ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Acroamatics Telemetry Systems 500V* at our website: ***Click HERE***

DN 6000252

USER'S MANUAL FOR THE • PCM FORMAT SIMULATOR •

Acroamatics, Inc.
70 South Kellogg Avenue
Goleta, CA 93117-3476

June 5, 2000

PROPRIETARY NOTICE

Information contained in this document is disclosed in confidence and may not be duplicated in full or in part by any person without prior written approval by Acroamatics, Inc., except as provided by separate contractual agreement. Its sole purpose is to provide the user with adequately detailed documentation in order to install, operate, maintain, and order spare parts for the equipment supplied. The use of this document for any other purpose is expressly prohibited.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ACROAMATICS DOCUMENT HISTORY

The following table indicates major changes made to *User's Manual for the PCM Simulator*, Acroamatics Document Number 6000252, released on June 5, 2000, and contains a record of all revisions made since that date.

DN6000252 CHANGE HISTORY			
Rev	**Date**	**Action**	**Name**
	6-5-00	Original Issue	DJM
	1-16-1	Mods to cover VME & PCI	DJM

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

USER'S MANUAL FOR THE
• PCM FORMAT SIMULATOR •

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION 1-1
DESCRIPTION 1-1
Format Description 1-1
Setup Mode 1-2
Definable Parameters 1-2
Mainframe & Subframe Description 1-3
Running the Simulator 1-3
PROGRAMMING LANGUAGE STRUCTURE 1-3
How Statements are Described in the Manual 1-4

CHAPTER 2 DATA SETUP 2-1
SIMULATOR SETUP 2-1
Clearing Memory 2-2
DATA SOURCE SETUP 2-2
Data Rate 2-2
Data Code 2-3
Word Properties 2-3
Parity Control 2-4
Fixed Data Words 2-4
Counters 2-5
Dynamic Data 2-5
Random Data 2-7
External Ports (hardware optional feature) 2-8
Program Counter 2-8
EXAMPLE 2-9

CHAPTER 3 FORMAT SETUP 3-1
WORD REFERENCES 3-1
Fill Words 3-2
End Command 3-2
SUBFRAMES 3-3
Subframe Words 3-3
Subcommutated Mainframe Complement Sync. 3-6
Using the Counter to Define Depth 3-7
Subframe Load Function 3-8
SUB-SUBCOMMUTATION 3-10
PROGRAMMING EXAMPLE 3-11

CHAPTER 4 RUN MODES AND MENUS 4-1
RUN MODES 4-1
SIMULATOR MENU 4-1
HELP MENU 4-2

APPENDIX A
INTRODUCTION TO PROGRAMMING THE SIMULATOR A-1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX B
PROGRAMMING THE PAM SIMULATOR B-1

APPENDIX C
SIMULATOR COMMAND INDEX C-1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

USER'S MANUAL FOR THE
• PCM FORMAT SIMULATOR •

CHAPTER 1
INTRODUCTION

DESCRIPTION

The Acroamatics PCM Simulator software works identically on both the Acroamatics Model 500V VME and Model 470 PCI printed circuit boards. This manual applies equally to both cards. The Simulator provides flexible static and dynamic data resources, output in a programmable format with frame and subframe structure. The Programmable Simulator is managed by a software program (executed in the Host) which allows you to define only that portion of the frame you require. The program loads default values into all programmable registers and format generation memory for use when a specific value is not required. This allows you to generate complex telemetry formats with minimal operator entries.

Format Description

The simulator provides a total of 1024 static and eight dynamic data words of various kinds. The program allows you to generate a PCM frame using 1024 static data words, the values of two 16-bit binary counters, a random word, two external words, the 16-bit program counter itself, and the contents of two 2k by 16-bit RAM memories, which are used for dynamic functions (preset to a sine and a ramp function). The output format is generated by specifying the sequence in which the words are output. The words have various properties which are program-defined and stored in a word definition memory. The word properties which are individually defined include the word length, the parity of the word, and the orientation for output (i.e., MSB or LSB first). Each static word may be from 1 to 32 bits in length; each dynamic word may be from 1 to 16 bits. Each is individually selected to have leading, trailing, or no parity generated by the Simulator. The static words are settable to user defined values. The dynamic words available include a pseudo-random word, two programmable functions provided by the random access memory, and the outputs of two 16-bit counters. Each random access memory is settable to a ramp function, a sine, triangle, or square wave, or you may word load them with any desired value. You can use the counters to generate ramping data values or to form data-word-synchronized subframes. Other dynamic words are two externally generated words and the program address register. The externally generated words permit external switch registers or externally generated data to be entered into the simulated output format. The program address register can be selected to provide unique identification of all words in the frame.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Setup Mode

The Simulator operates in Setup mode by translating setup commands in a symbolic language into information which is used to format the PCM data for simulation. The Simulator operates from a large memory which is memory-mapped into the Host address space. You can enter simulator setup programs via the keyboard or load them from a file. The Simulator program is a C program which is executed by the Host to convert the symbolic setup into memory-mapped, simulated formats. Once setup is complete, the Simulator is put into Run mode to generate PCM data.

Definable Parameters

The generated PCM data is next sent to and processed by the decommutation program you are testing or developing. Parameters which may be defined by Simulation setup commands include:

Data Rate in Hertz
Data transmit Code.
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
The Program counter

You can specify the data rate from 0.6Hz to 20MHz, the PCM output Code can be any of the codes shown below.

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 parity can be odd or even. Additional selections for the dynamic resources are: the random data generator may be reset on a frame or subframe recycle, or never; the two counters are presettable, limited up or down counters which increment or decrement each time they are output, or on a frame or subframe recycle. The two dynamic memories can either be incremented with each output

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

or at a specified rate.

Mainframe & Subframe Description

Once the data word sources have been setup for data generation, the format for the mainframe and subframes is described. The format describes the order in which the data will be simulated. The length of the mainframe may be specified, or if not, the mainframe is considered to be the greatest frame word referenced. To reference one of the data sources from the initial setup instructions, you name the mainframe word and specify the desired data source.

The subframe formats are declared within the mainframe declarations. A total of three subframes may be specified. You reference the subframe data sources in the same manner as the mainframe data sources. You can reference a subframe while in a subframe (sub-subcommutation). You can specify separate data sources to fill the mainframe and each subframe, thereby reducing the words in a frame that have to be specifically defined. After the format has been defined and the resulting program compiled, a matrix of the frames and data sources is available for display.

Running the Simulator

The Simulator operates in Run mode by translating the setup commands and format definitions into information which is used to output serial PCM data to the decommutator program under test. The Simulator has four states, a Stopped state, two Run states, and Insert Frame. There are two Run states to correspond to the two different simulation formats which can be setup and run alternately. Insert Frame allows you to insert one frame from the alternate format into the output stream.

Because all Simulator attributes are defined by the contents of a memory which is memory-mapped into the Host address space, the Host can address the Simulator frame structure, making all output data resources dynamically variable by the Host while the Simulator is running.

This manual contains both a Simulator software setup description and a guide to Simulator use.

PROGRAMMING LANGUAGE STRUCTURE

The Simulator command language consists of a stream of text which is used to define the setup of the PCM Simulator. The text is processed by a program which is referred in this manual as "The Compiler". Sections Two and Three of this manual explain how to use this language to setup the Simulator, with Section Two discussing the Simulator DATA setup and Section Three covering the FORMAT setup. Section Four describes the RUN modes and the MENUS.

A Simulator program consists of a series of lines of text containing Program Statements, which are strings of alphanumeric text constructed according to the rules described in this manual.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

How Statements are Described in the Manual

We show statements in two different ways in this manual.

1. A statement is sometimes represented schematically. The schematic representation of a statement defines the rules for construction of a real statement. The parameters of the statement are represented in a general way, using a specific notation. The following conventions apply to schematic statements.

   The upper case strings in any schematic representation are keywords and must appear literally as shown.

   The punctuation characters

   ```
   (  )  {  }
   ```

   are literals that must appear exactly as shown.

   The lower case italic strings

n	A Digit
nn	A Number

   represent variables that must conform to the formation rules described above. We show variables in italics and give them names that reflect their functions whenever possible.

   Optional elements of a line are shown in square brackets. The square brackets do not actually appear in the statement when the optional element is used.

2. Example statements show arbitrary typical statements constructed from these rules. No lower case characters appear in example statements. Examples are shown separated from the text and centered on the page. For example

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

   shows how an example statement appears in the text. We have used the upper case characters to make the statement stand out as an example statement; the Compiler sees the following as the same statement.

   ```
   dyn1 sin 1cm rate=3.5
   ```

   Text that appears after a bar (|) is commentary and will not affect the function of a program. Digits are the characters "0" through "9" and, when they are part of a hexadecimal number, the characters "A" through "F".

   Numbers are strings of digits, plus certain punctuation. A number preceded by a "/" is interpreted as hexadecimal. For some purposes a number may be a real number expressed as ±*i.f*, where *i* represents the integer part and *f* the fractional part.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 2
DATA SETUP

The Simulator translates Setup commands into information which formats the 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
```

SIMULATOR SETUP

The Simulator setup is divided into two sections, the data source setup and the format setup of the program covered in Section 3. To invoke the correct Simulator setup function, use the command

SIM *function*

where *function* must 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 cannot EDIT the format.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Clearing Memory

When the Simulator Setup program is invoked, you 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.

DATA SOURCE SETUP

The source setup operations may be specified 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*mm*	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.

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 .05% 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,

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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

You use an external clock by entering the command

`RATE=EXT`

INITIAL VALUE: RATE=100K

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 is 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

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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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 must be LSB*n* (Least Significant Bit) or MSB*n* (Most Significant Bit).
DEFAULT: MSBn

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

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

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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Counters

There are two programmable 16-bit counters that can keep track of the main-frame 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. The counters each have four sets of word properties.
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
```

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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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

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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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
```

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* must 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
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

External Ports (hardware optional feature)

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 affect only the next FORMAT 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
```

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
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 3
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.

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 must 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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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.

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.

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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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)
```

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.

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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The program above could 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.

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
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
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.

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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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
```

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$\{x_1 x_2 x_3 ... x_{nn}\}$
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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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 cannot load references to another subframe.

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

$$\text{DATA}\{x_1 x_2 x_3 \ldots x_{16}\}$$

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 be specified 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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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
```

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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
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).

PROGRAMMING EXAMPLE

The following is a PCM Simulator programming example.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
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
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
 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
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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)
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 4
RUN MODES AND MENUS

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 is 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.

SIMULATOR MENU

The Graphical User Interface program SIMGUI provides you with real-time control of the simulator operation parameters. You cannot use SIMGUI to describe the PCM format; for that you must use the language described elsewhere in this document. You can use SIMGUI to alter the value or properties of any data word in the format program, change the data rate and PCM code, and change the selected dynamic functions. You can also use SIMGUI to select the RUN and STOP modes of the simulator and select between the A and B programs. This gives you a powerful tool for testing your PCM format programming and your data processing programs. Used in combination with the Real-time Display, you can alter the data in the PCM stream and verify the computational results.

The icon *Simulator Menu* invokes the SIMGUI program or you can start it from the DOS prompt or a batch file with the command name simgui.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

HELP MENU

An extensive, user-friendly HELP menu (on the top level menu) contains operating instructions for the menu features. Refer to this HELP menu for information on the simulator GUI.

A toolbar on the menu gives quick access to the simulator GUI features. Tool tips, which appear when you pause the pointer over a tool, remind you of what the tool does.

Clicking on the rightmost tool icon < ↑ ? > and then clicking on one of the other tool icons will cause SIMGUI to automatically open the applicable HELP page for you.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX A
INTRODUCTION TO PROGRAMMING THE SIMULATOR

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX A
INTRODUCTION TO PROGRAMMING THE SIMULATOR

INTRODUCTION TO PROGRAMMING THE SIMULATOR

The purpose of this appendix is to acquaint new users of the PAM/PCM Simulator with its programming language by means of program examples which may actually be compiled and executed. It is not directed toward discussion of either the language syntax or the operation of the simulator, but is rather intended to be an introduction to the use of the simulator to assist in the checkout of the programming of complex PCM or PAM formats.

SIMULATOR RESOURCES

There are six separate types of data sources which may be used in simulations. The data output by the simulator from each of these sources depends upon:

1. The value to which the data source is initialized.
2. The length, orientation, and parity defined for the data source.
3. The behavior characteristics defined for the data source.

Length, Orientation and Parity

All of the data sources may have their length, orientation and parity defined. The maximum length of a data source is 16 bits for all sources except the fixed data words whose maximum length is 32 bits. A data source may be delivered LSB (Least Significant Bit) or MSB (Most Significant Bit) first. The parity for a data source may be LPA (Leading Parity), TPA (Trailing Parity), or NPA (No Parity). Up to four different sets of length, orientation and parity may be selected for the data sources, except for the data words.

Counting Words

There are two 16-bit programmable counters that may be used as minor frame counters, or as a uniform increasing or decreasing data source. These are referred to by CTR1 and CTR2. The event which causes the counter to be incremented or decremented is programmable to be the mainframe recycle, the subframe 1 or 2 recycle, or upon each reference to the counter.

Dynamic Data Words

The simulator provides two 16-bit dynamic data word memories. Each can be programmed to generate one of four built-in functions (a ramp, a sine wave, a triangle wave, and a square wave), or may be loaded with user defined data. These are referred to by DYN1 and DYN2. The data generated can be selected to be ones complement, twos complement, sign and magnitude, or offset binary. The frequency of the function (in cycles per second) is programmable.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Fixed Data Words

The simulator has 1024 user defined fixed data words with maximum lengths of 32 bits. The words are referred to as DATA0 through DATA1023.

Random Data Word

The simulator has a 16-bit pseudo-random number generator that may be used as a data source. This is referred to as RND. The event which resets the generator is programmable to the mainframe recycle, the subframe 1 or 2 recycle, or to never reset.

External Ports (hardware optional feature)

The simulator is equipped to handle up to two 16-bit external data ports. These are referred to as EXT1 and EXT2. External port 1 is a static (no strobe) word, while external port 2 has the option of input buffering the data through a FIFO.

Program Counters

Four separate program counters are used with the mainframe and three subframe instruction memories. When a word reference is to a program counter, the value of the current program counter (mainframe or one of the subframes) will be output. This data source is referred to as PC. The data output for the program counter may be 16 bits, but the two MSBs will be zero.

SIMULATOR FORMAT PROGRAMMING

The simulator has been designed to be capable of generating the extremely complex formats now being used in the transmission of PCM and PAM data. Though its capabilities are powerful, the task of programming a simulation is quite straightforward. A simulation program consists of:

1. A description of the data word sources which will be used.
2. A description of the format in which the data will be generated.

The examples in this document begin with simple formats and work toward the more complex. In fact, this method of beginning with simple cases and working towards the more complex is the recommended method of developing a new simulation program for any format which you may be required to deal with. No matter how complicated the format is, you can begin with a program which simply simulates the mainframe. Once that has been made to work, modify that program, one step at a time, to deal with the complexities such as subcommutation or sub-subcommutation.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example 1 - Mainframe Programming

In our first example, we will simulate a very simple format consisting of three 16 bit words, followed by a 16 bit sync pattern. The program is shown below.

```
LINE     STATEMENT
----  ---------------
  1    SIM PCM
  2    CLR
  3    DATA0=/1234
  4    DATA1=/1111
  5    DATA2=/2222
  6    DATA3=/3333
  7    FORMAT LEN=4
  8     WORD 1 DATA1
  9     WORD 2 DATA2
 10     WORD 3 DATA3
 11     WORD 4 DATA0
 12    END FORMAT
 13    END SIM
```

Looking at the program line by line:

1. SIM PCM This is a Compiler command, stating that the program is about to describe the setup for a PCM simulation.
2. CLR This command resets all of the programmable defaults to their initial values and clears the simulator of any previous setup. The initial values for both PAM and PCM simulations are listed in the User's Manual for the PCM/PAM Simulator.
3. DATA0=/1234 This command defines the first data word with the value 1234 (in hexadecimal), this word will be used as the sync word in our simulation. The default length is 16 bits, the default orientation is MSB justified, and the default parity is no parity. Lines 4 through 6 define the next three data words, similarly.
7. FORMAT LEN=4 This command begins the description of the simulation format, stating that the length of the mainframe is four words.
8. WORD 1 DATA1 This command declares that the first word in the frame is to be assigned data word 1 as a data source. Lines 9 through 11 assign the next three words in the frame, similarly.
12. END FORMAT This command marks the end of the programming for the format.
13. END SIM This command marks the end of the programming for the simulator.

Now we are ready to compile the example program. The compilation process will result in a listing like the one shown below.

You should always check the listing for any error or warning messages. If there are any messages, correct your program source and recompile it. The examples in this manual should produce no errors.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Next examine the program listing for a matrix that is generated to display the simulation format.

```
SIM PCM
CLR                     | reset programmable defaults
DATA0=/1234             | mainframe sync word
DATA1=/1111
DATA2=/2222
DATA3=/3333
FORMAT LEN=4            | format description
 WORD 1 DATA1
 WORD 2 DATA2
 WORD 3 DATA3
 WORD 4 DATA0
END FORMAT
END SIM
```

```
SIMULATOR FORMAT A:

MAINFRAME:

WORD:        1        2        3        4
        DA0001   DA0002   DA0003   DA0000
```

The program having compiled successfully, the simulator format memory A is set up and ready to be executed. To start the simulation, enter the command "RUN A". To halt the simulation, enter the command "STOP". The simulator has three states, an idle state and two run states, one for format A and one for format B.

You may change the resource words while the simulator is running. To change the length of DATA1, enter the compiler and change the length to 33. Notice the error message displayed. This entry is an error because the maximum length for a data word is 32 bits. The original length is unchanged. Now change the length to 15 bits. Now the simulator is generating one less bit per major frame.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example 2 - Word Length Changes

In this example, the mainframe consists of a PCM format which contains words of differing lengths. The frame consists of three 8 bit words, a 16 bit word, another 8 bit word, and a 16 bit sync word. The program listing for this format is shown below.

```
SIM PCM
CLR
DATA0=/1234
DATA1=/11 LEN=8
DATA2=/22 LEN=8
DATA3=/33 LEN=8
DATA4=/4444
DATA5=/55 LEN=8
FORMAT LEN=6
 WORD 1 DATA1
 WORD 2 DATA2
 WORD 3 DATA3
 WORD 4 DATA4
 WORD 5 DATA5
 WORD 6 DATA0
END FORMAT
END SIM
```

```
SIMULATOR FORMAT A:

MAINFRAME:

WORD:        1        2        3        4        5        6
        DA0001   DA0002   DA0003   DA0004   DA0005   DA0000
```

In this example, the default word length of 16 bits for the data words is used for DATA0 and DATA4. The length must be specified only for the data words which differ from the default length.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example 3 - Supercommutation

This example demonstrates the programming for a simulation of supercommutated data. The format is a mainframe which has six 8 bit words and a 16 bit sync word, but in this case there are two supercommutated words at mainframe word positions 5 and 6, as shown below.



The program listing for this format is shown below.

```
SIM PCM
CLR
DATA0=/1234
DATA1=/11 LEN=8
DATA2=/22 LEN=8
DATA3=/33 LEN=8
CTR1 PRE=1 LIM=/FF LEN=8
FORMAT LEN=7
 WORD 1 6 DATA1          | word 6 is supercommutated
 WORD 2 5 DATA2          | word 5 is supercommutated
 WORD 3 DATA3
 WORD 4 CTR1
 WORD 7 DATA0
END FORMAT
END SIM
```

```
SIMULATOR FORMAT A:

MAINFRAME:

WORD:        1        2        3        4        5        6        7
        DA0001   DA0002   DA0003   CT1(1)   DA0002   DA0001   DA0000
```

The number appearing in the parenthesis after CT1 specifies the set of length, orientation, and parity for that reference to the counter. When the set is not specified, as in our example, set 1 is used.

In the program listing we see that the statements directly follow the format diagram. The first line of the format description assigns data word 1 as the data source for mainframe word positions 1 and 6. Similarly, the second line of the format description assigns data word 2 as the data source for mainframe word positions 2 and 5.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example 4 - Parity

In this example, the mainframe consists of three 7 bit words, a 15 bit word, another 7 bit word, all with a trailing parity bit, plus a 16 bit sync word with no parity. The program defaults to odd parity, but even parity may be specified. The program listing for this format is shown below.

```
SIM PCM
CLR
DATA0=/1234
DATA1=8 LEN=7 TPA
DATA2=/11 LEN=7 TPA
DATA3=/19 LEN=7 TPA
CTR1 PRE=0 LIM=/7F LEN=7 TPA
DYN1 SIN SMG RATE=0.5 LEN=15 TPA
FORMAT LEN=6
 WORD 1 DATA1
 WORD 2 DATA2
 WORD 3 DATA3
 WORD 4 DYN1
 WORD 5 CTR1
 WORD 6 DATA0
END FORMAT
END SIM
```

```
SIMULATOR FORMAT A:

MAINFRAME:

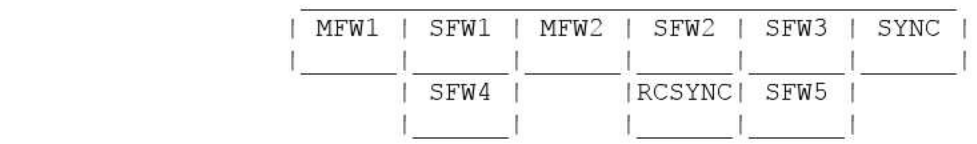
WORD:        1        2        3        4        5        6
        DA0001   DA0002   DA0003   DN1(1)   CT1(1)   DA0000
```

In this example, the suppression of parity for the data words is defaulted. The parity must be specified only for the data words which differ from the default parity.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example 5 - Recycle Subframes

With this example we begin examining the programming for subcommutated PCM data. The format to be simulated is diagramed below.



The frame consists of two 4 bit words, the second of which is subcommutated, then an 8 bit mainframe word, followed by two more 4 bit subcommutated words, and finally the 8 bit mainframe sync word. The subframe is recycle synchronized, and the sync word appears in the position shown as RCSYNC. The program listing for this format is shown below.

```
SIM PCM
CLR
DATA0=/B8 LEN=8
DATA1=/3 LEN=4
RND LEN=8
DYN1 LOAD LEN=4          | dynamic data word definition
 8 9 /A /B /F /C         | user defined function
END LOAD
FORMAT LEN=6
 WORD 1 DATA1
 WORD 2 4-5 SF1
 WORD 3 RND
 WORD 6 DATA0
 SF1 DEPTH=2             | subframe 1 description
  FOR 1-2                | both minor frames
   WORD 2 4-5 DYN1       | mainframe words
 END SF1
END FORMAT
END SIM

SIMULATOR FORMAT A:

MAINFRAME:

WORD:       1       2       3       4       5       6
       DA0001     SF1  RND(1)     SF1     SF1  DA0000

SUBFRAME 1:

              2       4       5
   1:   DN1(1)  DN1(1)  DN1(1)
   2:   DN1(1)  DN1(1)  DN1(1)
```

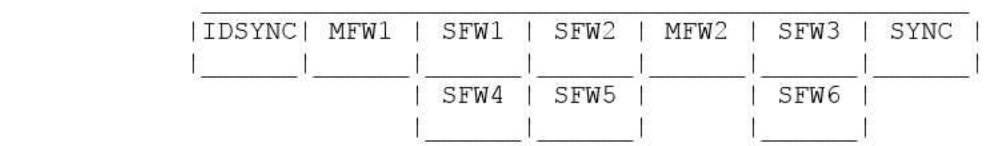
In this example, the dynamic data word is used as a data source for the subframe. The data that will be output by the dynamic data word is described by the user defined function listed on the line after the dynamic data word definition. The subframe description uses the dynamic data word as a data source for

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

mainframe word positions 2, 4, and 5 in both minor frames.

Example 6 - ID Subframes

This example demonstrates programming the simulator for ID synchronized subcommutated data. The format for this example shown below contains six 4 bit words, followed by an 8 bit mainframe sync word.



The first four bits are the counting ID sync word, and the third, fourth and sixth words of the frame are subcommutated. The program listing for this format is shown below.

```
SIM PCM
CLR
DATA0=/B8 LEN=8
DATA1=/3 LEN=4
DATA2=/7 LEN=4
CTR1 PRE=0 LIM=1 LEN=4   | counting word 1 definition
DYN1 LOAD LEN=4
 8 9 /A /B /C /D
END LOAD
FORMAT LEN=7
 WORD 1 CTR1             | counting ID sync word
 WORD 2 DATA1
 WORD 3-4 6 SF1
 WORD 5 DATA2
 WORD 7 DATA0
 SF1 DEPTH=2
  FOR 1-2
   WORD 3-4 6 DYN1
 END SF1
END FORMAT
END SIM
```

```
SIMULATOR FORMAT A:

MAINFRAME:

WORD:       1       2       3       4       5       6       7
       CT1(1)  DA0001     SF1     SF1  DA0002     SF1  DA0000

SUBFRAME 1:

             3       4       6
   1:   DN1(1)  DN1(1)  DN1(1)
   2:   DN1(1)  DN1(1)  DN1(1)
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

In this example, counting word 1 is used as a data source for the first mainframe word. This counting word will count from 0 to 1. The event for incrementing the counter is the default mainframe recycle.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX B
PROGRAMMING THE PAM SIMULATOR

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX B
PROGRAMMING THE PAM SIMULATOR

PROGRAMMING THE PAM SIMULATOR

To invoke the correct Simulator setup function, use the command

SIM *function*

where *function* may be PAM, PAMRZ, PAMNRZ, WFG, EDIT. If *function* is not specified at all, an error will result. If only the word PAM is entered, the system assumes PAMNRZ. If *function* is specified as WFG, the Simulator acts as a Waveform Generator, generating a PAM format with no sync or calibration signals. Initially a PAM function must be specified to set what type of data will be generated.

There are several setup operations that can be executed in any order. Many of these data source setup operations are similar to PCM Setup operations. These operations include

SYMBOL	SETUP FUNCTION
RATE	Channel rate in Hertz
CAL	Data Calibration
DATA*n*	Up to 3 user defined fixed data words
CTR*n*	2 User defined counters
DYN	Dynamic data
RND	Pseudo random data
EXT*n*	2 External data input port
PC	Program counter

The following subsections will describe the setup commands for PAM simulation.

Channel Rate

The PAMRZ channel data rate is programmable from 1 Hz to 1.25 MHz. For PAMNRZ and the WFG the rate can be from 1 Hz to 2.5 MHz. To set the channel rate use the command

RATE=*i.f*

where *i.f* is a number between 1 and 1250000 for PAMRZ, and is a number between 1 and 2500000 for PAMNRZ and WFG. The channel rate will then be set as near as possible to the desired rate. For example the command

```
RATE=14500
```

will set the PAM channel rate to 14.4997 kHz. The RATE command accepts the alpha characters H, K, and M which represent Hertz, Kilohertz and Megahertz, respectively. If no rate is given, the system defaults to 2K.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PAMRZ Calibration

The zero and full scale value of the PAMRZ data can be set with the command

CAL [ZERO=±*i.f* V] [FULL=±*i.f* V] [REF=±*i.f* V]

±*i.f* represents a voltage from −2.9V to +2.9V. The default value for the parameter ZERO is −1.45 volts. The default value for the parameter FULL is +2.9 volts. The parameter REF is the zero baseline or minus full scale, which defaults to −2.9 volts.

PAMNRZ and WFG Calibration

The zero and full scale value of the PAMNRZ and WFG data can be set with the command

CAL [ZERO=±*i.f* V] [FULL=±*i.f* V] [HALF=±*i.f* V]

±*i.f* represents a voltage from −2.9V to +2.9V. The default value for the parameter ZERO is −2.9 volts. The default value for the parameter FULL is +2.9 volts. The parameter HALF is the halfway point between ZERO and FULL, which defaults to 0.0 volts.

Defined Data

You may define three data words with values from 0% to 100% of the full scale PAM data. To define a particular data word the command schematic is

DATA*n*=*i.f* %

n must be an integer from 0 to 2 and the parameter *i.f* must be from 0.0 to 100.0. The percent symbol must be present. For example, to set data word number two to 57% enter the command

```
DATA2=57%
```

You may set the user-defined data words to voltages from −2.9 to +2.9 volts. To do this use the command schematic

DATA*n*=±*i.f* V

The parameter ±*i.f* may be a real number from −2.9 to +2.9. The number must be followed by the letter V.

Counters

There are two programmable counters that keep track of where the mainframe or subframe cycle is currently positioned, and can be used as a uniform increasing or decreasing data source. The counter's initial (preset) and terminal (limit) values must be specified. A counter will *increment* if its preset value is less than its limit value, and will *decrement* if the preset is greater than the limit. To set up the counters as a data source enter the command

CTR*n* [PRE=*nn*] [LIM=*nn*] [*freq*]

n must be either 1 or 2, specifying the counter number. PRE is the preset value of the counter, and *nn* may be a number from 0 to 4095, but since the data is sent to the DAC in 10-bit intervals, anything over 1023 will result in a warning message.
LIM is the limit of the counter, and *nn* may be a number from 0 to 4095. Again,

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

numbers over 1023 are valid, but will result in a warning message since they are more than 10 bits long.
The parameter *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.

Dynamic Data

The Simulator provides a Dynamic data word that has several user selectable functions. To set up this data word use the command

DYN *function* [RATE=*i.f*]

The parameter *function* may be one of the following

RMP	A Ramp function
SIN	A Sine wave
TRI	A Triangle wave
SQR	A Square wave
CAL5	A Five step Calibration sequence
CAL7	A Seven step Calibration sequence
CAL21	A Twenty-one step Calibration sequence
LOAD	A User defined function

The optional parameter RATE specifies the frequency (in cycles per second) of the function. The parameter *i.f* may be a positive real number, less than or equal to 1221, for all of the functions except LOAD. If RATE is not specified, the data read counter increments at every reference, making the wave a function of the reference rate. For example, to set up the dynamic data to be a sine wave with a period of 3.5 Hz, enter

```
DYN SIN RATE=3.5
```

The functions CAL5, CAL7, and CAL21 are PAM step functions.
CAL5 outputs five data words at equal percentage intervals: 0%, 25%, 50%, 75%, 100%.
CAL7 outputs: 0%, 20%, 40%, 50%, 60%, 80%, 100%.
CAL21 outputs: 0%, 5%, 10%, 15%, . . . 95%, 100%.

The parameter LOAD specifies a user defined function. When LOAD is used the data points of the function must be entered immediately following the DYN command line as shown below. Each data point is entered as a voltage or percentage as described previously for the defined data words.

```
DYN LOAD
10% 22.5% 90% 100% 90% 50% 10% -2V -1.3V 55% 56% 57% 58% 59% 60% 0% 10%
END
```

The END command terminates the LOAD definition. Since no RATE is specified the data counter is incremented every time the data is referenced. If a RATE is

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

specified, the number of data points will be calculated and the counter will step through the data points at the desired frequency. For the LOAD function, the frequency *i.f* must be less than or equal to 1250000 divided by the number of data words loaded.

Random Data

The Simulator has a pseudo-random number generator that can be a data word source. To use the Simulator pseudo-random data source, enter the command

RND [*recycle*]

The parameter *recycle* may be one of the following

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

Note that there is no Subframe Three recycle. This parameter specifies in which frame recycle the pseudo-random generator is reset. For example, to set up a random data word with reset at each Subframe One recycle, enter the command

```
RND SF1
```

External Ports (hardware optional feature)

The Simulator can handle two 16-bit external data ports. The command to use an external data source is

EXT*n*

The parameter *n* must be either 1 or 2, specifying which external port is to be used.

Program Counter

When the program counter is used as a data source for PAM simulation, you can record what word of the mainframe or subframe is currently being referenced. To use a program counter as a data source enter the command

PC

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PAM Example

The following sample program uses the PAM commands that have been described so far.

```
SIM PAMRZ                    | Invoke Simulator PAMRZ processor

CLR                          | Clear program and data sources

CAL ZERO=-2.0V FULL=+2.5V    | Calibrate: -2.0 volts is zero
                             | +2.5 volts is full

RATE=20K                     | Rate: 20 kHz

DATA0=.5%                    | DATA0 set to .5% of full
DATA1=20%                    | DATA1 set to 20% of full
DATA2=50%                    | DATA2 set to 50% of full

DYN CAL7                     | Dynamic data is 7 step
                             |  calibration function
```

The first line of the program tells the compiler that the following is a PAMRZ simulation setup program. Next, the clear command clears the memory and sets the Simulator default values. The range of the PAM data is then set to lie between -2.0 volts and +2.5 volts. The transmit rate is defined to be 20kHz, three user-defined data words are specified and finally, there is a dynamic data word defined to be a seven step calibration function.

Format Commands

PAM and PCM data format descriptions are the same with the following exception. The form to load DATA*n* words in the LOAD subframe statement

$$\text{DATA}\{x_1 x_2 x_3 \ldots x_{16}\}$$

uses different parameters x. If the LOAD function is used with PAMNRZ data or the WFG, and you want to use the brace format, the possible parameters are

DATA{ZFH012}

Z is ZERO, F is FULL, H is HALF, 0 is DATA0, 1 is DATA1, and 2 is DATA2. If PAMRZ is used, the possible parameters are

DATA{ZFR012}

Z is ZERO, F is FULL, R is REF, 0 is DATA0, 1 is DATA1, and 2 is DATA2. The output matrix containing any of the above references uses the mnemonics

ZR	ZERO
FL	FULL
RF	REF
HF	HALF

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PAM Programming Example

Below is the PAMNRZ format to be simulated.

```
CHNL  1 100%     CHNL 19  50%     CHNL 37  90%       CHNL 55  20%
CHNL  2  50%     CHNL 20   0%     CHNL 38  50%       CHNL 56  30%
CHNL  3  90%     CHNL 21  50%     CHNL 39 100%       CHNL 57  40%
CHNL  4  50%     CHNL 22  10%     CHNL 40 100%       CHNL 58  50%
CHNL  5  80%     CHNL 23  50%     CHNL 41   0%       CHNL 59  60%
CHNL  6  50%     CHNL 24  20%     CHNL 42   0%       CHNL 60  70%
CHNL  7  70%     CHNL 25  50%     CHNL 43 100%       CHNL 61  80%
CHNL  8  50%     CHNL 26  30%     CHNL 44  90%       CHNL 62  90%
CHNL  9  60%     CHNL 27  50%     CHNL 45  80%       CHNL 63 100%
CHNL 10  50%     CHNL 28  40%     CHNL 46  70%       CHNL 64  50%
CHNL 11  50%     CHNL 29  50%     CHNL 47  60%       CHNL 65   0%
CHNL 12  40%     CHNL 30  50%     CHNL 48  50%    -
CHNL 13  50%     CHNL 31  60%     CHNL 49  40%    s  CHNL 66   0%
CHNL 14  30%     CHNL 32  50%     CHNL 50  30%    y  CHNL 67 100%
CHNL 15  50%     CHNL 33  70%     CHNL 51  20%    n  CHNL 68 100%
CHNL 16  20%     CHNL 34  50%     CHNL 52  10%    c  CHNL 69 100%
CHNL 17  50%     CHNL 35  80%     CHNL 53   0%       CHNL 70  50%
CHNL 18  10%     CHNL 36  50%     CHNL 54  10%    -
```

There are 70 channels (or words) that must be simulated for this format. The program listing for this format is shown below. The PAMNRZ calibration data words are represented as ZERO, FULL, and HALF.

```
SIM PAMNRZ
CLR
RATE=25k
DYN LOAD
90% 80% 70% 60% 40% 30% 20% 10% 10% 20% 30%
40% 60% 70% 80% 90% 90% 80% 70% 60% 40% 30%
20% 10% 10% 20% 30% 40% 60% 70% 80% 90%
END
FORMAT LEN=70
 FILL DYN
 WORD 2 4 6 8 10 11 13 15 17 19 21 23 HALF
 WORD 25 27 29 30 32 34 36 38 48 58 64 HALF
 WORD 1 39 40 43 63 FULL
 WORD 20 41 42 53 65 ZERO
END FORMAT
END SIM
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SIMULATOR FORMAT A:

MAINFRAME:

WORD:	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	FL	HF	DN	HF	DN	HF	DN	HF	DN	HF	HF	DN	HF	DN
WORD:	15	16	17	18	19	20	21	22	23	24	25	26	27	28
	HF	DN	HF	DN	HF	ZR	HF	DN	HF	DN	HF	DN	HF	DN
WORD:	29	30	31	32	33	34	35	36	37	38	39	40	41	42
	HF	HF	DN	HF	DN	HF	DN	HF	DN	HF	FL	FL	ZR	ZR
WORD:	43	44	45	46	47	48	49	50	51	52	53	54	55	56
	FL	DN	DN	DN	DN	HF	DN	DN	DN	DN	ZR	DN	DN	DN
WORD:	57	58	59	60	61	62	63	64	65	66	67	68	69	70
	DN	HF	DN	DN	DN	DN	FL	HF	ZR	ZR	FL	FL	FL	HF

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX C
SIMULATOR COMMAND INDEX

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX C
SIMULATOR COMMAND INDEX

SIM *function*
Begin Simulator Programming, function=PCM/MEN/EDIT, requires END.

CLR
Clear data to defaults and clear formats.

DATA SETUP COMMANDS

CODE *type*
Set output code, type=NRZL/NRZM/NRZS/BIPL/BIPM/BIPS/
DBIPM/DBIPS/DMM/DMS/M2M/M2S/RNRZ11/RNRZ15/RNRZ17/RNRZ23

CTR*n* [PRE=*nn*] [LIM=*nn*] [*freq*] [LEN*n*=*nn*] [*orien*] [*parity*]
Setup Counters 1 & 2, Pre=0-65535, Lim=0-65535, freq=REF/MFR/SF1/SF2

DATA*nn*[=*val*] [LEN=*nn*] [*orien*] [*parity*]
Set data word 0-1023 values entered in decimal or preceeded by / for hex value.

DYN*n* [*function*] [*type*] [RATE=*i.f*] [LEN=*nn*] [*orien*] [*parity*]
Load canned function into Dynamic Memory 1 & 2, function=RMP/SIN/TRI/SQR,
type=1CM/2CM/SMG/OBN, RATE=EXT (for external clock) or .000003-12203.

DYN*n* LOAD [RATE=*i.f*] [LEN=*nn*] [*orien*] [*parity*]
Load user values into Dynamic Memory 1 & 2 from the following lines, requires END.

RPT *nn* PCM data points
Used only for Dynamic Load. Repeat nn times the following data points.

EPA
Set all output parity Even .

EXT1 [LEN*n*=*nn*] [*orien*] [*parity*]
External Data Source one.

EXT2 [*type*] [LEN=*nn*] [*orien*] [*parity*]
External Data Source two, type=NOR or FBD for frame buffered data.

OPA
Set all output parity Odd.

PC [LEN*n*=*nn*] [*orien*] [*parity*]
Program Counter.

RATE=*i.f*
Set Data Rate, for Rate=0.6Hz - 20MHz.

DEFINERS

LEN — Bit length of the output word,
maximum is 16 for all except fixed data words whose max. is 32.

orien — Orientiation of the output word, MSB or LSB.

parity — LPA (leading), TPA (trailing), NPA (no parity).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

FORMAT SETUP COMMANDS

FORMAT [*A/B*] [LEN=*nn*]
Setup Format A or B, Len=1-4096.

FILL *source*
Fill the format with the data source.

FOR nn_1 [nn_2-nn_3 nn_4 ... nn_{20}]
For subframe word nn.

SF*n* CTR*n* [MFCOMP]
Begin formatting Subframe 1-3, the depth is defined by the counter,
optional subcommutated mainframe complement sync, requires END.

SF*n* CTR*n* LOAD
Begin formatting Subframe 1-3, the depth is defined by the counter,
linear load of data sources into the subframe format, requires END.

SF*n* DEPTH=*nn* [MFCOMP]
Begin formatting Subframe 1-3,
Optional subcommutated mainframe complement sync, requires END.

SF*n* [DEPTH=*nn*] LOAD
Begin formatting Subframe 1-3,
Linear load of data sources into the subframe format, requires END.

RPT *nn* $source_1$ ...$source_{nn}$
Used only with Subframe Load. Repeat nn times the following sources listed on the line.

DATA{$x_1 x_2 x_3 ... x_{nn}$}
Used only with Subframe Load, load Fixed Data words.

WORD nn_1 [nn_2 nn_3-nn_4 ... nn_{20}] *source*
Specify frame word data source.

RUN MODE CONTROL COMMANDS

INS
Insert a frame from the alternate format.

RUN [*setup*]
Run format setup A or B.

STOP
Stop simulator.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP