; its parameter *nn* allows you to select the number of times the data points following it on that line are to be repeated.

Use **END** to terminate the **LOAD** command. For example

```
DYN2 LOAD 2CM LEN=12
/100 /101 /109 /10C /962 /928 /600 /10A /023 /10A /10B /114
RPT 3 /3f0 /111
END
```

will produce the above data points in 12-bit words. The **RPT** command will repeat /3f0 /111 three times, producing six words. Since **RATE** defaults to zero, the data address counter is incremented every time the data is referenced. If **RATE** is specified, the number of data points will be counted and the clock rate calculated such that the address counter will step through the data points at the desired frequency. When setting the **LOAD** function's frequency, the *i.f* value of the **RATE** parameter must be less than or equal to the maximum clock rate of 12500000 Hz divided by the number of data words loaded. This can be written

$$\text{RATE}= +\text{i.f} \le \frac{\text{maxclkrt}}{\text{no. words}}$$

For example, if 250 words have been loaded, the **LOAD** function maximum frequency (the *i.f* value of the **RATE** parameter) will be 12500000/250, which is 50000 Hz.

Since the two dynamic data rates share a common timer, the ratio of the two dynamic memory rates cannot be greater than 65,535 to 1. The changing of one rate may affect the other actual rate.

DEFAULT:
DYN1 SIN 2CM RATE=0 LEN=16 MSB1 NPA1 MSB2 NPA2
DYN2 RMP 2CM RATE=0 LEN=16 MSB1 NPA1 MSB2 NPA2

3.4.2.9 Random Data

The Simulator has a pseudo-random number generator which can serve as a data word source. Four sets of word properties can be selected. To set up the Simulator to have a pseudo-random data source, enter the command

RND [*recycle*] [**LEN***n*=*nn*] [*orien*] [*parity*]

The parameter *recycle* may be one of the following.

NR	Never recycle
MFR	Mainframe recycle
SF1	Subframe One recycle
SF2	Subframe Two recycle

Note there is no Subframe Three recycle. This parameter specifies the frame recycle that resets the pseudo-random generator. For example, to set up a random data word of length 12, least significant bit orientation, with reset at each Subframe One recycle, enter the command

```
RND SF1 LEN1=12 LSB1
```

DEFAULT:
RND NR LEN1=16 MSB1 NPA1
RND LEN2=16 MSB2 NPA2
RND LEN3=16 MSB3 NPA3
RND LEN4=16 MSB4 NPA4

3.4.2.10 External Ports

The Simulator is equipped to handle two 16-bit external data ports. For External port 1, the data is presented to the Simulator statically (no strobe), and has four sets of word properties. External port 2 has the option of input buffering the data through a FIFO. In mode **FBD** (Frame Buffered Data), the Simulator will insert the FIFO data starting on a new mainframe only if it has captured at least one frame of data and the FIFO is at least half full. If these conditions are not met, the data word **EXT1** will be substituted for the full frame. In mode **NOR** (NORmal) data is presented to the Simulator statically (no strobe). External port 2 has one set of word properties.

The commands to set up the external data sources are

EXT1 [**LEN***n=nn*] [*orien*] [*parity*]
EXT2 [*type*] [**LEN=***nn*] [*orien*] [*parity*]

The **EXT2** parameter *type* must be **NOR** or **FBD.** Because the *type* is encoded into the format instructions, whatever type is set before the **FORMAT** statement is the **EXT2** *type* that will be used. Changing the *type* will not be effective until the next **FORMAT** is loaded.

DEFAULT:

EXT1		LEN1=16	MSB1	NPA1
EXT1		LEN2=16	MSB2	NPA2
EXT1		LEN3=16	MSB3	NPA3
EXT1		LEN4=16	MSB4	NPA4
EXT2	NOR	LEN=16	MSB	NPA

3.4.2.11 Program Counter

A program counter is used with the mainframe and three subframe instruction memories. When a reference to the program counter occurs, the value of the current program counter (mainframe or one of the subframes) will be output. If a 16-bit word length is used, the two MSB bits will always be zero. The addresses for the formats run from 0 - 3FFF hex. Four sets of word properties may be selected. To set up the program counter as a data source, enter the command

PC [**LEN***n=nn*] [*orien*] [*parity*]

DEFAULT:

PC	LEN1=16	MSB1	NPA1
PC	LEN2=16	MSB2	NPA2
PC	LEN3=16	MSB3	NPA3
PC	LEN4=16	MSB4	NPA4

3.4.3 Program Example

The following is a sample program to review the commands that have been described thus far,

```
SIM                                | Invoke Simulator

CLR                                | Clear program and data sources

CODE BIPM                          | Code is Biphase Mark

RATE=2M                            | Rate: 2 MHz

DATA0=/11 LEN=8                    | Set DATA0 to /11, 8 bits long
DATA1=/22 LEN=8                    | Set DATA1 to /22, 8 bits long
DATA23=/15 LEN=8                   | Set DATA23 to /15, 8 bits long

CTR1 PRE=0 LIM=64000 SF1 LEN=16   | Counter 1, preset is 0, limit
                                   |  is 64000. Count every SF1
                                   |  recycle. 16 bits long

DYN1 SQR 1CM LEN=10                | Dynamic data, square wave,
                                   | ones complement, 10 bits long
                                   | increment on reference
                                   | MSB, no parity

PC LEN1=10  LEN2=8                 | Program counter
                                   | Set 1: 10 bits long, MSB, no parity
                                   | Set 2: 8 bits, MSB, no parity
```

First memory is cleared and the initialized values are set. Next, the code is specified as Biphase Mark and the transmit rate is defined to be 2 MHz. Three 8-bit user-defined data words are specified. A counter is set to increment (starting from 0, going to 64000) and reset every Subframe One recycle. There is a 10-bit dynamic data word defined to be a square wave in ones complement. Finally, there is a 10-bit and an 8-bit program counter. Blank lines have been included in the above example to help identify different functions, and are not required in an actual setup program.

3.4.4 Format Setup

The mainframe and subframe formats must be described, once the data word sources have been setup for data generation. Format description defines the order in which data will be simulated. To describe the format enter the command

FORMAT [*A/B*] [**LEN=***nn*] [**LOAD**]

It is optional to specify which program memory you wish to set up. If you do not specify which format it is, program memory A will be set up if SIM is in IDLE or if the B format is running, program memory B will be set up if the A format is running. The optional parameter **LEN** denotes the length of the mainframe. The maximum frame length is 4096. If **LEN** is not specified the length of the mainframe is considered to be that of the greatest frame word referenced. The optional **LOAD** specification follows the Subframe Load Function described below.

3.4.4.1 Word References

To reference one of the data sources from the initial setup instructions, use the command

WORD nn_1 [nn_2 nn_3-nn_4 ... nn_{20}] *source*

where nn_x may be any desired frame word (or range of words), and *source* is any data source. Any of the data sources may be preceded by ˜ (a tilde) to complement the data. Up to 20 different frame words (or ranges) may be specified per line. The word properties set number must be specified by (*n*) where applicable. If the set number is omitted, set 1 is assumed. For example, if the first word in the frame is to be the Program Counter, word properties set 2, enter

```
WORD 1 PC(2)
```

If a series or range of frame words have the same data source, a hyphen between the frame numbers will assign that source to the series. For example to assign Counter 1 to be the data source for words 4 through 17, use

```
WORD 4-17 CTR1
```

The word properties set for Counter 1 will be set 1 by default. The parameter *source* in the **WORD** command may also be a subframe reference. A subframe reference may be

SF1	Subframe One
SF2	Subframe Two
SF3	Subframe Three

For example, to specify that the second, third, fourth, and eleventh frame words are Subframe Two references, enter

```
WORD 2-4 11 SF2
```

All subframe descriptions must be in increasing word order. For instance, in the above example, you cannot specify word 11 to be **SF2**, and then on the same or a following line specify words 2-4 to be **SF2**.

Looking again at the example above, words 2-4 are three references and word 11 is one reference, for a total of four references to Subframe Two.

3.4.4.2 Fill Words

The mainframe and subframes are initialized to the pseudo-random data word, so a frame location that has no specific source assigned to it will be Random data unless a fill word is specified. To specify a fill word enter the command

FILL *source*

The parameter *source* may be any data word; it cannot be a subframe reference **SF1, SF2,** or **SF3**. If you use **FILL**, it should be the first command within a frame description because it will fill every word of the frame being described with the source, replacing any words already used to describe the frame. When **FILL** is used in the mainframe description, it fills the mainframe and all three subframes with the source. When used in a subframe description, it fills only that subframe with the source.

3.4.4.3 End Command

An **END** command is required to end the format setup, each subframe setup, and the Simulator setup. Only **END** is interpreted by the Simulator program; any words past **END** are ignored.

The following programming example uses the material covered so far.

```
SIM PCM
CODE BIPM
RATE=2M
DATA0=/11 LEN=8          | DATA SETUP
DATA1=/22 LEN=8
DATA2=/33 LEN=8
DYN1 SQR 1CM LEN=10
PC LEN1=10
                         | FORMAT SETUP
FORMAT LEN=14            | Begin format, 14 words long
FILL PC(1)               | Fill word is Program Counter set 1
WORD 1-4 DATA1           | MF words 1 through 4 are DATA1
WORD 5-7 DATA2           | MF words 5 through 7 are DATA2
WORD 9 DYN1(1)           | MF word 9 is Dynamic Data 1 set 1
WORD 12 DATA0            | MF word 12 is DATA0
END FORMAT               | End format setup
END SIM                  | End Simulator setup
```

This example shows the data source setup, then shows in the format section that the mainframe is 14 words long, and the words that are not specified (8,10,11,13,14) are the data source **PC**, the fill word, with word properties set one. Mainframe words one through four reference **DATA1**, as their data source, words five though seven reference **DATA2**, word nine is Dynamic Data 1, set one, and word twelve is **DATA0**.

When the Simulator setup is finished, a matrix is drawn to display the format. The previous example will look like

```
MAINFRAME

WORD:       1       2       3       4       5       6       7       8
       DA0001  DA0001  DA0001  DA0001  DA0002  DA0002  DA0002  PCT(1)
            9      10      11      12      13      14
       DN1(1)  PCT(1)  PCT(1)  DA0000  PCT(1)  PCT(1)
```

3.4.5 Subframes

The subframe formats must be declared within the mainframe declaration and should be defined only after all references to the subframe have been made. Use the command

SF*n* **DEPTH=***nn* [**MFCOMP**]

to start the subframe declaration. *n* must be 1, 2, or 3 to specify the subframe about to be described. **DEPTH** specifies the frame depth (number of frames), so *nn* must be a positive number. The optional parameter **MFCOMP**, Mainframe Complement, will automatically assign **DATA0** as the data source for every word in the last subframe reference, except for the final word of the last subframe which will contain the complement of **DATA0**. The last reference to this type of subframe must be left undefined in the subframe setup so that the mainframe complement function defines the data source.

3.4.5.1 Subframe Words

The subframe words are defined with a combination of two commands, the **FOR** command and the **WORD** command. The **FOR** command allows the user to define a different subframe word each time the mainframe references that subframe word. The **WORD** command has previously been described. The **FOR** command syntax is

FOR nn_1 [nn_2-nn_3 nn_4 ... nn_{20}]

where nn_x is the desired subframe number (or range of numbers). No more than 20 subframe numbers may be specified on one line. The **FOR** command may be followed by as many **WORD** commands as needed to describe the subframe words. For example

```
FORMAT              | Format Setup
WORD 1 DATA1        | Word 1 DATA1
WORD 2 SF1          | Word 2 is Subframe 1
WORD 3 DATA0        | Word 3 is DATA0

SF1 DEPTH=4         | Subframe 1 setup, 4 words deep
FOR 1 2             | Word 1 and Word 2 at
WORD 2 DATA1        | MF position 2, is DATA1
FOR 3 4             | Word 3 and Word 4 at
WORD 2 DATA2        | MF position 2, is DATA2
END SF1             | End Subframe 1 setup

END FORMAT          | End format setup
```

This setup says that at Mainframe Word 2, reference Subframe 1: Subframe 1 is four words deep; Subframe word 1 referenced at Mainframe word 2 is **DATA1**; Subframe word 2 referenced at Mainframe word 2 is **DATA1**; Subframe word 3 referenced at Mainframe word 2 is **DATA2**; and Subframe word 4 referenced at Mainframe word 2 is **DATA2**.
The mapping of this format looks like

```
MAINFRAME:

WORD:       1       2       3
       DA0001     SF1  DA0000

SUBFRAME 1:
           2
 1:   DA0001
 2:   DA0001
 3:   DA0002
 4:   DA0002
```

A shorter, more effective method of writing the above program is

```
FORMAT           | Format setup
WORD 1 DATA1     | Word 1 is DATA1
WORD 2 SF1       | Word 2 is Subframe 1
WORD 3 DATA0     | Word 3 is DATA0

SF1 DEPTH=4      | Subframe 1, 4 words deep
FILL DATA2       | Fill word is DATA2
FOR 1 2          | Words 1 and 2 at
WORD 2 DATA1     | MF position 2, are DATA1
END SF1          | End Subframe 1 setup

END FORMAT       | End format setup
```

The mapping of this format will be identical to the previous format because the subframe has a **FILL** word of **DATA2**, and the **FOR** statement specifies the first two frames of the subframe to be **DATA1**.

The following is a more complex example.

```
FORMAT                   | Format setup
  .
  .                      | MF word references
  .
WORD 3-6 SF1             |MF words 3 through 6 are Subframe 1.

SF1 DEPTH=5              |Subframe 1 is 5 words deep.
 FOR 1 2                 |Subframe word 1 and 2 referenced at:
  WORD 3 4 RND(1)        |  MF position 3 and 4 is Random data.
  WORD 5 6 DATA1         |  MF position 5 and 6 is Data 1.
 FOR 3                   |Subframe word 3 referenced at:
  WORD 3 4 RND(1)        |  MF position 3 and 4 is Random data.
  WORD 5 6 CTR1          |  MF position 5 and 6 is Counter 1.
 FOR 4 5                 |Subframe word 4 and 5 referenced at:
  WORD 3 4 DATA1         |  MF position 3 and 4 is Data 1.
  WORD 5 6 CTR1          |  MF position 5 and 6 is Counter 1.
END SF1                  |End Subframe 1.

END FORMAT
```

```
MAINFRAME:

WORD:     ...   3       4       5       6  ...
          ... SF1     SF1     SF1     SF1  ...

SUBFRAME  1:

         3        4        5        6
1:  RND(1)   RND(1)   DA0001   DA0001
2:  RND(1)   RND(1)   DA0001   DA0001
3:  RND(1)   RND(1)   CT1(1)   CT1(1)
4:  DA0001   DA0001   CT1(1)   CT1(1)
5:  DA0001   DA0001   CT1(1)   CT1(1)
```

Note that the subframe section of the matrix has the mainframe word position above it to clarify data word location. If an error occurs in the subframe setup, the matrix will be drawn with the word **ERROR** in the word position where the error occurs.

The program above may be simplified to look like this.

```
FORMAT
   .
   .
   .
WORD 3-6 SF1          |MF words 3 through 6 reference Subframe 1.

SF1 DEPTH=5           |Subframe 1 is 5 words deep.
 FILL DATA1           |The Subframe fill word is DATA1.
 FOR 1-3              |The Subframe words 1 through 3 referenced at
  WORD 3 4 RND(1)     |  MF position 3 and 4 are Random data.
 FOR 3-5              |The Subframe words 3 through 5 referenced at
  WORD 5 6 CTR1       |  MF position 5 and 6 are Counter 1.
END SF1               |End Subframe 1.

END FORMAT
```

The **FILL** command ensures that all subframe words which are not specifically defined are defined with **DATA1**.

3.4.5.2 Subcommutated Mainframe Complement Sync.

The following is a Simulator setup with two Subframe One references, the second reference being the Mainframe complement sync word.

```
FORMAT            | Format setup
WORD 1 DATA1      | Word 1 DATA1
WORD 2 SF1        | Word 2 SF1
WORD 3 DATA2      | Word 3 DATA2
WORD 4 SF1        | Word 4 SF1

SF1 DEPTH=4       | Subframe 1 setup, 4 words deep
FOR 1 2           | Word 1 and Word 2 at
WORD 2 DATA1      | MF position 2 is DATA1
WORD 4 DATA0      | MF position 4 is DATA0
FOR 3             | Word 3 at
WORD 2 DATA3      | MF position 2 is DATA3
WORD 4 DATA0      | MF position 4 is DATA0
FOR 4             | Word 4 at
WORD 2 DATA3      | MF position 2 is DATA3
WORD 4 ~DATA0     | MF position 4 is complement DATA0
END SF1           | End Subframe 1 setup

END FORMAT        | End format
```

```
MAINFRAME:

WORD:       1        2        3        4
       DA0001      SF1   DA0002      SF1

SUBFRAME 1:

          2        4
 1: DA0001   DA0000
 2: DA0001   DA0000
 3: DA0003   DA0000
 4: DA0003  ~DA0000
```

A much shorter way to write the above program using the **MFCOMP** command is

```
FORMAT               |Format setup
WORD 1 DATA1         | Word 1 is DATA1
WORD 2 SF1           | Word 2 is Subframe 1
WORD 3 DATA2         | Word 3 is DATA2
WORD 4 SF1           | Word 4 is Subframe 1

SF1 DEPTH=4 MFCOMP   |Subframe 1 setup, 4 words deep, Mainframe Comp.
FILL DATA3           | Fill word is DATA3
FOR 1 2              | Words 1 and 2 at
WORD 2 DATA1         | MF position 2, are DATA1
END SF1              | End Subframe setup

END FORMAT           |End format setup
```

The mapping of this format is identical to the previous one, because the subframe has a **FILL** word of **DATA3**, subframe words 1 and 2 at mainframe position 2 are set to **DATA1**, and the **MFCOMP** statement programs the rest of the subframe references. **MFCOMP** automatically assigns **DATA0** as the data source for every word in the last subframe reference, except for the final word of the last subframe, which will contain the complement of **DATA0**.

3.4.6 Using the Counter to Define Depth

The user defined counters may be used to specify the **DEPTH** of a subframe. The syntax is

SF*n* **CTR***n* [**MFCOMP**]

The parameter *n* in **SF***n* is 1, 2, or 3, and the parameter *n* in **CTR***n n* must be 1 or 2. When a counter is used to define the subframe depth, you must have already set up the counter. It is possible to have a frame number of 0 only when the counters are used to define the depth of a subframe. If the counter runs from 245 to 255, then you must use the numbers 245 - 255 to describe the frames in the **FOR** statement.
To illustrate the usage

```
SIM PCM                  |invoke the Simulator
CTR1 PRE=78 LIM=80       |define counter 1 for 78 - 80
FORMAT                   |start the Mainframe definition
WORD 2-3 SF1             |mainframe words 2 and 3 are Subframe 1
  .
  .
  .
SF1 CTR1 MFCOMP          |Subframe 1's depth defined by CTR1
FOR 78 79                |Subframe words 78 and 79
WORD 2 PC                |at mainframe word 2 is PC
FOR 80                   |Subframe word 80
WORD 2 EXT1              |at mainframe word 2 is EXT1
END                      |end the Subframe setup
END                      |end the format setup
END                      |end the Simulator setup
```

```
MAINFRAME:

WORD:      ...       2        3       ...
           ...      SF1      SF1      ...

SUBFRAME 1:
             2        3
  78:    PCT(1)   DA0000
  79:    PCT(1)   DA0000
  80:    EX1(1)  ~DA0000
```

3.4.6.1 Subframe Load Function

It is possible to have such a large number of references that **FOR/WORD** specifications will be too long. To alleviate this problem there is a **LOAD** function for the subframe word specifications. The data words are loaded into the subframe format corresponding to the order they are referred to in the mainframe. The **LOAD** function has the schematic form

SF*n* [**DEPTH=***nn*] **LOAD**
$source_1$ $source_2$ ...$source_{nn}$
[**RPT** *nn*] $source_1$ ...$source_{nn}$
DATA$\{x_1x_2x_3...x_{nn}\}$
END

MFCOMP is not an option when you use the subframe **LOAD** function. Again, the parameter *n* in **SF***n* is 1, 2, or 3. **DEPTH** specifies the number of words in the subframe. The depth may be defined by a counter instead of using **DEPTH**. In which case the subframe load line looks like this.

SF*n* **CTR***n* **LOAD**

The parameter *n* in **CTR***n n* is 1 or 2. When a counter is used to define the subframe depth, you must have already set up the counter. Only for the subframe **LOAD** can the depth be left undefined. If the depth is not specified at all, it is calculated at the end of the **LOAD** from the number of words entered and the number of references to the subframe being defined. If you specify a depth that defines the subframe to have more words than you load, the remaining words in the frame that you did not specify with your **LOAD** will be the mainframe **FILL** word, or the default fill of a random word. Also note it is possible to load a number of words that is not a multiple of the references made to the subframe.

In the **LOAD** subframe, as the word sources are read from left to right, the data pointer increments every subframe reference to store the next data word in the format. When the **REPEAT** command is used, the parameter *nn* specifies the number of times the text on that line is to be repeated. You may not load references to another subframe.

A form to load **DATA***nn* words unique to the **LOAD** subframe is the statement

DATA$\{x_1x_2x_3...x_{16}\}$

The individual parameter *x* is a number 0 through 9, or a character A through Z, which provides a unique, single character symbol for each of the first 36 user-defined **DATA** words 0 through 35. The maximum number of data references that can appear in a single set of braces is 16. There must not be blanks between the symbols, and the complement data tilde is not allowed in this form of **DATA** statement. As an example, below is a statement that would generate 64 words.

```
SF1 LOAD
RPT 2 DATA{10AN1230G234I110} DATA{012F4501BB450120}
END
```

The output matrix demonstrates how the subframe would look if it had 8 references to it.

SUBFRAME 1:

	1	2	3	4	5	6	7	8
1:	DA0001	DA0000	DA0010	DA0023	DA0001	DA0002	DA0003	DA0000
2:	DA0016	DA0002	DA0003	DA0004	DA0018	DA0001	DA0001	DA0000
3:	DA0000	DA0001	DA0002	DA0015	DA0004	DA0005	DA0000	DA0001
4:	DA0011	DA0011	DA0004	DA0005	DA0000	DA0001	DA0002	DA0000
5:	DA0001	DA0000	DA0010	DA0023	DA0001	DA0002	DA0003	DA0000
6:	DA0016	DA0002	DA0003	DA0004	DA0018	DA0001	DA0001	DA0000
7:	DA0000	DA0001	DA0002	DA0015	DA0004	DA0005	DA0000	DA0001
8:	DA0011	DA0011	DA0004	DA0005	DA0000	DA0001	DA0002	DA0000

Each brace enclosed **DATA** reference specifies 16 words, for a total of 32 words. The **REPEAT** command says to repeat the data that follows on that line two times. Reading from left to right, the first subframe reference will use **DATA1** as the data source, the second will use **DATA0**, the third (symbol A) will use **DATA10**, and the fourth (symbol N) will use **DATA23**, and so on to the final reference symbol.

Another example using the subframe **LOAD** is

```
SF2 DEPTH=11 LOAD
DATA1 DATA3 DATA4 CTR1(1) CTR2(1) DYN1(1)
RPT 4 DATA0 PC DATA1 EXT2
DATA{134} CTR1(1) CTR2(1) DYN1
RPT 4 DATA0 PC DATA{1M}
END SF2
```

If there were four references to the Subframe 2 description above the output matrix would be

```
MAINFRAME:

WORD:   . .  2      3      4      5. . .
            SF2    SF2    SF2    SF2
```

SUBFRAME 2:

	2	3	4	5
1:	DA0001	DA0003	DA0004	CT1(1)
2:	CT2(1)	DN1(1)	DA0000	PCT(1)
3:	DA0001	EX2	DA0000	PCT(1)
4:	DA0001	EX2	DA0000	PCT(1)
5:	DA0001	EX2	DA0000	PCT(1)
6:	DA0001	EX2	DA0001	DA0003
7:	DA0004	CT1(1)	CT2(1)	DN1(1)
8:	DA0000	PCT(1)	DA0001	DA0022
9:	DA0000	PCT(1)	DA0001	DA0022
10:	DA0000	PCT(1)	DA0001	DA0022
11:	DA0000	PCT(1)	DA0001	DA0022

3.4.7 Sub-Subcommutation

You can reference a subframe while in a subframe (sub-subcommutation). To do this, remember that all references to a subframe must be made before the subframe is defined. In the following example the comments on the right side of the page describe the setup of the subframes.

```
FORMAT
  .
  .
  .
WORD 10 SF1           |Word 10, reference Subframe 1
SF1 DEPTH=4           |Subframe 1, 4 words deep
 FOR 1-3              |Subframe words 1, 2, and 3 at
  WORD 10 DATA0       |MF position 10, reference DATA0
 FOR 4                |Word 4 at
  WORD 10 SF2         |MF position 10, reference Subframe 2
END SF1               |End Subframe 1
SF2 DEPTH=4           |SUBFRAME 2, 4 words deep
 FOR 1                |Word 1 at
  WORD 10.4 DATA3     |MF position 10, SF1 position 4, is DATA3
 FOR 2-4              |Words 2-4 at
  WORD 10.4 DYN1(1)   |MF position 10, SF1 position 4, are Dynamic Data 1
END SF2               |End Subframe 2
END FORMAT
```

The matrix will look like this.

```
MAINFRAME:

WORD:  . . .  10  . . .
       . . . SF1  . . .

SUBFRAME 1:
      10
1:  DA0000
2:  DA0000
3:  DA0000
4:     SF2

SUBFRAME 2:
      10
1:  DA0003
2:  DN1(1)
3:  DN1(1)
4:  DN1(1)
```

Since there are three subframes, up to three levels of subcommutation can be simulated. For example, if Subframe Two in the previous example had referenced a third subframe in its first word at 10.4, then the word references within Subframe Three would have been 10.4.1 (Mainframe position 10, Subframe One position 4, Subframe Two position 1).

3.4.8 Programming Example

The following is a PCM Simulator programming example.

```
SIM PCM

| Format:
|  16 word mainframe with ID subframe word 2 as the ID word.
|  Words 5, 6, 10-12 and 13,14 are ID subcommutated and
|  synchronized by a Mainframe Complement 5th frame.

| Data Setup
| Use DATA word default values

RATE=1000000              | Set rate
CODE DMM                  | Set PCM code
DATA19=/AAAAAA LEN=24     | Define five data words
DATA20=/CCCCCC LEN=24
DATA21=/111111 LEN=24
DATA22=/555555 LEN=24
DATA23=/FFFFFF LEN=24
DYN1 LOAD LEN=8           | Dynamic data 1 LOAD
 RPT 10  0 /FF            | Repeat ten times two 8-bit words
END                       | End the dynamic LOAD
CTR1 PRE=0 LIM=7 LEN=8    | Counter 1 (up counter) length is 8 bits
CTR1 LEN=16 MFR           | incremented on MF reference
RND LEN2=8 LEN1=16        | Pseudo-random word 8-bits and 16-bits
PC LEN1=8 MSB1 LEN2=8 LSB2 |Program counter, two sets of word properties

| Format Setup

FORMAT LEN=16            | Format 16 words long
FILL RND(1)              | Fill with 16-bit random word
WORD 2 CTR1              | Word 2 Counter 1
WORD 3 RND(2)            | Word 3, 8-bit random word
WORD 5 6 10-12 SF1       | Words 5, 6, 10, 11, 12, Subframe 1
WORD 13 14 16 SF2        | Subframe 13, 14, 16, Subframe 2

      SF1 CTR1 LOAD            | Subframe 1 (indexed by counter 1) LOAD
       DATA{1234} CTR1         | Frame 0*
       DATA{5678} CTR1         | Frame 1
       DATA{9abc} CTR1         | Frame 2
       DATA{defg} CTR1         | Frame 3
       DATA{hi00} CTR1         | Frame 4
       DATA{9abc} CTR1         | Frame 5
       DATA{1234} CTR1         | Frame 6
       RPT 4 EXT1              | Frame 7, first four words
       CTR1                    | End Frame 7
      END SF1                  | End of Subframe 1 description

      | * Subframe frame 0 exists only when indexed by a counter
```

```
 SF2 DEPTH=5 MFCOMP        | Subframe 2 description (complement sync)
  FOR 1                    | Words 1 through 3 at MF position
   WORD 13 DATA19          | 13, are DATA2
  FOR 2                    | Words 1 through 3 at MF position
   WORD 13 DATA20          | 13, are DATA2
  FOR 3                    | Words 1 through 3 at MF position
   WORD 13 DATA21          | 13, are DATA2
  FOR 4                    | Words 1 through 3 at MF position
   WORD 13 DATA22          | 13, are DATA2
  FOR 5                    | Words 1 through 3 at MF position
   WORD 13 DATA23          | 13, are DATA2
  FOR 1-5                  | Words 1 through 5 at MF position
   WORD 14 SF3             | 14 reference Subframe 3
 END SF2                   | End of Subframe 2 description

SF3 DEPTH=3                | Subframe 3 description
 FILL DYN1(1)              | Fill with Dynamic Data
 FOR 1 3                   | Word 1 and Word 3
  WORD 14.1 PC(1)          | MF word 14, SF2 frame 1, is Program Counter.
  WORD 14.2 CTR1           | MF word 14, SF2 frame 2, is Counter 1.
  WORD 14.3 PC(2)          | MF word 14, SF2 frame 3, is Program Counter.
  WORD 14.4 RND(2)         | MF word 14, SF2 frame 4, is Random Data.
  WORD 14.5 CTR1           | MF word 14, SF2 frame 5, is Counter 1.
END SF3                    | End of Subframe 3 description

END FORMAT                 | End of frame description
END SIM                    | Exit Simulator setup
```

The mainframe/subframe matrix is

SIMULATOR FORMAT A:

MAINFRAME:

WORD:	1	2	3	4	5	6	7	8
	RND(1)	CT1(1)	RND(2)	RND(1)	SF1	SF1	RND(1)	RND(1)

WORD:	9	10	11	12	13	14	15	16
	RND(1)	SF1	SF1	SF1	SF2	SF2	RND(1)	SF2

SUBFRAME 1:

	5	6	10	11	12
0:	DA0001	DA0002	DA0003	DA0004	CT1(1)
1:	DA0005	DA0006	DA0007	DA0008	CT1(1)
2:	DA0009	DA0010	DA0011	DA0012	CT1(1)
3:	DA0013	DA0014	DA0015	DA0016	CT1(1)
4:	DA0017	DA0018	DA0000	DA0000	CT1(1)
5:	DA0009	DA0010	DA0011	DA0012	CT1(1)
6:	DA0001	DA0002	DA0003	DA0004	CT1(1)
7:	EX1(1)	EX1(1)	EX1(1)	EX1(1)	CT1(1)

SUBFRAME 2:

	13	14	16
1:	DA0019	SF3	DA0000
2:	DA0020	SF3	DA0000
3:	DA0021	SF3	DA0000
4:	DA0022	SF3	DA0000
5:	DA0023	SF3	~DA0000

SUBFRAME 3:

	14	14	14	14	14
1:	PCT(1)	CT1(1)	PCT(2)	RND(2)	CT1(1)
2:	DN1(1)	DN1(1)	DN1(1)	DN1(1)	DN1(1)
3:	PCT(1)	CT1(1)	PCT(2)	RND(2)	CT1(1)

3.4.9 Run Modes

The Simulator operates in **RUN** mode by translating the setup commands and format definitions into information which is used to output the simulated PCM data to the decommutator program under test. Two Simulator programs may be programmed and loaded in a single Simulator setup. When the Simulator setup is first invoked it is not generating data, the Simulator run state is **STOP**. To put the Simulator into **RUN** mode, enter the command

RUN [*setup*]

The optional parameter *setup* is the letter **A** or **B** indicating which format should be run. If *setup* is not specified, the default is to run the format just entered when the command is entered in the Simulator setup. If the command is entered out of the Simulator setup, and the setup is not specified, the default is to run format A. When a format is running, to insert a frame of the nonrunning format into the data stream, enter the command

INS

The Simulator can be returned to the **STOPPED** state with the command

STOP

When the Simulator is running, the decommutator program under test will search the simulated data for the Sync Word and, when located, the program will CHECK the pattern and then LOCK onto the data. The decommutated data should be sent to an output port for post decommutation verification.

SECTION 4
THEORY OF OPERATION

4.1 BLOCK DIAGRAM

This chapter contains a block diagram of the FSIM card.

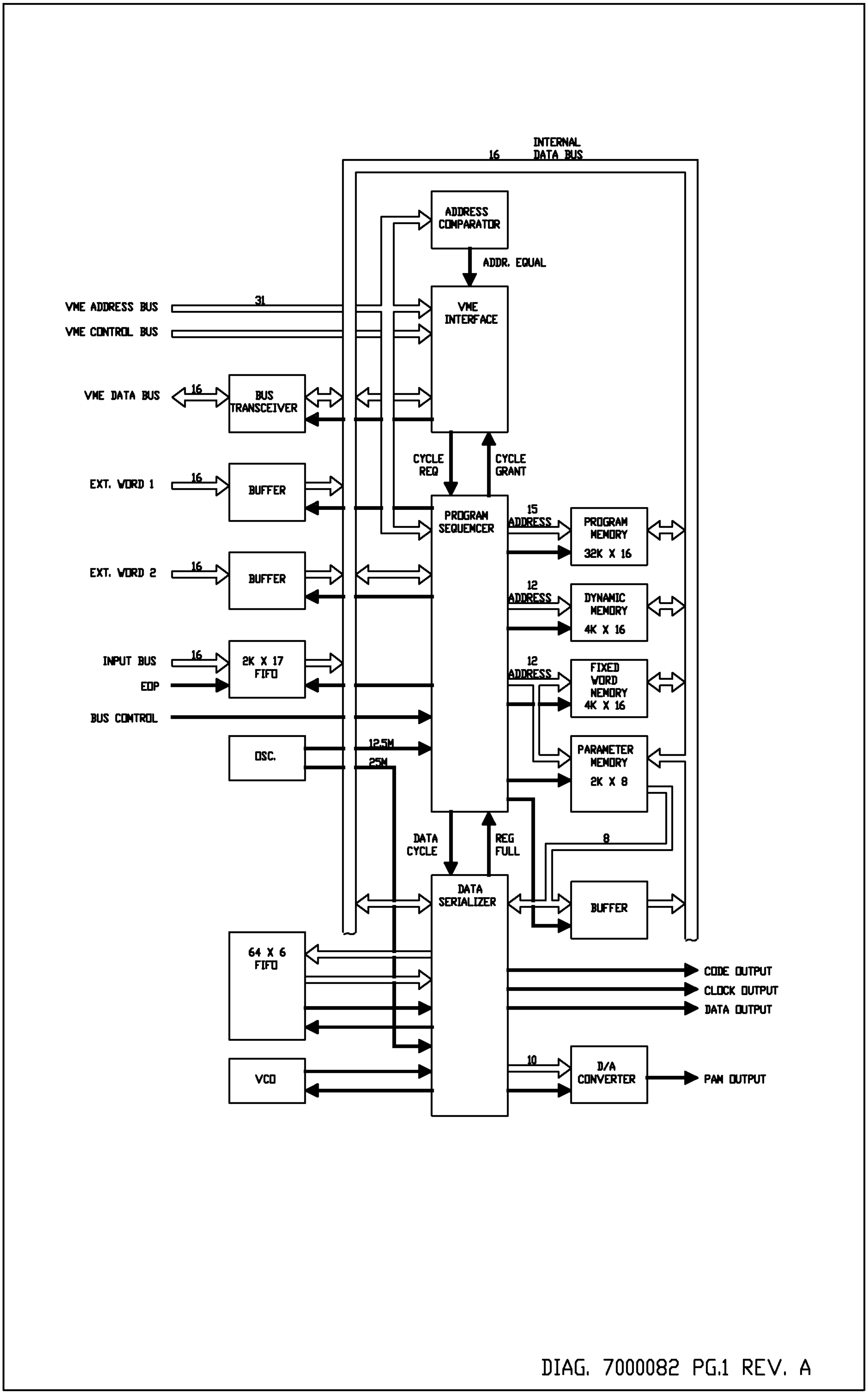

FIG 4-1 FSIM BLOCK DIAGRAM

SECTION 5
FSIM - ADJUSTABLE SWITCH & JUMPER SETTINGS

5.1 DESCRIPTION

The paragraphs below describe the selections available on the simulator card.

5.1.1 Address Select

The switches at U16 and U15 select the base of the 16 byte block of simulator registers in the A16:D16 address space A4 through A15. Switch U16 positions 1-8 select the eight MSBs and Switch U15 positions 1-4 select the four LSBs.

5.1.2 Input Selection

The simulator has a mode in which you can inject frames of decommutated data into the simulated format. The data is taken from the A bus using a Bus Device number specified by your JP1 jumper selection. The distribution DMA uses Device 0 to input data to the VME bus. Do not use Device 0 unless the output format is compatible with the simulator input (16 bit words with End Of Frame flag). Devices 1, 2, and 3 are normally available for digital outputs, and Device 3 is normally formatted to use with the simulator.

5.1.3 Framing Signals

Optional framing signals are jumper selectable at the SMB connectors J4, J5, and J8, using jumpers JP2, JP3, and JP5. The framing signals are First Bit flag, Minor Frame flag, and Major Frame flag.

5.1.4 External Clock

Using jumper JP4, you can select an external input clock (at J7) to drive the simulator. The maximum frequency is 25 MHz.

5.1.5 Bipolar Output Calibration

Perform the following steps to set the voltage you want at the bipolar output:

1. Connect BNC5 (4OUTCD1) to an oscilloscope with a 75 Ω termination.
2. Program the FSIM (any pattern) to output 1Mbps biphase-level (BipL) data.
3. Adjust VR2 to set the peak to peak amplitude to the level you want.
4. Adjust VR1 to remove any DC offset.

NOTE: minimum amplitude (p-p) is about 2.8V; maximum is about 5.5V.

5.1.6 Factory Settings

Figure 1 shows jumper and switch locations on the FSIM PCB, and provides a record of the factory settings when the board was shipped. Figure 2 indicates the functions of all the possible switch and jumper selections.