Programming Reference

HP 16520A/16521A
Pattern Generator Module

for the HP 16500A Logic Analysis System

HEWLETT
PACKARD

© Copyright Hewlett-Packard Company 1988

Manual Part Number 16520-90908

Printed in U.S.A. September 1988

Product Warranty

This Hewlett-Packard product has a warranty against defects in material and workmanship for a period of three years from date of shipment. During warranty period, Hewlett-Packard Company will, at its option, either repair or replace products that prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by Hewlett-Packard. However, warranty service for products installed by Hewlett-Packard and certain other products designated by Hewlett-Packard will be performed at the Buyer's facility at no charge within the Hewlett-Packard service travel area. Outside Hewlett-Packard service travel areas, warranty service will be performed at the Buyer's facility only upon Hewlett-Packard's prior agreement and the Buyer shall pay Hewlett-Packard's round trip travel expenses.

For products returned to Hewlett-Packard for warranty service, the Buyer shall prepay shipping charges to Hewlett-Packard and Hewlett-Packard shall pay shipping charges to return the product to the Buyer. However, the Buyer shall pay all shipping charges, duties, and taxes for products returned to Hewlett-Packard from another country.

Hewlett-Packard warrants that its software and firmware designated by Hewlett-Packard for use with an instrument will execute its programming instructions when properly installed on that instrument. Hewlett-Packard does not warrant that the operation of the instrument software, or firmware will be uninterrupted or error free.

Limitation of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by the Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. HEWLETT-PACKARD SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Exclusive Remedies

THE REMEDIES PROVIDED HEREIN ARE THE BUYER'S SOLE AND EXCLUSIVE REMEDIES. HEWLETT-PACKARD SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

Assistance

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For any assistance, contact your nearest Hewlett-Packard Sales and Service Office.

Certification

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Bureau of Standards, to the extent allowed by the Bureau's calibration facility, and to the calibration facilities of other International Standards Organization members.

Safety

This product has been designed and tested according to International Safety Requirements. To ensure safe operation and to keep the product safe, the information, cautions, and warnings in this manual must be heeded.

Table of Contents

Chapter 1

1 Programming the HP 16520A/16521A

Chapter 2

Module Level Commands

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Appendix A

Programming the HP 16520A/16521A

1

Introduction

This manual combined with the *HP 16500A Programming Reference* manual provides you with the information needed to program the HP 16520A/HP 16521A pattern generator module. Each module has its own manual to supplement the mainframe manual since not all mainframes will be configured with the same modules.

About This Manual

This manual is organized into six chapters. The first chapter contains:

- General information and instructions to help you get started
- Mainframe system commands that are frequently used with the pattern generator module
- HP 16520A/HP 16521A Pattern Generator command tree
- Alphabetic command-to-subsystem directory

Chapter two contains the module level commands. Chapters three through six contain the subsystem commands for the pattern generator.

Chapter six contains the Symbol Subsystem.

Error messages for the HP 16520A are included in generic system error messages and are in the *HP 16500A Programming Reference* manual.

Appendix A contains information on the SYSTem:DATA and SYSTem:SETup commands for this module.

Programming the HP 16520A Pattern Generator

This section introduces you to the basic command structure used to program the pattern generator. Also included is an example program that uses the two pods of the master card.

Selecting the Module

Before you can program the pattern generator, you must first "select" it, otherwise, there is no way to direct your commands to the pattern generator.

To select the module, use the system command :SELect , followed by t numeric reference for the slot location of the pattern generator(1...5 re to slot A...E respectively). For example, if the pattern generator master card is in slot E, then the command:

```
:SELect 5
```

would select this module. For more information on the select command, refer to the *HP 16500A Programming Reference Manual.*

Programming the Pattern Generator

A typical pattern generator program includes the following tasks:

- select the appropriate module
- set program parameters
- define a pattern generator program
- run the pattern generator program

The following example program generates a pattern using the two output pods of the master card:

```
10 OUTPUT XXX;":SELECT 1"
20 OUTPUT XXX;":FORMAT:REMOVE ALL"
30 OUTPUT XXX;":FORMAT:LABEL 1 'A',POSITIVE,127,0"
40 OUTPUT XXX;":FORMAT:LABEL 'B',POSITIVE,0,255"
50 OUTPUT XXX;":LIST:REMOVE ALL"
60 OUTPUT XXX;":LIST:PROG 1,NOOP,'#H7F','#HFF'"
70 OUTPUT XXX;":RMODE REPETITIVE"
80 OUTPUT XXX;":START"
90 END
```

Note

The three Xs (XXX) after the OUTPUT statement in the above example refer to the device address required for programming over either HP-IB or RS-232-C. Refer to your controller manual and programming language reference manual for information on initializing the interface.

Program Comments

Line 10 selects the pattern generator in slot A
Line 20 removes all labels previously assigned
Line 30 assigns label 'A', the output polarity and defines active channels of pod A3
Line 40 assigns label 'B' and defines active channels of pod A2
Line 50 removes all program lines
Line 60 lists the first line of the pattern generation program. Channel data may be specified in binary, octal, decimal, hexadecimal.
Line 70 Sets the RMODE to repetitive. If the program is to be run only once, select the :RMODE SINGLE command.
Line 80 Starts the program.

For more information on the specific pattern generator commands, refer to chapters two through six of this manual.

Mainframe Commands

These commands are part of the HP 16500A mainframe system and are mentioned here only for reference. For more information on these commands, refer to the *HP 16500A Programming Reference* manual.

CARDcage? Query

The CARDcage query returns a string which identifies the modules that are installed in the mainframe. The returned string is in two parts. The first five two-digit numbers identify the card type. The identification number for the HP 16520A pattern generator is 21 and the HP 16521A identification number is 22. A "-1" in the first part of the string indicates no card is installed in the slot.

The five single-digit numbers in the second part of the string indicate in which slots cards are installed and where the master card is located.

Example:

```
11,12,-1,-1,31,1,1,0,0,5
```

A returned string of 11,12,-1,-1,31,1,1,0,0,5 means that an oscilloscope timebase card (ID number 11) is loaded in slot A and the oscilloscope acquisition card (ID number 12) is loaded in slot B. The next two slots (C and D) are empty (-1). Slot E contains a logic analyzer module (ID number 31).

The next group of numbers (1,1,0,0,5) indicate that a two card module is installed in slots A and B with the master card in slot A. The "0" indicates an empty slot or the module software is not recognized or not loaded. The last digit (5) in this group indicates a single module card is loaded in slot E. Complete information for the CARDcage query is in the *HP 16500A Programming Reference* manual.

MENU
Command/query

The MENU command selects a new displayed menu. The first parameter (X) specifies the desired module. The optional second parameter specifies the desired menu in the module (defaults to 0 if not specified). The query returns the currently selected (and displayed) menu.

For the HP 16520A/HP 16521A Pattern generator:

- X,0 - Format Menu
- X,1 - Listing Menu
- X,2 - Macro 1 Menu
- X,3 - Macro 2 Menu
- X,4 - Macro 3 Menu
- X.5 - Macro 4 Menu

X = slot number that contains the pattern generator master card.

SELect
Command/query

The SELect command selects which module or intermodule will have parser control. SELect 0 selects the intermodule, SELect 1 through 5 selects modules A through E respectively. Parameters -1 and -2 select software options 1 and 2. The SELect query returns the currently selected module.

STARt
Command

The STARt command starts the specified module or intermodule. If the specified module is configured for intermodule, STARt will start all modules configured for intermodule.

STOP
Command

The STOP command stops the specified module or intermodule. If the specified module is configured for intermodule, STOP will stop all modules configured for intermodule.

RMODe
Command/query

The RMODe command specifies the run mode (single or repetitive) for a module or intermodule. If the selected module is configured for intermodule, the intermodule run mode will be set by this command. The RMODe query returns the current setting.

SYSTem:ERRor? Query

The SYSTem:ERRor query returns the oldest error in the error queue. In order to return all the errors in the error queue, a simple FOR/NEXT loop can be written to query the queue until all errors are returned. Once all errors are returned, the queue will return zeros.

SYSTem:PRINt Command/query

The SYSTem:PRINt command initiates a print of the screen or listing buffer over the current printer communication interface. The SYSTem:PRINt query sends the screen or listing buffer data over the current controller communication interface.

MMEMory Subsystem

The MMEMory Subsystem provides access to both internal disc drives for loading and storing configurations.

INTermodule Subsystem

The INTermodule Subsystem commands are used to specify intermodule arming between multiple modules.

Command Set Organization

The command set for the HP 16520A is divided into four separate subsystems. The subsystems are: FORMat, LISTing, MACRo, and the SYMBol subsystem. Each of the subsystems commands are covered in their individual chapters starting with chapter 2.

Each of these chapters contains a description of the subsystem, syntax diagrams and the commands in alphabetical order. The commands are shown in longform and shortform using upper and lowercase letters. For example, FORMat indicates that the longform of the command is FORMAT and the shortform is FORM. Each of the commands contain a description of the command and its arguments, the command syntax, and a programming example.

Figure 1-1 is the command tree for the HP 16520A pattern generator module.

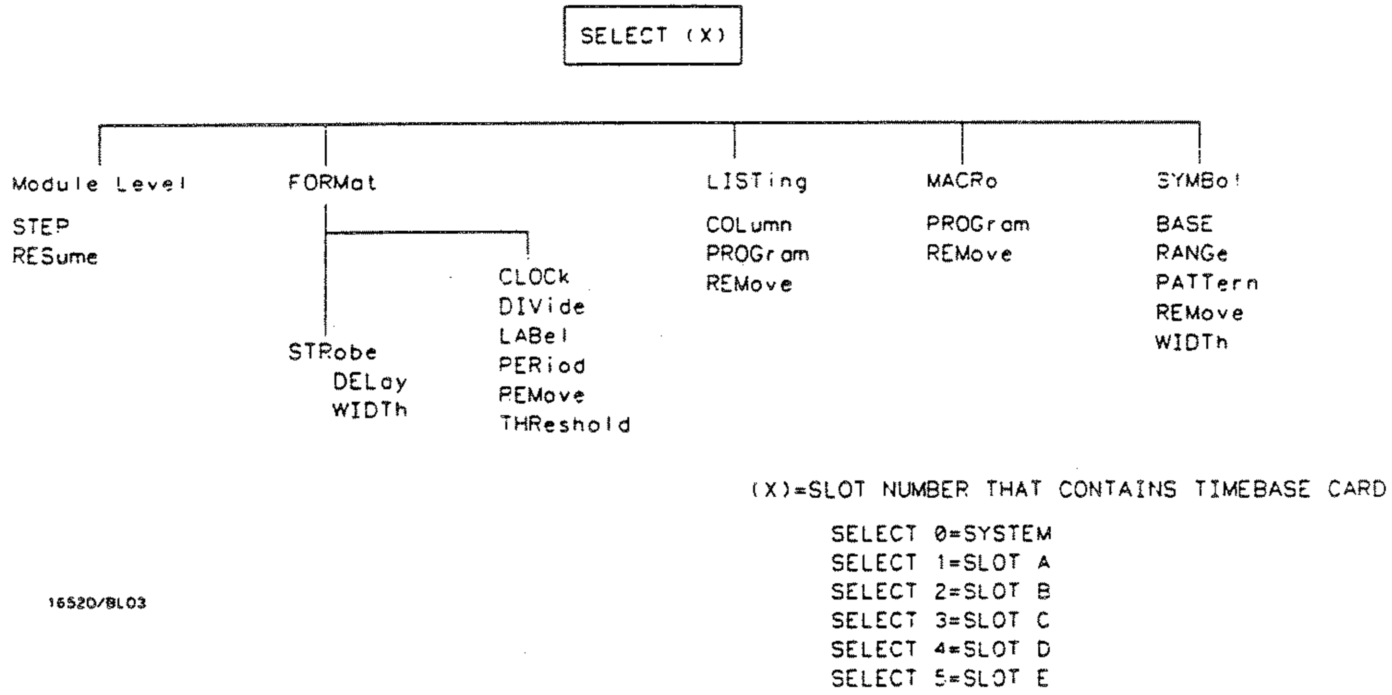
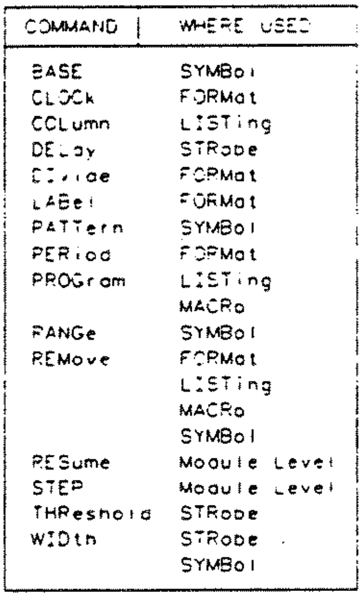


Figure 1-1. HP 16520A/HP 16521A Command Tree

Table 1-1. Alphabetical Command to Subsystem Directory.



COMMAND	WHERE USED
BASE	SYMBol
CLOCk	FORMat
CCLumn	LISTing
DELay	STRobe
DIVide	FORMat
LABel	FORMat
PATTern	SYMBol
PERiod	FORMat
PROGram	LISTing MACRo
RANGe	SYMBol
REMove	FORMat LISTing MACRo SYMBol
RESume	Module Level
STEP	Module Level
THReshold	STRobe
WIDth	STRobe SYMBol

16520/BL02

Module Status Reporting

Each module reports its status to the Module Event Status Register (MESR < N >) which in turn reports to the Combined Event Status Register (CESR) in the HP 16500A mainframe (see *HP 16500A Programming Reference* manual). The Module Event Status Register is enabled by the Module Event Status Enable Register (MESE < N >).

The following descriptions of the MESE < N > and MESR < N > commands provide the module specific information needed to enable and interpret the contents of the registers

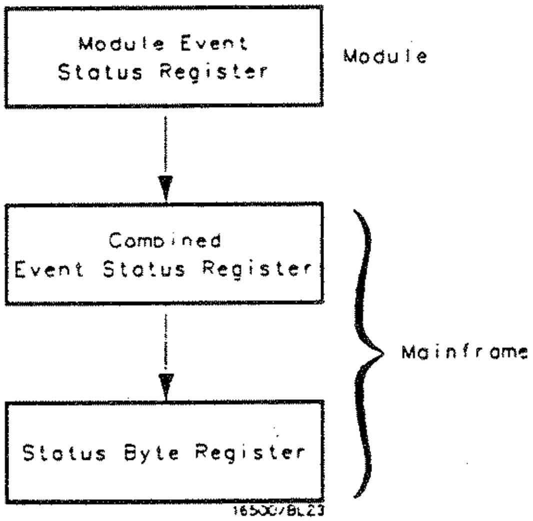


Figure 1-2. Module Status Reporting

MESE<N>

command/quer

The MESE<N> command sets the Module Event Status Enable registe bits. The MESE register contains a mask value for the bits enabled in the MESR register. A one in the MESE will enable the corresponding bit in the MESR register; a zero will disable the bit.

The first parameter after the command specifies the module (<N> = 1...5 refers to the module in slot A...E). The second parameter specifies the enable value.

The MESE query returns the current setting.

Refer to table 1-2 for information about the Module Event Status register bits, bit weights, and what each bit masks in the module. Complete information for status reporting is in the *HP 16520A Programming Reference* manual.

Command Syntax: :MESE<N> <enable_mask>

where:

<N> ::= {1|2|3|4|5} number of slot in which the module resides
<enable_mask> ::= integer 0 to 255

Example: OUTPUT XXX;":MESE5 2"

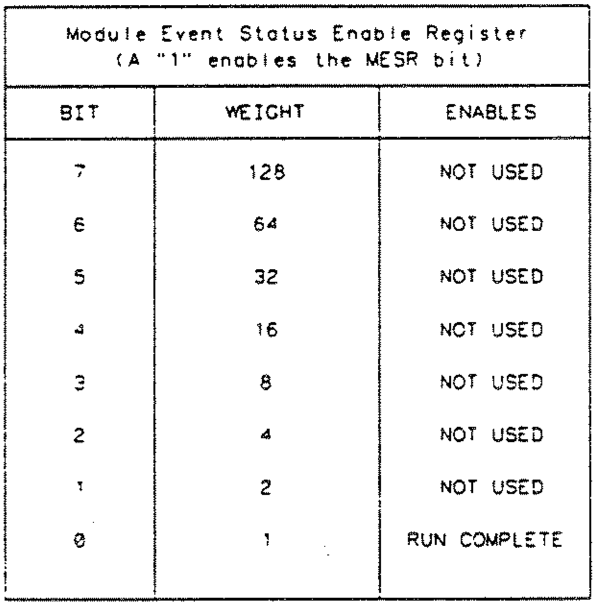
Query Syntax: :MESE<N>?

Returned Format: [MESE] <enable_mask> <NL>

Example:

```
10 OUTPUT XXX;":MESE2?"
20 ENTER XXX; Mes
30 PRINT Mes
40 END
```

Table 1-2. Module Event Status Enable Register



Module Event Status Enable Register (A "1" enables the MESR bit)		
BIT	WEIGHT	ENABLES
7	128	NOT USED
6	64	NOT USED
5	32	NOT USED
4	16	NOT USED
3	8	NOT USED
2	4	NOT USED
1	2	NOT USED
0	1	RUN COMPLETE

16520/BL04

The Module Event Status Enable Register contains a mask value for the bits to be enabled in the Module Event Status Register (MESR). A one in the MESE enables the corresponding bit in the MESR, a zero disables the bit.

MESR < N >

query

The MESR < N > query returns the contents of the Module Event Status register.

Note

Reading the register clears the Module Event Status Register.

Table 1-3 shows each bit in the Module Event Status Register and their bit weights for this module. When you read the MESR, the value returned is the total bit weights of all bits that are high at the time the register is read.

The parameter 1...5 refers to the module in slot A...E respectively.

Query Syntax: :MESR < N > ?

Returned Format: [MESR] < status > < NL >

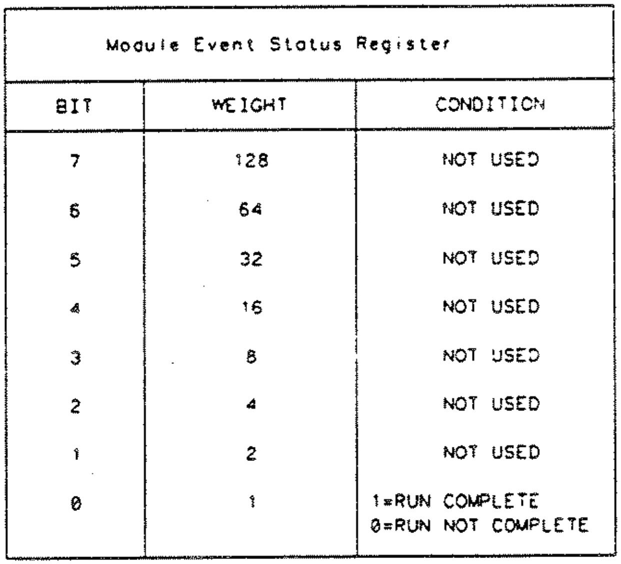
where:

< N > ::= {1|2|3|4|5} number of slot in which the module resides
< status > ::= 0 to 255

Example:

```
10 OUTPUT XXX;":MESR2?"
20 ENTER XXX; Mer
30 PRINT Mer
40 END
```

Table 1-3. Module Event Status Register



Module Event Status Register

BIT	WEIGHT	CONDITION
7	128	NOT USED
6	64	NOT USED
5	32	NOT USED
4	16	NOT USED
3	8	NOT USED
2	4	NOT USED
1	2	NOT USED
0	1	1=RUN COMPLETE 0=RUN NOT COMPLETE

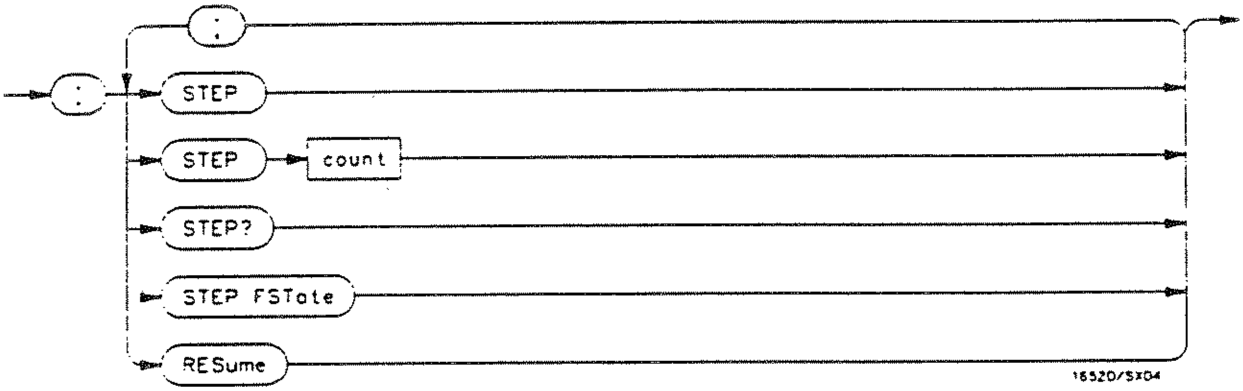
16520/BL05

Note

The MESR bit will be set at the end of the program or if a BREAK instruction is encountered within the program.

Module Level Commands 2

The Module Level Commands control the operation of pattern generator programs. The two Module Level Commands are STEP and RESume. Refer to figure 2-1 for the Module Level Syntax Diagram.



count = *an integer from 1 to 999, specifying number of program lines*

Figure 2-1. Module Level Syntax Diagram

STEP

Command/Query

The STEP command consists of four parts: the STEP command, the STEP Count command, the STEP query , and the STEP FSTate command .

The STEP command causes the pattern generator to step through the number of program lines specified by the STEP Count command. The valid program line range for the STEP Count command is from 1 to 999. The STEP count query returns the current count.

The STEP FSTate (step first state) command allows you to return to program line 0 of the current program.

Command Syntax:

For the STEP command `:STEP`

Example: `OUTPUT XXX;":STEP"`

Command Syntax:

For STEP Count command `STEP <count>`

where:

<count> ::= an integer from 1 to 999, specifying the number of program lines.

Example: `OUTPUT XXX;":STEP 20"`

STEP

Query Syntax: :STEP?

Returned Format: [STEP] <count>

Example:

```
10 DIM Sc$[100]
20 OUTPUT XXX;":STEP?"
30 ENTER XXX;Sc$
40 PRINT Sc$
50 END
```

Command Syntax:

For STEP FSTate command: :STEP FSTate

Example: OUTPUT XXX;":STEP FSTATE"

RESume

Commanc

When the pattern generator encounters a BREAK instruction, program execution is halted. The RESume command allows the program to continue until another BREAK instruction is encountered, or until the end of the program is reached.

Command Syntax: :RESume

Example: OUTPUT XXX;":RESUME"

FORMat Subsystem 3

Introduction

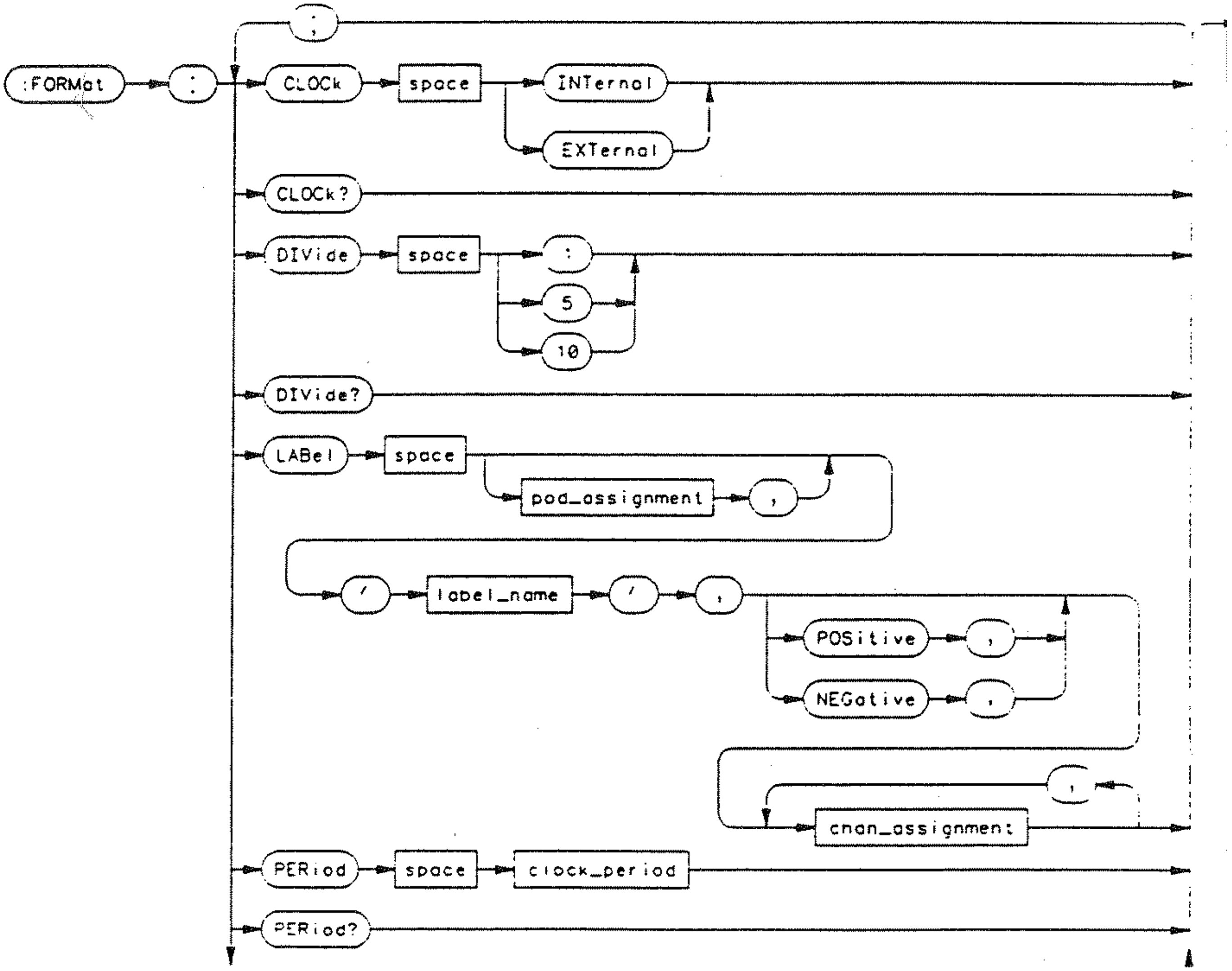
The commands of the Format subsystem control the pattern generator values such as data output rate, strobe width and delay, and the channels that you want to be active. The Format subsystem also lets you specify the clock source and allows you to group channels together under a common, user - defined name. Refer to Figure 3-1 for the Format subsystem syntax diagrams.

Strobe Sublevel Set

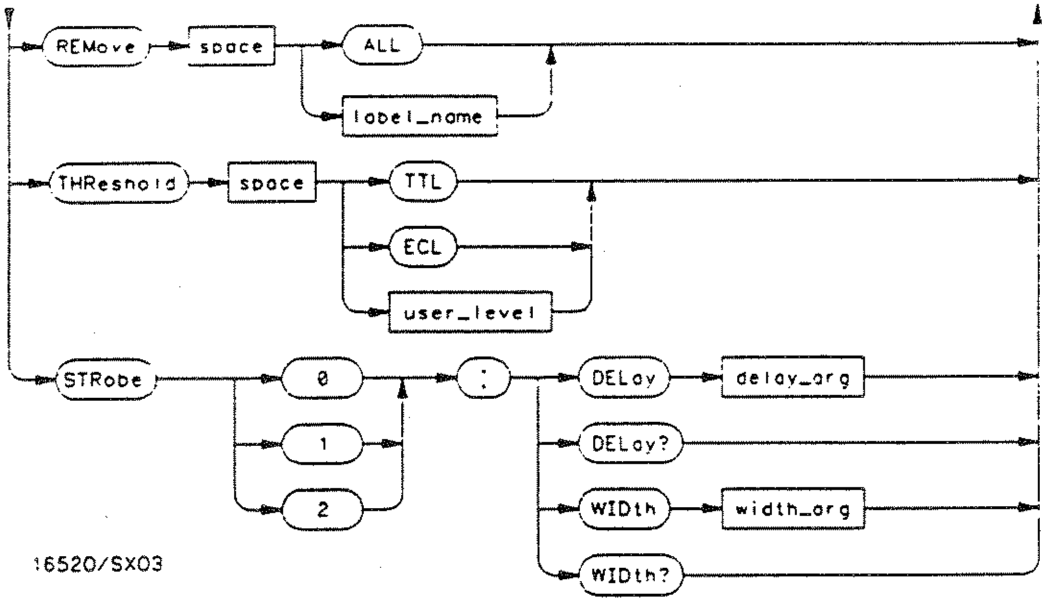
The commands of the Strobe sublevel are part of the Format subsystem and are used to set the strobe delay and strobe width.

Each pattern generator master card has three strobe outputs. The strobe outputs are data channels with selectable pulse width and pulse delay. While standard data channels can change state only at the start of an output clock cycle, strobes can change state after the clock transition or can change state in the middle of a clock cycle.

If the polarity of a strobe channel needs to be changed, use the LABEL command. To specify the polarity of strobe channels individually, rename each strobe under a different label.



P/O Figure 3-1. Format Subsystem Syntax Diagram



label name = *a string of up to 6 alphanumeric characters*
chan_assignment = *an integer from 0 to 255*
clock period = *a real number from 20 ns through 200 us*
user_level = *a voltage level from -9.9 V to + 9.9 V*
delay_arg = *a real number specifying strobe delay time*
width_arg = *a real number specifying strobe width*

P/O Figure 3-1. Format Subsystem Syntax Diagram

CLOCk

Command/Quer

The CLOCk command specifies the clock source for the pattern generator. The choices are INTernal or EXTernal. The clock specified by this command is the output data clock. Each time a new clock period starts, the pattern generator outputs go to their next state, as defined by the program listing. The internal clock pulse period may be varied using the PERIOD command. The maximum external clock rate is 50 MHz. The query returns the current clock choice.

Command Syntax: :FORMat:CLOCk {INTernal|EXTernal}

Example: OUTPUT XXX;":FORMat:CLOCK INTERNAL"

Query Syntax: :FORMat:CLOCk?

Returned Format: [:FORMat:CLOCk] {INTernal|EXTernal} < NL >

Example:

```
10 DIM Cl$[100]
20 OUTPUT XXX;":FORMAT:CLOCK?"
30 ENTER XXX;Cl$
40 PRINT Cl$
50 END
```

DIVide

Command/Query

The DIVide command allows you to divide the external clock frequency by 1, 5 or 10. When divide by 1 is chosen, the output strobes are not available. The divide by 5 and divide by 10 division parameters determine the resolution by which the strobe width and delay paramcters can be set. In divide by 5, the width and delay may be set in 1/5 increments of the clock period. In divide by 10, the width and delay may be set in 1/10 increments of the clock period. The query returns the current division ratio.

Command Syntax: :FORMat:DIVide <divide by ratio>

where

<divide by value> ::= 1, 5, or 10

Example:

```
OUTPUT XXX;":FORM:DIV 1"
```

Query Syntax: :FORMat:DIVide?

Returned Format: [FORMat:DIVide] <divide by ratio> <NL>

Example:

```
10 DIM Di$[100]
20 OUTPUT XXX;":FORMAT:DIVIDE?"
30 ENTER XXX;Di$
40 PRINT Di$
50 END
```

LABel

Command/Quer\

The LABel command inserts a new label or modifies the contents of an existing label. If more than 20 labels are specified, and an attempt is made to insert another new label, the last label (bottom label) will be modified.

Stimulus channels can be assigned to only one label at a time. If duplicate assignments are made, the last channel assignments take precedence.

The first parameter is optional and is used to specify the first pod that is to have channels assigned. If the first parameter choice is not made, then the STROBE/DATA pod of the master card is assumed. The pods are numbered in the same order as they appear in the format menu, with zero representing the STROBE/DATA pod of the master card. The second parameter sets the channel polarity. If the polarity is not specified, the last polarity assignment is used. The last parameters assign the active chann-'- for each pod.

Each assignment parameter is a binary encoding of the channel assignments of the pod. A "1" in a bit position means that the associated channel in that pod is included in the label. A "0" in a bit position excludes the channel from the label. The minimum value for any pod specification is 0, the maximum value for all pods except the STROBE/DATA pod is 255. The maximum value for the STROBE/DATA pod is 127. A value of 255 includes all channels of a pod assignment. The query must specify a label name and returns the current pod assignments and channel polarity for that label.

Command Syntax: :FORMat:LABel [<pod assignment>,]<label name>[<polarity>,]<channel assignment>, ... <channel assignment>

where:

<pod assignment> ::= an integer from 0 to 26, depending on how many expansion cards are used.
<label name> ::= string of up to 6 alphanumeric characters
< polarity> ::= polarity of the channel outputs,NEGative or POSitive

<channel assignment> ::= a string in one of the following forms:
'B01...' for binary
'#Q01234567..'for octal
'#H0123456789ABCDEF... for hexadecimal
'0123456789...' for decimal.

Example:
```
OUTPUT XXX;":FORMAT:LABEL 1,'A',POSITIVE,255,0"
```

Query Syntax: :FORMat:LABel? <label name>

Returned Format: [:FORMat:LABel] <label name>,<polarity>,<channel assignment>, ... <channel assignment> <NL>

Example:
```
10 DIM La$[100]
20 OUTPUT XXX;":FORMAT:LABEL? 'A'"
30 ENTER XXX;La$
40 PRINT La$
50 END
```

PERiod

Command/Query

The PERiod command specifies the internal clock period. The range limits are from 20 ns to 200 us in a 1, 2, 5 sequence. The query returns the current clock period.

Command Syntax: :FORMat:PERiod <clock period>

where:

<clock period> ::= a real number from 20 ns to 200 us, in a 1, 2, 5 sequence

Example:

```
OUTPUT XXX;":FORMAT:PERIOD 1.0E-6"
```

Query Syntax: :FORMat:PERiod?

Returned Format: [:FORMAT:PERIOD] <clock period> <NL>

Example:

```
10 DIM Cp$[100]
20 OUTPUT XXX;":FORMAT:PERIOD?"
30 ENTER XXX;Cp$
40 PRINT Cp$
50 END
```

REMove

Command

The REMove is used to delete a single label, or all labels from the format menu. If a label name is specified, it must match a label name currently active in the format menu.

Command Syntax:

```
:FORMat:REMove {ALL| <label name> }
```

Example:

```
OUTPUT XXX;":FORMAT:REMOVE ALL"
```

THReshold

Command/Query

The THReshold command sets the input threshold levels for the pattern generator input pod. The selection may be TTL, ECL or User Defined. The user defined input may range from -9.9 V to +9.9 V. The query returns the current setting.

Command Syntax:

```
:FORmat:THReshold {TTL|ECL|<value>}
```

where:

<value> ::= voltage (real number) -9.9 to +9.9

Example

```
OUTPUT XXX;":FORMAT:THRESHOLD 5.2V"
```

Query Syntax:

```
:FORMAT:THRESHOLD?
```

Returned Format:

```
[FORMAT:THRESHOLD] <value>
```

Example:

```
10 DIM Th$[100]
20 OUTPUT XXX;":FORMAT:THRESHOLD?"
30 ENTER XXX;Th$
40 PRINT Th$
50 END
```

STRobe

Selector

The STRobe Selector is used as part of a compound header to set the output strobe parameters.

It always follows the FORMAT selector because it selects a branch below the Format level in the command tree. When setting strobe parameters, the strobe number must always be specified (strobe 0 through strobe 2).

Command Syntax: :FORMat:STRobe < strobe number > : < strobe parameter >

where:

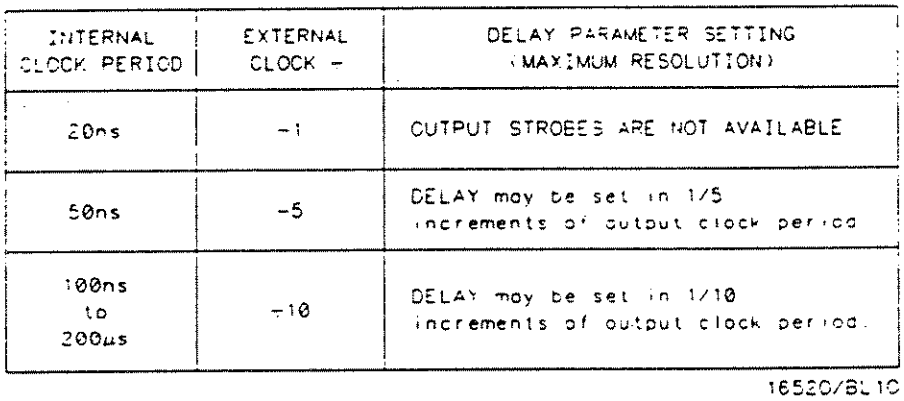
< strobe number >	:: = strobe number 0, 1, or 2
< strobe parameter >	:: = strobe parameter may either be DELay or WIDTh command

Example: OUTPUT XXX;":FORMAT:STROBE1:DELAY 10E-9"

DELay

Command/Query

The DELay command sets the strobe delay value. The delay value set is with respect to the rising edge of the output clock. In other words, the delay value tells the pattern generator to delay the start of the strobe from the rising edge of the output clock. The output strobes are not available at the clock period of 20 ns or external clock divide by 1. Strobe delay time and strobe width are related to the clock period. The delay time plus width can not exceed the one clock period. The delay parameters may be set as shown in Table 3-1. The query reports the current delay setting.



INTERNAL CLOCK PERIOD	EXTERNAL CLOCK ÷	DELAY PARAMETER SETTING (MAXIMUM RESOLUTION)
20ns	÷1	OUTPUT STROBES ARE NOT AVAILABLE
50ns	÷5	DELAY may be set in 1/5 increments of output clock period
100ns to 200µs	÷10	DELAY may be set in 1/10 increments of output clock period.

16520/BL10

Table 3-1. Strobe DELAY Parameter Setting

Command Syntax: :FORMat:STRobe < strobe number > :DELay < delay value >

where:

< strobe number > ::= strobe number 0, 1, or 2

< delay value > ::= a real number from 0 to the current output data rate if internally clocked

or

an integer between 0 and 9 if output is externally clocked

Example:

```
OUTPUT XXX;":FORMAT:STROBE1:DELAY 1E-6"
```

Query Syntax: FORMat:STRobe < strobe number > :DELay?

Returned Format: [FORMAT:STROBE1 < strobe number > :DELAY] < delay value > < NL >

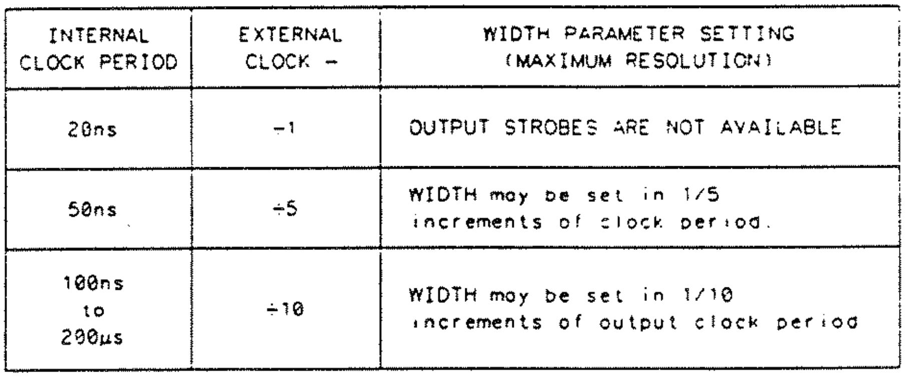
Example:

```
10 DIM Sd$[100]
20 OUTPUT XXX;":FORMAT:STROBE1:DELAY?"
30 ENTER XXX;Sd$
40 PRINT Sd$
50 END
```

WIDTh

Command/Quer

The WIDTh command sets the strobe width value. The width parameter are set in the same manner as the strobe delay parameters. Refer to Table 3-2 for an explanation of the strobe width parameter settings. The query returns the current width setting.



INTERNAL CLOCK PERIOD	EXTERNAL CLOCK ÷	WIDTH PARAMETER SETTING (MAXIMUM RESOLUTION)
20ns	÷1	OUTPUT STROBES ARE NOT AVAILABLE
50ns	÷5	WIDTH may be set in 1/5 increments of clock period.
100ns to 200µs	÷10	WIDTH may be set in 1/10 increments of output clock period

16520/BLC9

Table 3-2. Strobe WIDTH Parameter Settings

Command Syntax: :FORMat:STRobe<strobe number>:WIDTh <width value>

where:

<strobe number>	::= strobe number 0, 1, 2
<width value>	::= a real number between 0 and the output data rate if internal clock is used or an integer between 0 and 10 if external clock is used

Example: OUTPUT XXX;":FORMAT:STROBE0:WIDTH 500ns"

WIDTh

Query Syntax: :FORMat:STRobe:WIDTh?

Returned Format: [FORMAT:STROBE<strobe number>:WIDTh]<width value><NL>

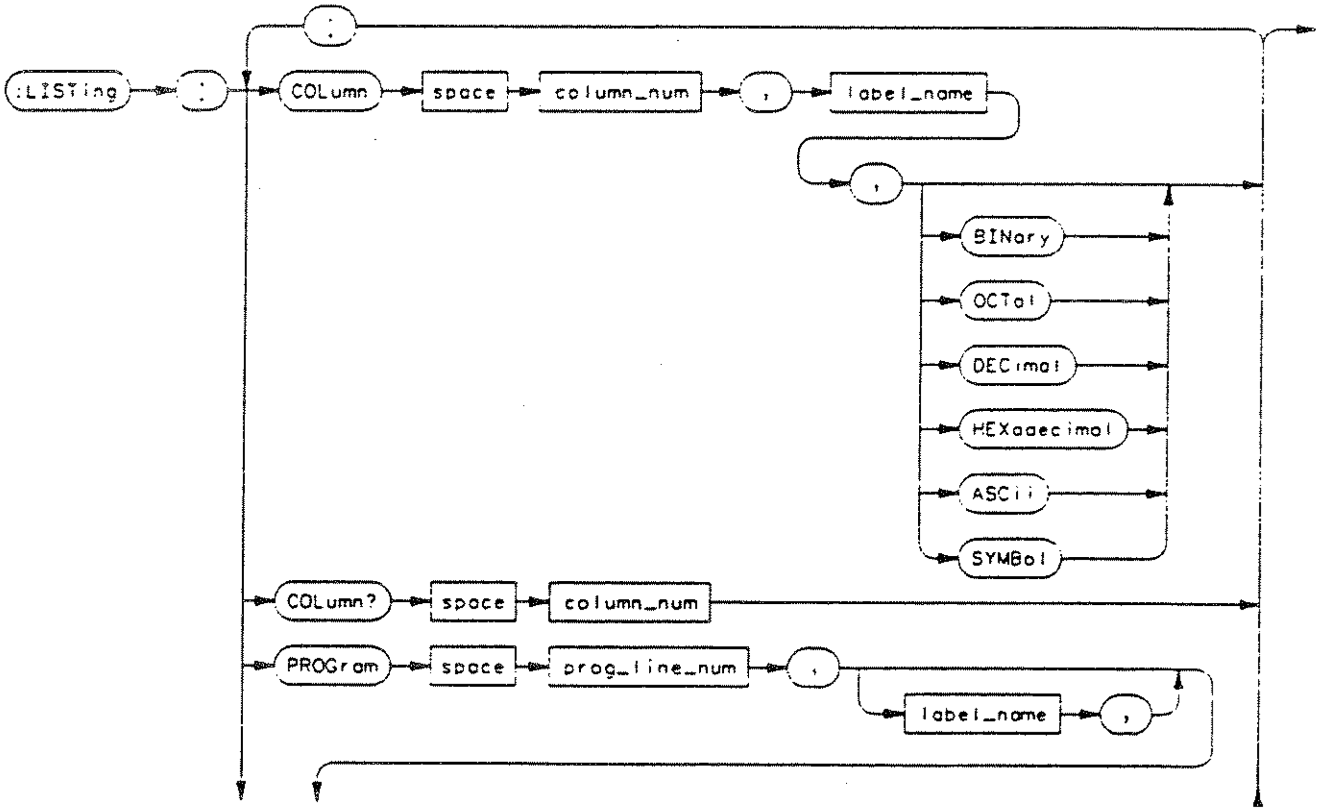
Example:

```
10 DIM Wv$[100]
20 OUTPUT XXX;":FORMAT:STROBE1:WIDTH 500ns"
30 ENTER XXX;Wv$
40 PRINT Wv$
50 END
```

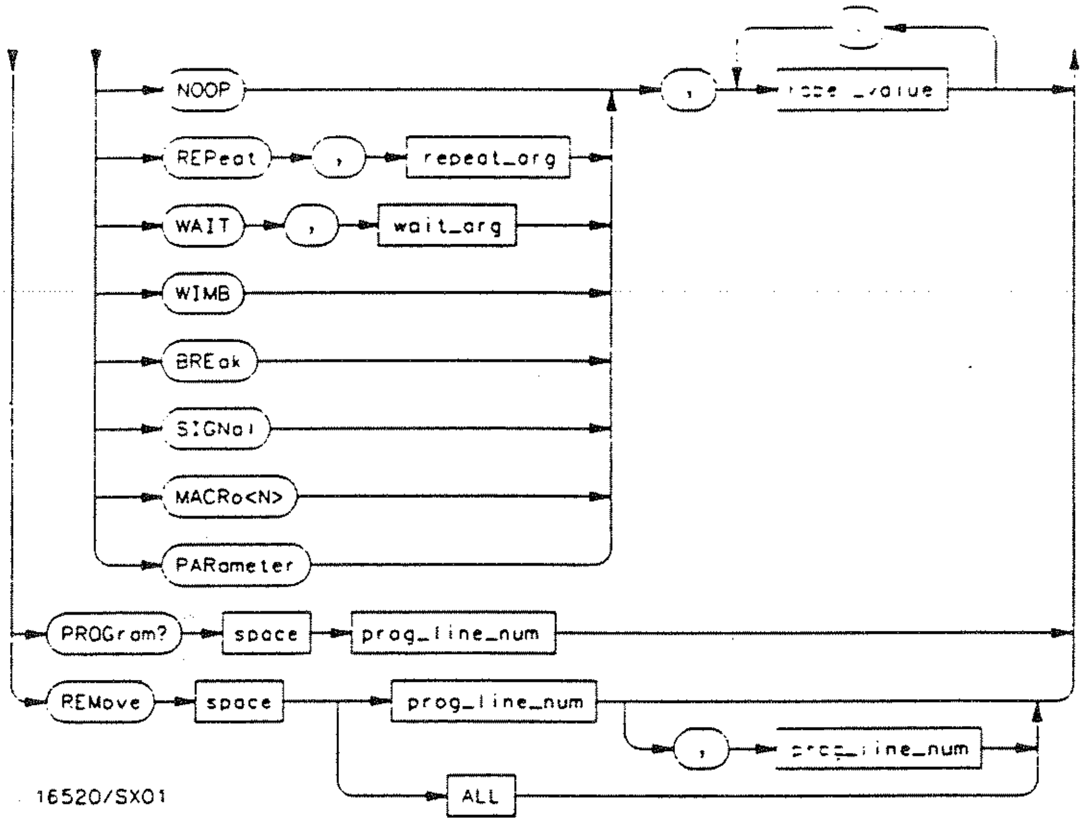
LISTing Subsystem 4

Introduction

The commands of the Listing subsystem allow you to write a pattern generator program using the parameters set in the Format subsystem.



P/O Figure 4-1. LISTing Subsystem Syntax Diagram



column_num = *an integer specifying the column that is to receive the new label*
label_name = *the label name that is to be removed*
prog_line_num = *an integer specifying the program line number*
label_value = *a string in one of the following forms:*
'#B01...' for binary
'Q01234567...' for octal
'H0123456789ABCDEF...' for hexadecimal
'0123456789...' for decimal
repeat_arg = *an integer from 1 through 256*
wait_arg = *an integer from 0 through 255*

P/O Figure 4-1. LISTing Subsystem Syntax Diagram

COLumn

command/Query

The COLumn command allows you to reorder the labels in the listing menu and set the numerical base for each label. The order of the labels in the format menu is not changed when the COLUMN command is used.

The first parameter of the command specifies the column number, followed by a label name and an optional number base. If a number base is not specified, the current number base for the label is used.

The query must include a column number and returns the label in that column and its base.

Command Syntax

```
:LISTing:COLumn <column number>,'<label
name>'[,{BINary|OCTal|DECimal|HEXadecimal|ASCii|SYMBol}]
```

where:

<column number> ::= an integer specifying the column that is to receive the new label
<label name> ::= the label name that is to be moved

Note

To move the Instruct column, use INSTRUCT as the label name without quotation marks.

Example:

```
OUTPUT XXX:":LIST:COL 1,'A',HEX"
```

COLumn

Query Syntax; :LISTing:COLumn? <column number>

Returned Format: [LISTING:COLUMN] <column number>,<label name>, {BINary|OCTal|DECimal|HEXadecimal|ASCII|SYMBol}

Example:

```
10 DIM Co$[100]
20 OUTPUT XXX;":LIST:COL? 1"
30 ENTER XXX;Co$
40 PRINT Co$
50 END
```

PROGram

Command/Query

The PROGram command adds pattern generator program lines, or modifies an existing line. The first parameter is the program line number. If the line number specified is beyond the last program line currently entered, a new line is added to the program. If the line number reference is a line within the current program, the existing program line is modified. The valid range of line numbers is 0 to 4094.

The labels are programmed in the same order as they are specified in the format menu regardless of their order in the listing.

If macros are invoked in the main program, the PROGram command line numbers may not correspond with the line numbers shown on the listing menu. This is because the macro program is inserted in the main program list. The PROGRAM command however, compensates for this and allows contiguous line numbering.

The second parameter is an optional string parameter. It specifies the starting label for the pattern strings that follow. This parameter is useful when long program strings are to be separated into several commands.

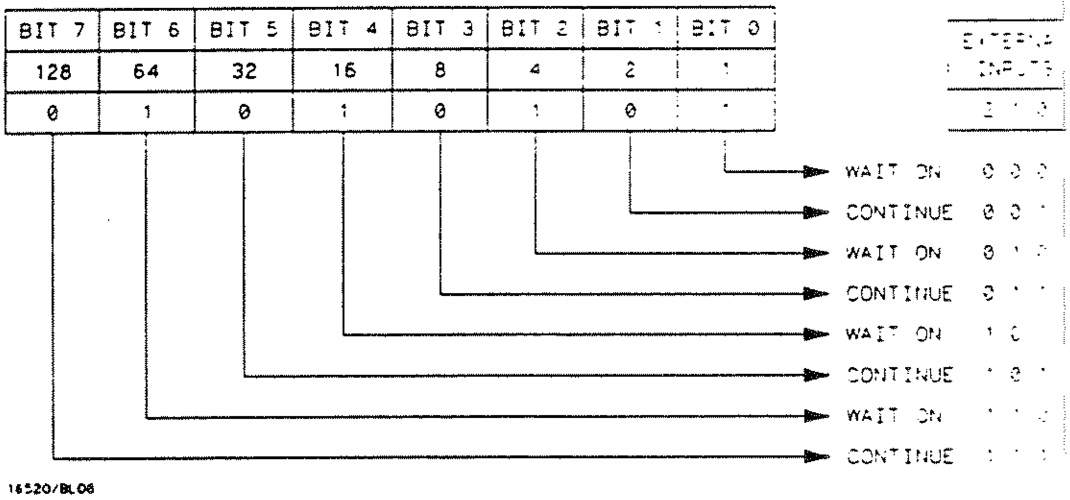
The next parameter may be one of five instructions or a call to one of four user defined macros. The instructions that may be used in a program are: NOOP, REPeat, WAIT, WIMB (Wait IMB), BREak, SIGNal, and MACRO#.

NOOP The NOOP instruction places a no instruction into the program line.

REPeat The REPéat instruction allows you to repeat a program line up to 256 times.

WAIT Along with an external clock, there are three external input qualifiers available with each master card. The WAIT instruction causes the pattern generator to wait at the current program line until the three external inputs go to a predefined state. When the predefined state is met, the program proceeds to the next program line.

When the wait parameter is represented in binary, each bit determines whether the associated state on the external inputs will be included or excluded from the wait condition. If all the wait bits are 1's, the pattern generator output is stopped, while all 0's allow the pattern generator to continue to the end of the program. A wait parameter of 01010101 is used in the example of Figure 4-2).



BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
128	64	32	16	8	4	2	1
0	1	0	1	0	1	0	1

Figure 4-2. WAIT Condition Example

WIMB (Wait IMB) Any module in the HP 16500A can signal the other modules through the Intermodule Bus (IMB).

If the pattern generator encounters a WIMB instruction in the program, it will hold the data outputs at their current state, while the output data clock and the strobes continue to run. The pattern generator will not continue to the next program line until it sees a signal on the IMB. In other words, the pattern generator will wait until another module tells it to continue.

BREak When a BREak is encountered in a program line, the pattern generator will stop. To advance to the next program line, use the RESume command

SIGNal (Signal IMB) The SIGNal instruction is the complement of the WIMB instruction. When the pattern generator program encounters a SIGNal instruction, it will output a signal to the Intermodule Bus (IMB). This signal is used to trigger other modules that are linked through the IMB, or the Port Out BNC.

MACRO< N > The macro instruction field lets you call one of four macros into a program. The macro programs are written in a similar manner as the main pattern generator programs. The MACRO subsystem section of this manual explains how to generate macros.

The program query returns the content of a program line and must include a program line number.

Command Syntax:

```
:LISTing:PROGram <program line number>,[<'label
name',]{NOOP|REPeat,<repeat arg>,|WAIT,<wait
arg>,|WIMB|BREak|SIGNal|MACRO#|PARameter}<'label value'>[,<'label
value>.....]
```

where:

<program line number> ::= an integer specifying the program line number

<label name> ::= string of up to six alphanumeric characters

<repeat arg> ::=an integer from 1 through 256

<wait arg> ::= an integer from 0 through 255

<label value> ::= a string in one of the following forms:

'#B01...' for binary

'#Q01234567...'for octal

'#H0123456789ABCDEF...'for hex

'0123456789...'for decimal

PROGram

Examples:

```
OUTPUT XXX;":LIST:PROG 0,REPEAT,255,'#B01X10111'"
OUTPUT XXX;":LIST:PROG 1,NOOP,'0'"
OUTPUT XXX;":LIST:PROG 2,SIGNAL,'1234'"
OUTPUT XXX;":LIST:PROG 3,WAIT,#B01010101,'#H2XBC'"
OUTPUT XXX;":LIST:PROG 4,MACRO2,'#H3X45'"
OUTPUT XXX:":LIST:PROG 5,PARAMETER,'#B0101111100001111'"
```

Query Syntax: :LISTing:PROGram? <program line number>

Returned Format: [LISTING:PROGRAM] <program line number>,{NOOP|REPeat<repeat arg>|WAIT<wait arg>|WIMB|BREak|SIGNal|MACRO<N>|PARameter},<label value>[,<label value>.....]

Example:

```
10 DIM A$[100]
20 OUTPUT XXX;":LIST:PROG? 1"
30 ENTER XXX;A$
40 PRINT A$
50 END
```

REMove

Command

The REMove command allows you to remove one or several lines from the main pattern generator program. If only one parameter number is given, that line number is deleted. If two numbers are given, the range of lines between those two values inclusive is deleted. The command REMove ALL deletes the entire program.

Command Syntax: `LISTing:REMove{ <program line number[,<program line range> |ALL> }`

where:

<program line number> ::= an integer specifying the program line to be removed

<program line range> ::= two integers separated by a comma, specifying the program line range to be removed.

Example: `OUTPUT XXX;":LIST:REM 1,4"`

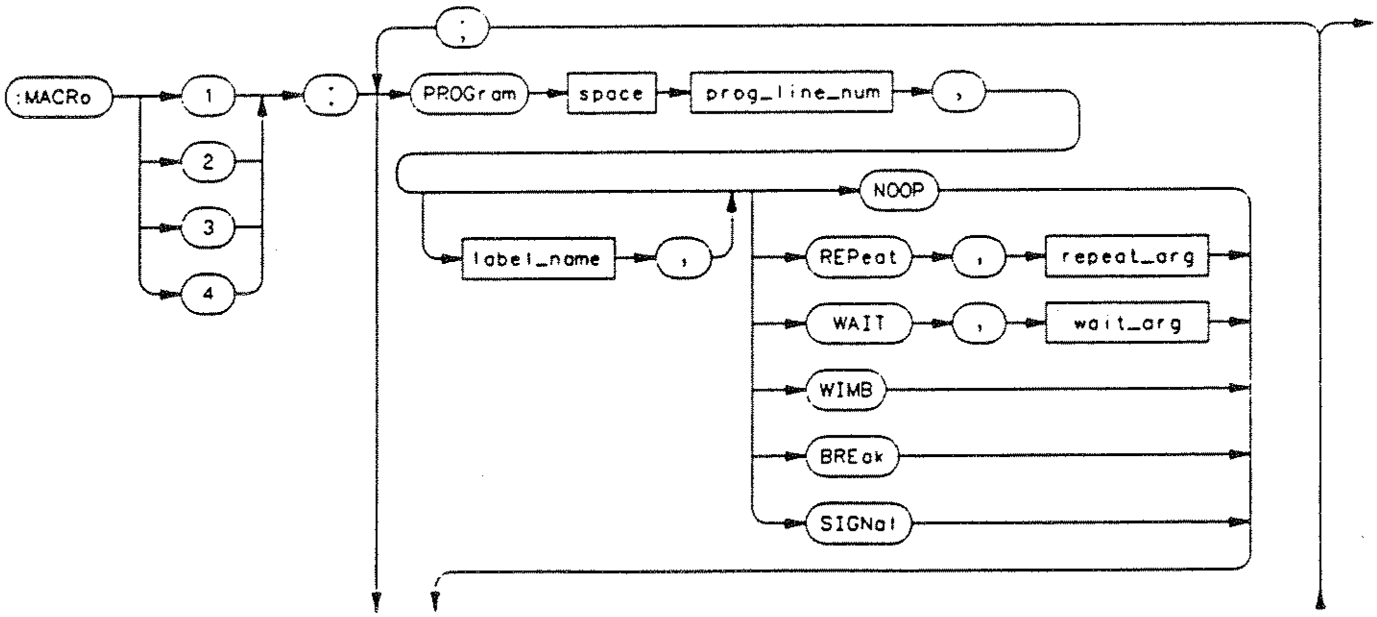
MACRo Subsystem 5

Introduction

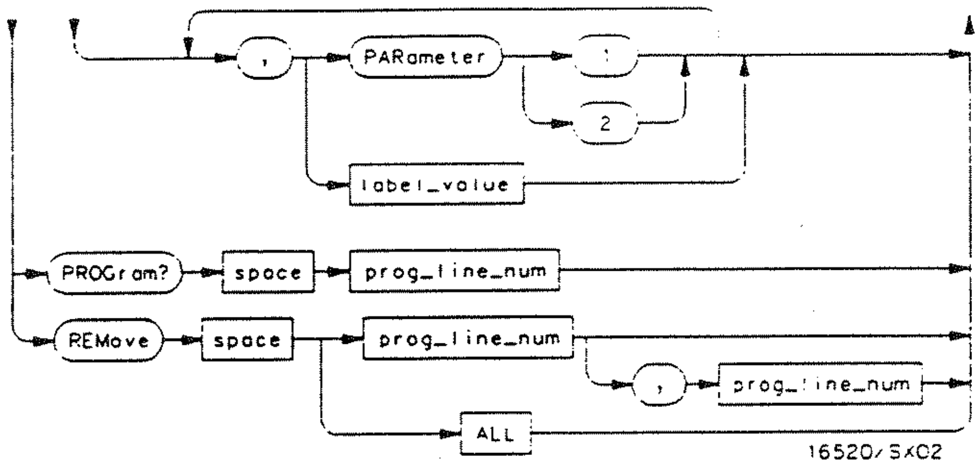
The commands of the MACRo subsystem allow you to write and edit macros for use in the main pattern generator program. Up to four macros may be called into the main listing program. The macros are labeled MACRO1 through MACRO4 and cannot be renamed over the interface bus.

The query returns the content of a program line and must include a program line number.

Refer to figure 5-1 for the MACRo subsystem syntax diagram.



PO/ Figure 5-1. MACRo Subsystem Syntax Diagram



prog_line_num = *an integer specifying the program line number*
repeat_arg = *an integer from 1 through 256*
wait_arg = *an integer from 0 through 255*
label_value = *a string in one of the following forms:*
'#B01...' for binary
'#Q01234567...' for octal
'#H012345679ABCDEF...' for hexadecimal
'0123456789...' for decimal

P/O Figure 5-1. MACRo Subsystem Syntax Diagram

PROGram

Command/Query

The PROGram command adds macro program lines, or modifies an existing line. This command is identical to the LISTing:PROGram command, with two exceptions:

- MACRo and PARameter are not included as choices for the instruction parameter because a macro cannot be invoked from another macro.
- The pattern generator allows you to pass parameters between the main listing program and the macros using the PARAM1 and PARAM2 key words. These key words may be substituted for any label value string

Setting Pass Parameters

There are two parameters available for each label in the macro list. They are labeled PARAM1 and PARAM2. In the example of figure 5-2, a macro call is made at line three of the main listing program.

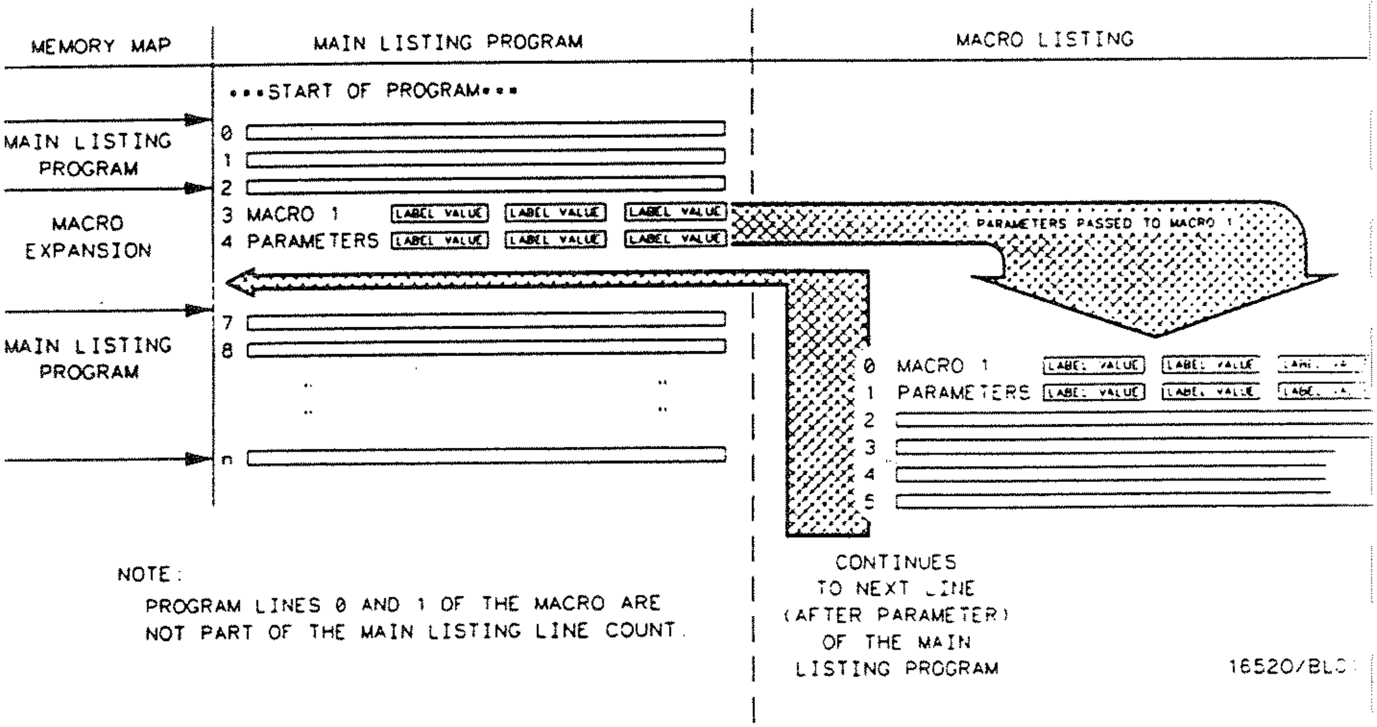


Figure 5-2. Setting Pass Parameters

The data of lines three and four are passed into the macro lines zero and one and are labeled PARAM1 and PARAM2. At the last line of the macro, the program continues to the next line in the main listing program. The main listing program line numbering is not consecutive. This is because the macro is placed in memory at the location of the macro command. Also note that lines zero and one of the macro are not part of the main listing line count.

PROGram

Command Syntax:

```
:MACRo <macro number> :PROGram <program line number> ,[<label_ name> ,]
{NOOP|REPeat, <repeat arg> |WAIT, <wait arg> |WIMB|BREak|SIGNal|},
{PARameter <1|2> | <label value> }[,PARameter <1|2> | <label value> ....]
```

where:

<macro number> ::= an integer from 1 through 4
<program line number> ::= an integer specifying the program line number
<label_name> ::= string of up to six alphanumeric characters
<repeat arg> ::= an integer from 1 through 256
<wait arg> ::= an integer from 0 through 255
<label value> ::= a string in one of the following forms:

'#B01...' for binary
'#Q01234567...' for octal
'#H123456789ABCDEF...' for hexadecimal
'0123456789...' for decimal

Examples:

```
OUTPUT XXX;":MACRo1:PROG 0,NOOP,'#B010110010','#B000100101'"
OUTPUT XXX;":MACRO1:PROG 1,REPEAT,127,PAR1,PAR2"
OUTPUT XXX;":MACRO1:PROG 2,NOOP,'#B01X10X10',PAR2"
```

Query Syntax:

```
:MACRO <macro number> :PROGram? <program line number>
```

Returned Format:

```
[:MACRo <macro number> :PROGram <program line
number> ],{NOOP|REPeat, <repeat arg> |WAIT, <repeat
arg> |WIMB|BREak|SIGNal}|{PARameter <1|2> | <label
value> }[,PARameter <1|2> | <label value> ....]
```

Example:

```
10 DIM A$[100]
20 OUTPUT XXX;":MACRO1:PROGRAM? 1"
30 ENTER XXX;A$
40 PRINT A$
50 END
```

REMove

Command

The REMove allows you to remove one or several lines from the macro. If only one parameter is given, only that line is deleted. If two numbers arc specified, the range of lines between those values, inclusive, is deleted. The command REMove ALL deletes the entire program.

Command Syntax: :MACRo <macro number> :REMove <program line number> [, <program line range> |ALL

where:

<macro number>	::= an integer from 1 through 4
<program line>	::= an integer specifying the program line to be removed
<program line range>	::= two integers separated by a comma, specifying the program lines to be removed

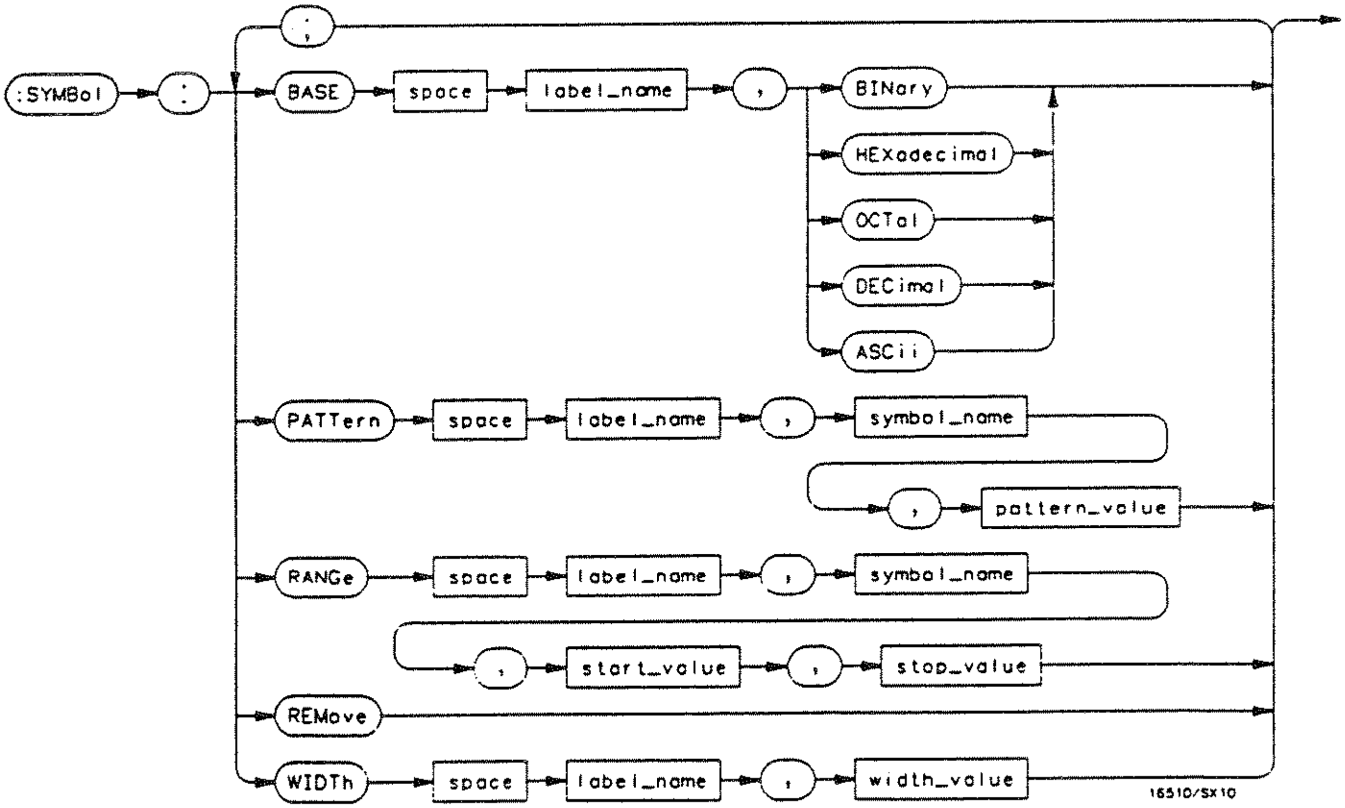
Example: OUTPUT XXX;":MACRO1:REM 1,3"

SYMBol Subsystem 6

Introduction

The SYMBol subsystem contains the commands that allow you to define symbols on the controller and download them to the HP 16520A/HP 16521A Pattern Generator module. The commands in this subsystem are:

- BASE
- PATTern
- RANGe
- REMove
- WIDTh



P/O Figure 6-1. SYMBol Subsystem Syntax Diagram

<label_name> = *string of up to 6 alphanumeric characters*

<symbol_name> = *string of up to 16 alphanumeric characters*

<pattern_value> = *string of one of the following forms:*

'#B01X...' for binary

'#Q01234567X..' for octal

'#H0123456789ABCDEFX...' for hexadecimal

'0123456789...' for decimal

<start_value> = *string of one of the following forms:*

'#B01...' for binary

'#Q01234567..' for octal

'#H0123456789ABCDEF...' for hexadecimal

'0123456789...' for decimal

<stop_value> = *string of one of the following forms:*

'#B01... for binary

'#Q01234567..' for octal

"#H0123456789ABCDEF..." for hexadecimal

'0123456789...' for decimal

<width_value> = *integer from 1 to 16*

P/O Figure 6-1. SYMBol Subsystem Syntax Diagram

BASE

command

The BASE command sets the base in which symbols for the specified labcl will be displayed in the symbol menu. It also specifies the base in which the symbol offsets are displayed when symbols are used.

Note

BINary is not available for labels with more than 20 bits assigned. In this case the base will default to HEXadecimal.

Command Syntax: `:SYMBol:BASE <label_name>,<base_value>`

where:

<label_name>	::= string of up to 6 alphanumeric characters
<base_value>	::= {BINary | HEXadecimal | OCTal | DECimal | ASCii}

Example:

```
OUTPUT XXX;":SYMBol:BASE 'DATA',HEXadecimal"
```

PATTern

command

The PATTern command allows you to create a pattern symbol for the specified label. The pattern may contain "don't cares" in the form of XX...X's.

Command Syntax:

```
:SYMBol:PATTern <label_name>,<symbol_name>,<pattern_value>
```

where:

<label_name> ::= string of up to 6 alphanumeric characters

<symbol_name> ::= string of up to 16 alphanumeric characters

<pattern_value> ::= string of one of the following forms:

'#B01X...' for binary
'#Q01234567X..' for octal
'#H0123456789ABCDEFX...' for hexadecimal
'0123456789...' for decimal

Example:

```
OUTPUT XXX;":SYMBol:PATTern 'STAT', 'MEM_RD','#H01XX'"
```

RANGe

command

The RANGe command allows you to create a range symbol containing a start value and a stop value for the specified label.

Note

Don't cares are not allowed in range symbols.

Command Syntax:

```
:SYMBol:RANGe <label_name>,<symbol_name>,
<start_value>,<stop_value>
```

where:

```
<label_name>    ::= string of up to 6 alphanumeric characters
<symbol_name>   ::= string of up to 16 alphanumeric characters
<start_value>   ::= string of one of the following forms:
                      "#B01..." for binary
                      '#Q01234567..' for octal
                      '#H0123456789ABCDEF...' for hexadecimal
                      '0123456789...' for decimal
<stop_value>    ::= string of one of the following forms:
                      "#B01..." for binary
                      '#Q01234567..' for octal
                      '#H0123456789ABCDEF...' for hexadecimal
                      '0123456789...' for decimal
```

Example:

```
OUTPUT XXX;":SYMBol:RANGe 'STAT', 'IO_ACCESS','#H0000','#H000F'"
```

REMove

command

The REMove command deletes all symbols from the symbol menu.

Command Syntax: :SYMBol:REMove

Example: OUTPUT XXX;":SYMBol:REMove"

WIDTh

command

The WIDTh command specifies the width (number of characters) in which the symbol names will be displayed when symbols are used.

Note

The WIDTh command does not affect the displayed length of the symbol offset value.

Command Syntax: `:SYMBol:WIDTh <label_name>,<width_value>`

where:

<label_name>	::= string of up to 6 alphanumeric characters
<width_value>	::= integer from 1 to 16

Example:

```
OUTPUT XXX;":SYMBol:WIDTh 'DATA',9 "
```

Data and Setup Commands A

Introduction

The DATA and SETup commands are system commands that allow you to send and receive instrument configuration, setup and program data to and from a controller in block form. This is useful for saving block data for re-loading the pattern generator. This appendix explains how to use these commands.

The block data for the DATA command is broken into byte positions and descriptions. The SETup command block data is not described in detail. No changes should be made to the "config" section of the block data.

Definition of Block Data

Block data in the # format is made up of a block length specifier and a variable number of sections.

```
<block length specifier> <section 1> ... <section N>
```

The block length specifier is defined as follows:

```
#8<length>
```

where:

<length> ::= the total length of all sections in byte format (must be represented with 8 digits)

For example, if the total length of the block (all sections) is 14506 bytes, the block length specifier would be "#800014560" since the length must be represented with 8 digits.

Sections consist of a section header followed by the secti lata as follow

< section header > < section data >

where:

< section header > ::= 10 bytes for the section name
1 byte reserved (always 0)
1 byte for the module ID code (21 for pattern generator)
4 bytes for the length of the data in bytes

The section data format varies for each section and may be any length.

Note

The total length of a section is 16 (for the section header) plus the length of the section data. Thus, when calculating the length of a block of configuration data, don't forget to add the length of the headers.

HP-IB Example:

```
10 DIM Block$[3200]  !allocate enough memory for block data
20 DIM Specifier$[2]
30 OUTPUT XXX;"EOI ON"
40 OUTPUT XXX;"SYSTEM:HEAD OFF"
50 OUTPUT XXX;"SELECT 1"  !select module
60 OUTPUT XXX;"SYSTEM:DATA?  !send the data query
70 ENTER XXX USING"#,2A";Specifier$  !read in #8
80 ENTER XXX USING"#,8D",Blocklength  !read in block length
90 ENTER XXX USING"-K",Block$  !read in data
```

SYSTem:SETup

The SETup command for the pattern generator module is used to configure system parameters, such as the pod and bit assignment, input thresholds, strobe values, and clock rates.

The "CONFIG" section consists of 1128 bytes of information which fully describe the main parameters for the pattern generator. The total length of the section is 1144 bytes (recall that the section header is 16 bytes).

The data in this section of the block should not be changed to ensure proper pattern generator operation.

Command Syntax: :SYSTem:SETup <block data in # format>

Query Syntax: :SYSTem:SETup?

Returned Format: [:SYSTem:SETup] <block data in # format> <NL>

SYSTem:DATA

The DATA command is used to send and receive the pattern generator main program listings and the macro listings. The complete pattern generator data block consists of five sections not counting the SYMBOL section. The sections are:

Section 1	"MAINPROG"
Section 2	"MACRO1"
Section 3	"MACRO2"
Section 4	"MACRO3"
Section 5	"MACRO4"

Command Syntax:

```
:SYSTem:DATA <block data in # format>
```

Query Syntax:

```
:SYSTem:DATA?
```

Returned Format:

```
[:SYSTem:DATA] <block data in # format> <NL>
```

Section 1 "MAINPROG"

The Main Program section contains the program listing data. The length of this section depends on the length of the program listing and the number of expansion cards connected to the master card.

The data for this section is as follows:

1 **16 bytes** - section header "MAINPROG"

17 **2 bytes** - number of pods - The total number of pods for which the program is written. Valid values are 2 to 26 in increments of 6 because the master card has 2 pods and each expansion card has 6 pods.

19 **2 bytes** - total program length - The total length of the pattern generator program with macros expanded. Valid values are 1 to 4095.

21 **2 bytes** - edit line index - The index of the current editing line on screen. Valid values are 0 to the total program length - 1.

23 **6 bytes** - reserved - The values should be set to zero.

29 **2 bytes** - total program lines - The total number of program lines with macros not expanded. Valid values are 1 to total program length.

Note

Macro calls require two program lines.The first line contains the MACRO opcode and the values for PARAM1 for each label. The second line contains the PARAMETER opcode and the values for PARAM2 for each label.

31 **number of bytes = total program lines** (N) - opcode list - This block contains a list of the opcodes for the main program in order of ascending line numbers. The opcode for each main program line occupies one byte, with the opcode for line N preceding the opcode for line N + 1 in the structure. The valid opcodes are:

Note

A macro opcode must be followed by the appropriate macro parameter opcode.

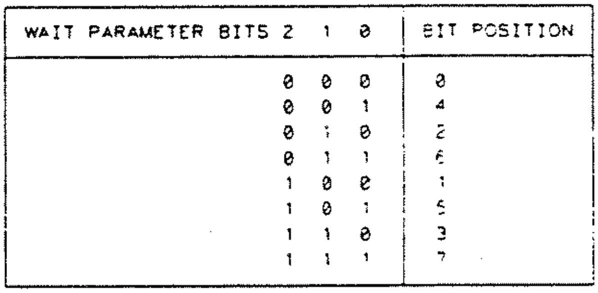
0 - NOOP
1 - WAIT IMB
2 - WAIT EXTERNAL
3 - REPEAT
4 - SIGNAL IMB
8 - BREAK
16 - MACRO1
17 - MACRO2

18 - MACRO3
19 - MACRO4
20 - MACRO1 PARAMETER
21 - MACRO2 PARAMETER
22 - MACRO3 PARAMETER
23 - MACRO4 PARAMETER

Note

Byte position from here on varies with total program line length.

31 + N **number of bytes = total program lines (N)** - parameters - This block contains a list of the parameters for the main program in order of ascending line numbers. The parameter for each main program line occupies one byte, with the parameter for line N preceding the parameter for line N + 1 in the structure. These bytes will only be valid when they correspond to main program lines containing either a REPEAT or a WAIT instruction. The valid values for the WAIT instruction are 0 to 255 where 0 in a bit position means continue and a 1 in a bit position means wait. The WAIT bit positions are defined as follows:



WAIT PARAMETER BITS 2	1	0	BIT POSITION
0	0	0	0
0	0	1	4
0	1	0	2
0	1	1	6
1	0	0	1
1	0	1	5
1	1	0	3
1	1	1	7

16520/BL07

All other instructions parameters have no effect and should be zero.

31 + 2N **number of bytes = total program lines (N) * number of pods (P)** - data array - The data array block contains the 0/1 pattern information for each main program line and each pod. Program line number is the primary array index and the pod number is the secondary index, hence P bytes of pod data are sent for a given line before any data from the next line. For a given line, pod data is sent in order of descending pod numbers with master pod data before expansion pod data. When more than one expansion pod is installed, the data is sent in order of ascending slot numbers 1-5 or A-E.

With this organization, data will be sent out in the same order as if read from the LISTing menu as English text from left to right, then top to bottom. A "1" in the data array means generate a "1" on the corresponding channel of the output pod, assuming positive label polarity.

31+2N+PN **number of bytes = total program lines (N) * number of pods (P)** - auto-fill array - This array contains auto-fill/no auto-fill information for each main program line and each pod. This array is organized exactly as the data array described above, therefore bits map directly across from one to the other. A "1" in this array means output the last specified pattern for the corresponding bit from the data array when the auto-fill bit was 0.

Note

The easiest way to send a program is to indicate all the data in the data array and to send all 0's in the autofill array.

Section 2 "MACRO1"

The "MACRO1" section contains all the program listing for MACRO1. The length of this section varies depending on the length of the macro listing and the number of expansion cards connected to the master card.

1 **16 bytes** - section header "MACRO1"

17 **1 byte** - number of pods - The total number of pods for which this macro is defined. Valid values are 2 to 26 in increments of 6 because the master card has 2 pods and each expansion card has 6 pods.

18 **1 byte** - length of macro - The total number of program lines in the macro. Valid values are 2 to 62.

19 **1 byte** - macro references - The total number of times the macro is referenced by the main program. Valid values are 0 to 127.

20 **7 bytes** - macro name - This is the name of the macro. The name may be up to 6 alphanumeric characters long. The last byte must be a null (0).

27 **1 byte - macro number** - This is the current macro number. Valid values are 0 through 3.

28 **280 bytes** - parameter names - This structure contains the parameter name for each parameter in the macro. The 280 bytes are organized as 7 bytes for each name * 20 labels for PARAM1 data, followed by 7 bytes for each name * 20 labels for PARAM2 data. The name may be up to 6 alphanumeric characters long and the seventh byte for each name must be null (0).

308 **6 bytes * macro lines (M) + 2** - parameter values - This represents the parameter usages within the macros and should all be zeros.

Note

Byte position from here on varies with the macro program length.

310 + 6M **number of bytes = macro lines (M)** - opcode list - This is a list of the opcodes for the macro program. There should be one opcode for each line in the macro program. Refer to the "MAINPROG" opcode list for the description of opcodes.

310 + 7M **number of bytes = macro lines (M)** - parameters - This is a list of the parameters for the WAIT and REPEAT instructions used within the macro. Refer to "MAINPROG" parameters for a description of this structure.

310 + 8M **number of bytes = macro lines (M)** - data array - This is the 0/1 pattern information for each pod. A "1" in the data array means generate a "1" on the associated output line, subject to the polarity of that label. Refer to the "MAINPROG" data array for the description of this structure.

310 + 8M + PM **number of bytes = macro lines (M) * number of pods** auto-fill array - This represents the auto-fill/no auto-fill information. A "1" means output the last specified pattern for that bit when the auto-fill array was 0. Refer to the "MAINPROG" autofill array for the description of this structure.

Sections 3, 4, 5 "MACRO2", "MACRO3", "MACRO4"

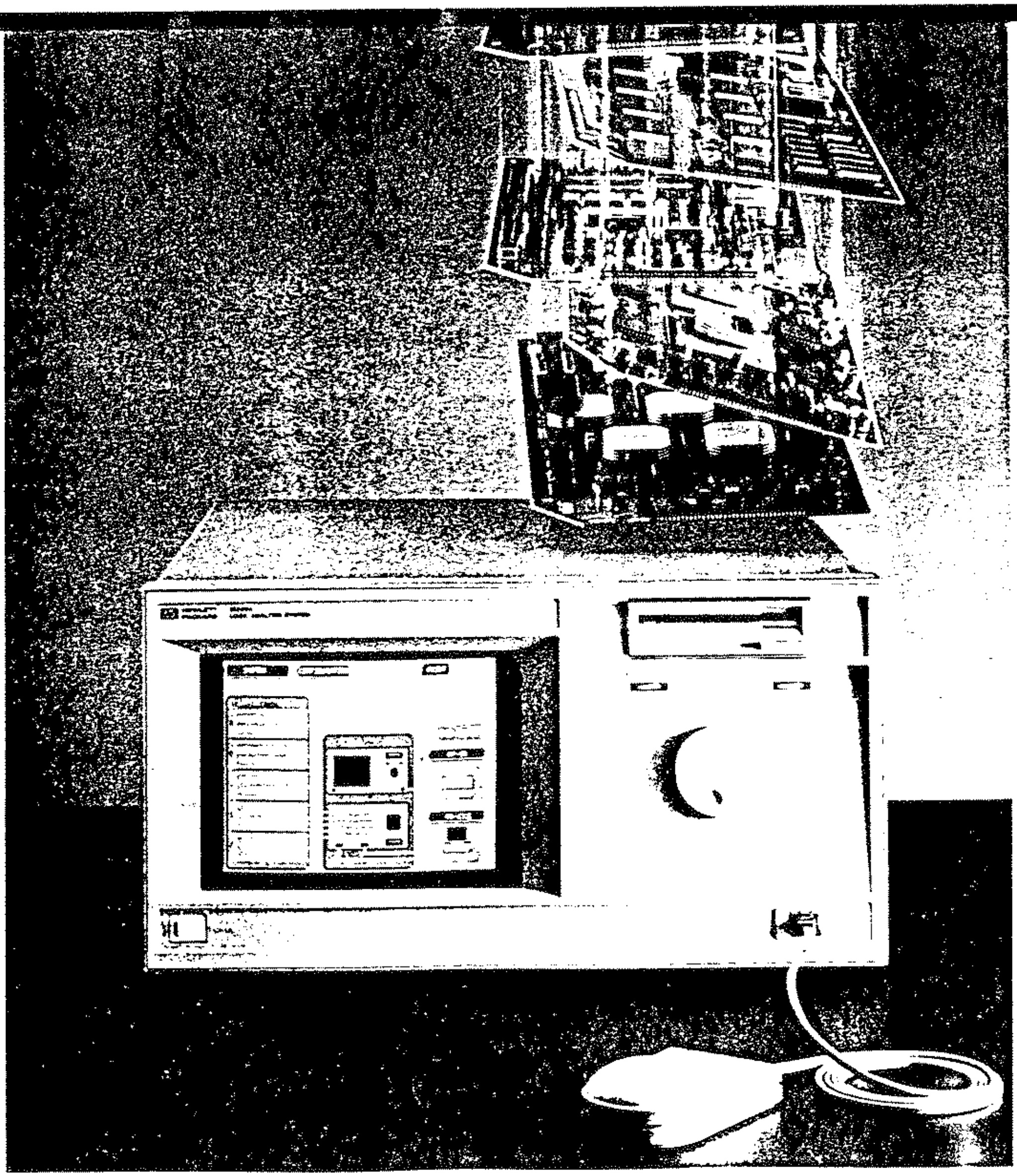
The program listing for Macros 2 through Macros 4 are identical to Macro 1. The length of these sections vary with the length of the macro listing and the number of expansion cards connected to the master card. Refer to Section 2 of this appendix for details of the section definitions.

Index

HP 16520A/16521A

PATTERN GENERATOR MODULE
for THE HP 16500A LOGIC ANALYSIS SYSTEM

Front-panel Operation Reference



HEWLET
PACKAR

Front-Panel Operation Reference

HP 16520A/16521A Pattern Generator Module

for the HP 16500A Logic Analysis System

HEWLETT PACKARD

© Copyright Hewlett-Packard Company 1987

Manual Set Part Number 16520-90902
Microfiche Part Number 16520-90802

Printed in U.S.A. October 1987

Printing History First Edition - October 1987 Printed in U.S.A.

Product Warranty

This Hewlett-Packard product has a warranty against defects in material and workmanship for a period of one year from date of shipment. During warranty period, Hewlett-Packard Company will, at its option, either repair or replace products that prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by Hewlett-Packard. However, warranty service for products installed by Hewlett-Packard and certain other products designated by Hewlett-Packard will be performed at the Buyer's facility at no charge within the Hewlett-Packard service travel area. Outside Hewlett-Packard service travel areas, warranty service will be performed at the Buyer's facility only upon Hewlett-Packard's prior agreement and the Buyer shall pay Hewlett-Packard's round trip travel expenses.

For products returned to Hewlett-Packard for warranty service, the Buyer shall prepay shipping charges to Hewlett-Packard and Hewlett-Packard shall pay shipping charges to return the product to the Buyer. However, the Buyer shall pay all shipping charges, duties, and taxes for products returned to Hewlett-Packard from another country.

Hewlett-Packard warrants that its software and firmware designated by Hewlett- Packard for use with an instrument will execute its programming instructions when properly installed on that instrument. Hewlett-Packard does not warrant that the operation of the instrument, software, or firmware will be uninterrupted or error free.

Limitation of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by the Buyer, Buyer-supplied software, or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. HEWLETT-PACKARD SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Exclusive Remedies

THE REMEDIES PROVIDED HEREIN ARE THE BUYER'S SOLE AND EXCLUSIVE REMEDIES. HEWLETT-PACKARD SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

Assistance

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For any assistance, contact your nearest Hewlett-Packard Sales and Service Office. Addresses are provided at the back of this manual.

Certification

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Bureau of Standards, to the extent allowed by the Bureau's calibration facility, and to the calibration facilities of other International Standards Organization members.

Safety

This product has been designed and tested according to International Safety Requirements. To ensure safe operation and to keep the product safe, the information, cautions, and warnings in this manual must be heeded.

How to Use This Manual

About This Manual...

This manual is organized in encyclopedic form, with each chapter covering a subject. It does assume some basic knowledge of the HP 16500A and its user interface. If you are unfamiliar with the user interface, chapter 3 of this manual gives a brief overview of its operation.

Chapter 1 gives a brief pictoral explanation on the process of writing a pattern generator program, as well as the part played by each of the menus.

Chapters 4, 5, and 6 describe the functions in the three main menus of the pattern generator. Each includes pictoral index on the second page of the chapter. This pictoral index names each of the fields in the menu and gives the page or chapter number in this manual where you will find a detailed explanation of its use.

At the start of each major section in the chapters you'll find headings that look like this:

Menu: Format
Field: Clock Selection (6)

Notice that below the bar there are two lines that say **Menu** and **Field**. The **Menu** line tells you which menu the function is in. The **Field** line tells you which field in the menu to touch to get to the function. Directly after the **Field** line you'll see a number in parenthsis. This number refers back to the pictoral index in chapter 4, 5, or 6. The number serves as a cross reference, allowing you to look back at the menu pictures and see what field is being discussed. If you are unsure of the field being discussed in the text, turn to the pictoral index for the menu listed, and look at the picture to see which numbered field is explained. In the example above you would turn to the **Format** menu index in chapter 4, and look up field 6, which is the Clock Selection field. If there is no number following the field name, it means that the field may be found in more than one menu, such as the **Print** or **Run** field.

Table of Contents

Introduction

Chapter 1:

Chapter 2:

Chapter 3:

Chapter 4:

Setting Pattern Generator Values

Chapter 5:

Entering and Editing Output Data Patterns

Chapter 6:

The Macro List Menus

Chapter 7:

Running and Stopping the Pattern Generator

Chapter 8:

Creating a Symbol Table

Chapter 9:

Defining and Using Strobes

Introduction

Welcome to the new generation of HP logic analyzers! The HP 16500A Logic Analysis System has been designed to make it easier to use than any previous Hewlett-Packard logic analyzer. And, because of its configurable architecture, it can easily be tailored to your specific logic design and debug needs.

The user interface of the HP 16500A was designed for the most intuitive operation possible. The use of "pop-up" windows and color graphics helps lead you through set ups and measurements without having to memorize a lot of steps.

If you haven't already read "How To Use This Manual" at the front of this book, please do so. It will give you some important information about the structure of this manual and how to get the most out of it.

1 What is the HP 16520A/16521A?

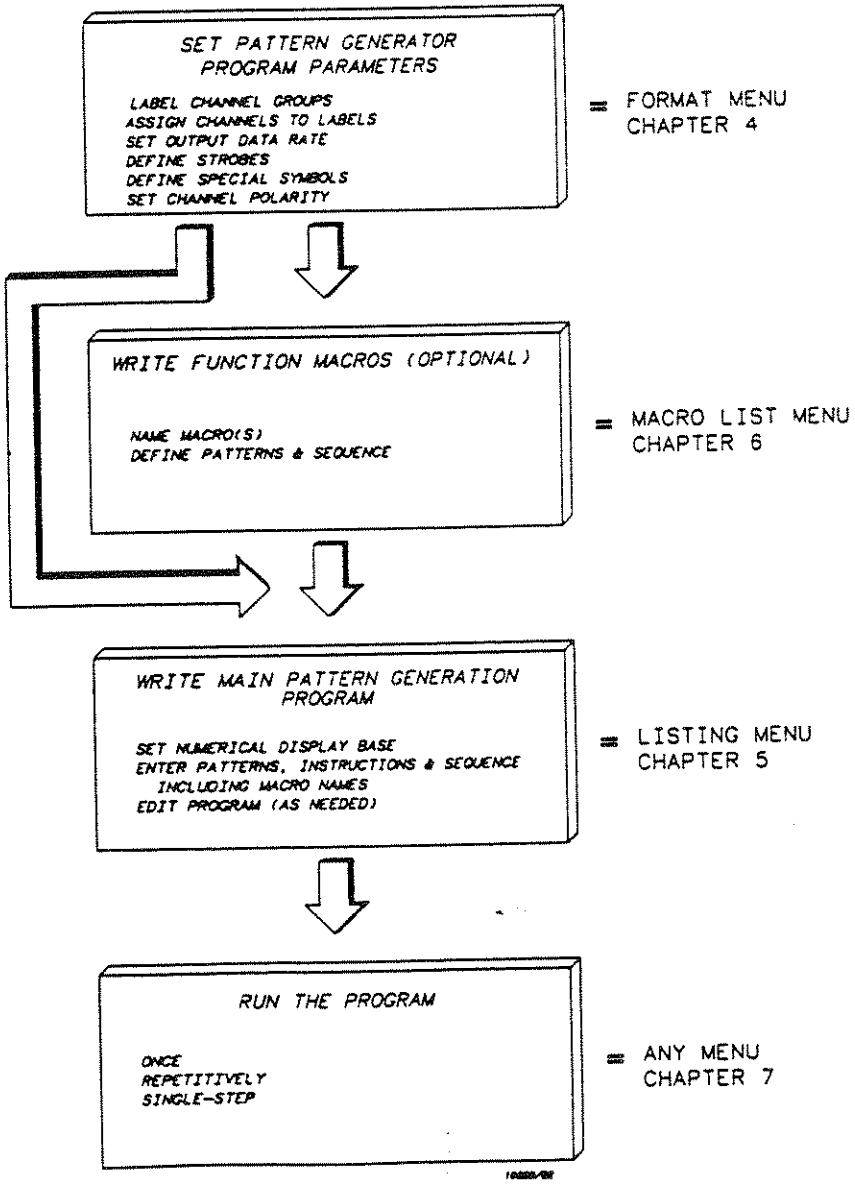
The HP 16520A/16521A Pattern Generator is a programmable, 50 Mbit/s pattern generation module that plugs into the HP 16500A Logic Analysis System. The HP 16520A is a master card, and can support up to four of the HP 16521A expansion cards.

The key features of the pattern generator are:

- 50 Mbit/s data rate
- 12 NRZ data channels on each HP 16520A master card
- 48 NRZ data channels on each HP 16521A expansion card
- Up to 4095 program steps
- Three 20 MHz RZ strobe channels on each HP 16520A master card
- Lightweight, passive probes
- Gives the HP 16500A Logic Analysis System up to 204 data channels with one master card and four expansion cards installed
- TTL or ECL output
- External clock input
- Up to four user-definable macros
- External qualifier inputs

Pattern Generation Process

The following diagram illustrates the process of writing a pattern generation program on the HP 16520A/16521A Pattern Generator cards. It also shows the chapters in this manual that cover those subjects.


SET PATTERN GENERATOR PROGRAM PARAMETERS
LABEL CHANNEL GROUPS
ASSIGN CHANNELS TO LABELS
SET OUTPUT DATA RATE
DEFINE STROBES
DEFINE SPECIAL SYMBOLS
SET CHANNEL POLARITY
= FORMAT MENU CHAPTER 4
WRITE FUNCTION MACROS (OPTIONAL)
NAME MACRO(S)
DEFINE PATTERNS & SEQUENCE
= MACRO LIST MENU CHAPTER 6
WRITE MAIN PATTERN GENERATION PROGRAM
SET NUMERICAL DISPLAY BASE
ENTER PATTERNS, INSTRUCTIONS & SEQUENCE INCLUDING MACRO NAMES
EDIT PROGRAM (AS NEEDED)
= LISTING MENU CHAPTER 5
RUN THE PROGRAM
ONCE
REPETITIVELY
SINGLE-STEP
= ANY MENU CHAPTER 7

The pattern generator menus are designed so that they share as many operations as possible with the logic analyzer modules for the HP 16500A. That means that once you've learned how to use the pattern generator, learning the logic analyzer is made much simpler, and vice versa.

2 Cables and Probes

What Cables and Probes are Included?

The cables listed below are already connected to the instrument when you receive it and exit via the rear panel of the instrument.

Each HP 16520A master card comes with the following:

- ✓ One - 1.52 m (5 ft), 2 by 10 pin output data ribbon cable with violet label
- ✓ One - 1.52 m (5 ft), 2 by 10 pin strobe/data ribbon cable with violet label
- One - 1.52 m (5 ft), 2 by 10 pin input qualifier ribbon cable with gray label P/N 16520-64501
- Two bags of probes and leads for the output data/strobe cables
- One bag of probes and leads for the input qualifier cable
- Three clip-on label holders and sheet of cable and probe labels.

Each HP 16521A expansion card comes with the following:

- Six - 1.52 m (5 ft), 2 by 10 pin output data ribbon cables with violet labels
- Six clip-on label holders and sheet of cable and probe labels.

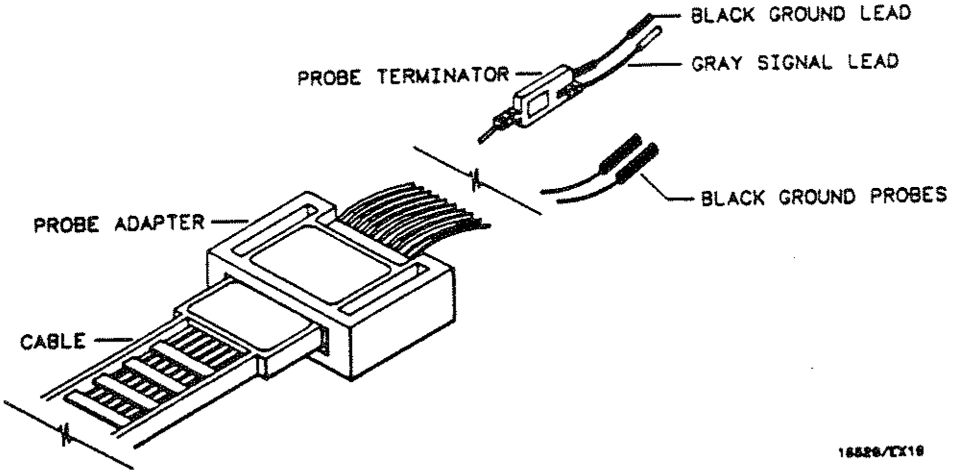
Each bag of probes and leads for the output data/strobe cables contains the following:

- One probe adapter pod
- Eight output data probes (violet tip)
- Eight 51 mm (2 in.) signal ground leads (black)
- Two 152 mm (6 in.) pod ground leads (black).

Each bag of probes and leads for the input qualifier cable contains the following:

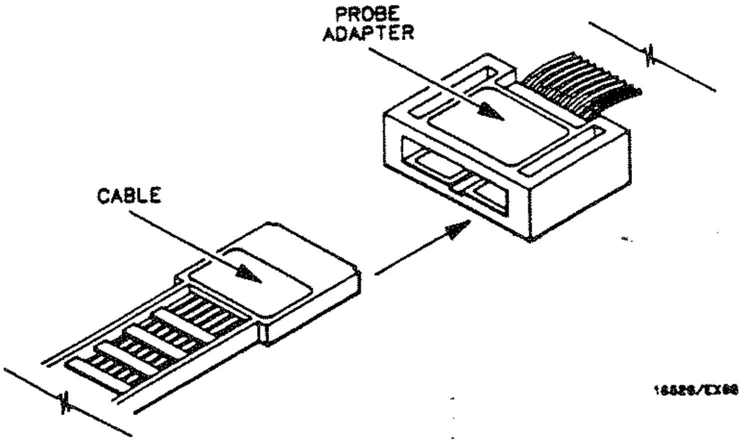
- Eight input probes (gray)
- Eight 51 mm (2 in.) signal ground leads (black)
- Two 152 mm (6 in.) pod ground leads (black).

The illustration below identifies all the probes and assemblies.



Connecting the Probe Adapter Pods

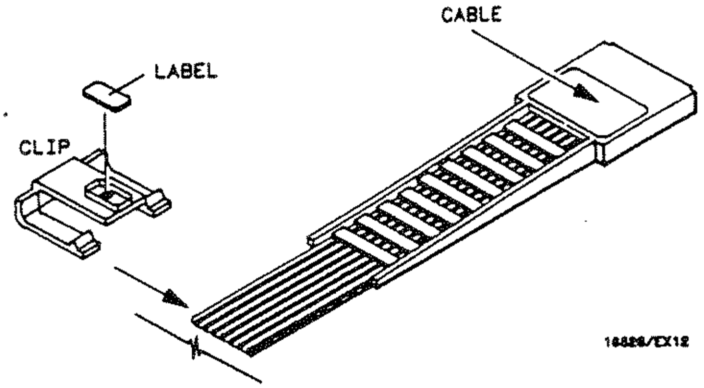
The probe adapters fit onto the end of the cables to provide an alternate means of connecting to your target system. There are ten probes on each adapter. To attach a probe adapter to a cable, simply push the adapter onto the end of the cable. Both the cable and adapter are keyed such that they will go together only one way.



Attaching Labels to the Cables

Since you may have as many as 27 cables and 270 probes attached to an HP 16500A with one master card and four expansion cards, it is helpful to have some method of quickly identifying them. Clip-on label holders for each cable and stick-on labels for each probe are provided for just this reason.

To attach the clip-on label holder to a cable, just slide it on to the edge of the cable as shown below. Then remove the appropriate label from the sheet provided and stick it into the recessed rectangle on the label holder. Notice that there are labels that conform to the slots in the mainframe and the cable number on the card. For instance, if you have a pattern generator master card in slot A of the mainframe, and you are using only the output data and strobe channels, you'll need to label the two cables A2 and A3 since there are two output cables from a master card. The A indicates that the card is in slot A. Looking at the card from the rear, cable numbering is from left to right. Thus, cable two is in the middle and cable three is on the right of the card. You'll want to attach a clip-on label holder to each and label cable two (eight data channels) with a red A2 sticker and cable three (data/strobe channels) with an orange A3 sticker.



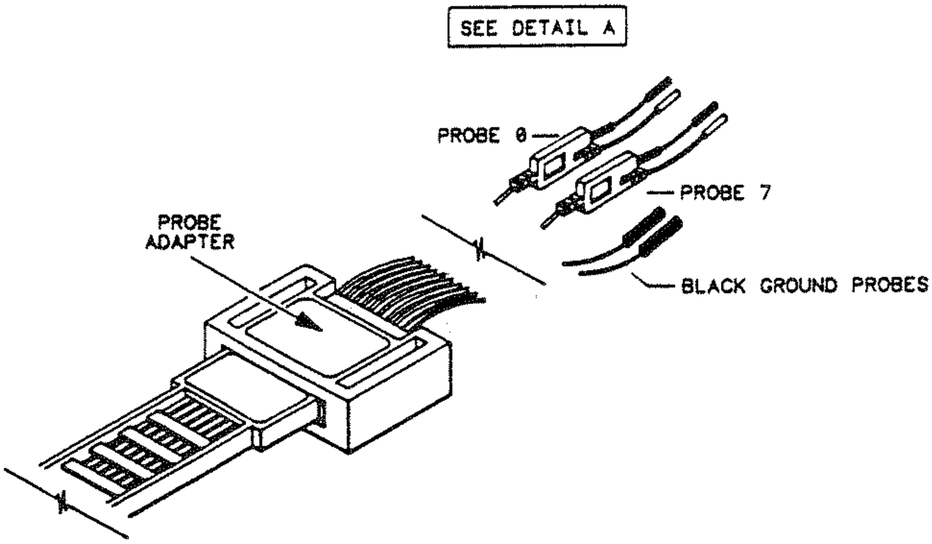
In you are using an external clock or input qualifiers, you'll need to connect cable one to the external input connector, which is on the left of the card when looking at the rear of the board. Attach a clip-on label holder to the cable and the brown A1 label.

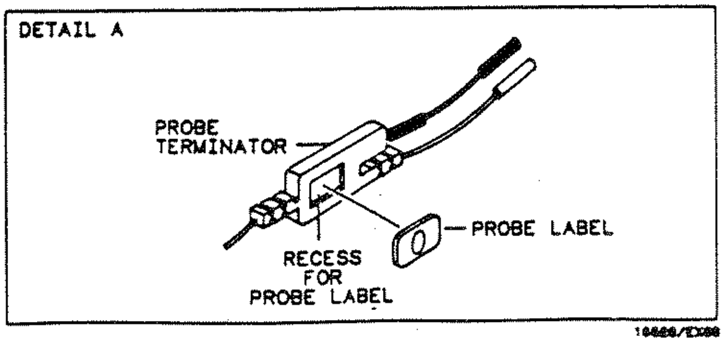
The diagram on page 2-8 shows all the connectors on master and expansion cards along with numbering and function.

Attaching Labels to the Probes

Each pattern generator card is supplied with a sheet of probe numbering labels. The probe labels are color coded to match the cable labels. If you have a red A2 label on a cable, there are eight red labels numbered 0 - 7 that go on the probes of that cable. This color coding makes it easy to identify which probes belong to what cable in case you have numerous probes intertwined on your target system.

Each of the violet probe tips has a recess on one side to allow for a probe number label. To label a probe, remove a stick-on label provided and place it in the recess on the probe tip. Start with probe 0, which is furthest from the pod ground leads.





There are also labels for the input qualifier probe. These are for external clock (**CLK**), input wait qualifiers (**W0 - W2**), and test (**T0 - T3**).

Connecting Probes to Your Target System

Of the ten probes on each probe adapter, eight are for signals and two are for connection to the circuit ground. All signal probes are gray, with a violet probe tip. The pod ground probes are black and do not have colored probe tips.

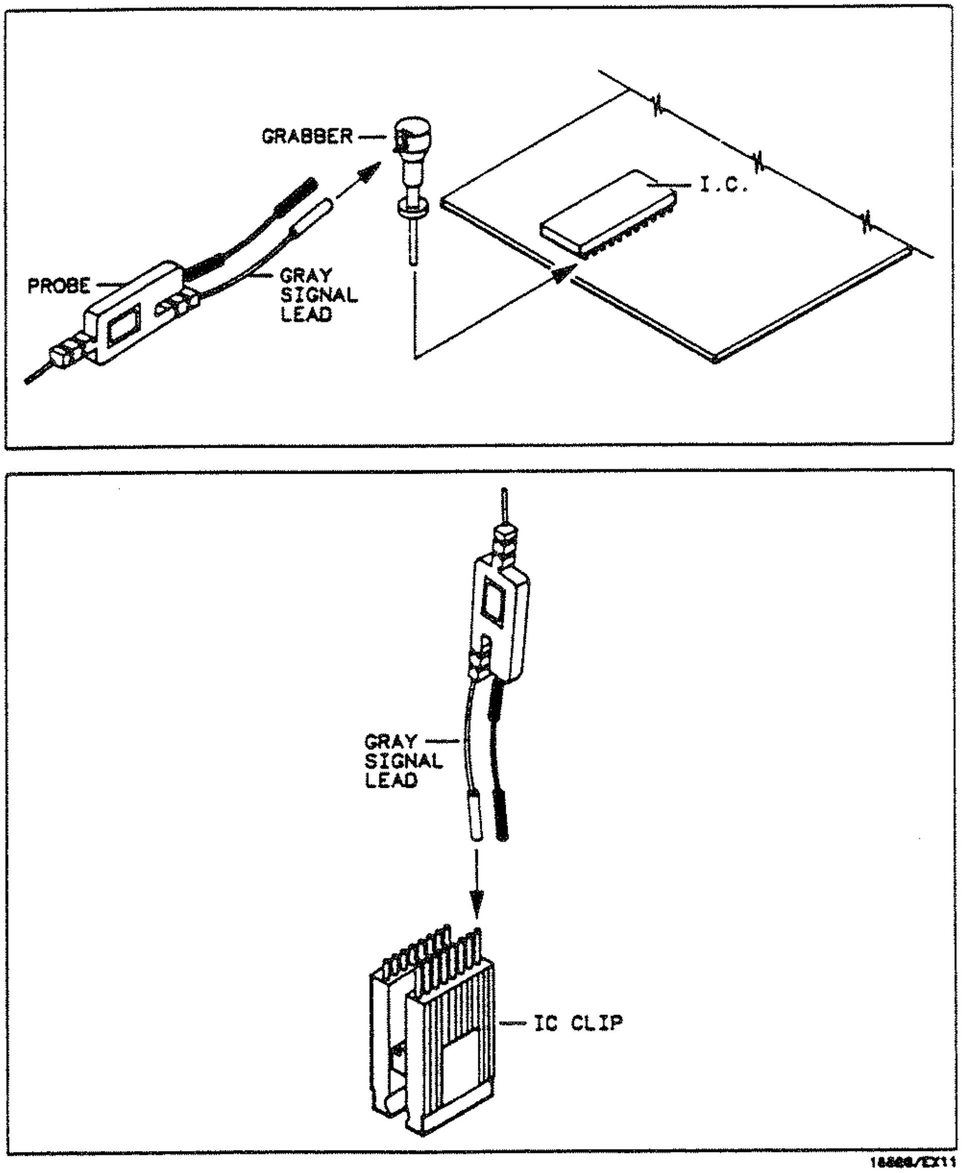
The colored probe tip at the end of each probe has a gray lead to connect the output signal to your target system. There is also a detachable signal ground for each to help maintain signal fidelity. Whenever practical, we recommend that you use the signal grounds.

There are several ways to connect to your target system. First, the probe leads will connect directly to an IC clip or round pins with a diameter from 0.66 mm (0.026 in.) to 0.84 mm (0.033 in.).

You may also clip directly onto your circuit using the optional grabbers with the probes. To connect the grabbers to the probes, simply push the probe lead onto the pin in the head of the grabber.

If you have a pin strip header or square pin connector on your board, you can remove the probe adapter and plug the cable directly onto your connector. The connector must have pin spacing of 2.54 mm (0.1 in.), pin size of 0.63 mm (0.025 in.) and pin height of at least 5.97 mm (0.235 in.). A polarized connector equivalent to 3M* part number 3592-500X or 3592-600X is suggested. A non-polarized pin strip header will also work provided it meets the spacing and pin size requirements given above.

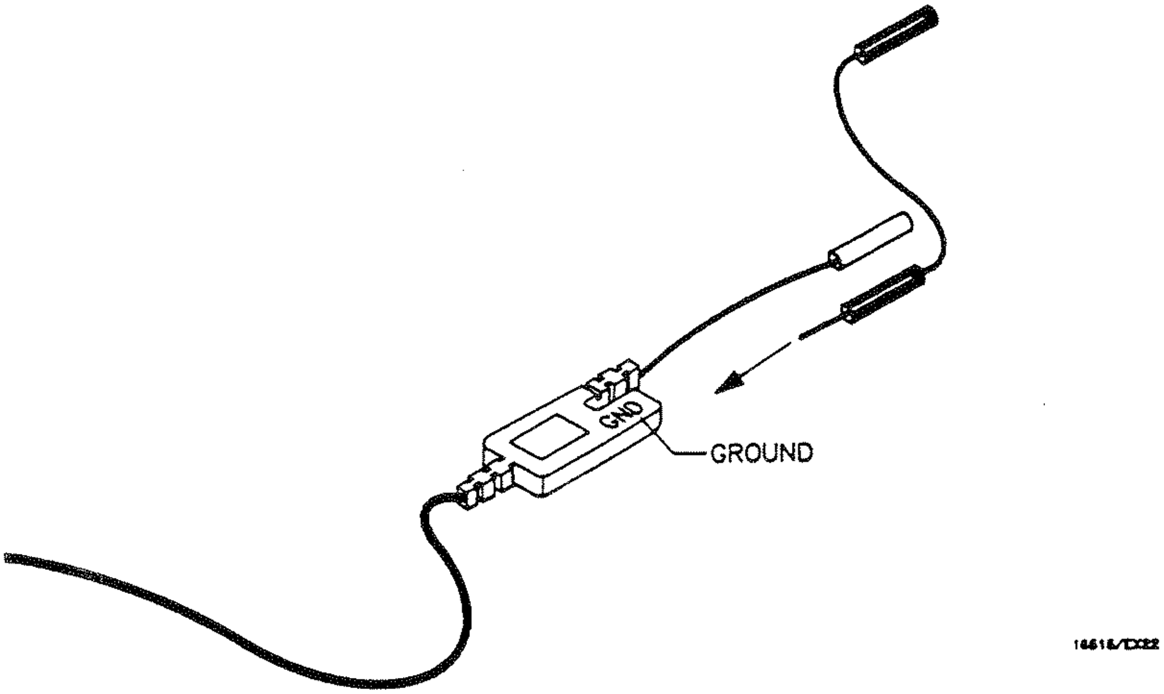
*3M is a registered trade mark of Minnesota Mining and Manufacturing Co.


GRABBER
I.C.
PROBE
GRAY SIGNAL LEAD
GRAY SIGNAL LEAD
IC CLIP
16866/EX11

Using the Pod and Signal Grounds

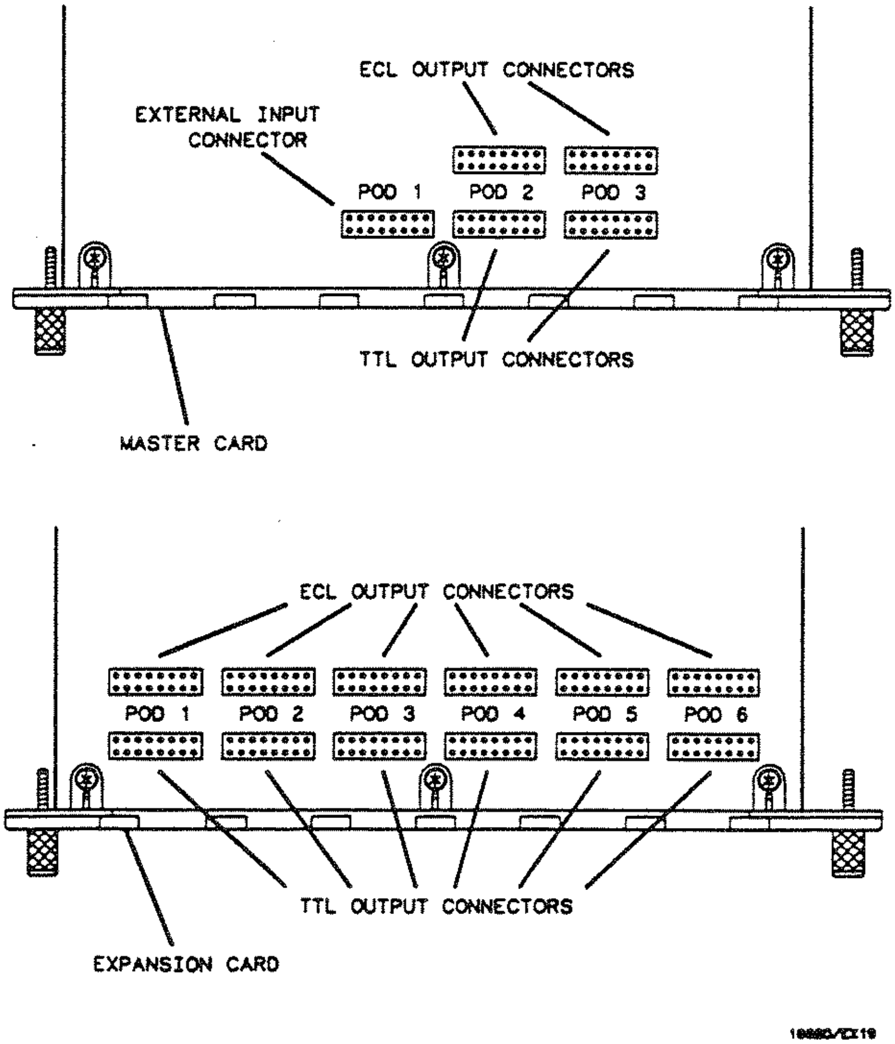
The probe adapter has two separate ground leads that allow you to connect all the signal grounds to a common ground. These pod grounds may be connected by plugging directly onto pins or by means of the grabbers.

At higher frequencies, using only the two pod grounds may affect the edge slew of the output signals. If you are concerned about the possible slewing of output signal edges, we recommend you use the individual signal grounds provided with each signal lead. The signal grounds should be connected as closely as possible to the individual signal leads on your target system. The signal ground leads are connected to each probe as shown below.



Connecting Cables for ECL or TTL Output

On each pattern generator board there are two sets of output connectors, one for TTL output and one for ECL output. The following diagram shows the location of each.



The cables are connected to the TTL outputs from the factory on all factory-installed boards. The procedure for connecting the standard or optional cables is the same and is as follows.

1. Remove the cable restraint by taking out the Torx head (number 10) screws that hold the restraint to the board.

2. Plug the cables onto the TTL or ECL board connectors, depending on the type of output you need. Each cable is "keyed" and will go on only one way. The key on each cable should face toward the rear of the board, i.e., toward the endplate.

3. Lay the cable restraint over the cables. Make sure that all cables are routed through the notches in the restraint. If the cables are not routed through the notches, they may get pinched when the restraint screws are tightened.

4. Replace the cable restraint screws.

Looking at the back of the boards, the pod cables are numbered from left to right, as shown on the previous page.

On the HP 16520A master card, pod 3 contains one clock channel, four data channels and three strobe channels. All eight channels on pod 2 are data channels. Pod 1 contains the external clock and input qualifier channels. The violet or gray label on the cable shows for what each input or output is used.

On the HP 16521A expansion card, all six pod connectors are output data, with eight channels per pod.

Connecting External Inputs

Patterns for the input qualifiers are set in the **Listing** menu with the **Instruction** field. If you use the input qualifiers and external clock, you'll need to use the Input Qualifier Probe Cable, HP Part number 16520-61601.

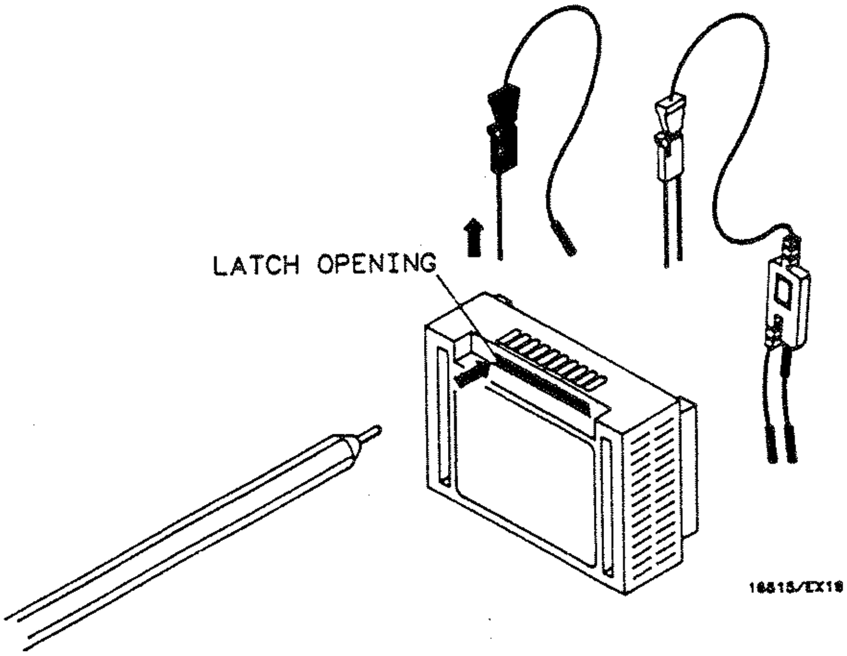
Replacing a Cable

If you need to replace a cable, follow the procedure given under the heading "Connecting Cables for ECL or TTL Output" earlier in this chapter.

Removing or Replacing a Probe Lead

Should a probe break and need to be replaced, or you want to remove unused probes to keep them out of the way, use the following steps:

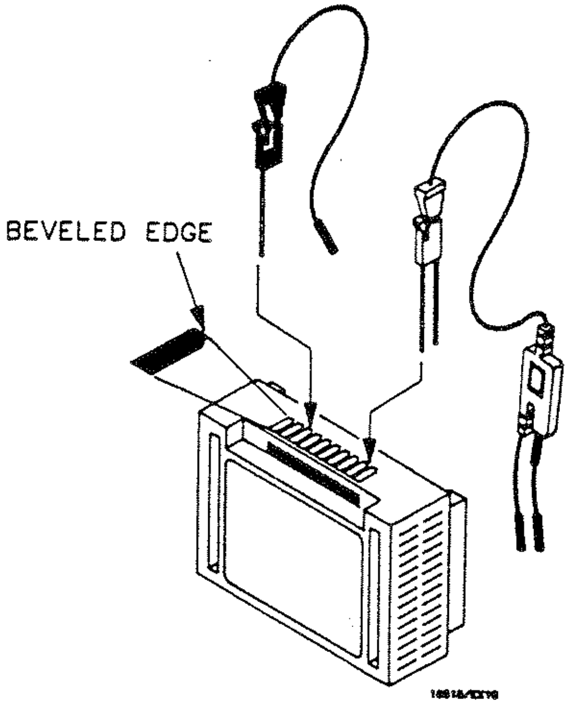
1. Hold the probe adapter with the label facing you.



2. Insert a pen or other pointed object into the notch of the

probe you want to remove. The notch is located at the point where the probe goes into the probe adapter. Press firmly into the notch while pulling gently on the probe lead. The probe lead should pop out.

Notice that one edge of the probe lead is beveled so that it goes into the probe adapter only one way. To reinstall a probe lead, simply push the metal prongs of the lead into the probe adapter until the lead snaps into place. When the lead is in place, you should not be able to pull it out.



The data probes will have two metal prongs, while the black ground probes have only one.

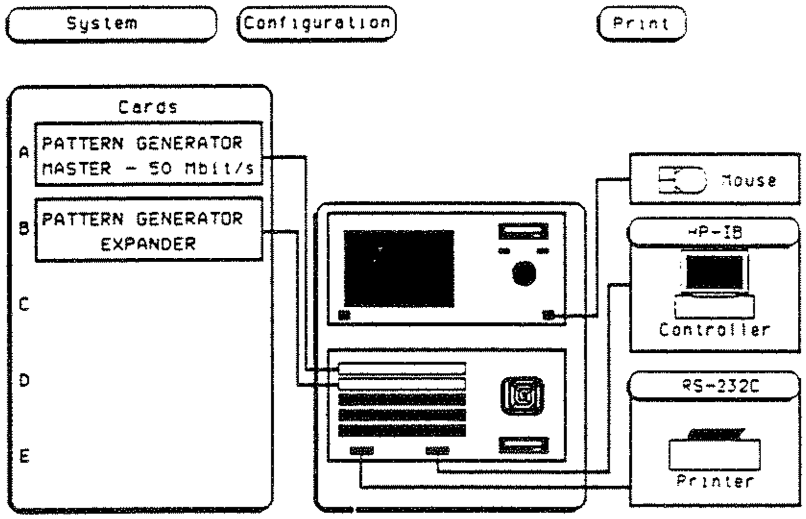
3 Basic User Interface Information

User Interface Devices

The HP 16500A has three user interface devices: the knob on the front panel, the touch-sensitive screen, and the optional mouse. If you are unfamiliar with any of these, this chapter covers the basic concepts of their use. For more detailed information, refer to the *HP 16500A Front Panel Operation Reference.*

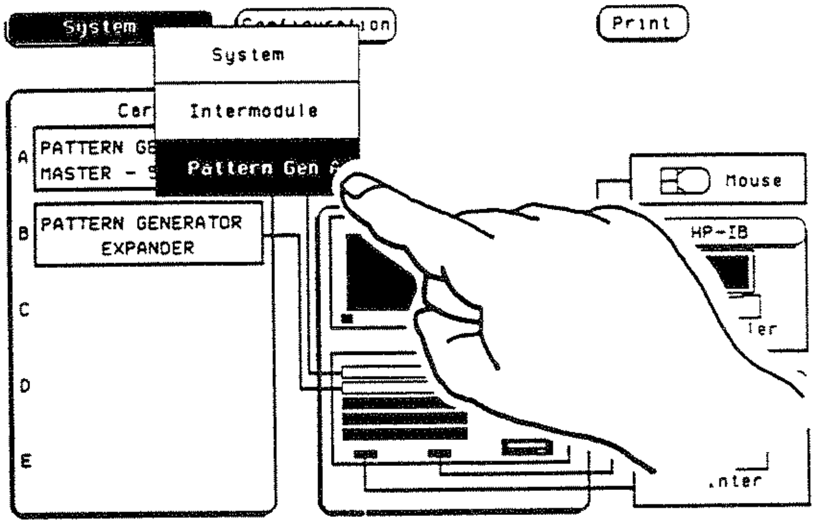
System Power Up

When the HP 16500A system is powered up, the menu you see should look similar to the one shown below.


System
Configuration
Print
Cards
A PATTERN GENERATOR MASTER - 50 Mbit/s
B PATTERN GENERATOR EXPANDER
C
D
E
Mouse
HP-IB
Controller
RS-232C
Printer

Using the Touch Screen

Any dark-blue field on screen is a "touchable" field. That is, if you touch a dark-blue field, the field will toggle to another value, or a pop-up will appear allowing you to select another function. For example, touch the dark-blue field labeled **System** in the upper left of the screen. A pop-up appears showing all the modules and software options of the mainframe. The actual order and content of this pop-up may vary depending on the modules you have installed and which slots the modules are in.

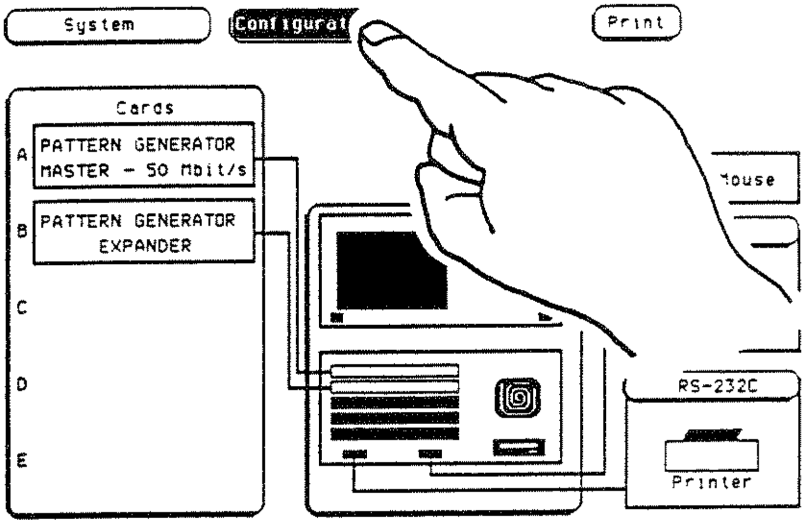


Notice that the **System** field in the pop-up is highlighted in light blue. This tells you that you are in a system menu. To move to any module in the list, touch that field in the pop-up. The pop-up will close and the module chosen will appear on screen.

If you are in any other module menu, you can return to **System** by touching the module field in the upper left of the screen. When the pop-up appears, notice again that the module you are in is highlighted in light blue. Remember that the dark-blue field in the upper left of the screen allows you to move among the modules.

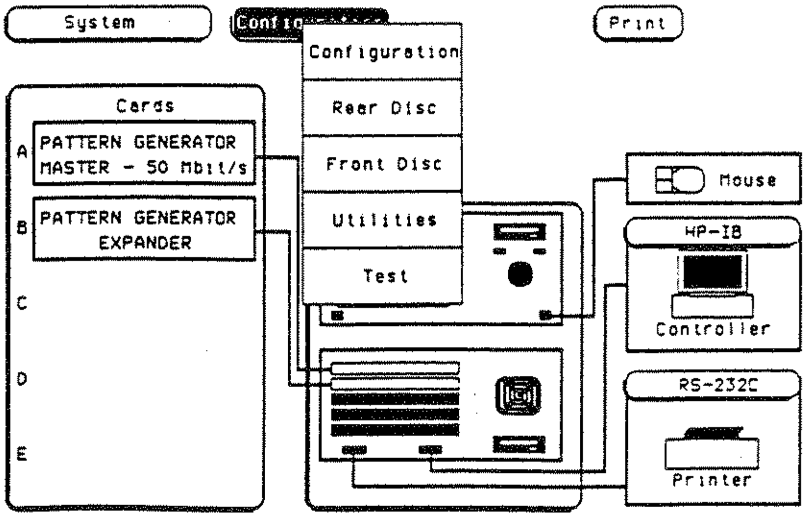
Module Menus

Each module may have several menus within it. To see these menu selections, touch the dark-blue field second from the left at the top of the screen.



System
Configurat
Print
Cards
A
PATTERN GENERATOR MASTER - 50 Mbit/s
B
PATTERN GENERATOR EXPANDER
C
D
E
Mouse
RS-232C
Printer

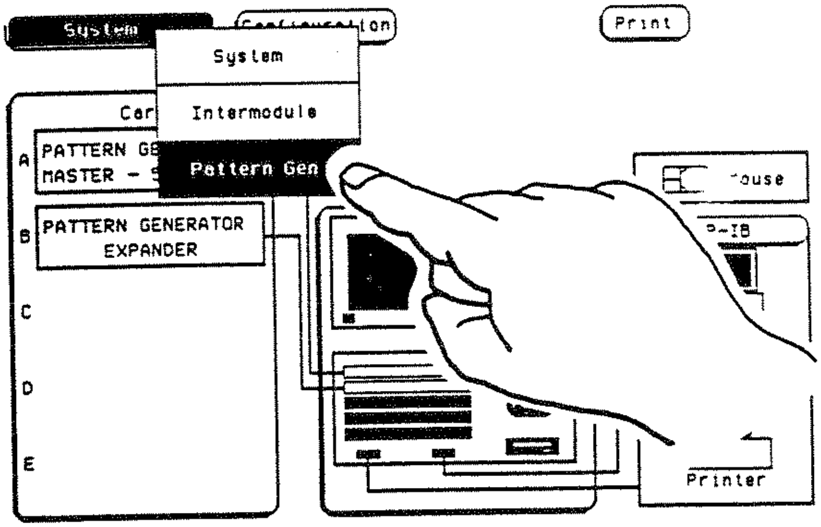
This menu field works the same as the module field to the left of it, except instead of showing all the modules, the menus within each module are displayed. For instance, if you touch this menu field while you are in **System**, you'll get a pop-up that looks like the one shown below.



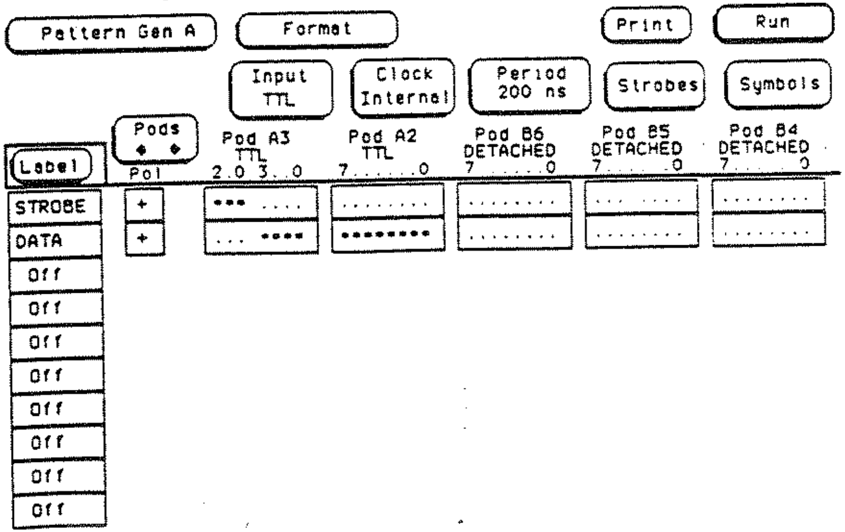
Moving to the Pattern Generator

Touch the module field in the upper left of the screen. A pop-up will appear similar to that shown below displaying all the modules and software options in the mainframe. The actual order and content of the pop-up will vary depending on the modules you have installed and their slots. The capitalized letter to the right of the module name refers to the slot in the mainframe where the module is installed.

In this example, to get to the pattern generator menus, you would touch the field labeled **Pattern Gen A**.

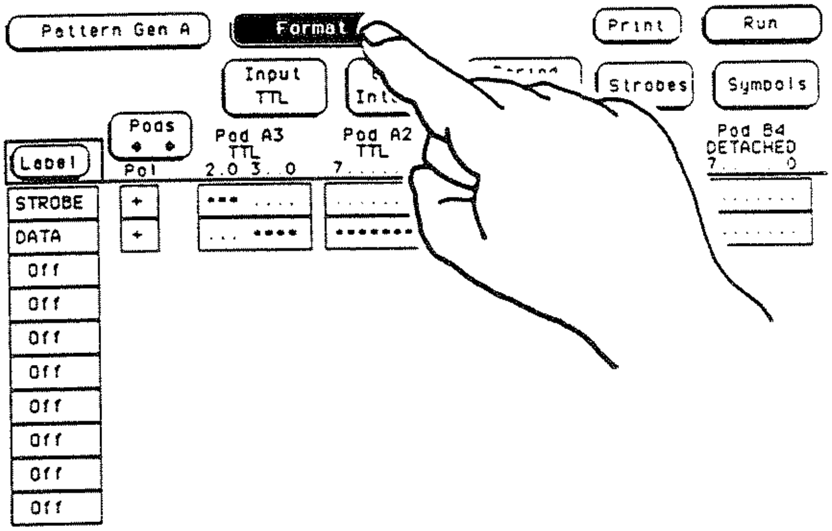


This will bring up the pattern generator **Format** menu.

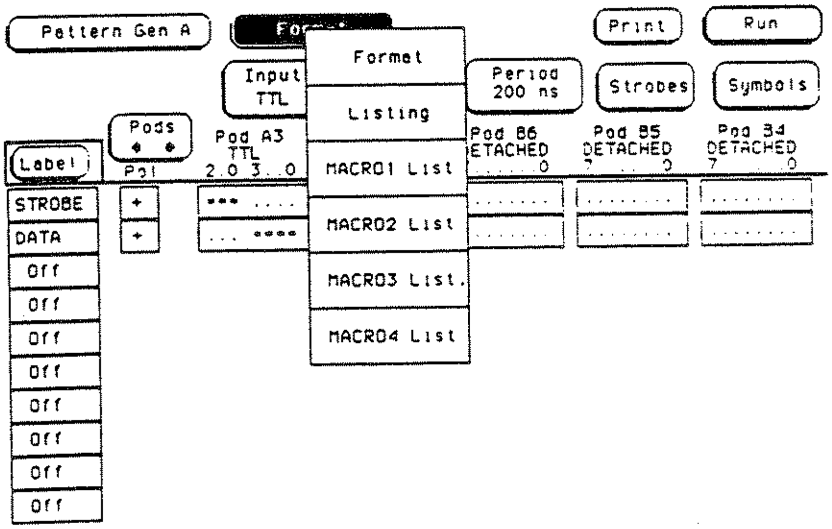


Pattern Generator Menus

The pattern generator has six menus. You can access them by touching the menu field to the right of the **Pattern Gen A** field. Touch the **Format** menu field which is currently being displayed.



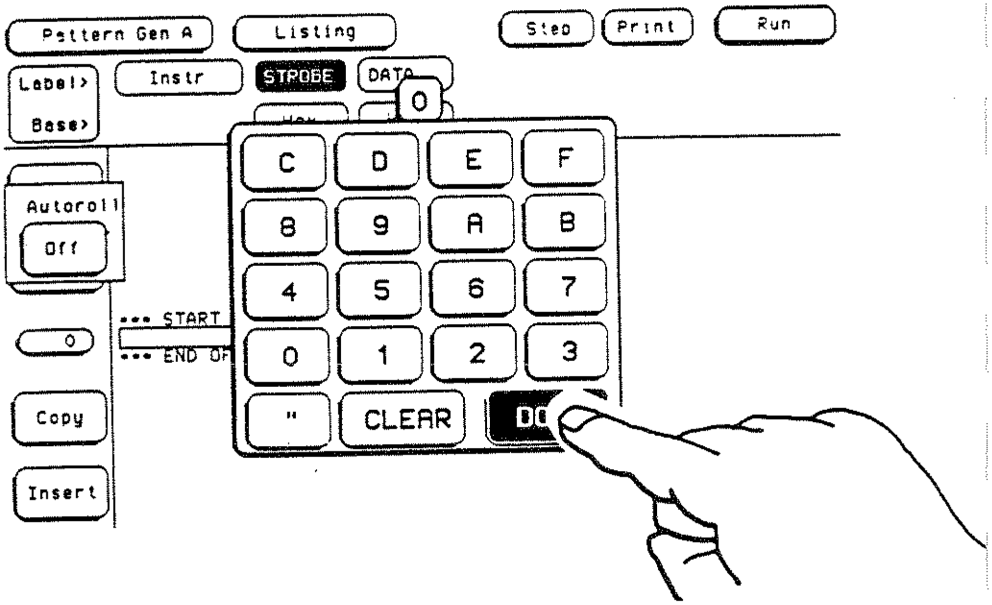
A pop-up appears with all the pattern generator menu selections.



The following chapters will familiarize you with these pattern generator menus. For now, touch the **Format** field in the pop-up menu to return to the **Format** menu. Just remember that the field next to **Pattern Gen** always shows which pattern generator menu is displayed.

Data Entry Fields

When you need to enter alphanumeric or numeric data in some fields, a pop-up keypad will appear on screen to allow you to enter the information. In this pop-up, there is a field labeled **DONE**. This field lets the instrument know that you are finished entering data. The keypad pop-up will not close until you touch the **DONE** field.



What's the Knob For?

To the right of the screen is a knob. Turning the knob allows you to roll the screen up and down for lists, left to right when getting to information off screen, or for positioning the cursor when entering information from a keypad.

Using the Mouse

Everything that can be done with the touch screen and knob on the HP 16500A can also be done with the optional mouse. The mouse plugs into the connector in the lower right of the front panel. As soon as the mouse is plugged in, it is active.

When the mouse is plugged in, a white cursor (cross) appears on screen. Moving the mouse causes the cursor to move. To "touch" a field with the mouse, move the cursor to the field and press the left mouse button.

To use the mouse to perform the functions of the front-panel knob, hold down the right mouse button and move the mouse. When you release the right button on the mouse, the function returns to the cursor.

4 Setting Pattern Generator Values

Introduction

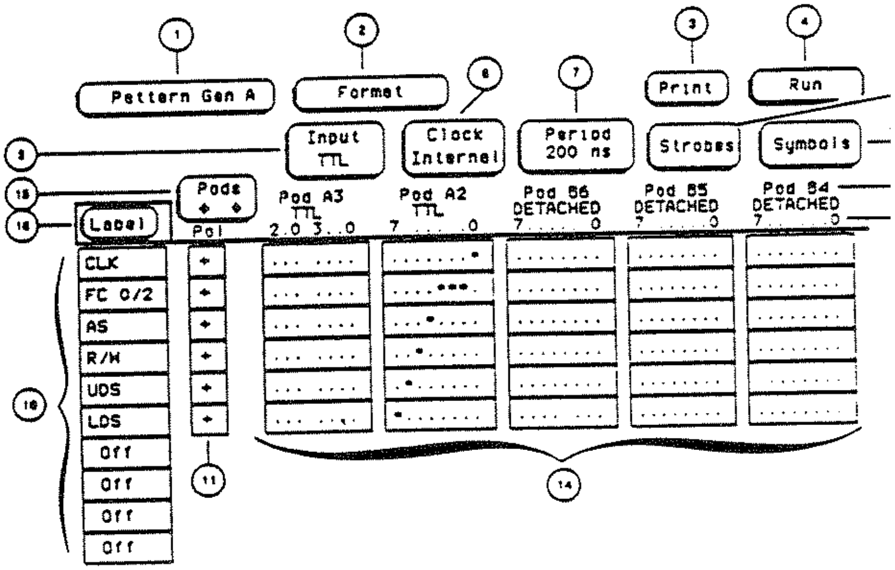
In the HP 16520A/16521A Pattern Generator, similar functions are generally placed together under a single menu. For instance, the operating values of a pattern generation program are under the **Format** menu, where you set the format of your data. The **Listing** contains the list of patterns and the sequence in which they are to go out. And the **Macro menus** let you write macros to eliminate entering redundant pattern sequences.

This chapter tells you how to set the pattern generator program values like data output rate, strobe width and delay, and the channels you want to be active. It also tells you how to group channels together under a common, user-defined name. All of these tasks are done in the **Format** menu.

The pictoral index on the next page gives you a visual map of the **Format** menu. It gives you the name of each field in the menu, along with the page or chapter number where you'll find more information about its function.

All the pictures in this manual were taken from an HP 16500A with one HP 16520A master card and one HP 16521A expansion card. If the screens on your instrument differ from the pictures in this manual, it simply means that you have a different card configuration. All other functions will work the same except where noted.

Format Menu Field Index



	Description	Manual Page or Chapter
1	Module Field	3-2 to 3-3
2	Menu Field	3-2, 3-4
3	Print Field	Chapter 11
4	Run Field	Chapter 7
5	Input Type Selection Field	4-14 to 4-15
6	Clock Source Selection Field	4-10 to 4-12
7	Clock Period Selection Field	4-10 to 4-12
8	Strobe Definition Field	Chapter 9
9	Symbol Definition Field	4-15, Chapter 8
10	Label Fields	4-3 to 4-6
11	Output Polarity Field	4-9
12	Channel Numbering	4-6
13	Pod Numbering	4-6
14	Channel Assignment Fields	4-7 to 4-8
15	Horizontal Roll Field*	4-13 to 4-14
16	Label Roll Field	4-12 to 4-13

* Appears only if one or more HP 16521A expansion modules are installed.

Naming Channel Groups

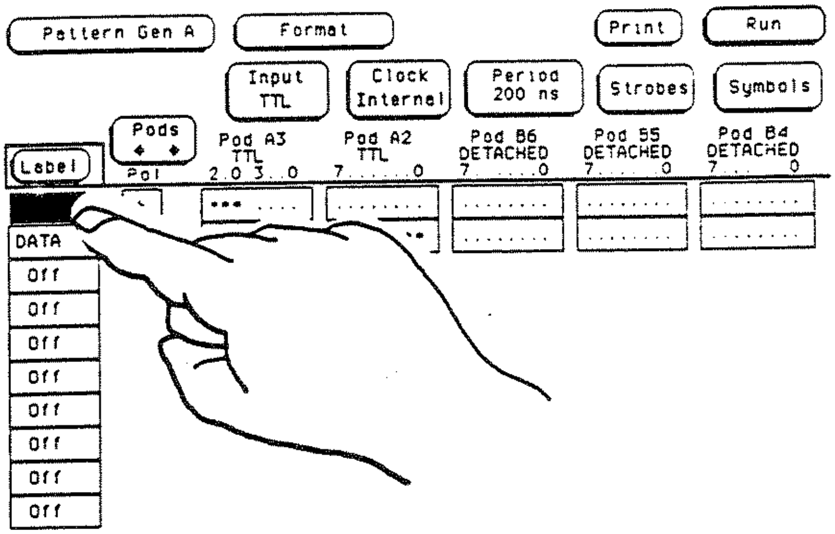
Menu: Format
Fields: Label (9)

Each channel you're going to use must be assigned to a label. A label is a name for a channel or group of channels, like **ADDRSS** or **DATA**. You can give the label any name you want, up to a maximum of six alphanumeric characters. All the labels appear in the leftmost column of the screen.

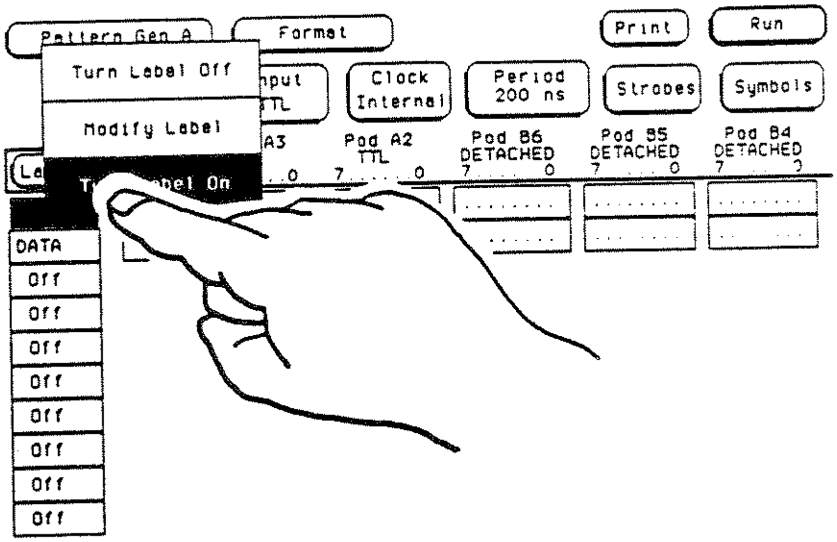
When the **Format** menu first comes up, it has two labels already assigned. You'll notice the label **STROBE** in the upper label field and **DATA** in the field below it. These are the default labels. The labels can be changed, but the default labels appear at first because every label must have a name (i.e., you can't have a label that contains only blank spaces).

Turning Labels On

To turn a label on, you need to touch the label field you want to turn on.

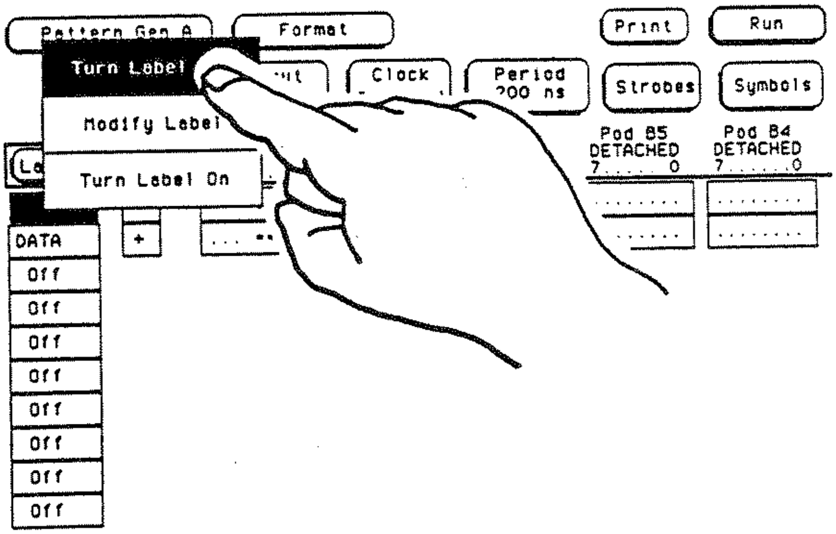


When a label field is touched, a pop-up appears with three choices: **Turn Label On, Modify Label,** and **Turn Label Off**. Touching the **Turn Label On** field turns the label on and assigns a default label. If the label has been previously defined and then turned off, the previous label will show when you turn the field on again.



Turning Labels Off

If you touch a label field, a pop-up with three choices appears on screen: **Turn Label On**, **Modify Label**, and **Turn Label Off**. To turn a label off, touch the field labeled **Turn Label Off**.

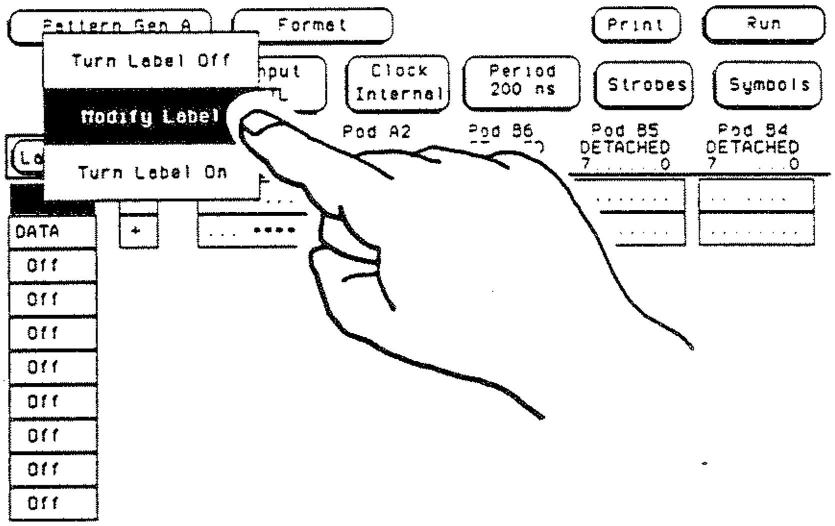


Turning a label off does not destroy the label name you have defined. If you turn the label on again, the name will still be there.

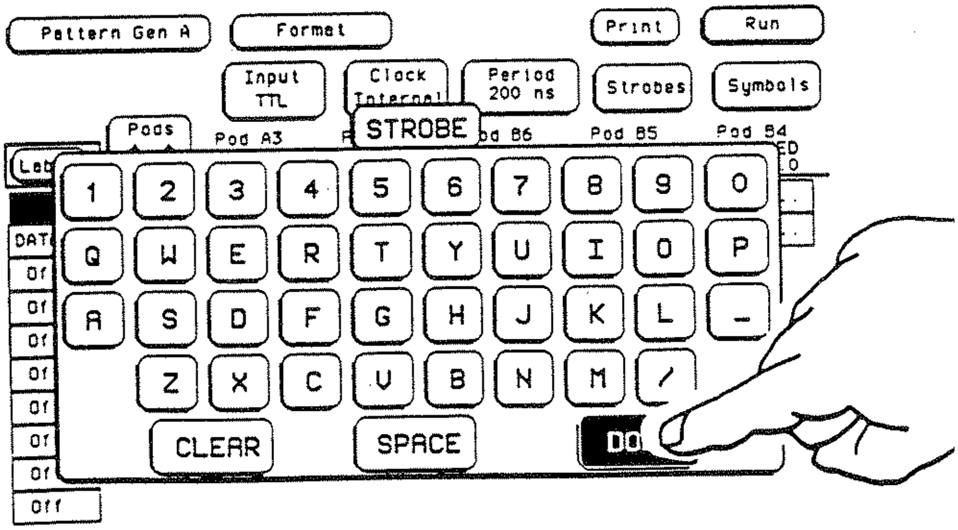
Turning off a label causes all assigned output channels to go to their disabled state.

Modifying Labels

When you touch a label field, a pop-up appears with three choices: **Turn Label On, Modify Label,** and **Turn Label Off**. Touching the **Modify Label** field causes a keypad to come up on screen.



From this keypad you enter the new label. Spaces and any other special character on the keyboard are allowed in the label. To clear the label and start over, touch the **CLEAR** key. If you make an error while entering a label, you can move to the character you want to change with the front-panel knob and enter the correct character. When you are finished with the label, touch the **DONE** key.



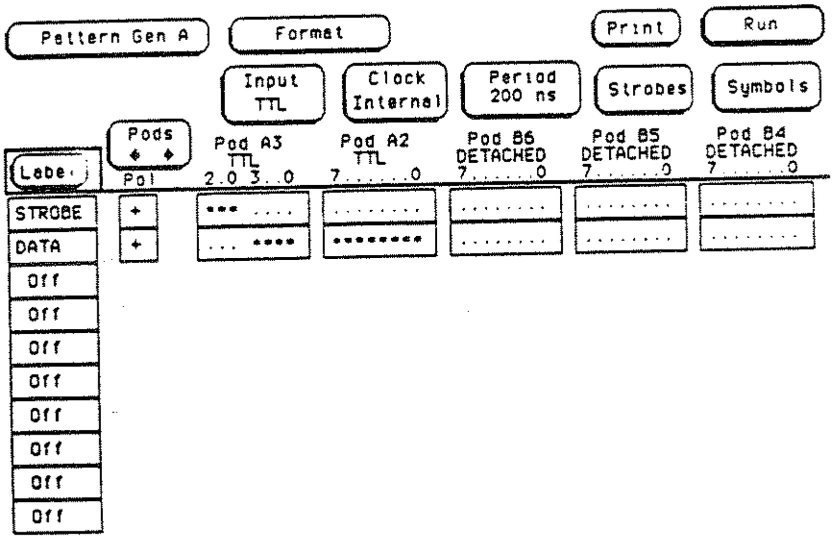
If you touch a label field that says **Off**, you don't need to turn the label field on and then modify the default label. If you touch a label that is turned off, just touch **Modify Label** when the pop-up appears and enter the desired label.

Pod and Channel Numbering

Menu: Format
Fields: Pod, Channel Numbering (11,12)

Above the channel enable fields are the pod and channel numbers. These tell you where each channel is located. The channels of each pod are numbered from right to left, starting with channel 0. Together the channel numbers across the top and the labels along the left side of the screen make up a matrix. While the numbers across the top indicate the physical grouping, the labels on the left show the logical grouping.

This area also tells you if the output cables are connected to the TTL or ECL connectors, or if the cables are disconnected.



Input TTL
Clock Internal
Period 200 ns

Label	Pol	Pod A3 TTL 2.0 3..0	Pod A2 TTL 7......0	Pod B6 DETACHED 7......0	Pod B5 DETACHED 7......0	Pod B4 DETACHED 7......0
STROBE	+	***	....	....	....	....
DATA	+	... ****	********	....	....	....
Off						
Off						
Off						
Off						
Off						
Off						
Off						
Off						

Assigning Output Channels to Labels

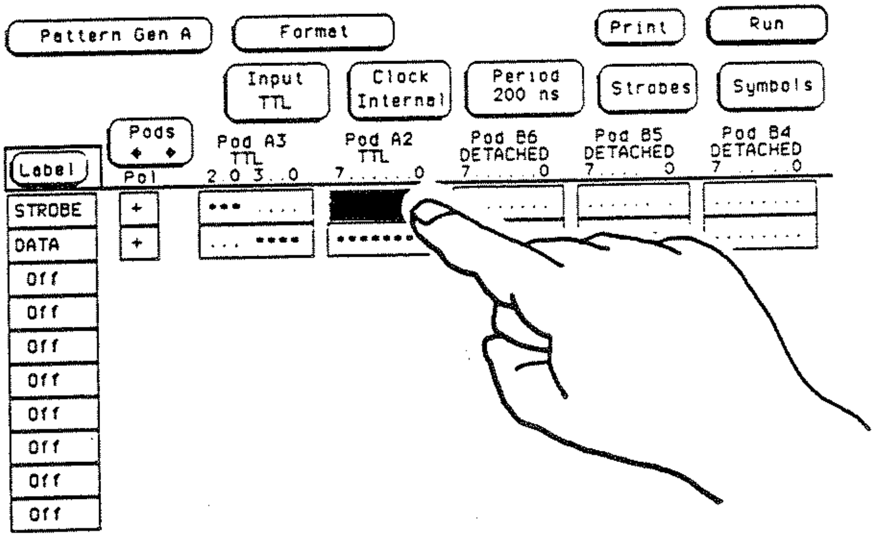
Menu: Format
Fields: Channel Assignment (13)

The process of assigning channels tells the pattern generator which channels are active and to which label each channel belongs.

Each label can have more than one channel assigned to it. For instance, you may have 16 channels assigned to the single label **DATA**. However, a channel can be assigned to only one label.

Enabling Output Channels

To the right of each label are fields allowing you to specify which channels from each pod are associated with that label. In other words, each label may have several channels assigned to it, but those channels need not be on the same physical pod. The channel enable fields let you select where you want each signal to go.

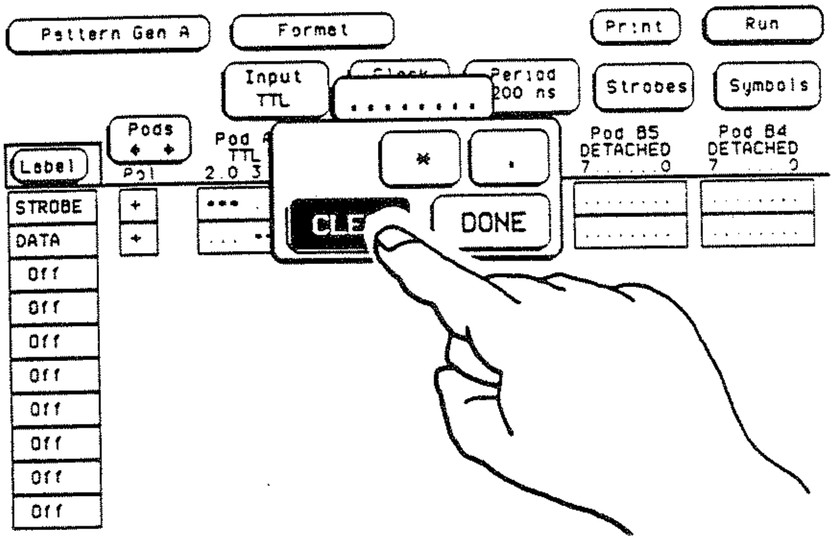


Touching a channel enable field causes a pop up window to appear. The pop-up has two fields with "*" and "." characters on them. The "*" (asterisk) causes a channel to be enabled. For each channel you want enabled, you need to enter a "*" in the channel enable field. You can move to each channel by using the front-panel knob.

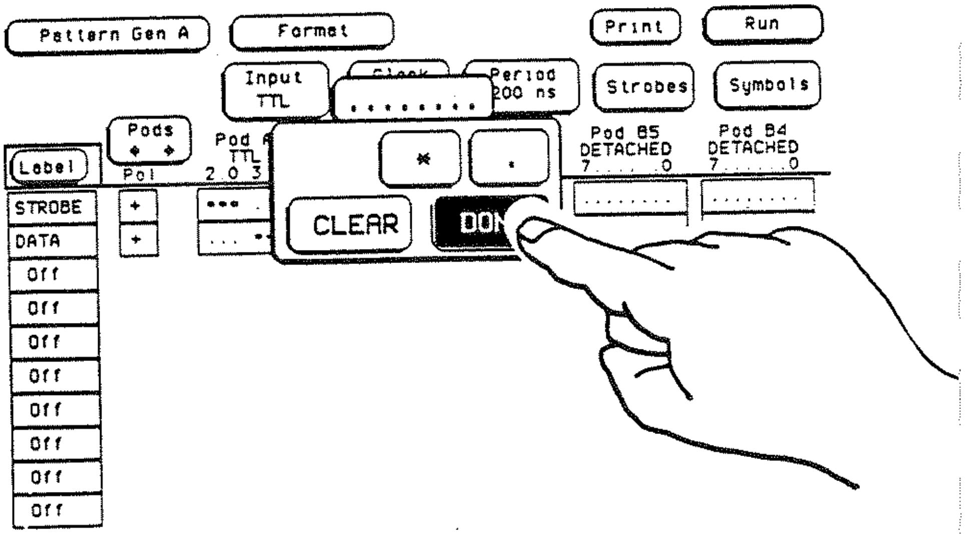
After entering "*" for each channel you want enabled, touch the **DONE** key to close the pop-up.

Disabling Output Channels

Touching a channel enable field causes a pop-up window to appear. The pop-up has two fields with "*" and "." characters on them. The "." (period) causes a channel to be disabled. For each channel you want disabled, you need to enter a "." in the channel enable field. You can move to any channel by using the front-panel knob. Or, you can easily disable all the channels in the pod by touching the **CLEAR** field.



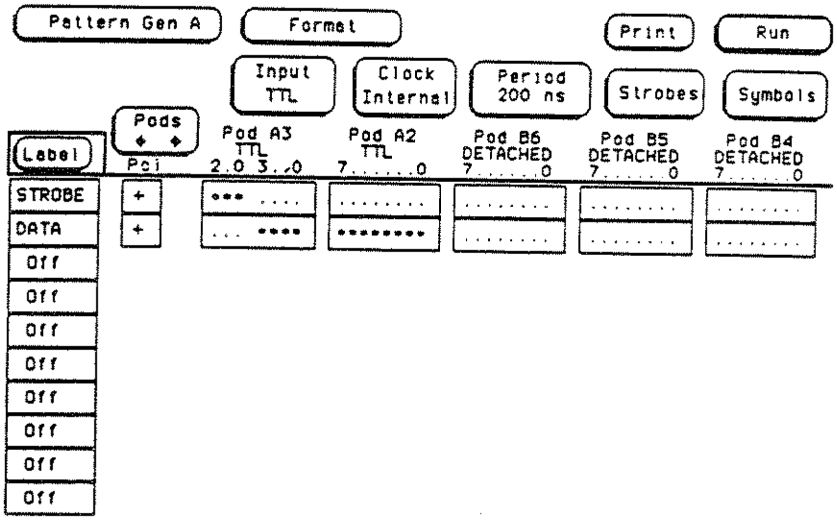
After entering "." for each channel you want disabled, touch the **DONE** key to close the pop up.



Specifying Output Polarity

Menu: Format
Fields: Output Polarity (10)

The fields between the label and channel enable fields specify the logic polarity of the pattern generator output. The field toggles between "+" and "-" when touched. There is one field for each label. For data channels, a positive sign ("+") tells the pattern generator to send out signals that are a high voltage if the pattern requested is a 1. The negative sign ("-") specifies a low voltage if the requested pattern is a 1.


Pattern Gen A
Format
Print
Run
Input TTL
Clock Internal
Period 200 ns
Strobes
Symbols
Pods ◆ ◆

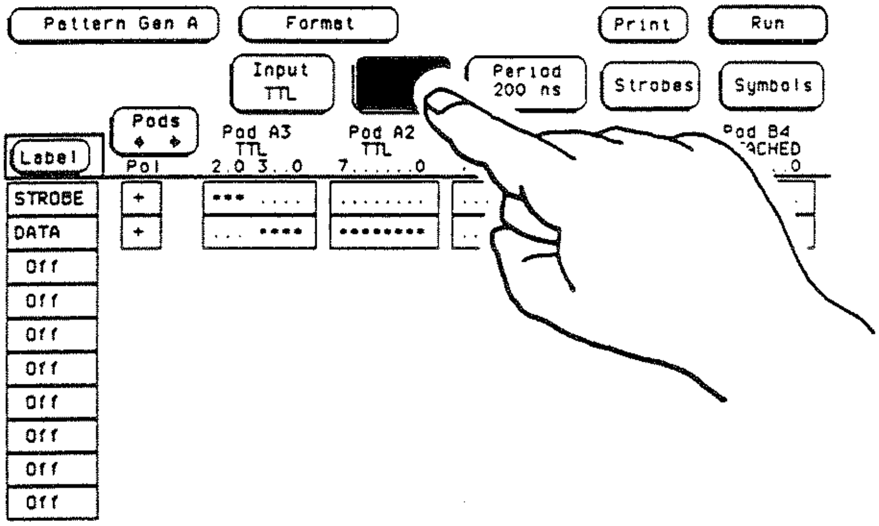
Label	Pol	Pod A3 TTL 2.0 3..0	Pod A2 TTL 7......0	Pod B6 DETACHED 7......0	Pod B5 DETACHED 7......0	Pod B4 DETACHED 7......0
STROBE	+	•••	....	....	....	....
DATA	+	... ••••	••••••••	....	....	....
Off						
Off						
Off						
Off						
Off						
Off						
Off						
Off						

Setting the Data Output Rate (Internal Clock)

Menu: Format
Field: Clock Source Selection (5)

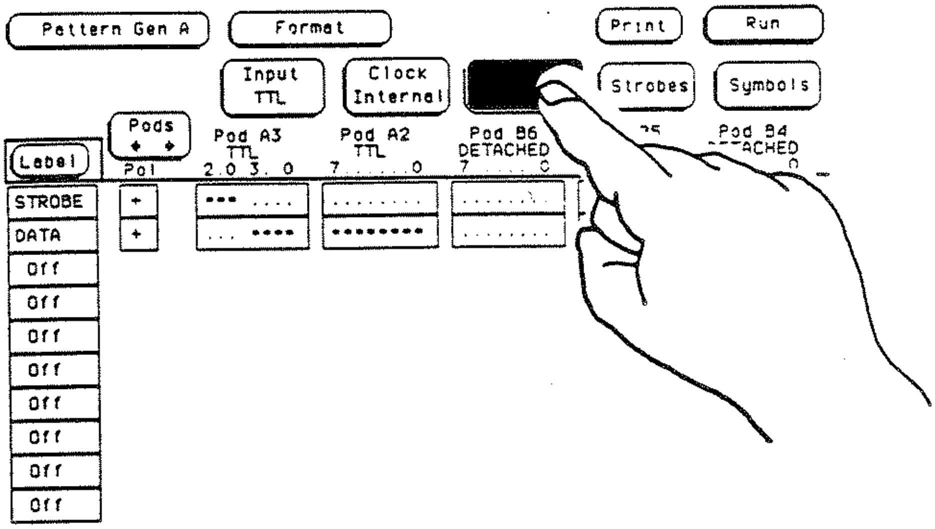
The clock, or what might more correctly be called the output data clock, drives the pattern generation hardware. Each time a new clock period starts, the pattern generator outputs go to their next state, as defined by the listing you specify in the **Listing** menu.

The clock field allows you to select an internal or external clock. When you touch the field, it will toggle between internal and external.



Specifying an Internal Clock Period

The HP 16520A/16521A powers up with the **Clock Internal** field showing. This means that the clock driving the pattern generator is coming from inside the instrument. The internal clock has a selectable period, via the **Period** field. When you touch the **Period** field, a pop-up appears with all the internal clock period choices. The periods are in a 1, 2, 5 sequence from 20 ns to 200 μs. Touch any field to select a period and the pop-up will close.



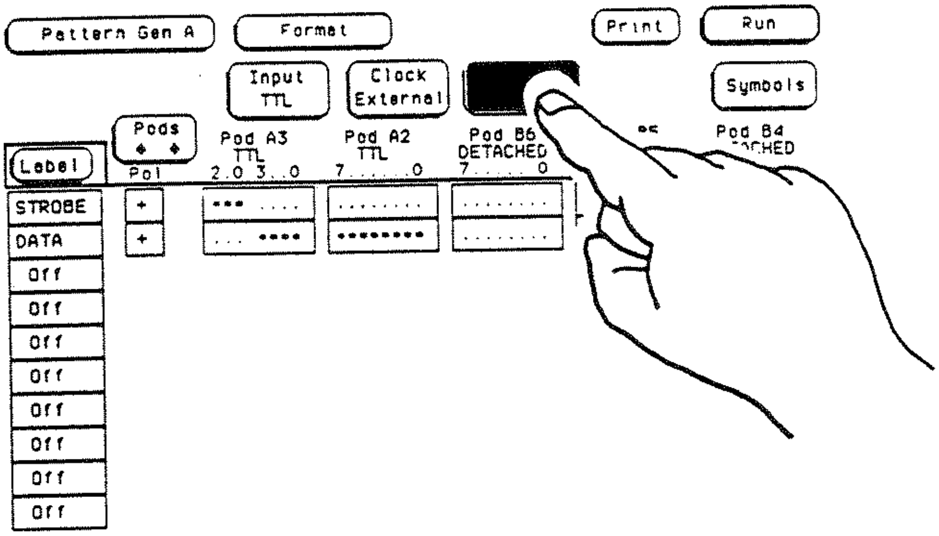
Setting the Data Output Rate (External Clock)

Menu: Format
Field: Clock External (5)

If you touch the **Clock Internal** field, it toggles to **Clock External**. The HP 16520A/16521A can be driven from a user specified external clock. The clock is supplied through the **EXT CLK** pin of pod 1 on the master card (HP 16520A). Pod 1 of the master card must be the HP 16520-61601 Input Qualifier Probe. The pattern generator changes data on the rising edge of the external clock. There will be some propagation delay from the rising edge of the external clock to when data is output.

Specifying an External Clock Period

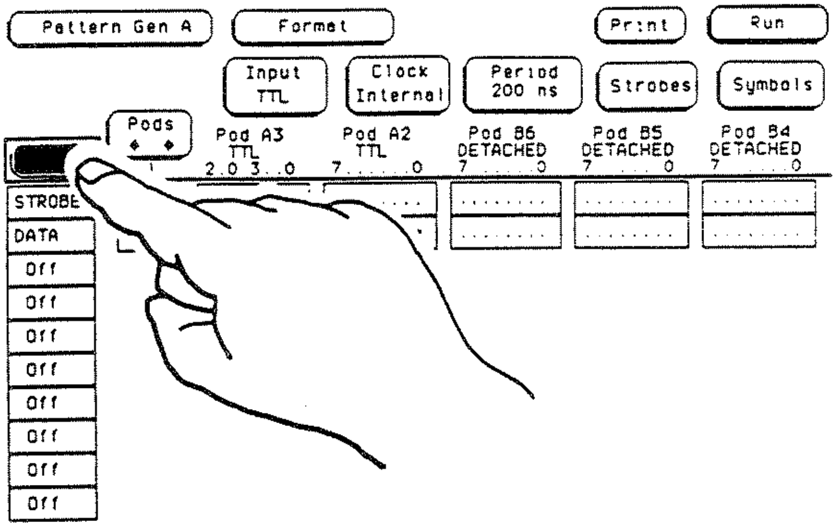
Any external clock is run through a divide circuit on the pattern generator master card. When you choose **Clock External**, the field to the right of the clock field changes from **Period** to **Divide by 1**. By touching this field, you can also select **Divide by 5** or **Divide by 10**. This gives you more capability for strobes. For a complete explanation, see Chapter 9 of this manual, "Defining and Using Strobes."



Rolling the Screen Vertically

Menu: Format
Field: Label Roll (15)

Up to 20 labels can be assigned in the pattern generator. Since only ten labels can be displayed on screen at one time, you can use the knob to roll the list of labels up and down to display any group of labels you like. Above the label fields there is a field that says **Label**. When this field is light blue, the knob will roll the labels up and down. If the field is dark blue, touch it and it will turn light blue.



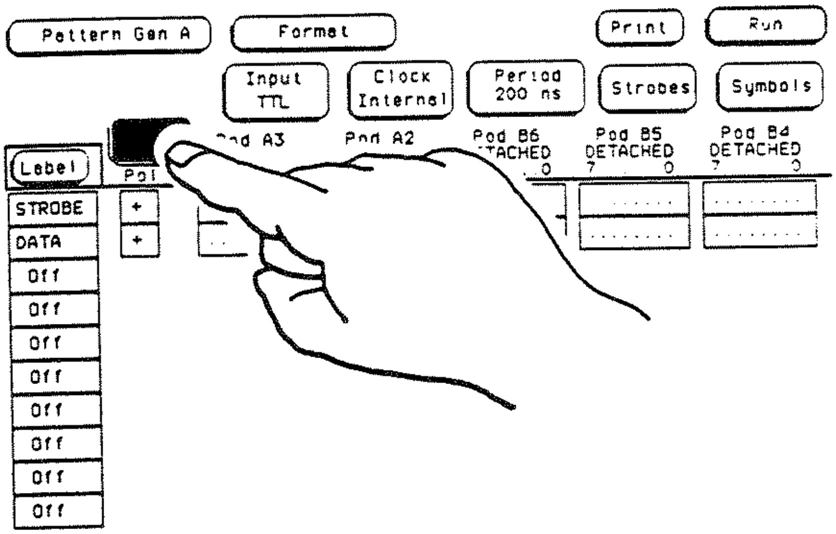
Rolling the Screen Right or Left

Menu: Format
Field: Pods ← → (14)

If one or more expansion cards are installed in the mainframe, the knob serves an additional function. With a master card connected to at least one expansion card in the HP 16500A, there are more channels than can be displayed on screen at one time. The additional channels are off screen to the right. To get to these channels, touch the field in the upper left of the screen labeled **Pods ← →**. This field will turn light blue, indicating that it is assigned to the front-panel knob. Thus, when you turn the knob, the screen will roll left and right. If the **Label** field, discussed on the previous page, is light blue, the knob will roll the screen up and down. If the **Pods ← →** field is light blue, the knob will roll the screen left and right.

Note

The **Pods ← →** *field will not appear if you have no expansion cards installed.*

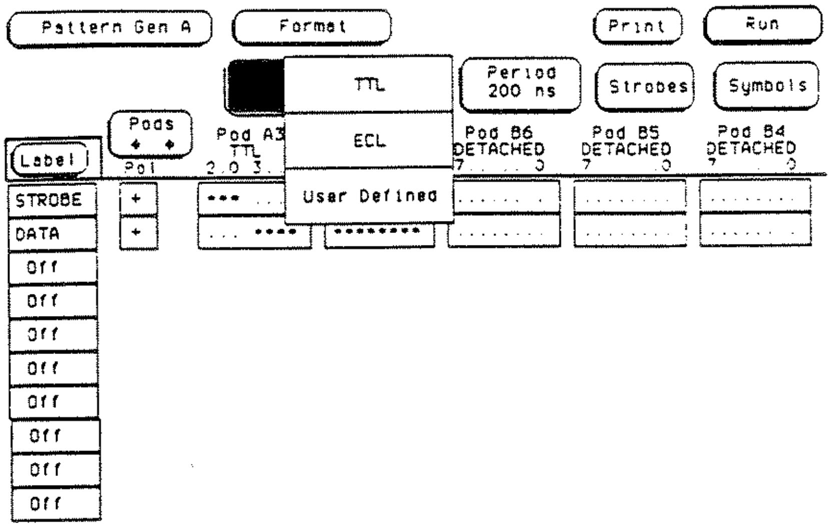


Setting the External Input Signal Type

Menu: Format
Field: Input Type (4)

If you are putting a signal into the pattern generator, like an external clock or input qualifiers, there is a field in the **Format** menu to allow you to set the threshold of the incoming signal. Touch the input type selection field, and a pop-up will appear with three choices for input signal threshold: **TTL, ECL,** and **User Defined.**

Touching **User Defined** will cause a numeric keypad to pop up on screen. From this keypad you can enter the threshold voltage of the external clock and external input qualifiers.



For more information on using and setting input qualifiers, see Chapter 10, "Setting Instructions", and the section entitled *WAIT*.

Using Symbols

Menu: Format
Field: Symbols (8)

Symbols are defined in the **Format** menu, but are used in the **Listing** menu. Because of this, symbols are covered in a separate chapter. Please see Chapter 8, "Creating a Symbol Table."

5 Entering and Editing Output Data Patterns

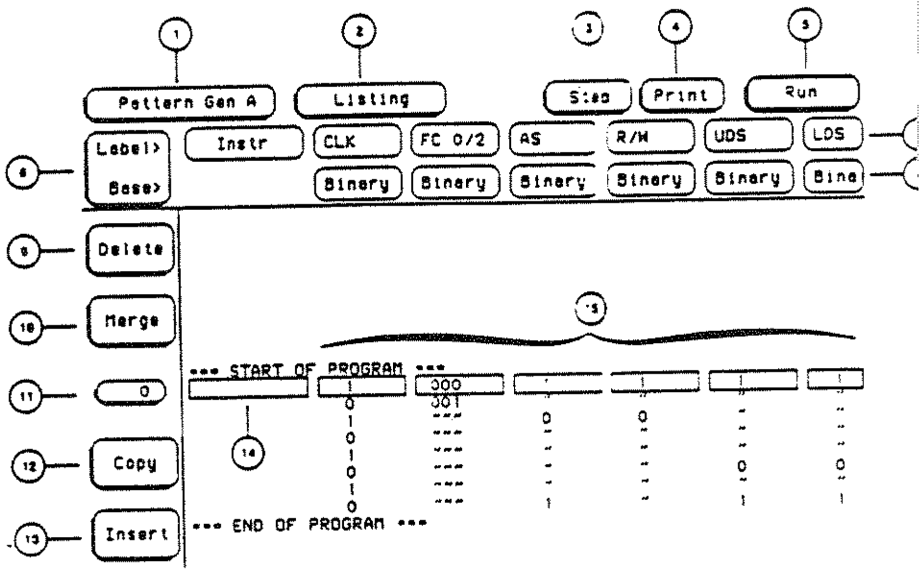
Introduction

In the HP 16520A/16521A Pattern Generator, similar functions are generally placed together under a single menu. For instance, the **Listing** menu contains the list of patterns and the sequence in which they are to go out. The operating values for that list of patterns, such as output rate, are set in the **Format** menu, where you set the format of your data. And the **Macro** menus let you write macros to eliminate entering redundant pattern sequences.

This chapter tells you how to enter and edit a pattern generation list. These functions are performed in the **Listing** menu.

The pictoral index on the next page gives you a visual map of the **Listing** menu. It gives you the name of each field in the menu, along with the page or chapter number on which you'll find information about its function.

Listing Menu Field Index



	Description	Manual Page or Chapter
1	Module Field	3-2 to 3-3
2	Menu Field	3-2, 3-4
3	Step Run Field	Chapter 7
4	Print Field	Chapter 11
5	Run Field	Chapter 7
6	Column Roll Field	5-17
7	Label Fields	5-4 to 5-5
8	Numerical Display Base Fields	5-3
9	Line Delete Field	5-8 to 5-9
10	Program Merge Field	5-12 to 5-14
11	Line Number Field	5-6
12	Line Copy Field	5-10 to 5-11
13	Line Insert Field	5-8
14	Instruction Field	Chapter 10
15	Data Entry Fields	5-7, 5-14 to 5-16

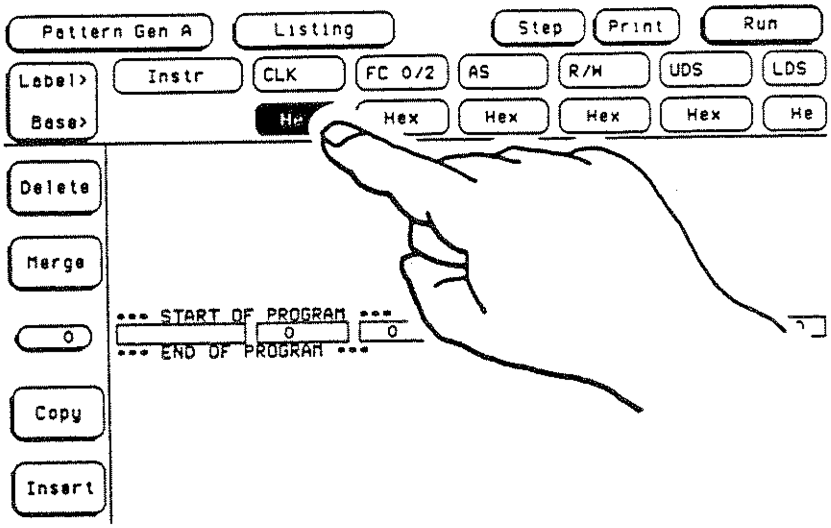
Setting the Display Base

Menu: Listing
Field: Numerical Base (8)

Immediately below each of the labels is a field showing the numerical base used to display the pattern for that label. You may display the channels in binary, octal, decimal, hex, ASCII or as a user-defined symbol.

For example, assume you have a label with three channels assigned to it. If you want a pattern **1 1 1** for those three bits, you can enter it in binary as shown, or in hex as **7**. The number bases allow you to enter and display the program data in a convenient form.

The default number base is hex. If you want to change to another base, touch the numerical base field and select the base you want from the pop-up.



*** START OF PROGRAM ***
*** END OF PROGRAM ***

Reading Labels

Menu: Listing
Field: Label (7)

Labels are defined and output channels are assigned in the **Format** menu. Each of the labels are displayed across the top of the screen in the **Listing** menu, left to right in the same order as in the **Format** menu.

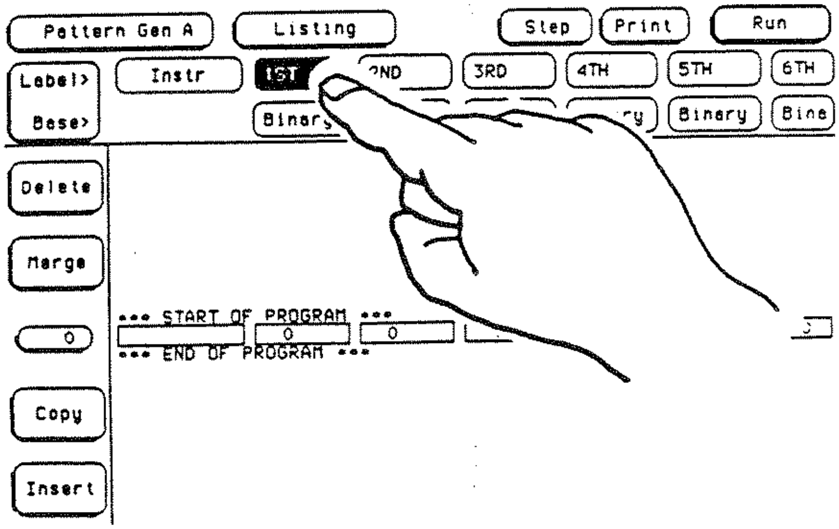
The **Label> Base>** field in the upper left of the screen point to the rows that contain the label and numerical base fields respectively.

All data for the **Listing** menu is entered into the data fields below each label.

Changing Column Order

Menu: Listing
Field: Label (7)

Even though you defined the label order in the **Format** menu, the order may not be convenient for the **Listing** menu. You can change the order of columns in the **Listing** menu without affecting the order defined in the **Format** menu. To change the order of columns, touch the label field.

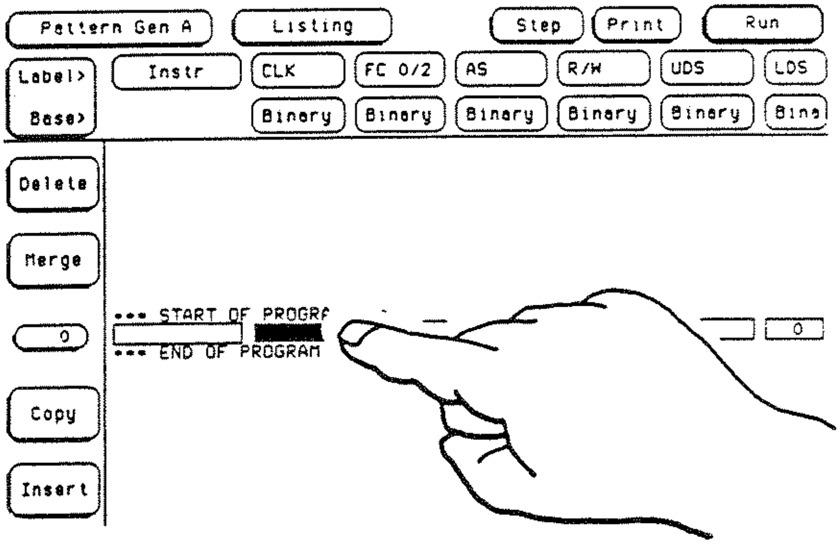


```
*** START OF PROGRAM ***
*** END OF PROGRAM ***
```

Entering Data Output Patterns

Menu: Listing
Field: Data Entry (15)

Output data is entered into the data fields by touching them. When a data field is touched, a pop-up appears on screen, allowing you to enter pattern data. The pop-up will vary depending on what value is shown in the numerical base field. For example, if the numerical base field is set to **Hex**, the pop-up will allow you to enter data in the range of 0 to F. If you select **Binary**, the pop-up displays only zero and one.



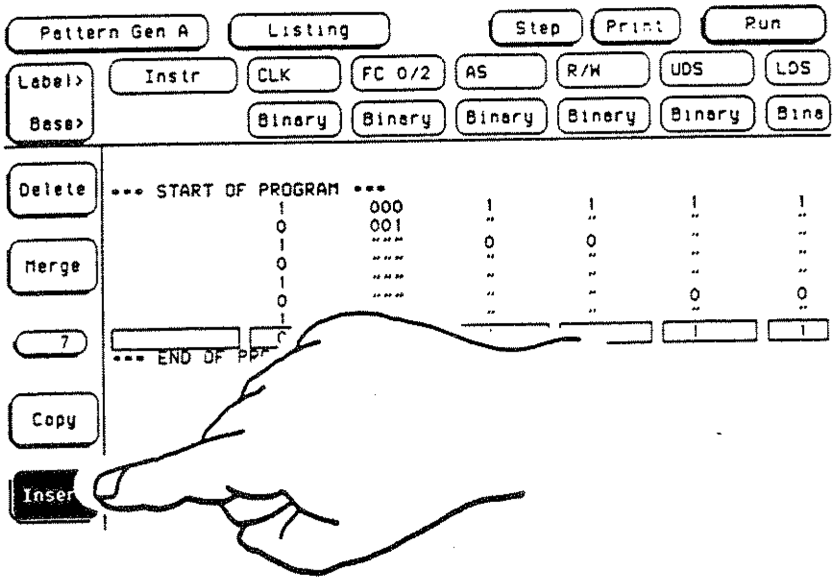
```
*** START OF PROGRA
*** END OF PROGRAM
```

If you try to put a value into a data field that is larger than the maximum value that the field can accomodate, the field will truncate the entry, displaying as much of the entry as it can. That is, if you have the numerical field set to **Hex**, but you only have one bit assigned to the label, the only legal entries are **0** and **1**. If you enter 5 and touch the **DONE** field, the pattern generator will truncate the entry, leaving only the least significant number, which is 1.

Inserting Program Lines

Menu: Listing
Field: Insert (13)

When the pattern generator is powered up, there is only one program line, line 0. To add lines to a program, use the **Insert** field at the lower left of the screen. Each time you touch the **Insert** field, the pattern generator will add one line immediately after the one shown in the line number field. That is, if the line number field shows 7, and you touch the **Insert** field, a line will be inserted immediately after line 7.

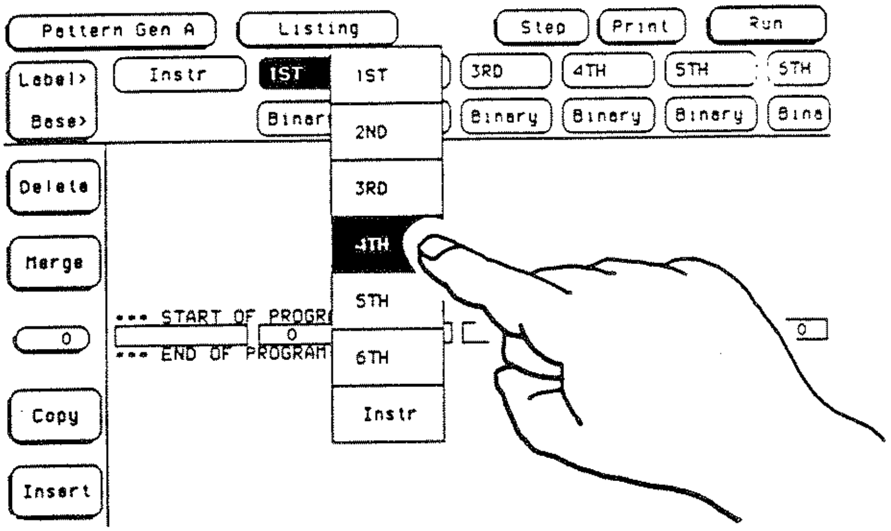


Deleting Program Lines

Menu: Listing
Field: Delete (9)

You can delete one or more lines from a pattern generation program with the **Delete** field to the left of the screen. First, position the line you want to start or end the deletion with in the line number field at the center of the screen. When you touch the **Delete** field, two red

A pop-up will appear with all the labels listed. Touch the field in the pop-up that you want to move.



The pop-up will close. Notice that the label field that you first touched and the label you touched in the pop-up have exchanged places.

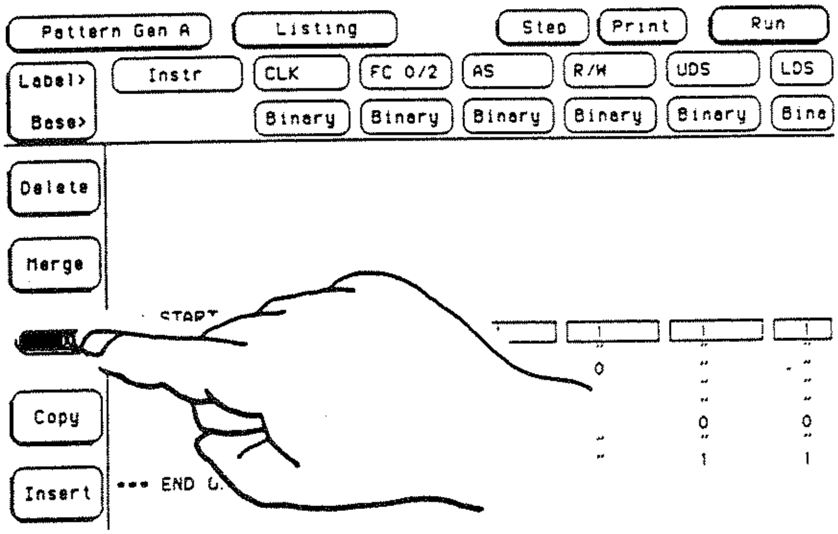
Program Line Numbering

Menu: Listing
Field: Line Number (11)

The field at the center left of the screen shows the program line number. This field shows the current line that can be edited. When the pattern generator is powered up, line 0 is the only one in the new program. And, all data fields are zero.

Each program will start at line 0.

You may move to any line in a program with the front-panel knob. Since the knob may also be used to move the screen left and right, make sure the program line number is light blue before you try to move to a line. If the **Label> Base>** field is light blue, the screen will move left and right instead of up and down. When the program line number is light blue, the knob will roll the program up and down.

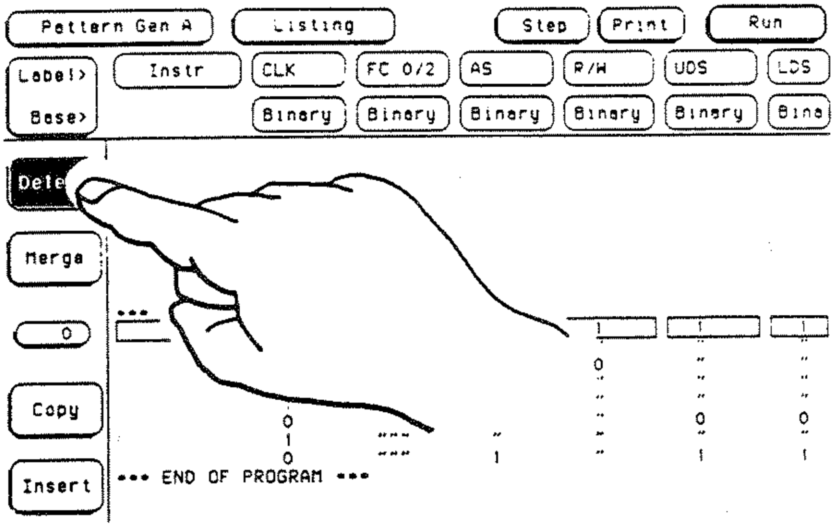


In some cases, as when a program is several hundred or thousand lines long, using the knob may not be the most convenient way to move through the program. In such cases you can touch the program line number (provided it is light blue) and a keypad will pop-up on screen.

Note

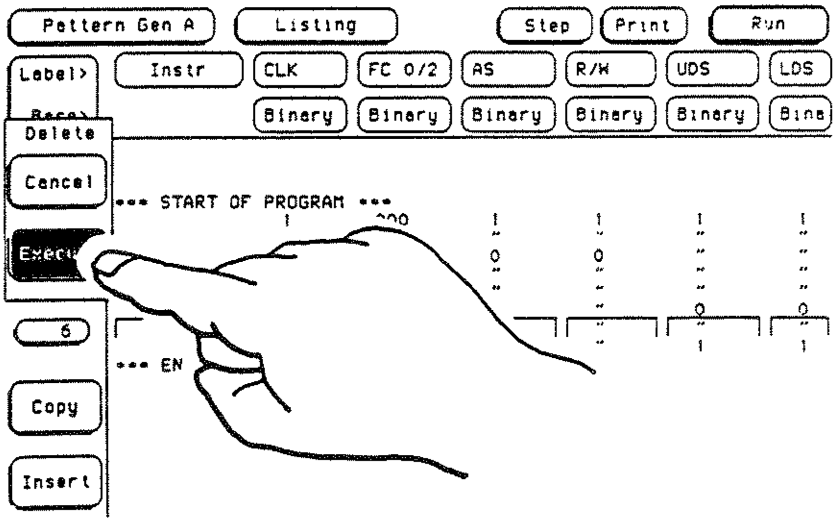
If the line number field is dark blue, touch it once to use the knob and twice to bring up the keypad.

From this keypad you can enter the line number you want to move to. When you finish entering the line number, touch the **DONE** key. The keypad will close and the program will jump to the specified line number. This feature can be used within the copy, delete, or merge functions described later in this chapter.



*** END OF PROGRAM ***

horizontal lines will appear on screen, one above and one below the line in the line number field. A pop-up will also appear at the left of the screen with two fields in it, **Cancel** and **Execute**. Use the front-panel knob, or the pop-up keypad from the line number field to scroll up or down in the program, until the two red lines encompass the line(s) you want to delete. Then touch the **Execute** field in the pop-up. The lines indicated will be deleted and the program will be renumbered.



*** START OF PROGRAM ***

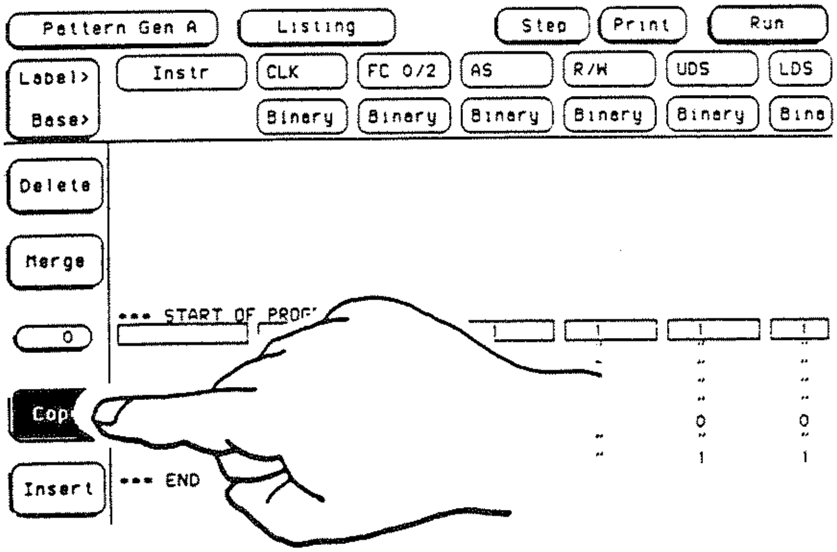
*** EN

If you decide not to delete anything, you can touch the **Cancel** field. This will cancel the delete operation.

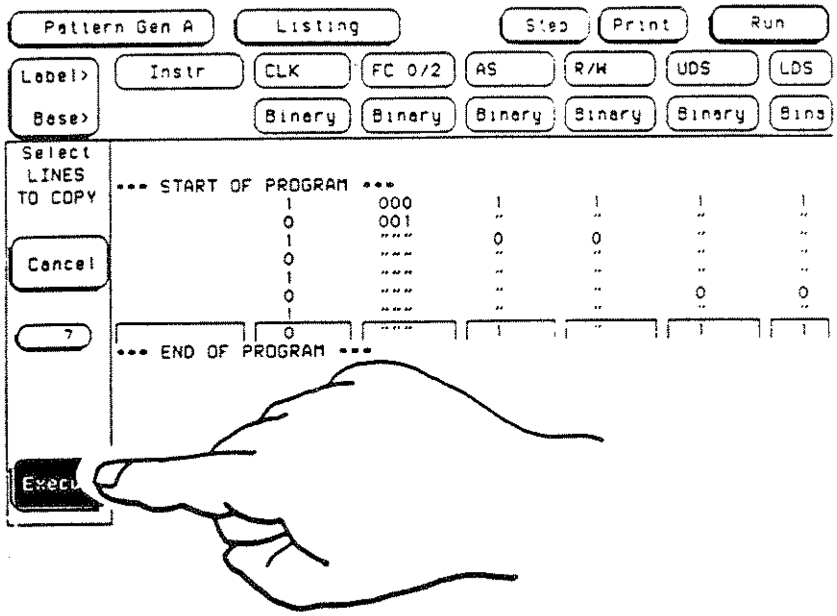
Copying Program Lines

Menu: Listing
Field: Copy (12)

You can copy any portion of a program, provided you are not already at the program limit size (4095 lines). The pattern generator allows you to select the number of copies as well as where you want those copies in the program.

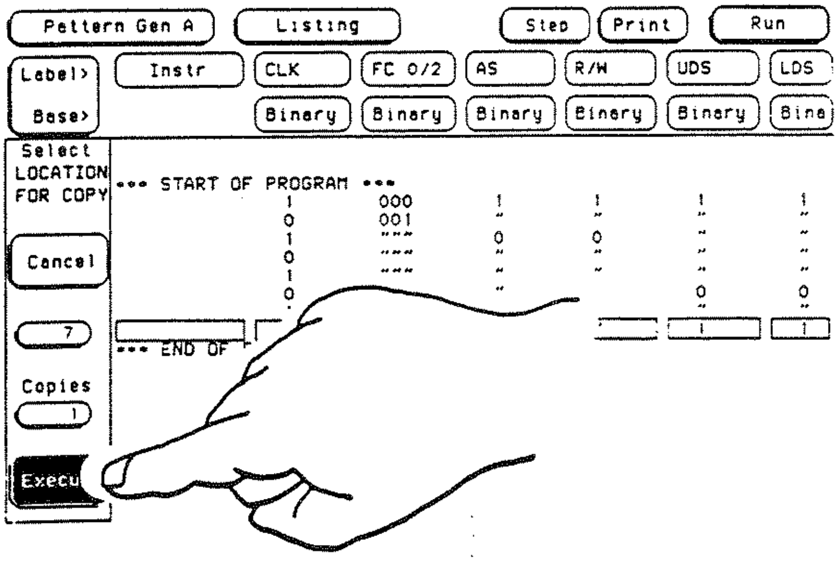


To copy program lines, position the first or last line of the section to be copied in the line number field. Touch the **Copy** field at the left of the screen. A pop-up will appear at the left of screen, with fields labeled **Execute** and **Cancel**. Two red, horizontal lines will also appear, above and below the line. Use the front-panel knob or the pop-up keypad from the line number field to roll up and down in the program until the red lines enclose the section you want to copy, and then touch the **Execute** field. Now, if you want more than one copy, touch the oval field that specifies the number of copies. A pop-up keypad allows you to enter the number of copies. When you have finished entering the number of copies, touch the **Execute** field.



```
*** START OF PROGRAM ***
                   1       000        1         1         1         1
                   0       001        "         "         "         "
                   1       """        0         0         "         "
                   0       """        "         "         "         "
                   1       """        "         "         "         "
                   0       """        "         "         0         0
                   1       """        "         "         "         "
                   0       """        1         "         1         1
*** END OF PROGRAM ***
```

Now that you've told the pattern generator what you want to copy, it needs to know where to put the copies. The pop-up at the left of the screen now prompts you to move to the location in the program where you want the copy(s). Move to the location with the front-panel knob or by touching the line number field and entering a line number. When you reach the point where the copies are to go, touch the **Execute** field again. All the copies specified will be added directly below the line in the line number field.



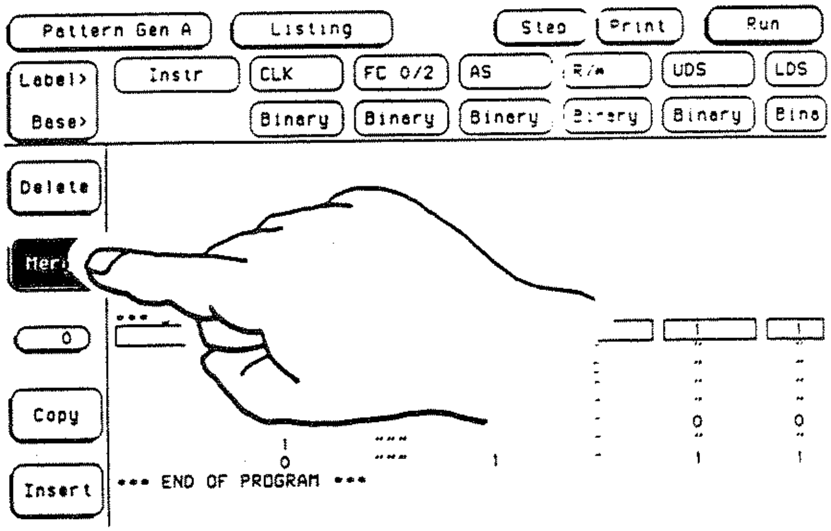
```
*** START OF PROGRAM ***
                   1       000        1         1         1         1
                   0       001        "         "         "         "
                   1       """        0         0         "         "
                   0       """        "         "         "         "
                   1       """        "         "         "         "
                   0                  "                   0         0
                                                            "         "
                                                            1         1
*** END OF
```

If you decide not to copy anything, touch the **Cancel** field at the left of the screen to cancel the copy operation.

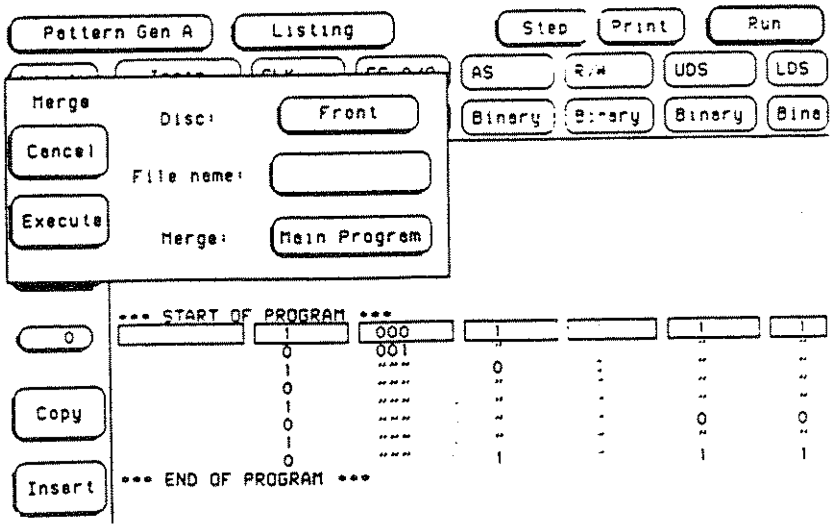
Merging Program Lines

Menu: Listing
Field: Merge (10)

The **Merge** field allows you to merge a pattern generator program stored on disc with the program you are develcoing. You may also merge one of the four macros from the program on disc with your program.



Position the line number to where you want to merge the file in your program. Touch the **Merge** field and a pop-up with five fields appears.



Disc: Front

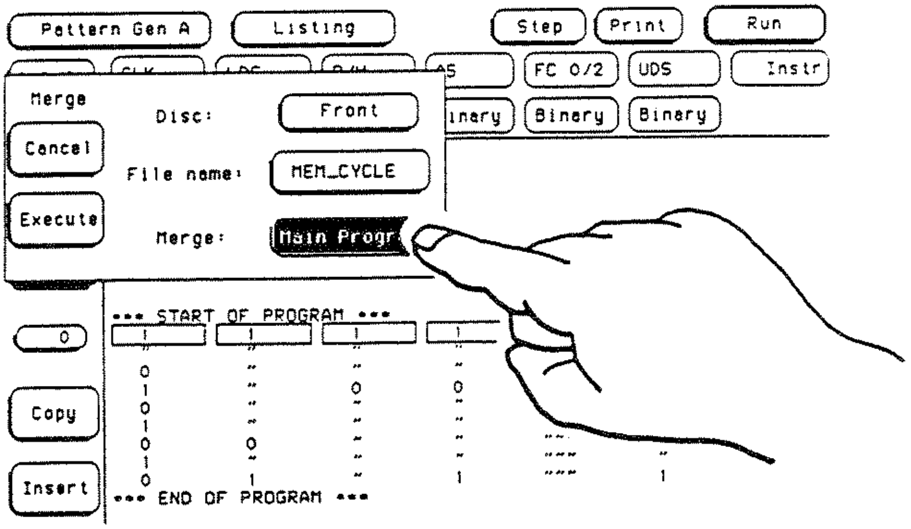
File name:

Merge: Main Program

The **Disc** field in the pop-up lets you select whether the program is to come from the front or rear disc.

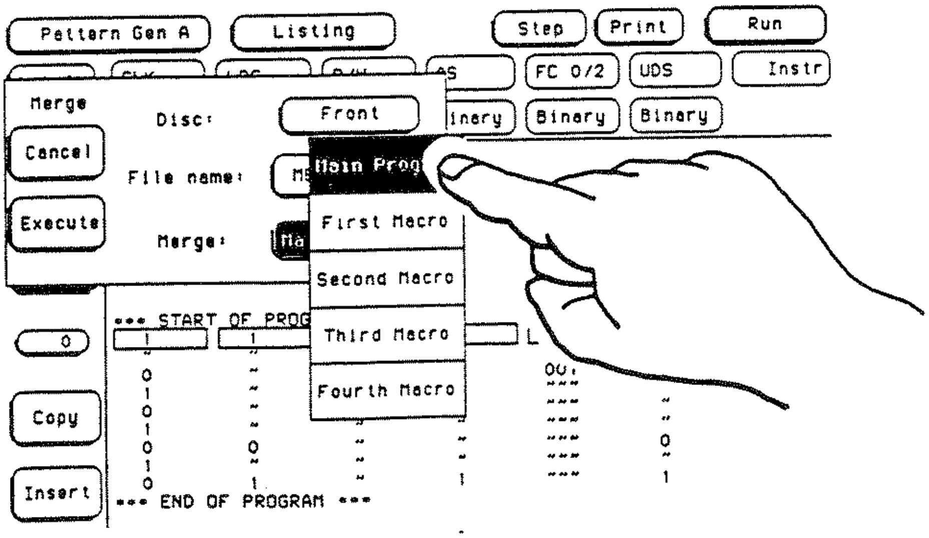
Touch the **File name** field in the pop-up and a keypad appears for entering the disc file name. When you finish entering the file name, touch the **DONE** field in the keypad.

At the bottom of the pop-up is a field labeled **Merge**. This field lets you specify whether you want to merge a main program on disc or one of four macros that may be stored with that program.

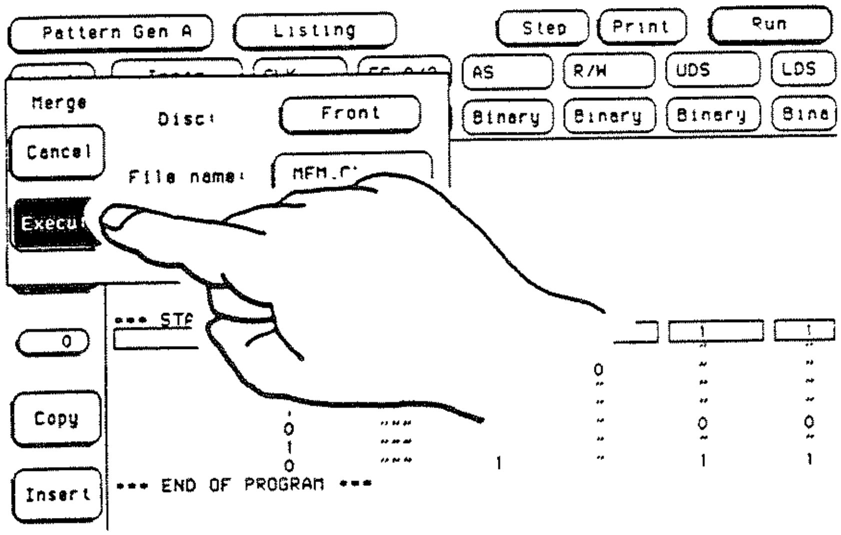


Disc: Front
File name: MEM_CYCLE
Merge: Main Progr

Touch one of the choices to select what you are merging from the disc.



Two fields on the right side of the first pop-up allow you to cancel the merge function or to execute it after you've entered the source of the program, its name, the portion to be merged. If you do not want to merge anything, touch the **Cancel** field. If you touch the **Execute** field, the program or macro will be pulled from the disc and added after the current line of the program or macro being edited.



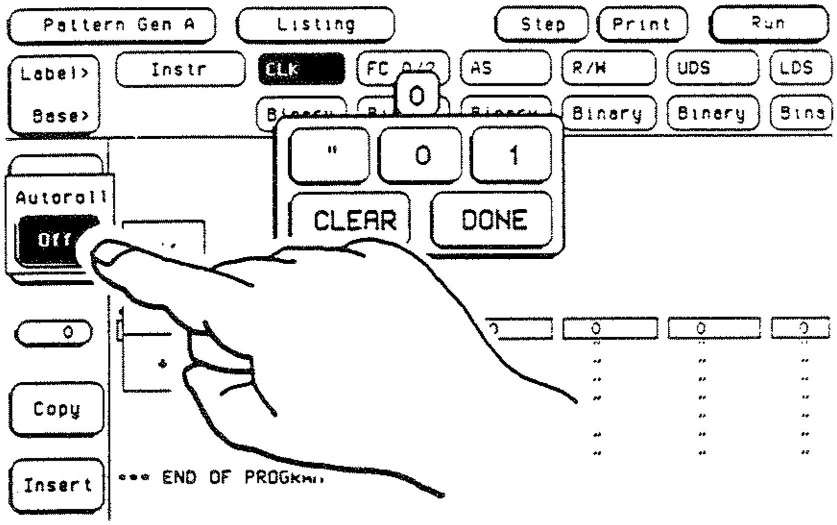
If you do not want to merge anything, touch the **Cancel** field.

Using Autoroll to Enter Data

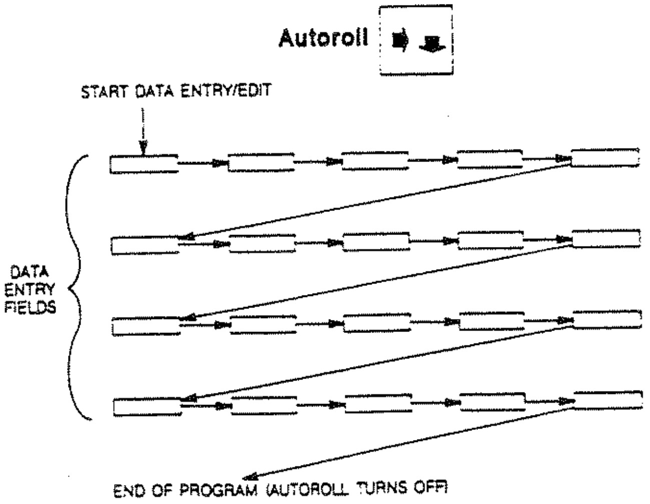
Menu: Listing
Field: Data Entry (15)

When entering pattern generation data, you may go across the screen, filling each data field in the line before going to the next program line. Or, you may want to fill in all the data in a column and then go to the next column. Whatever method you choose, **Autoroll** makes moving from one data entry field to another easier than touching each in succession.

When you touch a data field and the pop-up appears, notice that a field also appears at the left side of the screen labeled **Autoroll**. The default for Autoroll is **Off**. When you touch the **Autoroll** field, another pop-up appears with three choices: **Off**, ← ↓, and ↓.



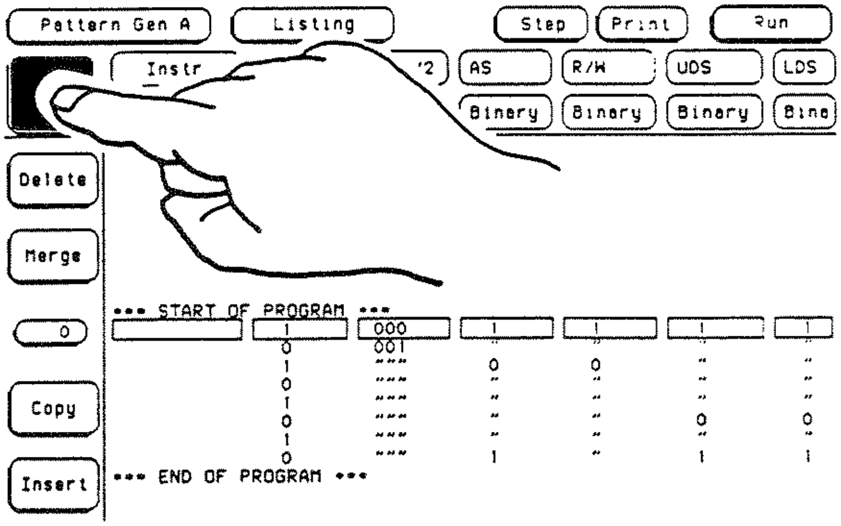
If the ➡ ↓ field is touched, the pop-up will close and **Autoroll** through the fields in the program line from left to right each time you finish entering data (touch the **DONE** field). When you finish entering data into the last field on a line, the pattern generator will automatically jump to the first field in the next line. This continues until the pattern generator reaches the end of the program or until you turn **Autoroll** off.



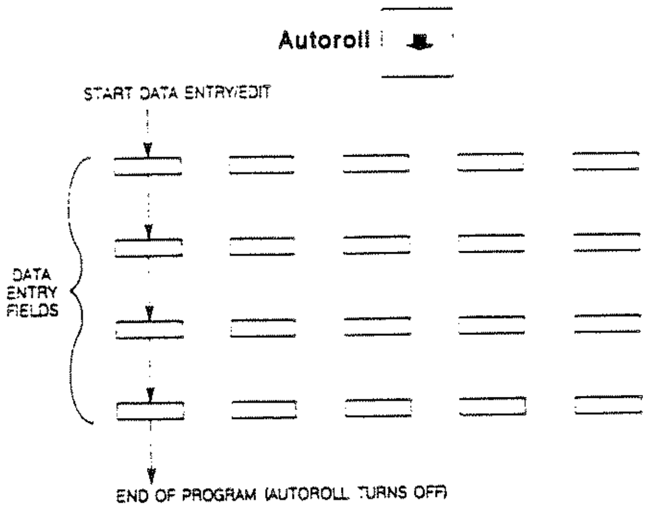
Moving the Screen Left or Right

Menu: Listing
Field: Label> Base> (6)

Since you can have more labels than can be displayed on screen horizontally at one time, the pattern generator has a **Label> Base>** field. This field, located in the upper left fo the screen, allows you to move to those labels that are off screen either to the left or right. Touch the **Label> Base>** field and use the knob to roll the screen in either direction.



To return the screen to the vertical scroll mode, simply touch the program line number field.



If you touch the ↓ field, the pattern generator will move down the column you are in, advancing to the next field under it each time you finish entering data (touch the **DONE** field), as shown below. This continues until the pattern generator encounters the end of the program or you turn **Autoroll** off.

6 The Macro List Menus

Introduction

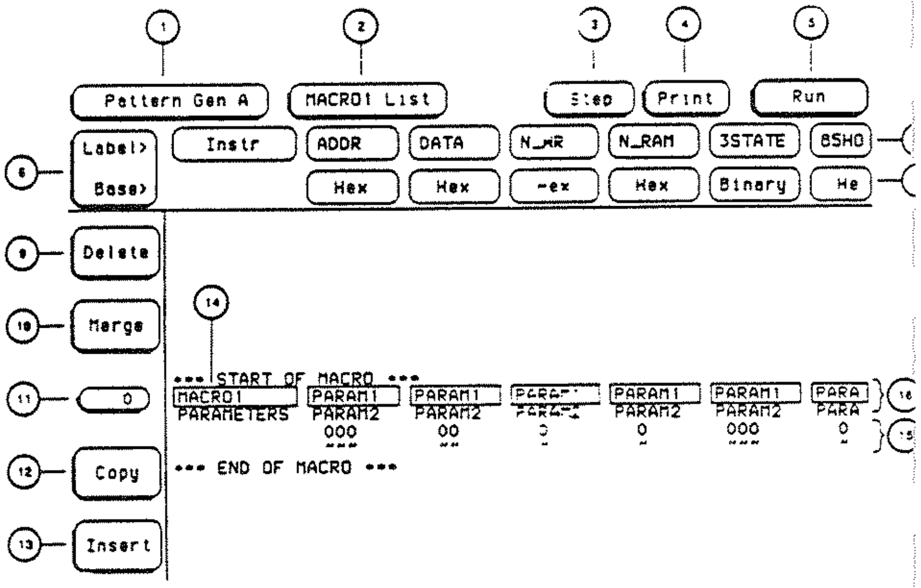
In the HP 16520A/1621A Pattern Generator, similar functions are generally placed together under a single menu. For instance, the **Macro List** menus let you write macros to eliminate entering redundant pattern sequences. The operating values of a pattern generation program are under the **Format** menu, where you set the format of your data. And the **Listing** contains the list of patterns and the sequence in which they are to go out.

This chapter tells you how to write and edit macros for use in the main pattern generator program. Macros are written on one of the four macro list menus. Each of the four menus is the same, so the information in this chapter applies to all.

Most of the functions in the macro list menus are identical to those in the **Listing** menu. This chapter explains only those function that are unique to the macro list menus. If you are uncertain about how to create a pattern generator list or program, refer to Chapter 5 of this manual, "Entering and Editing Output Data."

The pictoral index on the next page gives you a visual map of the **MACRO List** menus. It gives you the name of each field in the menu, along with the page or chapter number where you'll find more information about its function.

MACRO List Menu Field Index



	Description	Manual Page or Chapter
1	Module Field	3-2 to 3-3
2	Menu Field	3-2, 3-4
3	Program Step Run Field	Chapter 7
4	Print Field	Chapter 11
5	Run Field	Chapter 7
6	Column Roll Field*	5-17
7	Label Fields	5-4 to 5-5
8	Numerical Display Base Fields	5-3
9	Line Delete Field	5-8 to 5-9
10	Program Merge Field	5-12 to 5-14
11	Line Copy Field	5-6
12	Line Insert Field	5-10 to 5-11
13	Line Number Field	5-8
14	Instruction Field	Chapter 10
15	Data Entry Fields	5-7, 5-14 to 5-16
16	Parameter Fields	6-6 to 6-7

* Appears only if there are more columns than can be displayed on screen at once.

What is a Macro?

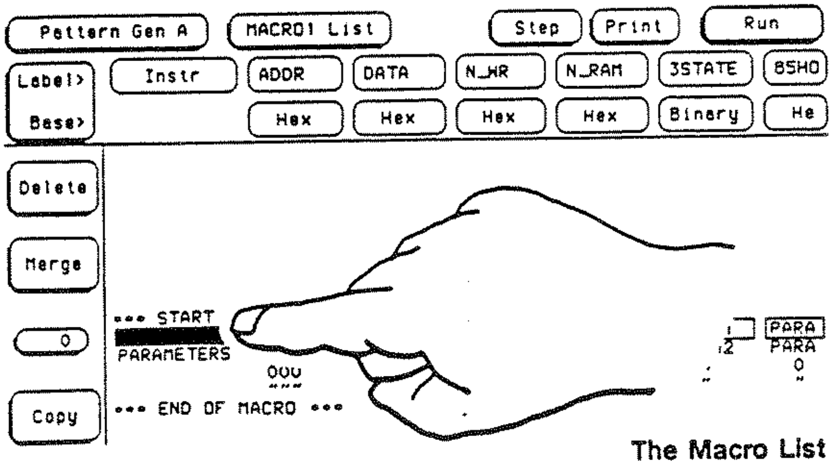
Macros are separate programs that can be called by the main pattern generation program. Often you may have a sequence of patterns that is repeated many times within the pattern generator program, like a handshake sequence. Rather than having to enter this pattern sequence in the main program every time you want a handshake, macros allow you to define the sequence once in one of the four macro lists, give the sequence a name, then call that macro by name in the main program. So, instead of having to enter the pattern sequence each time, you can specify the macro name, and the patterns in the macro will be output, saving both effort and program space.

The pattern generator also allows you to pass parameters to the macros. For example, suppose you are doing a lot of writing to the memory of your system and you'd like to define a macro to take care of actually writing to the RAM. Each memory access differs only in address and data. With the ability to pass parameters, you may define the access routine in a macro and then pass the address and data to the macro. This makes the macros much more general purpose while cutting down on the amount of time it takes to develop pattern generation programs.

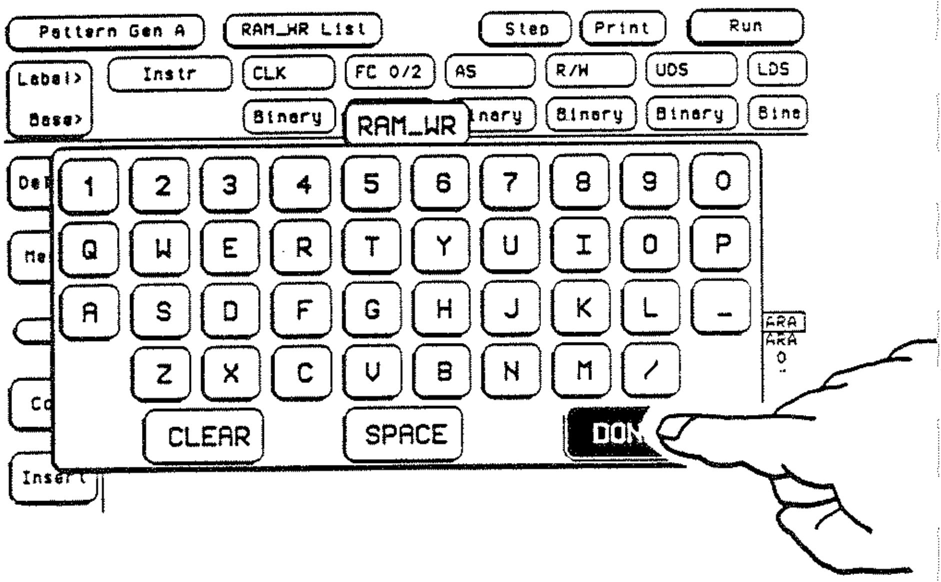
Naming a Macro

Menu: MACRO List
Field: Instruction (14)

Macros can be given any six character name you want. To name a macro, touch the **Instruction** field in line 0 of the macro list.



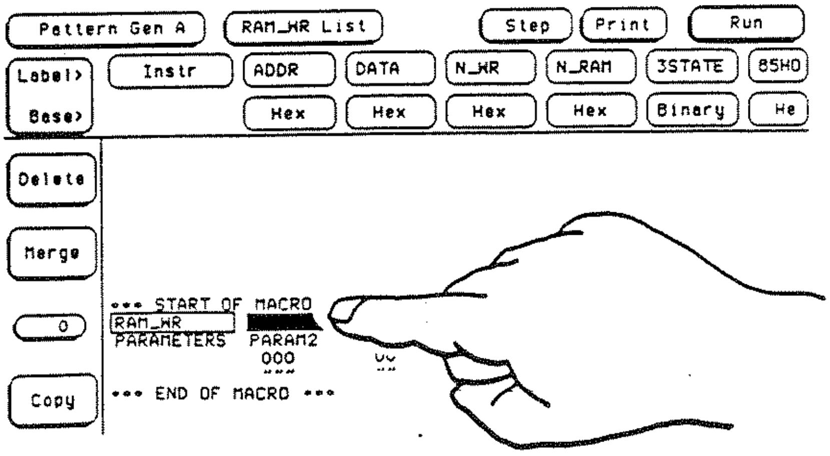
A keypad will pop up, allowing you to enter the macro name. When you finish entering the name, touch the **DONE** key. When the keypad closes, the macro name will appear in the **Instruction** field and the **Menu** field at the top of the screen.



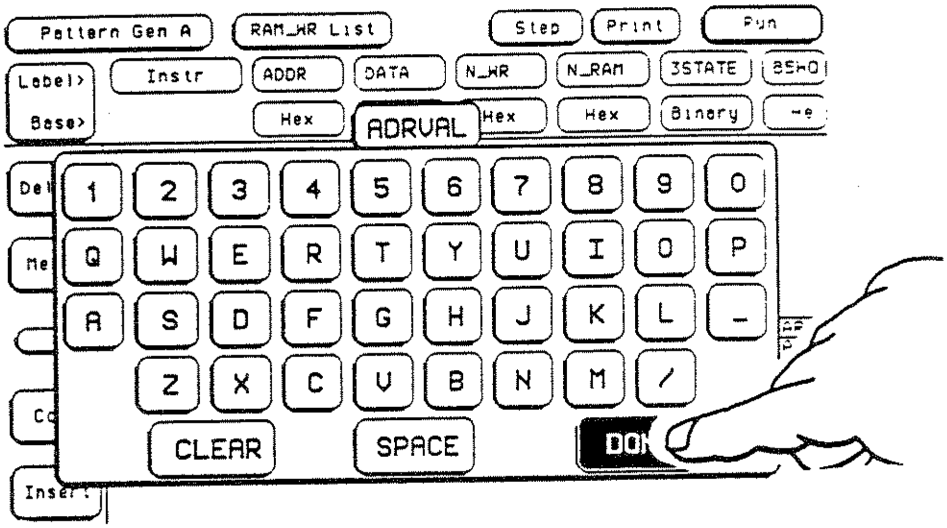
Setting Pass Parameters

Menu: MACRO List
Field: Parameter (16)

The pattern generator allows you to pass parameters between the main program and the macros. There are two parameters available for each label in the list. The parameters fields appear in lines 0 and 1, in what would normally be data entry fields. They are labeled **PARAM1** and **PARAM2.**



The parameter fields differ from data entry fields in that they will not accept numeric input like a data entry field. Instead, when you touch a parameter field a pop-up keypad appears, allowing you to enter a name. The name can be up to six characters, and the parameter can be referred to in the rest of the macro by this name.

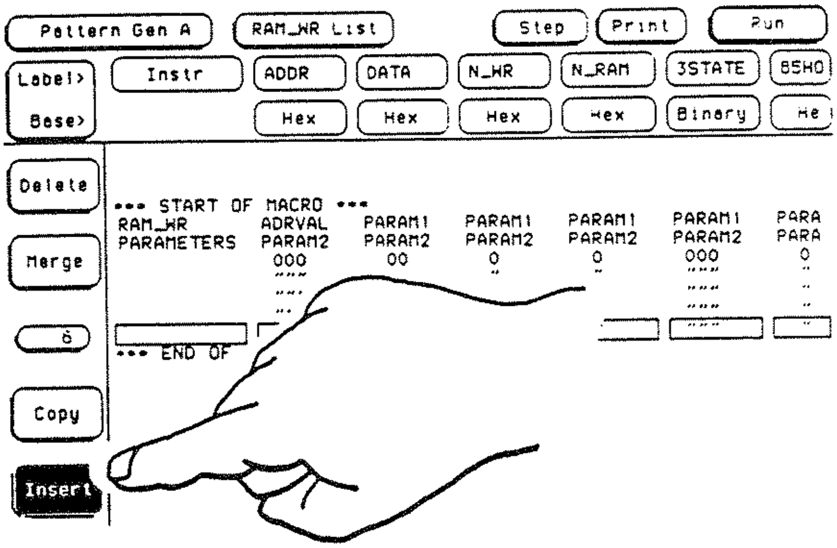


The parameter name can be used in the rest of the macro when referring to the parameter. When naming a parameter, you are naming a variable that is passed into the macro from the main pattern generation program. As an example, suppose you have a label in the main program entitled **ADDR**. You'd like to pass an address value into the macro, so you might want to name one of the parameters under the label **ADDR**, **ADRVAL**, for ADDRESS VALUE. You can then refer to this parameter (variable) by its new name, **ADRVAL**, rather than just **PARAM1**.

Parameters do not have to be renamed. You may use the default names of **PARAM1** and **PARAM2** if you wish.

Editing a Macro

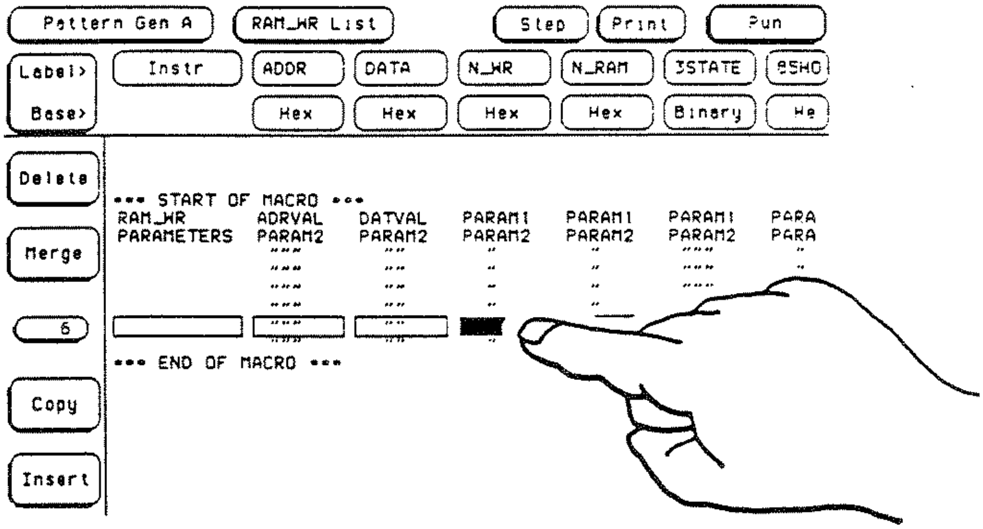
Creating and editing a macro is virtually the same as creating and editing a program in the **Listing** menu. To insert, delete, or copy merge lines in a macro, use the **Insert, Delete, Merge** or **Copy** fields at the left of the screen, just as you would in the main program in the **Listing** menu. The functionality of these fields is the same in the macro or main programs, with one exception. You can not delete, edit, or insert within lines 0 - 3 of the macro.



```
*** START OF MACRO ***
RAM_WR        ADRVAL   PARAM1   PARAM1   PARAM1   PARAM1   PARA
PARAMETERS    PARAM2   PARAM2   PARAM2   PARAM2   PARAM2   PARA
               000       00       0        0       000       0
*** END OF
```

You may also use instructions in a macro, to repeat a line, wait for external input qualifier values, wait for an IMB signal, break, or send out an IMB signal. The instructions are inserted in the **Instruction** field, just as in the **Listing** menu. However, you can not insert instructions into the first two lines of the macro, in the **Parameter** fields.

Entering patterns into the data entry fields is also done in the same manner as in the **Listing** menu. To enter patterns into a field, simply touch the field and use the pop-up keypad.

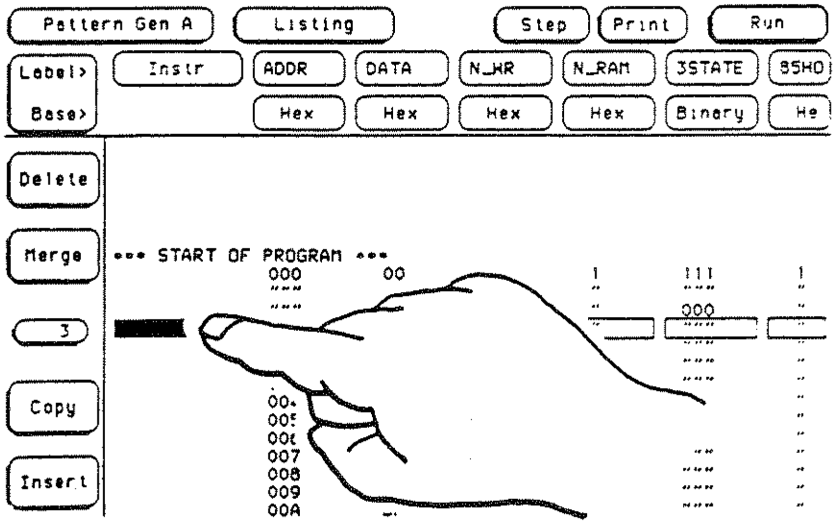


```
*** START OF MACRO ***
RAM_WR        ADRVAL   DATVAL   PARAM1   PARAM1   PARAM1   PARA
PARAMETERS    PARAM2   PARAM2   PARAM2   PARAM2   PARAM2   PARA
*** END OF MACRO ***
```

Calling a Macro from the Main Program Listing

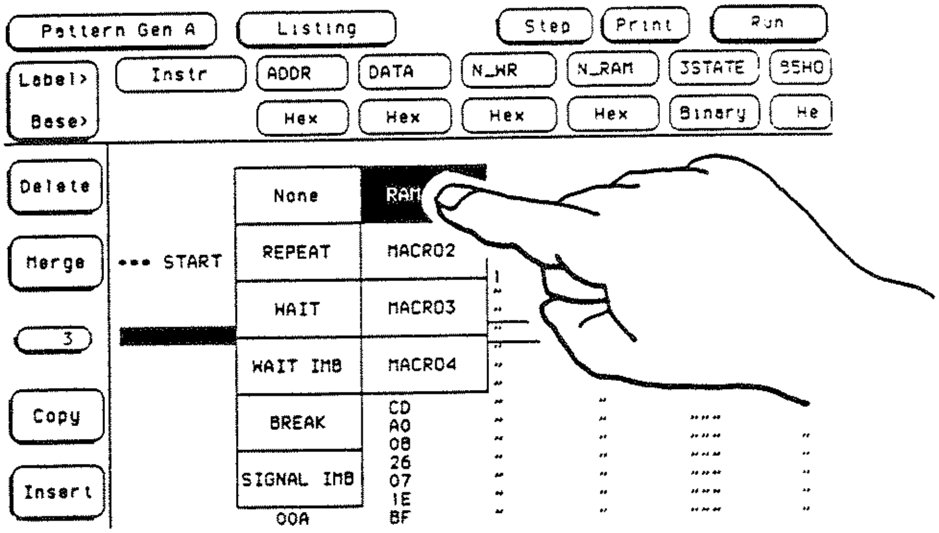
Menu: Listing
Field: Instruction (14)

To call a macro from the main program, touch the **Instruction** field of the line in which you want to invoke the macro.



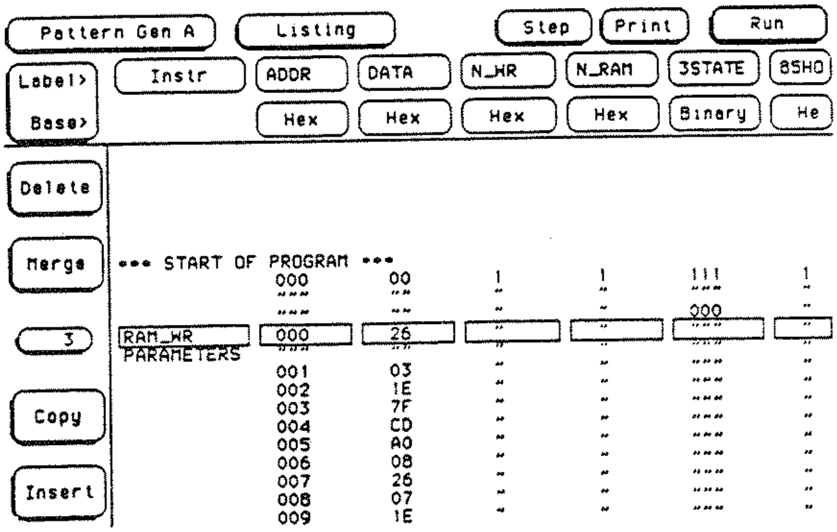
```
*** START OF PROGRAM ***
              000      00                1   111   1
                                             000
              004
              005
              006
              007
              008
              009
              00A
```

When the pop-up appears, select one of the **MACROX** fields, depending on the list where your macro is. If you have named your macro, that name will appear in the pop-up.



```
*** START
                  CD
                  A0
                  08
                  26
                  07
                  1E
         00A      BF
```

Whenever you select a macro in one of the **Instruction** fields, another line will appear below it. The word **Parameters** will appear in the **Instruction** field of this second line.

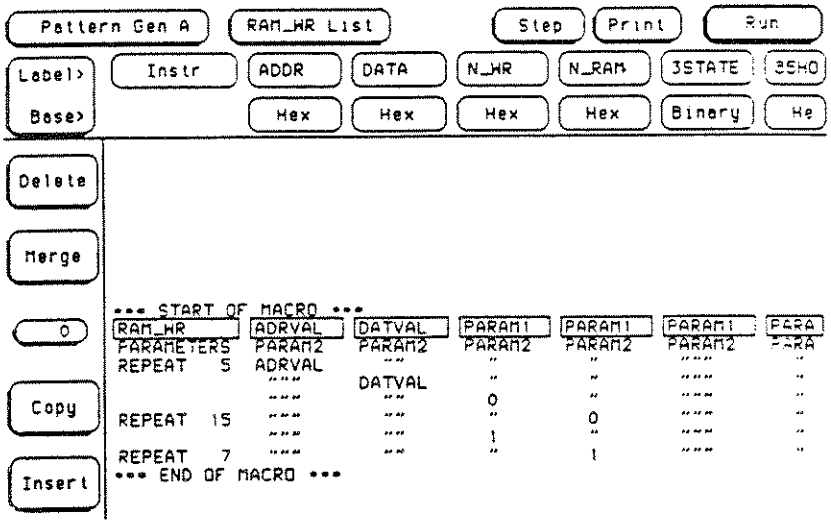


```
*** START OF PROGRAM ***
             000     00     1     1    111    1
RAM_WR       000     26
PARAMETERS
             001     03
             002     1E
             003     7F
             004     CD
             005     A0
             006     08
             007     26
             008     07
             009     1E
```

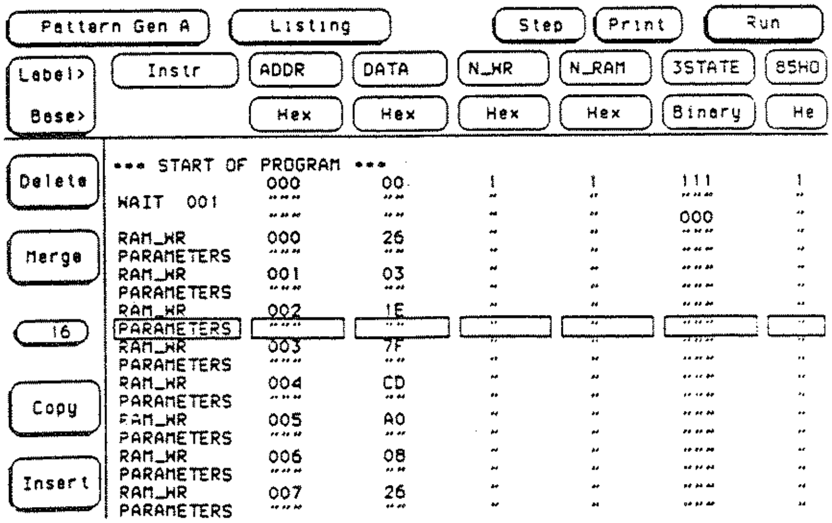
The data entry fields of these two lines (the one in which the macro is called and the parameter line below it) can now be used to enter the values that will be passed to the macro. There are two values that can be passed to the macro for each label in the listing.

An Example Macro

The following program segments are examples of a simple macro for writing to a RAM and the program that calls it. Notice that data and address values are passed into the macro from the main program. The macro has been renamed **RAM_WR**. Also notice that there are two parameters in the macro, under the labels **ADDR** and **DATA** that they have been renamed to ADRVAL and DATVAL, for ADDRESS VALUE and DATA VALUE, respectively. The second parameter for each label, **PARAM2** is not used, and is allowed to assume the same value as **PARAM1, ADRVAL,** or **DATVAL.**



```
*** START OF MACRO ***
RAM_WR        ADRVAL    DATVAL    PARAM1    PARAM1    PARAM1    PARA
PARAMETERS    PARAM2    PARAM2    PARAM2    PARAM2    PARAM2    PARA
REPEAT   5    ADRVAL    ""        "         "         """       "
              """       DATVAL    "         "         """       "
              """       ""        0         "         """       "
REPEAT  15    """       ""        "         0         """       "
              """       ""        1         "         """       "
REPEAT   7    """       ""        "         1         """       "
*** END OF MACRO ***
```



```
*** START OF PROGRAM ***
              000       00        1         1         111       1
WAIT  001     """       ""        "         "         """       "
              """       ""        "         "         000       "
RAM_WR        000       26        "         "         """       "
PARAMETERS    """       ""        "         "         """       "
RAM_WR        001       03        "         "         """       "
PARAMETERS    """       ""        "         "         """       "
RAM_WR        002       1E        "         "         """       "
PARAMETERS    """       ""        "         "         """       "
RAM_WR        003       7F        "         "         """       "
PARAMETERS    """       ""        "         "         """       "
RAM_WR        004       CD        "         "         """       "
PARAMETERS    """       ""        "         "         """       "
RAM_WR        005       A0        "         "         """       "
PARAMETERS    """       ""        "         "         """       "
RAM_WR        006       08        "         "         """       "
PARAMETERS    """       ""        "         "         """       "
RAM_WR        007       26        "         "         """       "
PARAMETERS    """       ""        "         "         """       "
```

7 Running and Stopping the Pattern Generator

Introduction

This chapter will show you how to run and stop the pattern generator and what the various run modes mean. It also discusses what happens to the output channels when a program is finished or stopped.

The pattern generator can be run from any menu. The procedure for running is basically the same for all menus, so the **Listing** menu is used as an example.

Run Modes

There are two basic run modes in the pattern generator, single and repetitive. In addition, the pattern generator can run independent of other modules in the mainframe or in conjunction with other modules through the IMB (Intermodule Bus). In total there are four possible run modes.

Independent Run Single- pattern generator runs once and stops. Running is independent of other modules in the mainframe. Data, strobe and clock outputs are held at the state defined by the last line of the program.

Independent Run Repetitive- pattern generator runs continuously until stopped. Time between the last program line and the first is the same as all other program steps. Running is independent of other modules in the mainframe.

Group Run Single- pattern generator runs once and stops. Running is in conjunction with other modules through the IMB. Data, strobe and clock outputs are held at the state defined by the last line of the program.

Group Run Repetitive- pattern generator runs continuously with other modules in the mainframe until stcpped. At the end of each run it halts and waits for an indication through the IMB that the other modules have finished their acquisitions. The pattern generator then starts another run at the beginning of the program.

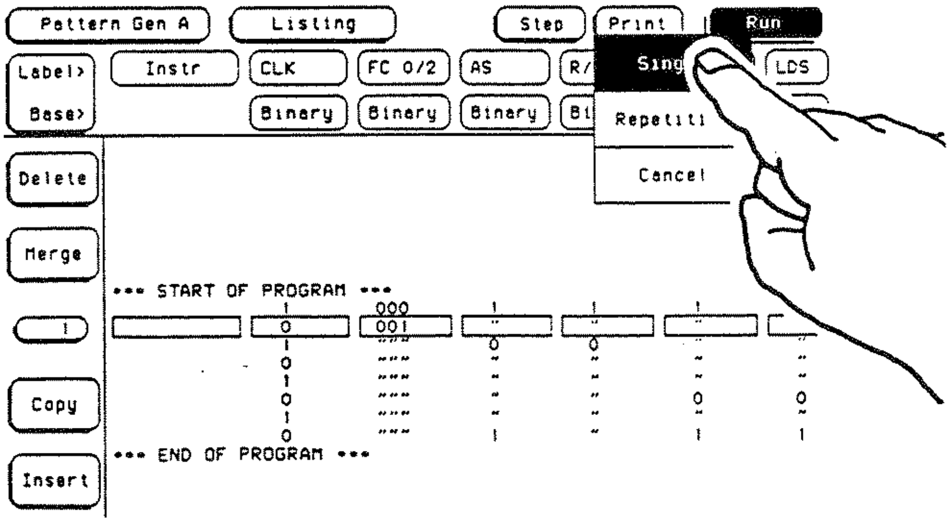
An intermodule menu lets you tell the mainframe which module is to send an arming signal and which modules are to act upon it. If you need more information on how to use the intermodule capabilities, see the chapter in the *HP 16500A Reference Manual* entitled "Intermodule Measurements."

If the field in the upper right of your screen says **Run**, the pattern generator is set to run independently of all other modules. If, however, the field says **Group Run**, the pattern generator is tied to another module or modules through the intermodule menu.

Running the Pattern Generator Once

Menu: Any
Field: Run

To run the pattern generator once, touch the **Run** field. When the pop-up appears, move your finger to the **Single** field in the pop-up without lifting your finger from the screen. When the **Single** field turns white, lift your finger from the screen. The pattern generator will run once and stop.



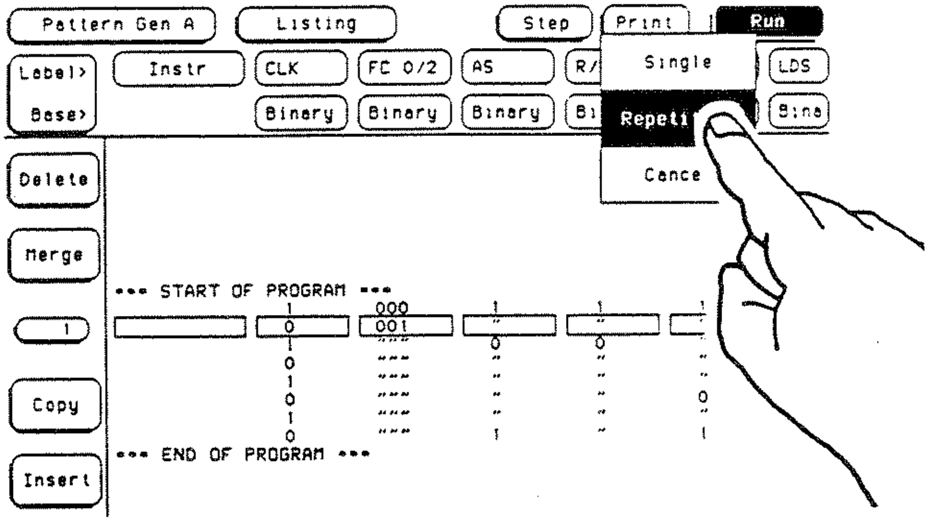
The run mode (**Single** or **Repetitive**) will stay in the last one selected until you change it. That is, if you choose **Single** as the run mode, it will stay in **Single** each time you touch **Run** until you change it to **Repetitive**.

At the end of the run, the pattern generator data, clock and strobe outputs will remain in the state defined by the last line of the program.

Running the Pattern Generator Repetitively

Menu: Any
Field: Run

To run the pattern generator repetitively, touch the **Run** field. When the pop-up appears, move your finger to the **Repetitive** field in the pop-up without lifting your finger from the screen. When the **Repetitive** field turns white, lift your finger from the screen. Each time the pattern generator reaches the end of the program, it starts at the beginning again. The time between the last line of the program and the first is the same as if the lines were contiguous.



The run mode (**Single** or **Repetitive**) will stay in the last one selected until you change it. That is, if you choose **Repetitive** as the run mode, it will stay in **Repetitive** each time you touch **Run** until you change it to **Single**.

Single-Stepping the Pattern Generator

Menu: Listing, Macro List
Field: Step (3)

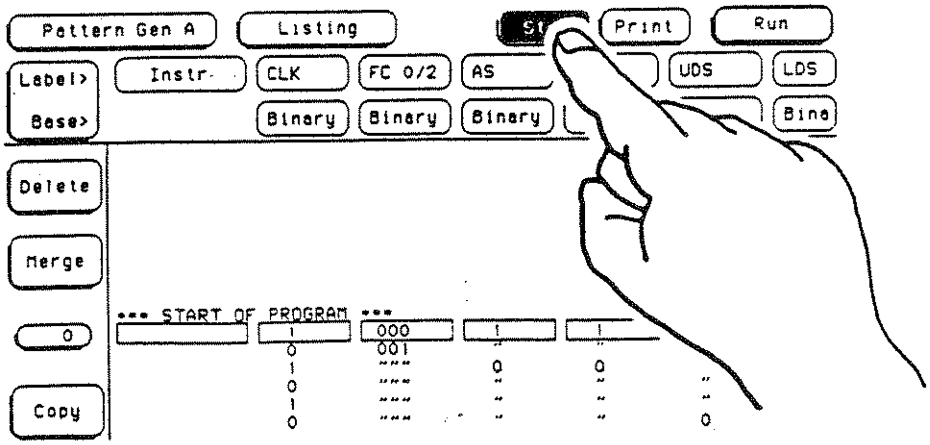
It is often necessary to have control over when the pattern generator steps to its next state. For example, if you are using the pattern generator to stimulate a prototype while checking the response with a logic analyzer or scope, it is particularly helpful to be able to halt the pattern generator when an error is found with the target system. It is also helpful to be able to continue the pattern generation program from that point, rather than having to run the whole program over again.

The single-step function in the HP 16520A/16521A allows you to do just that. You can insert a **BREAK** instruction into your program at any point, and single step through one or more program lines at a time from there. The pattern generator can then return to its single-run mode. For more information on how to use the **BREAK** instruction, see chapter 10 of this manual, "Using Instructions."

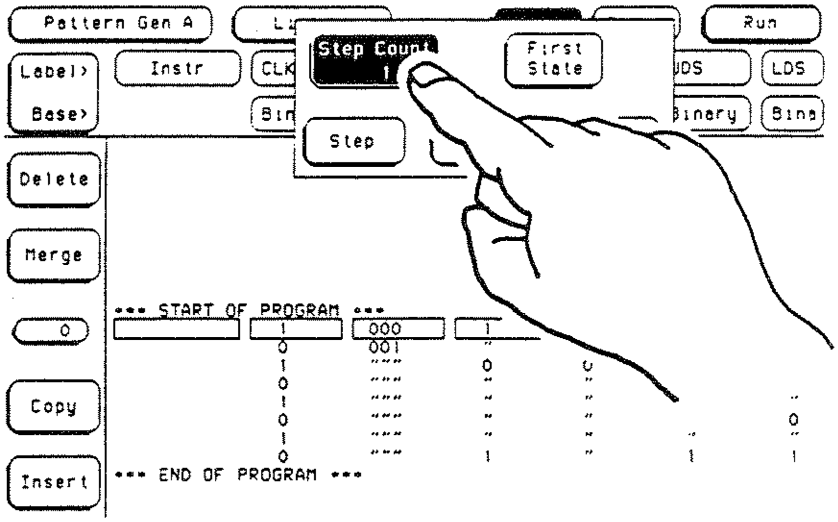
Note

> *The single-step mode is only usable in* **Independent Run Single** *and* **Group Run**. *Both* **Independent Run Repetitive** *and* **Group Run Repetitive** *do not halt at a* **BREAK** instruction.

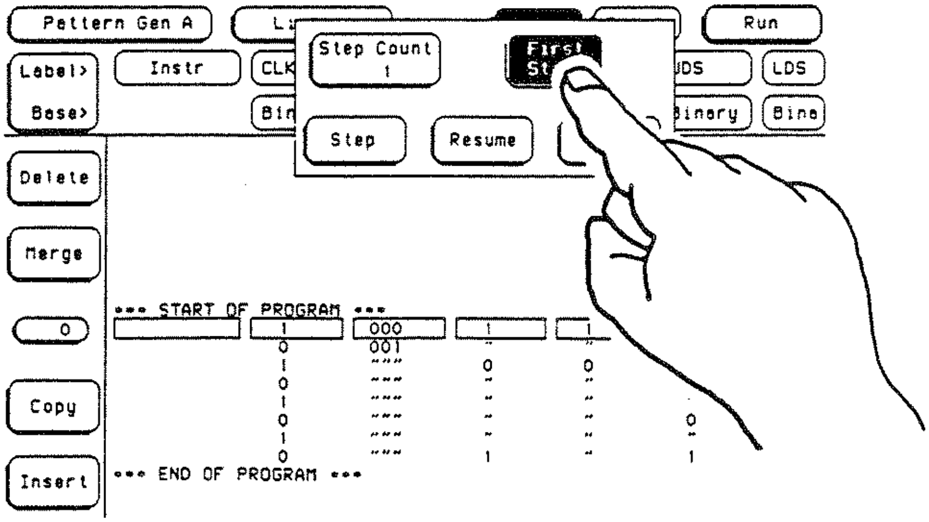
A program can be single stepped from the **Listing** or any of the **Macro List** menus with the **Step** field at the top of the screen. Touch **Step** and a pop-up appears.



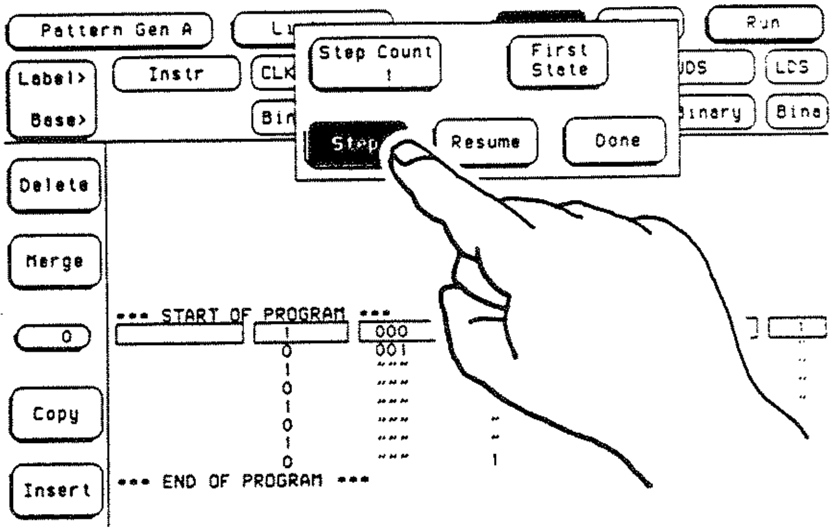
There are five fields in the pop-up. The first, **Step Count**, lets you set the number of states or program lines that will be stepped through each you touch the **Step** field. Use the front-panel knob to change the step count or touch the **Step Count** field and a keypad will pop up from which you can enter a new number. The default is one state per step.



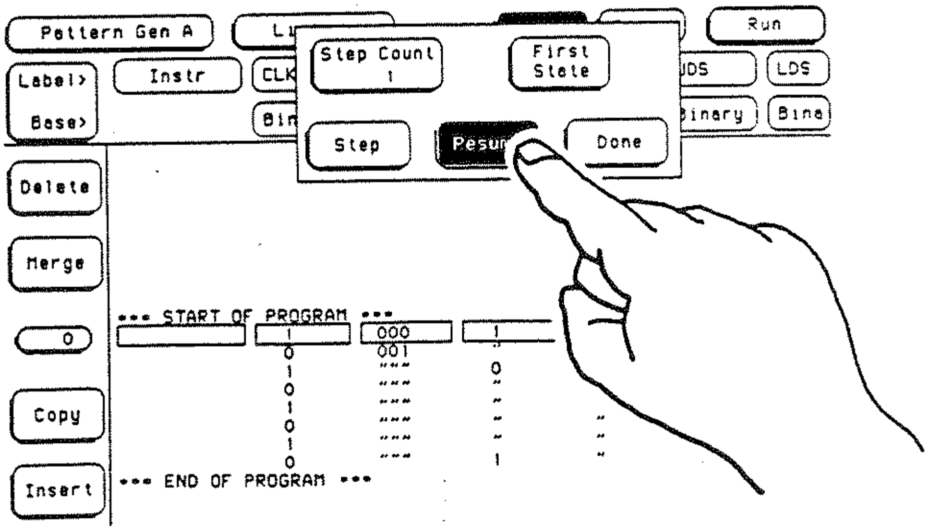
Touch the **First State** field and the pattern generator will jump to line 0 in the program. When the program jumps to line 0, you may single step the program from there, even if there is no **BREAK** instruction at that line.



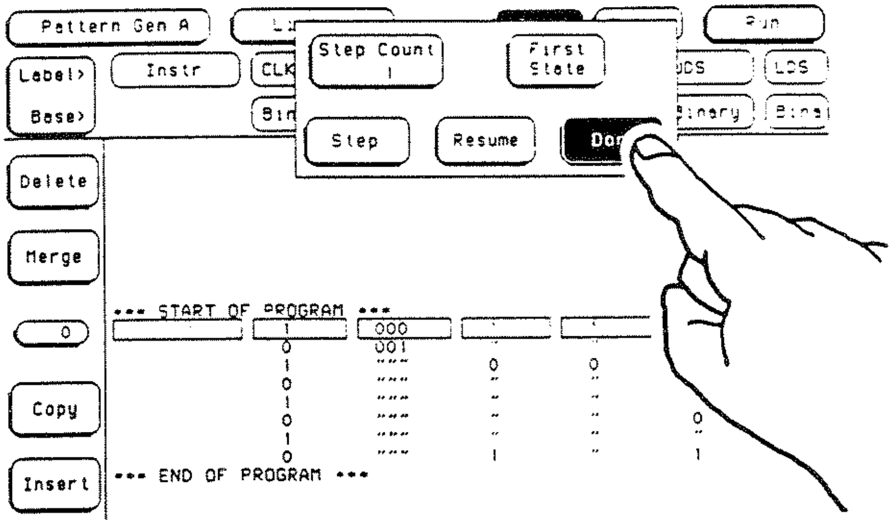
The **Step** field, found at the bottom of the pop-up, controls when the pattern generator steps to the next program line. Each time you touch **Step**, the pattern generator goes to the next line and waits. The output data, strobes, and clock are held at the state defined by that line until you advance to the next line.



The **Resume** field tells the pattern generator to stop single stepping and return to normal operation. This is the only way to discontinue the single-step mode.



When you are finished with the single-step run, touch the **Done** field at the lower right of the pop-up.

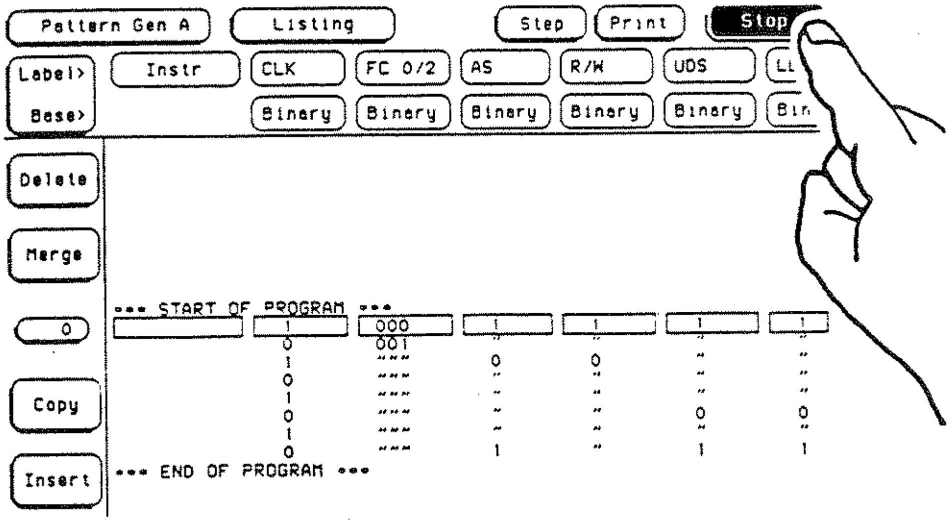


```
*** START OF PROGRAM ***
            1      000
            0      001
            1      """     0     0
            0      """     "     "
            1      """     "     "
            0      """     "     "     0
            1      """     "     "     "
            0      """     1     "     1
*** END OF PROGRAM ***
```

Stopping the Pattern Generator

Menu: Any
Field: Run

When the pattern generator is running repetitively, the **Run** field is replaced by **Stop**. To stop a program that is running repetitively, touch the **Stop** field.



```
*** START OF PROGRAM ***
            1      000     1     1     1     1
            0      001     "     "     "     "
            1      """     0     0     "     "
            0      """     "     "     "     "
            1      """     "     "     "     "
            0      """     "     "     0     0
            1      """     "     "     "     "
            0      """     1     "     1     1
*** END OF PROGRAM ***
```

The **Stop** field also comes up during a single run. but you may not see it since the program is usually finished by the time the **Stop** field is displayed. However. if your program is several hundred or thousand lines long, or if the output data rate is low, the **Stop** field will show for several seconds.

8 Creating A Symbol Table

What is a Symbol?

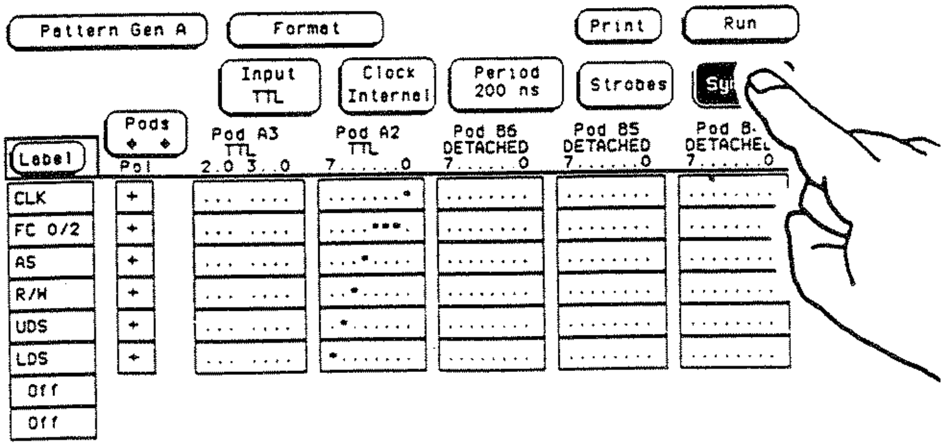
Because long strings of binary patterns are difficult for the human mind to recognize, we often give these patterns functional or symbolic names, to make them easier to remember. A good example of this is microprocessor assembly language. Rather than have to deal with patterns like 0011 0110, we can give the pattern a name like Jump or Compare. By looking at a list of these symbolic names in sequence, we can decipher what a state machine or processor is doing. It would be much more difficult to look at a list of binary codes and get the same information. Appropriately enough, we call these symbolic names Symbols.

The HP 16520A/16521A pattern generator allows you to create a table of such symbols. You can enter the symbol names into the program without having to remember the binary or hex code for each. The pattern generator can then display these names in the program listing for ease of reading.

Getting to the Symbol Table

Menu: Format
Field: Symbol (9)

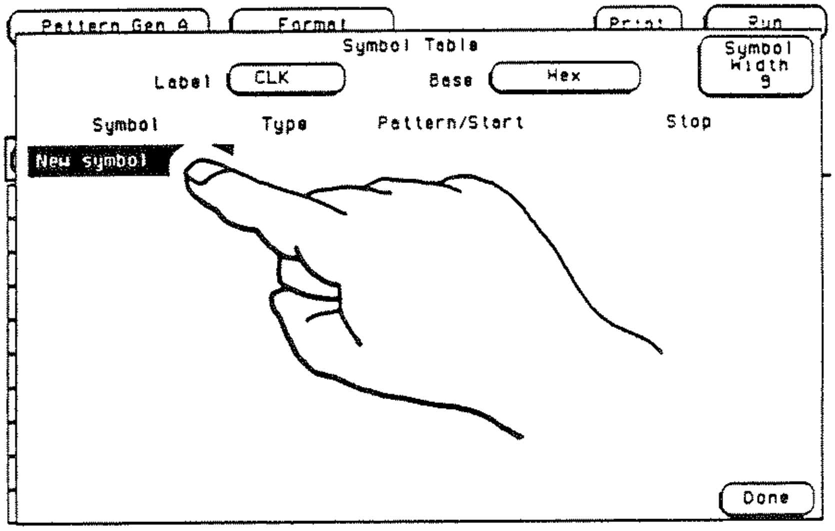
To get to the symbol table, go to the **Format** menu. Touch the field labeled **Symbols** in the upper right of the screen.



Entering Symbol Names

Menu: Format
Field: Symbol (9)

To enter a symbol name, touch the label field at the left that says **New Symbol**.



Label CLK

Base Hex

Note

> *If you have previously created symbols for this label, those symbols will be displayed. The field* **New Symbol** *will not appear. See the section in this chapter entitled* "Adding, Modifying or Deleting a Symbol."

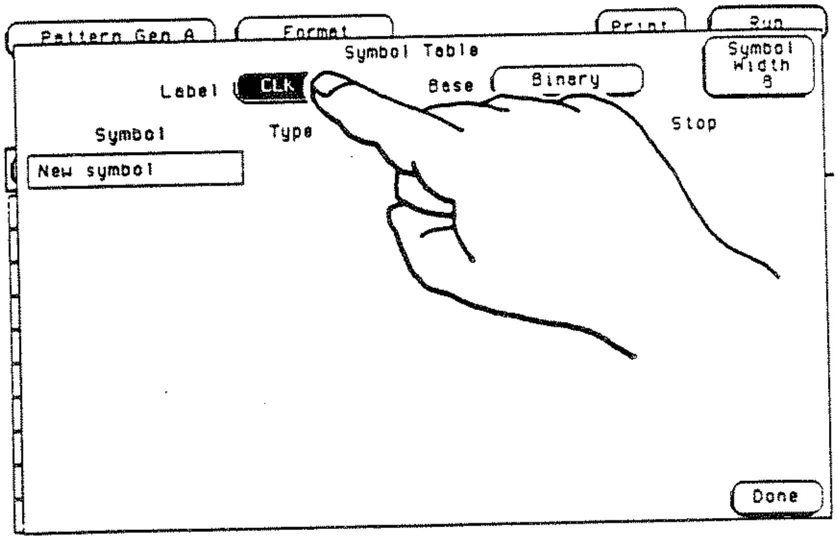
A pop-up keypad will appear to allow you to enter the symbol name. If you make a mistake and need to backspace, use the front-panel knob. When you've finished entering the name, touch the **DONE** key.

Entering a Pattern Symbol

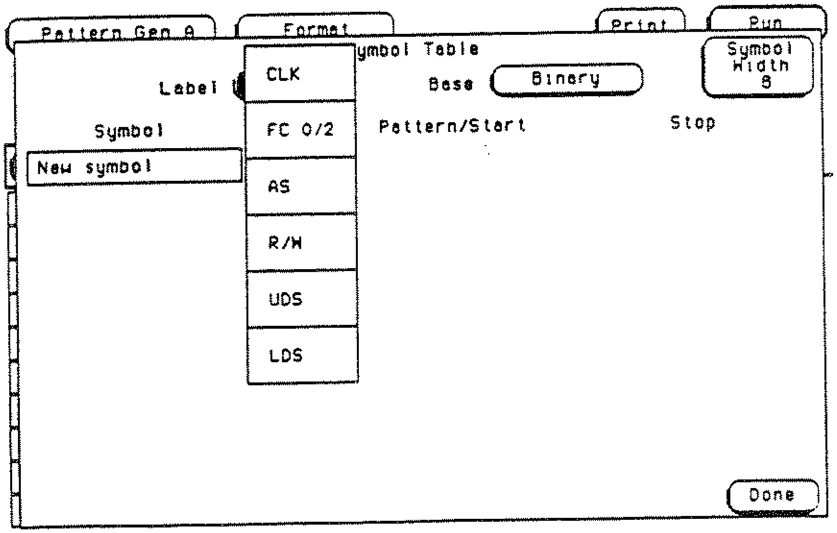
Menu: Format
Field: Symbol (9)

The process of entering symbol data has two parts: specifying the channels on which to put the symbol patterns and entering the symbol pattern or range.

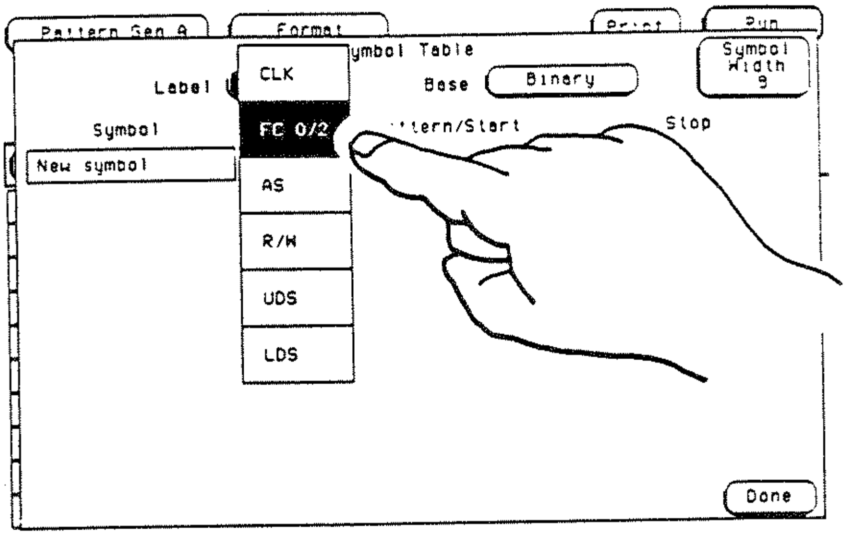
Touch the field that says **Label**.



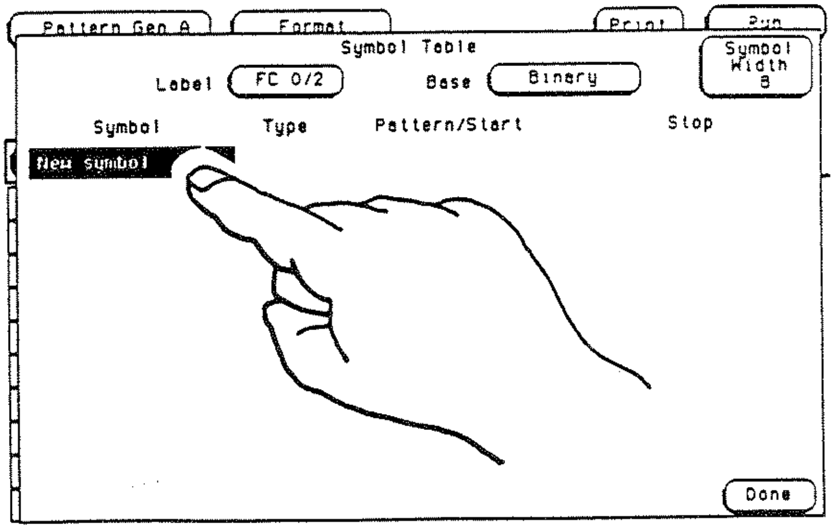
A pop-up appears showing all the labels defined in the **Format** menu. Each label has data channels assigned to it, so when you enter a symbol under a label, the pattern generator knows on which channels to output the symbol data. In this example, there are six label names, **CLK, FC 0/2, AS, R/W, UDS,** and **LDS**.



Suppose, for example, you're going to define a symbol named **USER_DATA** that should go out on the **FC 0/2** channels. Touch the label **FC 0/2** from the pop-up. This tells the pattern generator that you want to build a symbol table for label **FC 0/2**.

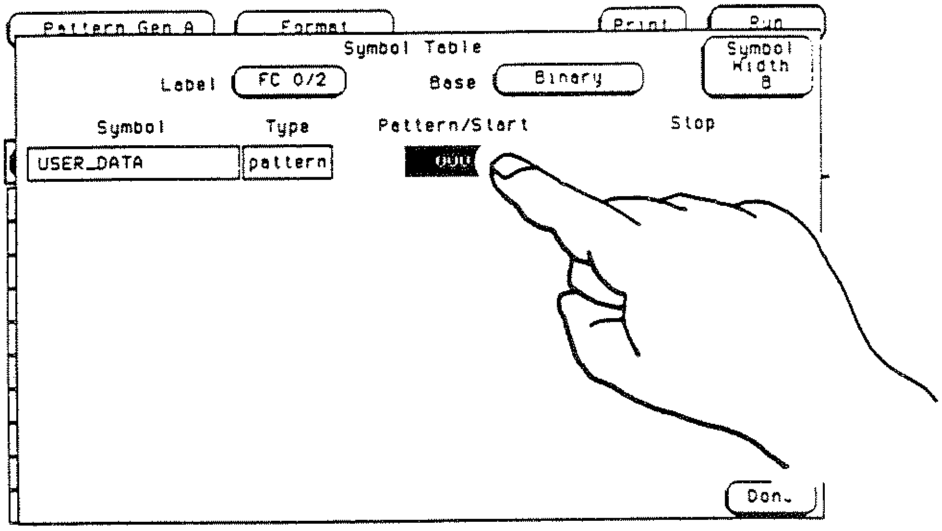


Touch the field labeled **New Symbol** and enter the name **USER_DATA** using the pop-up keypad.



Now that you've told the pattern generator where to put the symbol pattern, you'll need to tell it what the **USER_DATA** pattern should be.

Touch the field labeled **Pattern/Start**. A keypad will pop up, allowing you to enter the pattern. After the pattern is entered, touch the **DONE** key.



Label FC 0/2   Base Binary

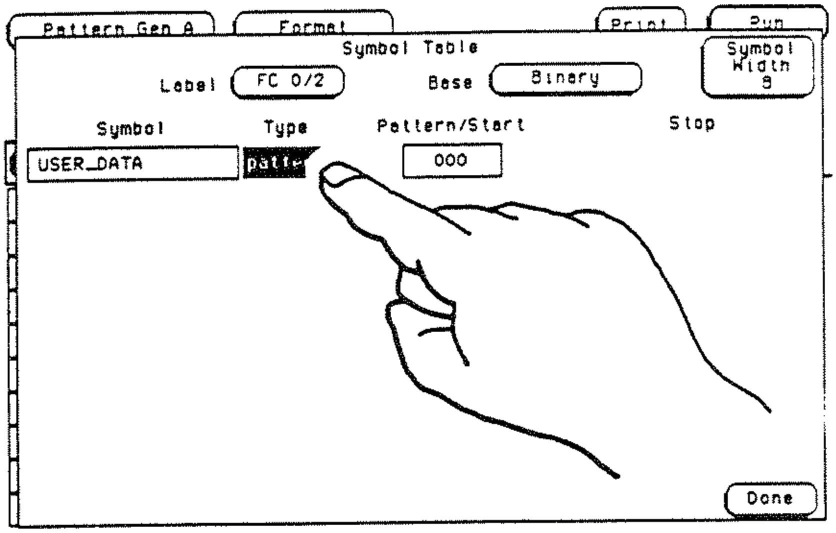
Symbol	Type	Pattern/Start	Stop
USER_DATA	pattern	000	

Setting a Value Range

Menu: Format
Field: Symbol (9)

In certain cases it may be useful to define a symbol as a value plus some offset. The range term allows you to specify a value range for the symbol, and an offset from the start of the range. The symbol pattern then becomes the start address of the range added to the offset.

Touch the **Type** field and it will toggle from **pattern** to **range**.

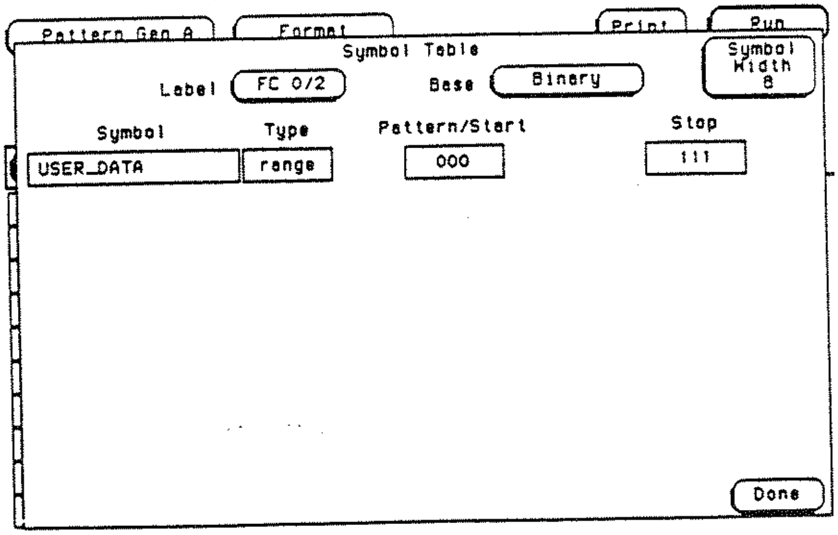


Symbol Table

Label FC 0/2 Base Binary Symbol Width 8

Symbol	Type	Pattern/Start	Stop
USER_DATA	patte	000	

A second field will appear in the **Stop** column to the right. The **Pattern/Start** column allows you to enter the start of the range while the **Stop** column lets you enter the end of the range.



Symbol Table

Label FC 0/2 Base Binary Symbol Width 8

Symbol	Type	Pattern/Start	Stop
USER_DATA	range	000	111

To specify an offset, perform the following steps:

1. Go to the **Listing** menu.

2. Touch the numeric base field below the label you want symbols

displayed on.

3. When the pop-up appears, touch the field that says **Symbol**.

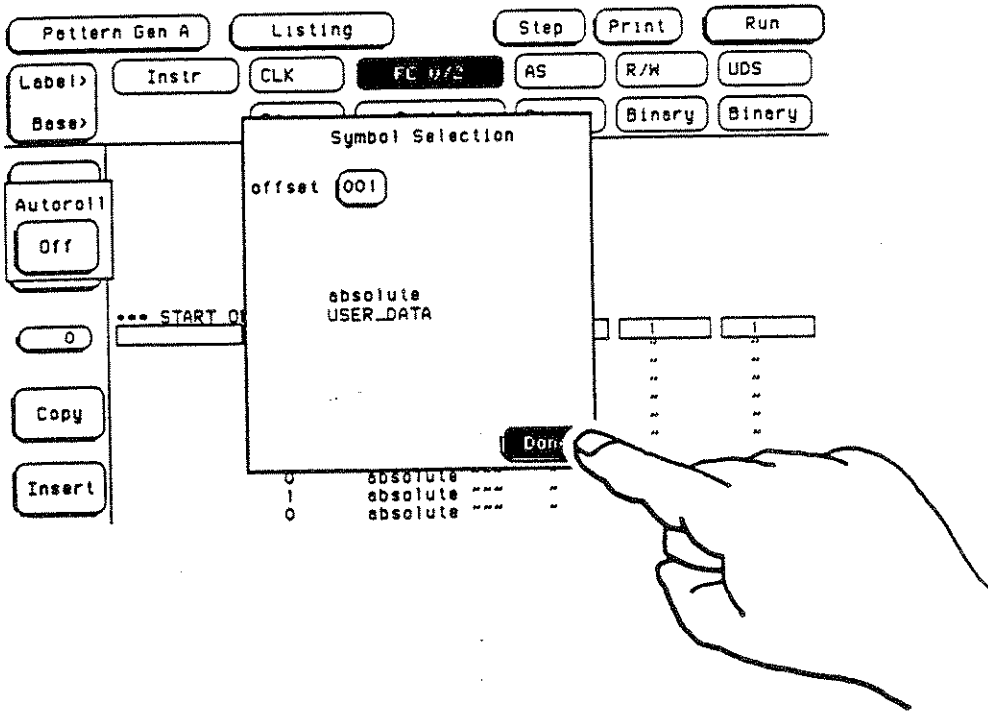
4. Touch the data-entry field below the label you've set to display symbols.

5. A pop-up labeled **Symbol Selection** pops up with all the symbols for that label listed. The list also includes an entry labeled **absolute**. Using the front-panel knob, roll until **absolute** appears in the highlighted bar.

6. A field labeled **offset** appears at the top of the pop-up. Touch the numeric field and enter the offset from the keypad.

7. Touch the **Done** field in the keypad and symbol selection pop-ups to complete the offset selection.

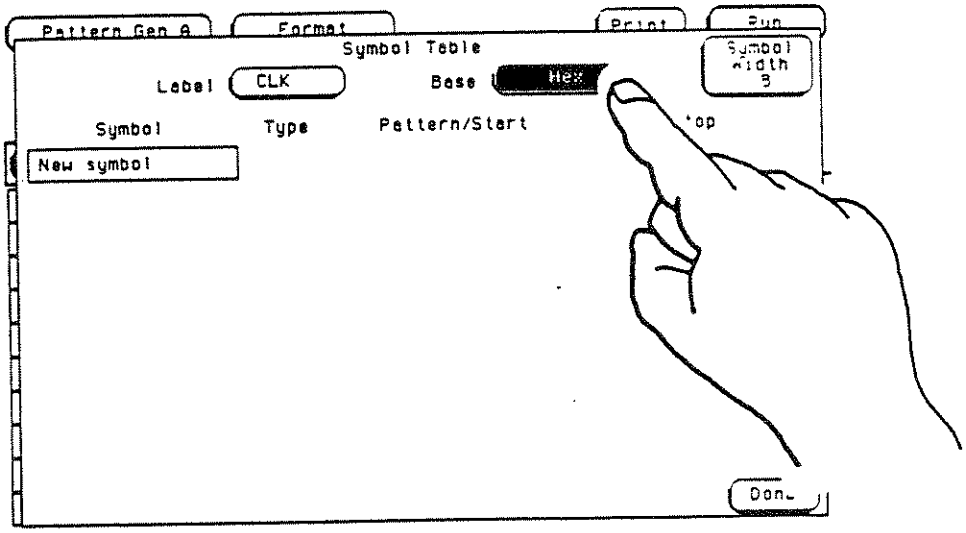
The value range of the symbol cannot exceed the number of bits assigned to the label. For example, there are three bits assigned for the label **FC 0/2**, so the range for a symbol would be 8 (0 through 7).



Setting the Numerical Base

Menu: Format
Field: Symbol (9)

The field labeled **Base** allows you to select the number format for entering patterns. The default is **Hex**. Touch the field and a pop-up appears with fields that allow you to select **Hex, Binary, Octal, Decimal,** and **ASCII.**

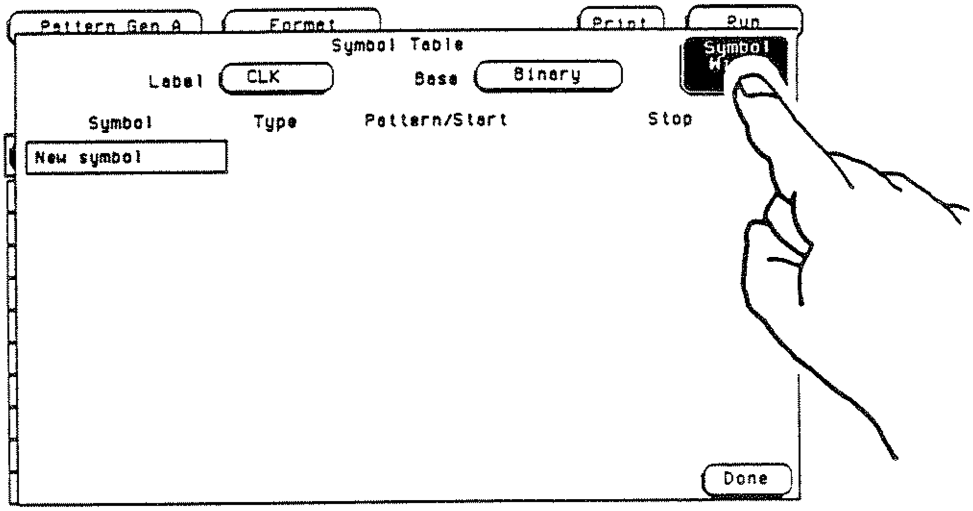


Touch the number base you want and the pop-up will close.

Setting the Symbol Width

Menu: Format
Field: Symbol (9)

In the upper right of the **Symbol Table** screen is a field labeled **Symbol Width**. This field allows you to specify how many characters of the defined symbol name are to appear in the **Listing** menu. The default value is eight. Keeping the symbol width as small as possible will keep the horizontal width of the **Listing** menu smaller. This is important if you have a large number of labels. If you have several symbol names that all have the same first four letters, and you set the symbol width to 4, all the symbols will appear the same in the **Listing** menu. In such a case, you'd need to set the symbol width to at least five.



Symbol Table

Label CLK

Base Binary

Symbol Type Pattern/Start Stop

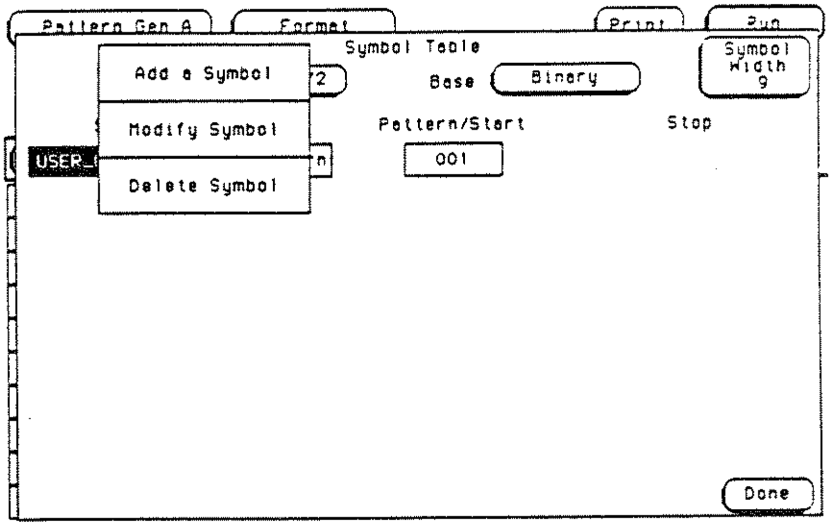
New symbol

Adding, Modifying, or Deleting a Symbol

Menu: Format
Field: Symbol (9)

When a symbol table has no entries, the **Symbol** field will say **New Symbol**. However, after the first symbol name is entered, the **New Symbol** indicator disappears and is replaced by the first symbol name.

To add more symbols, touch the **Symbol** field. A pop-up will appear with three choices: **Add a Symbol, Modify Symbol,** and **Delete Symbol.**



To add a symbol, touch the field labeled **Add a Symbol.** A new symbol line will be added directly after the symbol field you touched.

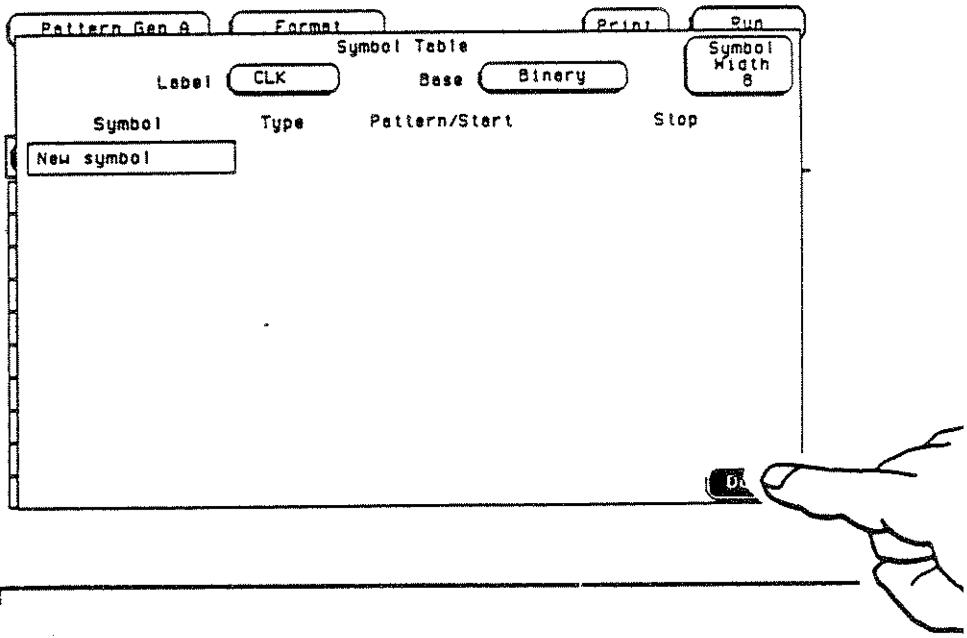
To modify an existing symbol, touch the name of the symbol you want to change. When the pop-up appears, touch the field labeled **Modify Symbol.** A keypad will appear to allow you to modify the current name. The front-panel knob can be used to move the cursor to any part of the old name for editing.

To delete any symbol in the table, touch the symbol you want to get rid of and a pop-up will appear. Touch the **Delete Symbol** field in the pop-up and the chosen symbol will be deleted from the table.

Getting Out of the Symbol Table

Menu: Format
Field: Symbol (9)

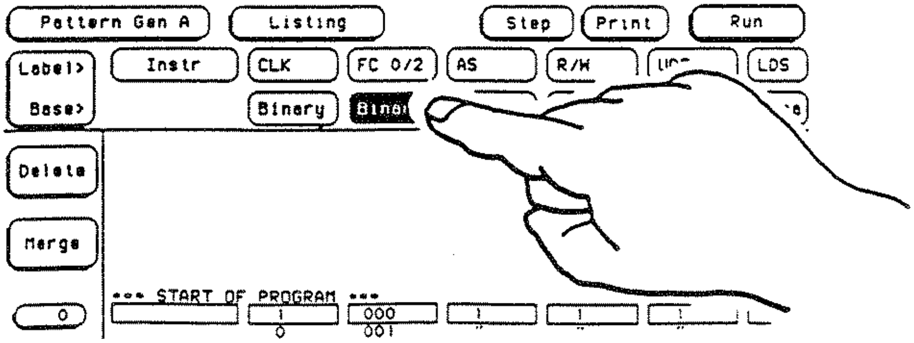
To leave the **Symbol Table** screen, touch the field at the lower right of the screen labeled **Done**.



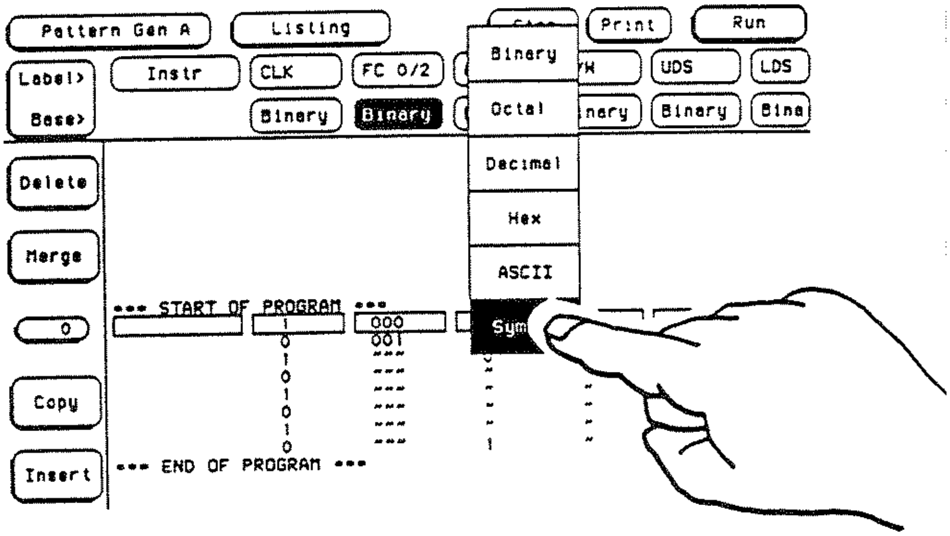
Displaying Symbols in a Program

Menu: Listing
Field: Numerical Base (8)

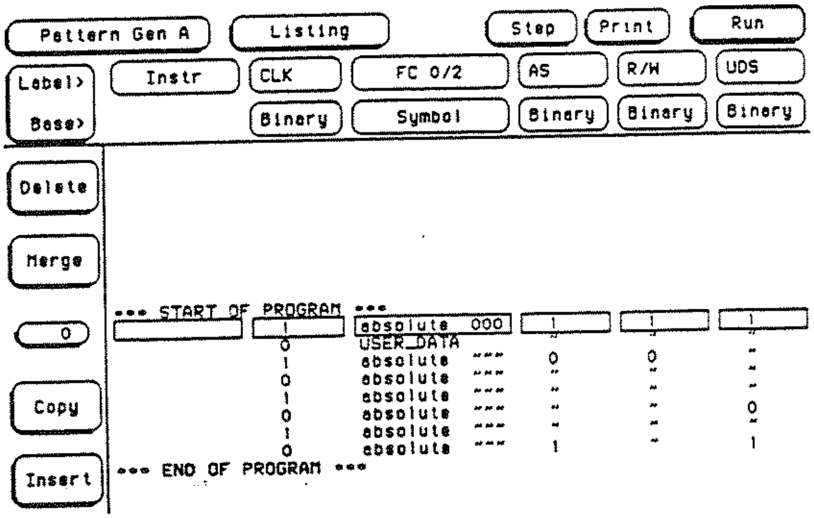
To display symbols in a program, touch the **Numerical Base** field in the **Listing** menu. A pop-up will appear with all the choices for number base.



Touch the field in the pop-up labeled **Symbol**.



The pattern generator will look at all patterns in the program under the label you've chosen to display symbols. If any of the patterns match those in the symbol table, the symbol name will be displayed for that pattern.



All patterns in the column that don't match any of the symbol patterns will be displayed as **absolute xxx**. The numbers that follow the word **absolute** will be the actual pattern for that line, displayed in the number base used in the symbol table.

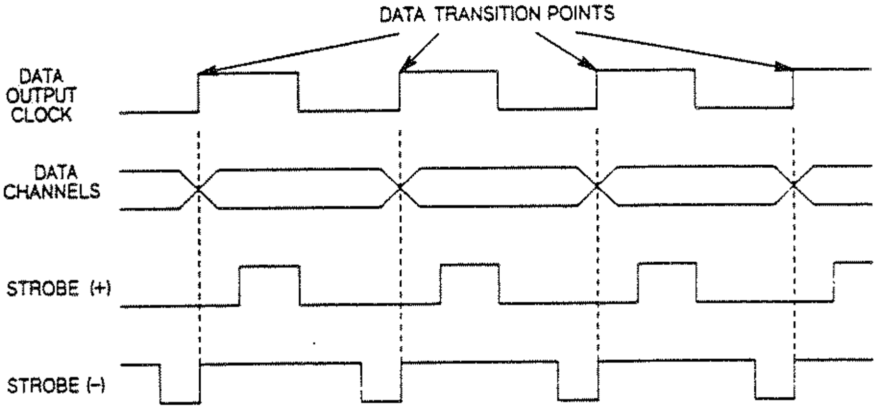
Note

The ditto character (") is used in a program listing to indicate that the pattern is the same as that immediately above it. However, the symbol table will interpret a ditto as an absolute value, and will display it as **absolute** ".

9 Defining and Using Strobes

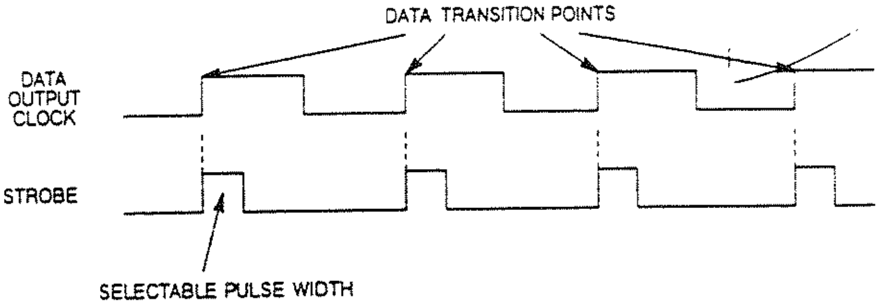
What Is a Strobe?

Strobes are data channels with selectable width and delay in the HP 16520A/16521A Pattern Generator. While standard data channels can change state only at the start of an output clock cycle, strobes can start after the clock transition and can pulse even in the middle of a clock cycle. Because of their selectable pulse width and start delay, the strobes in the HP 16520A can be used in such applications as creating asymmetrical clocks with greater or less than 50% duty cycle.

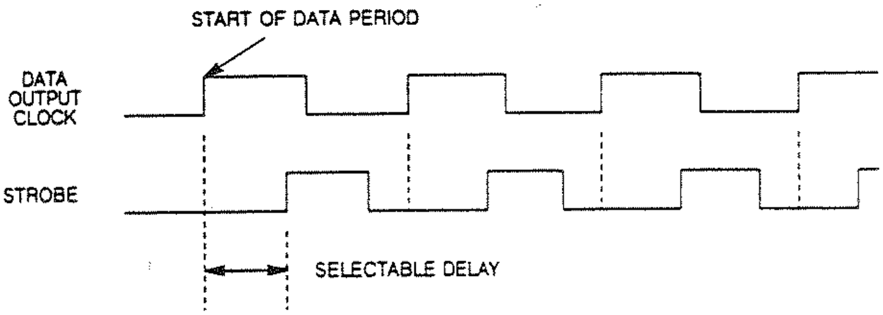


In the pattern generator, all standard data channels are referenced to the data output clock, whether that clock is internal or external. All data transitions occur on the positive edge of the clock. That means the data pulse width cannot be less than one clock period.

Strobes are a special class of data channel, allowing you to specify transitions in increments of one-fifth to one-tenth the data output clock rate. For rates greater than 20 Ms/s, the strobe rate is the same as the data output clock. For rates less than 20 Ms/s but greater than 10 Ms/s, strobes can be adjusted in increments of one-fifth the clock rate. And for rates less than 10 Ms/s, strobe width and delay may be specified in increments of one-tenth the clock rate. Pulse width may be set from one-tenth to one full clock period.



In addition, the start of the strobe pulse can be delayed from the start of the output data clock in increments of one-fifth or one-tenth the clock period.



Strobe channels are defined in the **Format** menu and controlled in the **Listing** menu. For each strobe channel, a "1" in the **Listing** menu tells the pattern generator to output the strobe as defined in the **Format** menu, while a "0" disables the strobe. By putting a one or zero in the program listing, strobes can be enabled or disabled for each output data cycle.

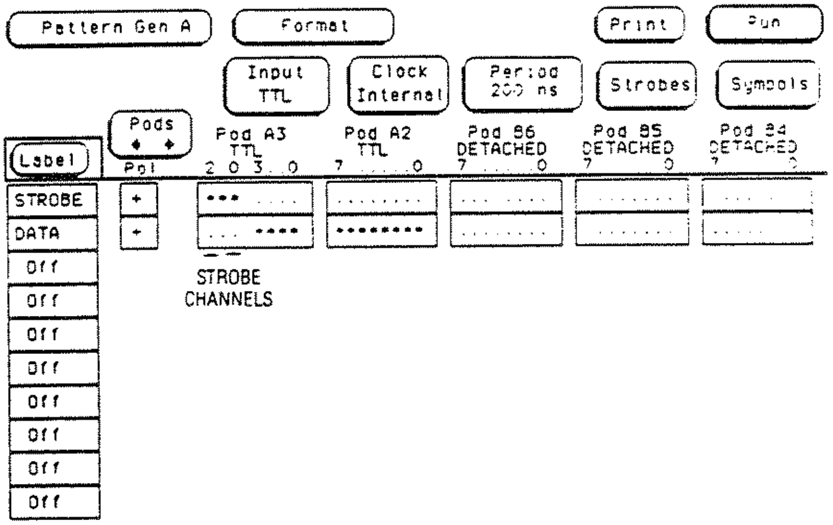
Assigning Strobe Channels

Menu: Format
Field: Channel Assignment (14)

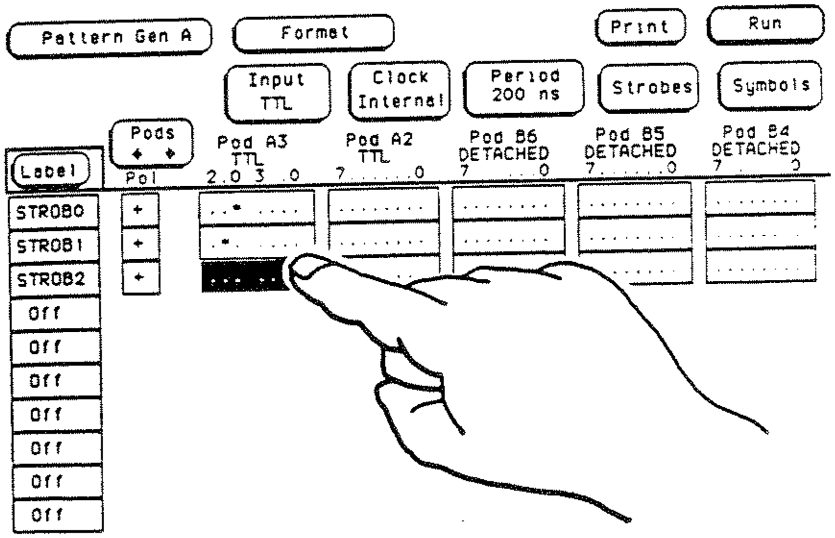
Each HP 16520A master card has three strobes. They are physically located on pod 3 of the master card.

Strobes are assigned in the **Format** menu, as are all the data channels.

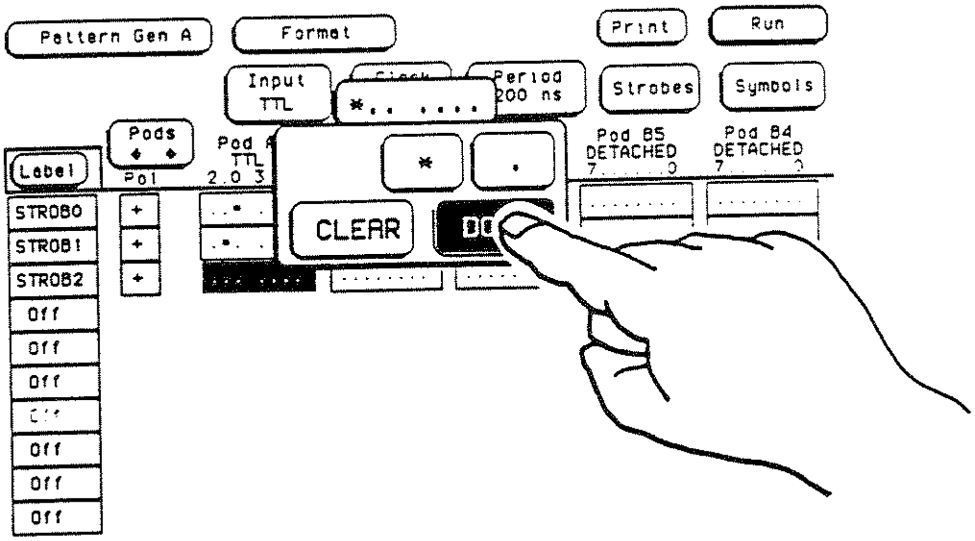
Pod 3 of the master card has two channel groupings, 0-3 and 0-2. The right-most group of pod 3 (3..0) is data channels, as you can see in the **Format** menu below. The left-most group of pod 3 (2.0) is comprised of the strobe channels.



Strobes are assigned using the same procedure as for data channels. Touch the channel assignment field and a pop-up appears with "." and "*". Assigned channels have a "*" while unassigned channels have a ".".



Using the pop-up keypad or the front-panel knob, move the cursor to the strobe channel you want to assign. Touch the "*" field to assign the channel and then the **DONE** field.

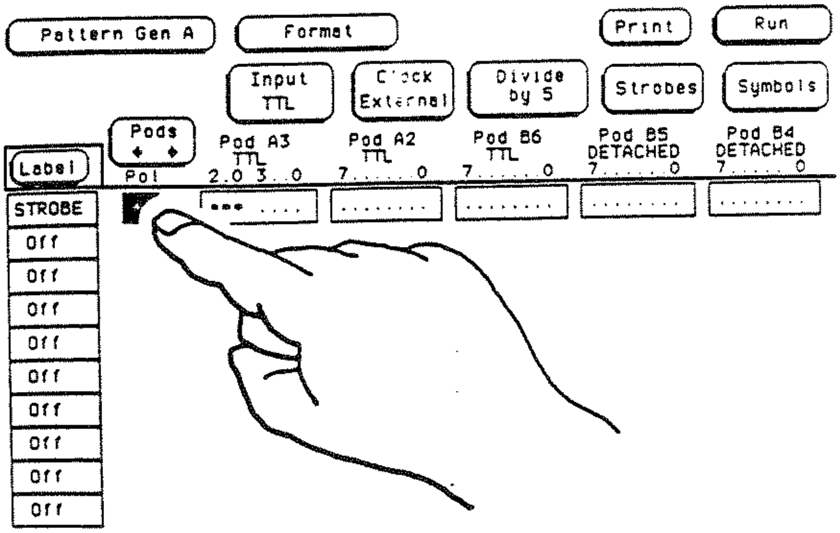


Strobes that are not assigned to a label will be output disabled.

Setting Strobe Polarity

Menu: Format
Field: Polarity (11)

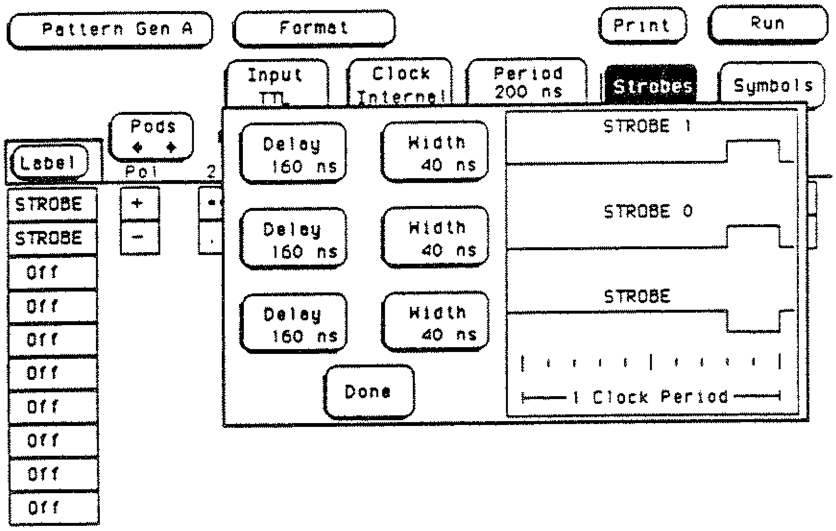
You can select negative or positive polarity for strobes. The polarity for strobes is set the same way as for data channels. Touch the polarity field for the strobe channels and it toggles between negative and positive.



For data channels, selecting negative polarity causes the output to be inverted from the program listing. If there is a **0** in the listing, the output at the probes will be a **1**.

If you select positive polarity for strobe channels, a **0** in the pattern listing tells the pattern generator to disable the strobe output, while a **1** tells it to output the strobe according to your definition in the **Strobes** menu.

If you select negative polarity, a **1** in the listing menu will disable the strobe output and a **0** will enable the output. When the output is disabled (1 in the program listing), the strobe channel will return to 1 (R1) instead of 0 (RZ). When the output is enabled (0 in the program listing), the strobe output will be inverted. Whenever the polarity is changed to negative for a strobe channel(s), the waveform in the **Strobes** menu for that strobe(s) will also be inverted.


Input TTL
Clock Internal
Period 200 ns
STROBE 1: Delay 160 ns, Width 40 ns
STROBE 0: Delay 160 ns, Width 40 ns
STROBE: Delay 160 ns, Width 40 ns

Label	Pol
STROBE	+
STROBE	−
Off	
Off	
Off	
Off	
Off	
Off	
Off	
Off	

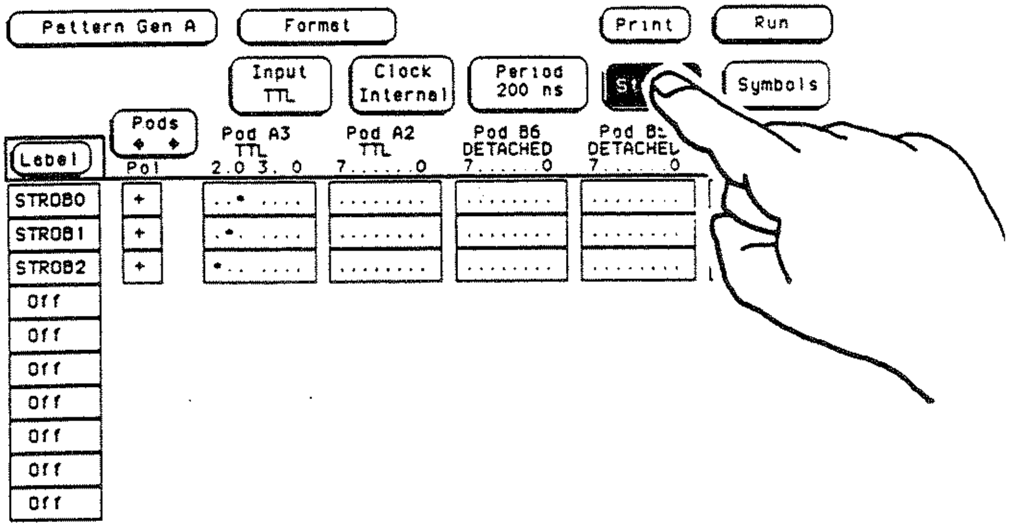
The following chart summarizes the effects of polarity on the strobe channels.

Pattern (strobe bits)	Polarity	Output (at probes)
0	+	disabled/RZ
1	+	enabled/non-inverted
0	-	enabled/inverted
1	-	disabled/R1

Setting Strobe Delay and Width

Menu: Format
Field: Strobes (8)

Strobes are defined by touching the **Strobes** field in the **Format** menu.



Input TTL

Clock Internal

Period 200 ns

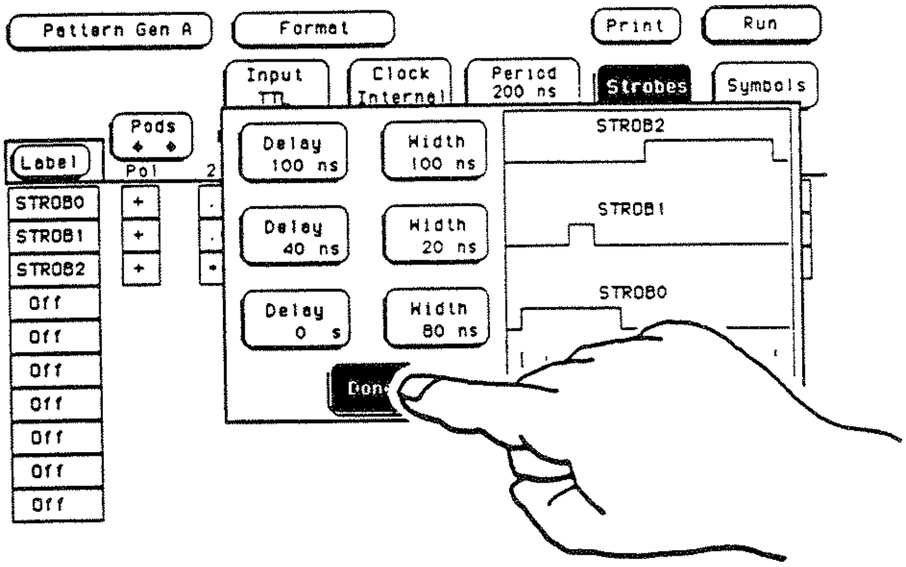
Label	Pol	Pod A3 TTL 2.0 3. 0	Pod A2 TTL 7.....0	Pod B6 DETACHED 7.....0	Pod B: DETACHEL 7.....0
STROB0	+	..•	....	....	....
STROB1	+	.•.	....	....	....
STROB2	+	•..	....	....	....
Off					
Off					
Off					
Off					
Off					
Off					
Off					

A pop-up will appear with all the strobes shown in waveforms at the right of the pop-up. The left half of the pop-up has fields for setting the delay and width of each strobe.

Note

The **Strobes** *field will not appear if the period is set to 20 ns (internal clock) or if the* **Divide by 1** *field is showing (external clock).*

Touch any dark blue **Delay** or **Width** field and it will turn light blue, indicating the value in it can be changed with the front-panel knob. You can also touch the light blue field again and a keypad will pop up on screen, allowing you to enter delay or width directly. Any changes in delay or width made with the knob or keypad will be reflected in the waveforms. When you have finished setting the delay and width of the strobes, touch the **Done** field at the bottom of the pop-up.



Input TTL

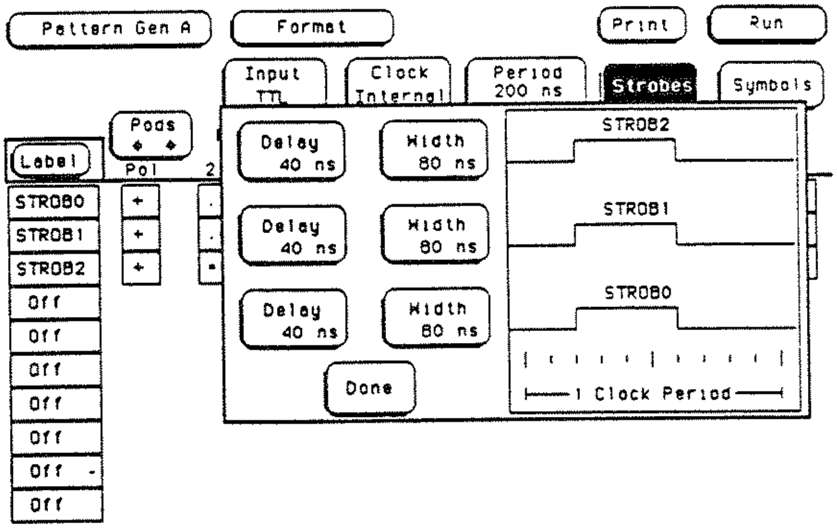
Clock Internal

Period 200 ns

Label	Pol
STROB0	+
STROB1	+
STROB2	+
Off	
Off	
Off	
Off	
Off	
Off	
Off	

Delay 100 ns	Width 100 ns	STROB2
Delay 40 ns	Width 20 ns	STROB1
Delay 0 s	Width 80 ns	STROB0

Notice the line labeled **1 Clock Period** below the strobe waveforms. This line shows the relationship among the strobes and output data clock.

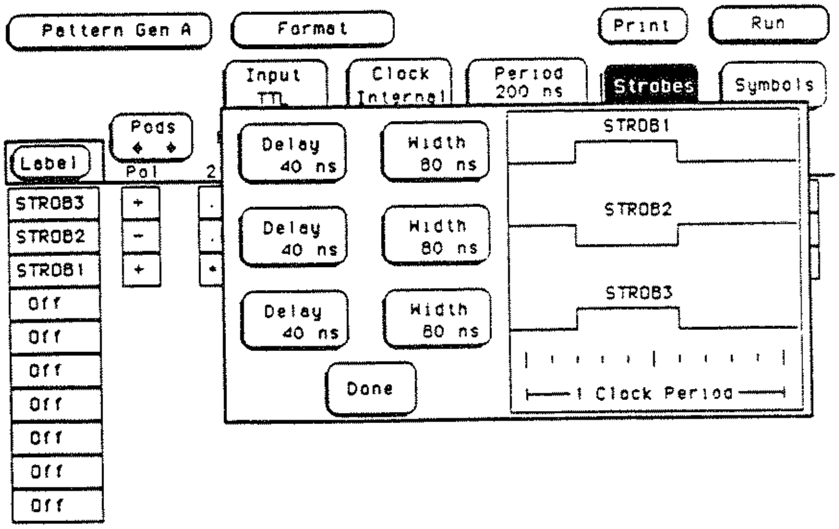


Input TTL
Clock Internal
Period 200 ns

Label	Pol
STROB0	+
STROB1	+
STROB2	+
Off	
Off	
Off	
Off	
Off	
Off	
Off	

Delay 40 ns Width 80 ns
Delay 40 ns Width 80 ns
Delay 40 ns Width 80 ns

If you set the polarity of any strobe to negative, the pop-up will reflect the change and the waveform in the pop-up will be inverted, as shown below.

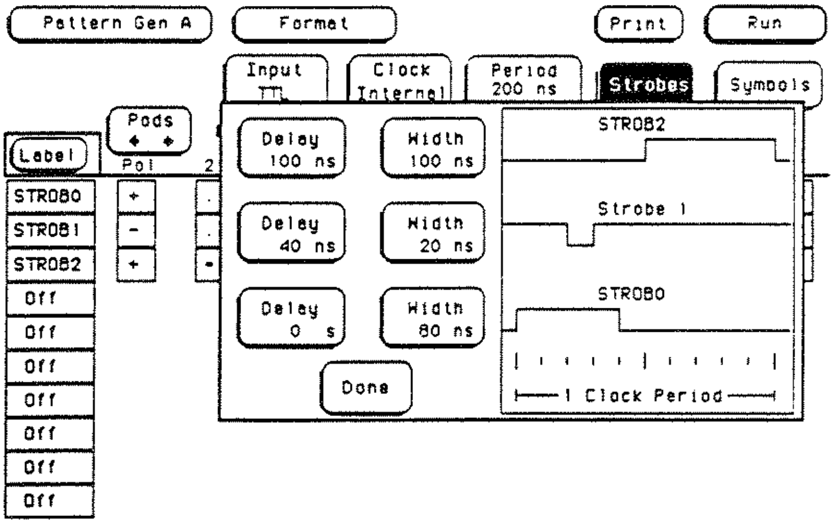


Input TTL
Clock Internal
Period 200 ns

Label	Pol
STROB3	+
STROB2	-
STROB1	+
Off	
Off	
Off	
Off	
Off	
Off	
Off	

Delay 40 ns Width 80 ns
Delay 40 ns Width 80 ns
Delay 40 ns Width 80 ns

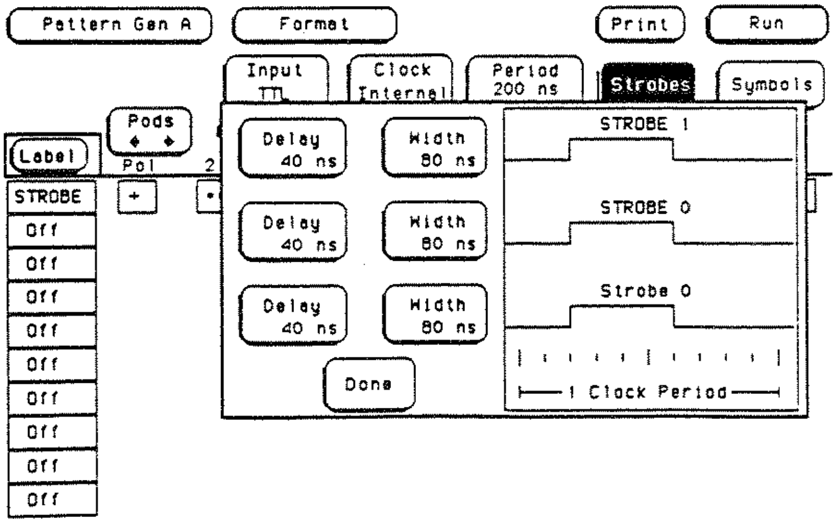
Unassigned strobes will be displayed in the pop-up but the label above the waveform will be in lower case text. Strobes that are assigned to a label will also have waveforms in the menu but the name of the label they are assigned to will be in capital letters. Notice in the screen below that the label above the middle waveform is in lower case letters, indicating that it is unassigned.



Input TTL
Clock Internal
Period 200 ns
Delay 100 ns Width 100 ns
Delay 40 ns Width 20 ns
Delay 0 s Width 80 ns

Label	Pol
STROB0	+
STROB1	-
STROB2	+
Off	
Off	
Off	
Off	
Off	
Off	
Off	

An unassigned strobe will also be displayed in the pop-up with a default label as **Strobe 0, Strobe 1**, or **Strobe 2**. It is possible to have two strobes with the same name in the pop-up if you use **STROBE** as a label. However, notice in the picture below that even though there are two Strobe 0s, the lower label is in lower case text, indicating that it is unassigned.



Input TTL
Clock Internal
Period 200 ns
Delay 40 ns Width 80 ns
Delay 40 ns Width 80 ns
Delay 40 ns Width 80 ns

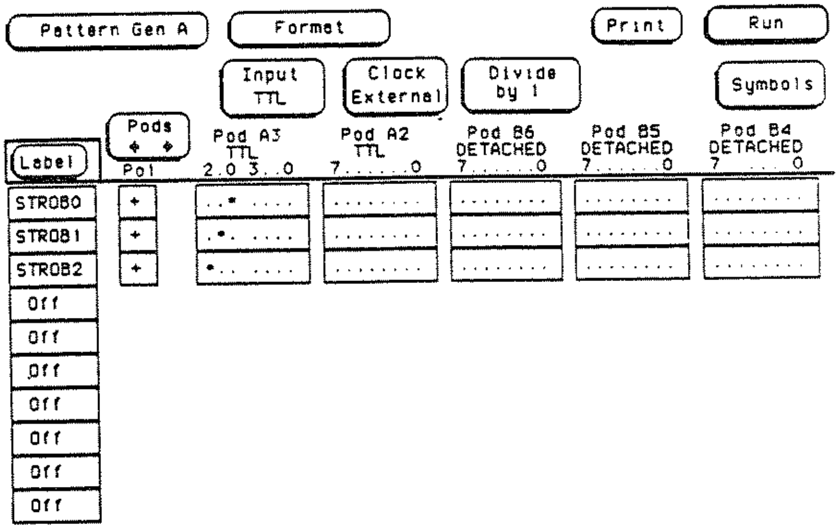
Label	Pol
STROBE	+
Off	
Off	
Off	
Off	
Off	
Off	
Off	
Off	
Off	

Specifying Strobes With an External Clock

Menu: Format
Fields: Strobes, Clock External, Divide By (8, 6, 7)

If you are using an external clock to run the pattern generator, the strobes are still available. Just as the period of the internal clock determines the strobe rate, width and delay, so does the period of the external clock.

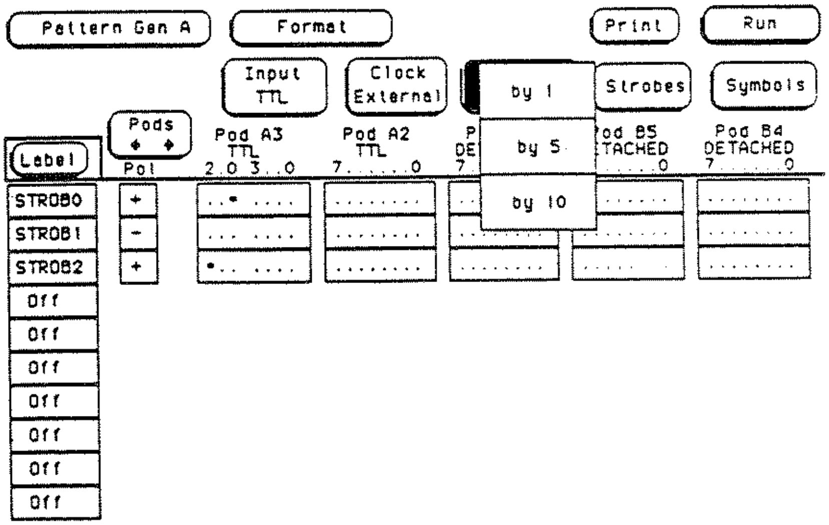
To specify an external clock, touch the **Clock Internal** field. The field will toggle to **Clock External**. When you switch to **Clock External**, the **Period** field changes to **Divide by 1**.



Label	Pol	Pod A3 TTL 2.0 3..0	Pod A2 TTL 7.......0	Pod B6 DETACHED 7.......0	Pod B5 DETACHED 7.......0	Pod B4 DETACHED 7.......0
STROB0	+	..•	....	....	....	....
STROB1	+	.•.	....	....	....	....
STROB2	+	•..	....	....	....	....
Off						
Off						
Off						
Off						
Off						
Off						
Off						

When you specify **Clock External**, the **Divide by** field gives you a choice of three different divisors: **Divide by 1, Divide by 5,** and **Divide by 10**.

These cause the external clock to be divided down by one, five, or ten respectively. The pattern generator gives the options of supplying an external clock which is five or ten times faster than needed to output data. The external clock goes through an internal divider before becoming the data clock. However, the external clock goes directly to the strobes without division. This enables the pattern generator to adjust edges of the strobes to one-fifth or one-tenth of the clock period. That, in turn, allows you to adjust the delay and width of the strobes in increments of one-fifth or one-tenth the data output clock.

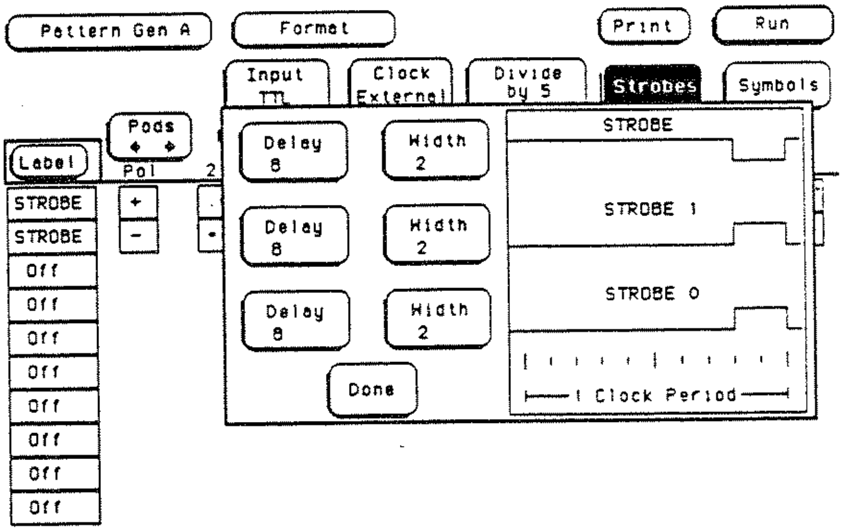


If you select **Divide by 1**, the **Strobes** field disappears, since the strobe and data clock rate would be the same. However, for rates of less than 20 MHz, strobe channels can be used as additional data channels by selecting **Divide by 1**.

Note

When using **Divide by 5**, *the strobe width and delay can only be adjusted in increments of one-fifth the output clock rate. If you specify* **Divide by 10**, *width and delay are selectable in increments of one-tenth the output clock.*

If you specify **Clock External**, strobe width and delay are shown in integer form with no units of time measure. This is because the pattern generator has no way of knowing what the external clock rate is.



Input TTL

Clock External

Divide by 5

Delay 8 | Width 2

Delay 8 | Width 2

Delay 8 | Width 2

What Happens at the End of a Program?

If the pattern generator is in **Indepedent Run Single**, the strobes are held at their last-defined state at the end of the program.

The first time the pattern generator is run in **Independent Run Single,** the hardware disables the output data clock and all the strobe output channels as the first state of the program is loaded into the output latches. of the program is loaded into the output latches. The remainder of the program is then loaded into the program memory. When the memory load is complete, the program is executed. During the time that the program is being loaded into memory, the strobes and output data clock assume the levels of the first state of the program, and are then disabled. However, the output data from the first state in the program will be on the output data channels. If this data causes problems for your system, you will need to tri-state the data channels or enter a data pattern that does not cause problems as the first state in the program.

If the pattern generator is being run in **Repetitive Run Group** or **Independent Run Repetitive**, there are not delays as with **Independent Run Single**. In other words, the first state of the program follows the last with no delay in **Independent Run Repetitive**.

10 Using Instructions

Introduction

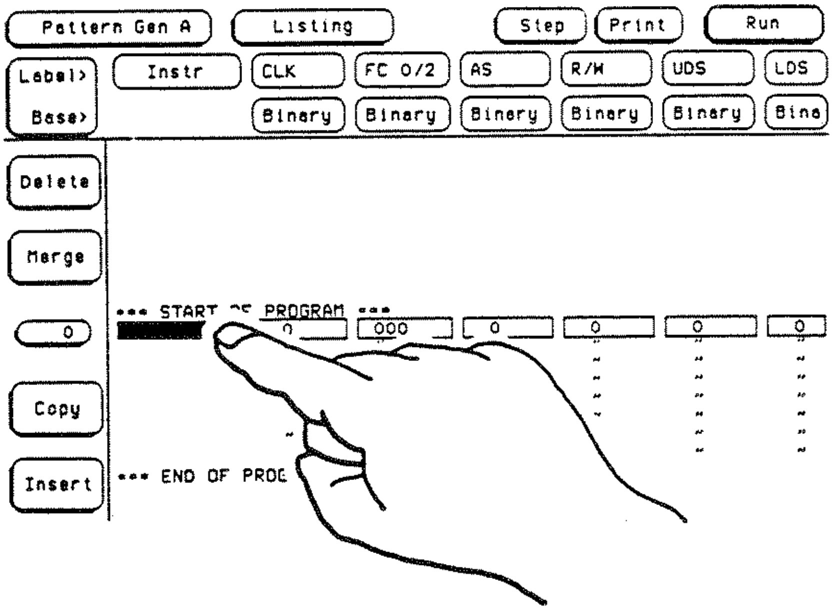
Menu: Listing
Field: Instruction (14)

The pattern generator provides five instructions for use in a program, and the ability to call any of four user-defined macros. Instructions and macros are called from the **Instruction** field in the **Listing** menu.

To insert an instruction or macro call into a program line, use the front-panel knob or pop-up keypad to move the line to the line number field. Touch the **Instruction** field and a pop-up with all the instructions appears.

Note

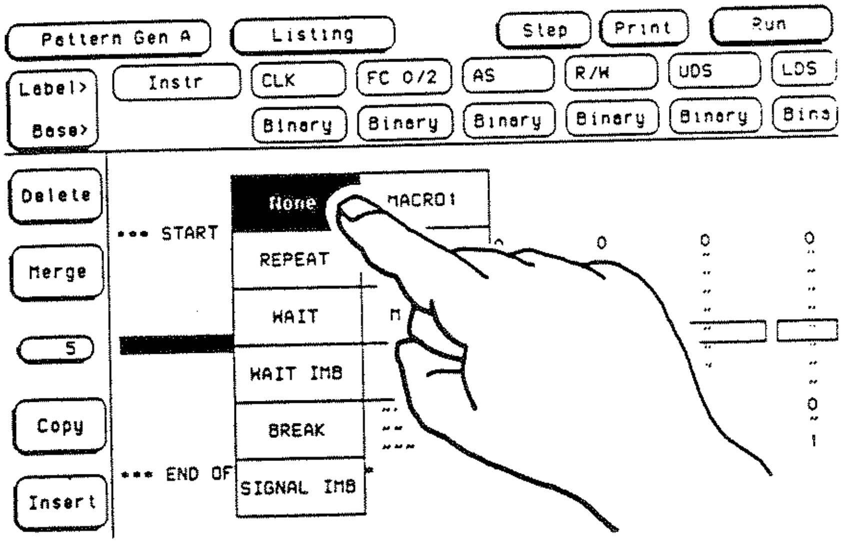
> *The* **Instruction** *field will always appear below the label* **Instr** *at the top of the screen.*



None

Menu: Listing
Field: Instruction (14)

The first field in the **Instruction** pop-up is **None**. This field does exactly what its name implies, by putting no instruction in the field. If you touch **None** and there is no previous instruction in the program line, it will close the pop-up and do nothing. This allows you to exit the **Instruction** pop-up in case you decide you do not want to put an instruction in the current program line.

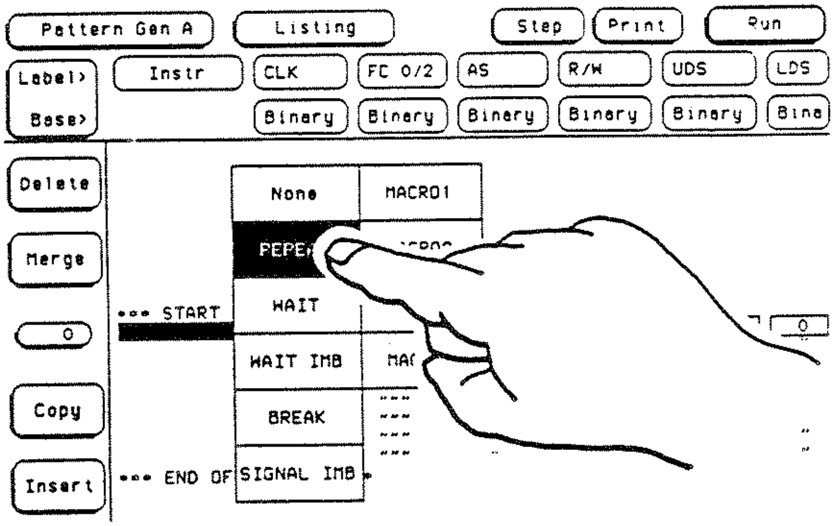


If there is already an instruction in the line, touching **None** will remove the instruction and close the pop-up.

REPEAT

Menu: Listing
Field: Instruction (14)

The **REPEAT** instruction lets you repeat a program line up to 256 times. When you touch **REPEAT**, a numeric keypad will appear to allow you to enter the number of times you want to repeat the line. When you've entered the number, touch the **DONE** key. The pop-up will close and **REPEAT XXX** (where XXX is the decimal number of repeats) will appear in the **Instruction** field.

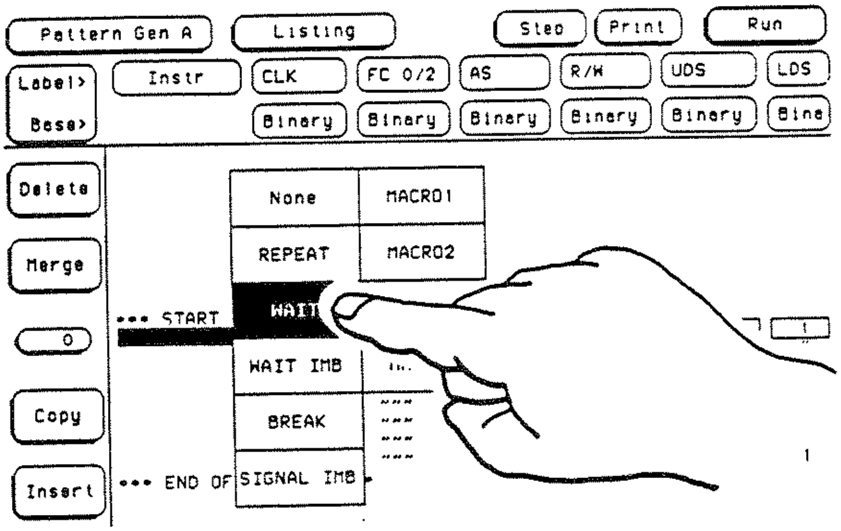


WAIT

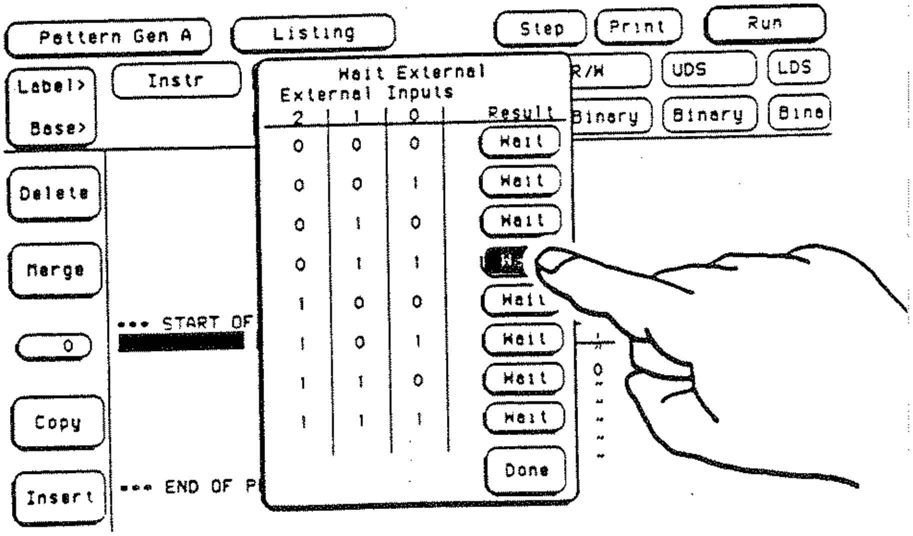
Menu: Listing
Field: Instruction (14)

Along with an external clock, there are three external input qualifiers available with each master card. The **WAIT** instruction causes the pattern generator to wait at the current program line until the three external inputs go to a pre-defined state that allow the program to go to the next program line.

When you touch **WAIT**, a table entitled **Wait External** pops up on screen. The table contains the eight binary combinations for the three external inputs, along with a **Result** column on the right. The table lets you specify on which of the three-bit conditions to wait and on which to continue program execution.



For each bit combination, there are two possible results: **Cont** (Continue) and **Wait**. To change the result for a pattern, touch the corresponding field in the **Result** column. The field will toggle to the next value.



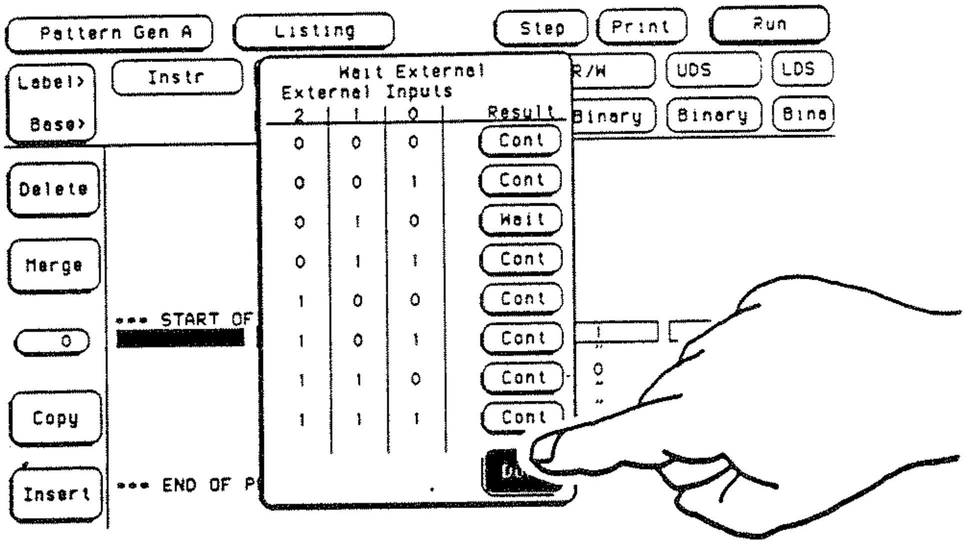
Wait External

External Inputs			
2	1	0	Result
0	0	0	Wait
0	0	1	Wait
0	1	0	Wait
0	1	1	W
1	0	0	Wait
1	0	1	Wait
1	1	0	Wait
1	1	1	Wait

You can have from zero to eight wait conditions. The default value is all eight conditions set to **WAIT**. When you have finished specifying the wait conditions, touch the **Done** field in the lower right of the table.

Note

The default values of the **Wait External** *table will cause it to wait on any input condition. Therefore, before running a program with a* **WAIT** *instruction, you'll need to change the wait conditions to whatever values you need.*

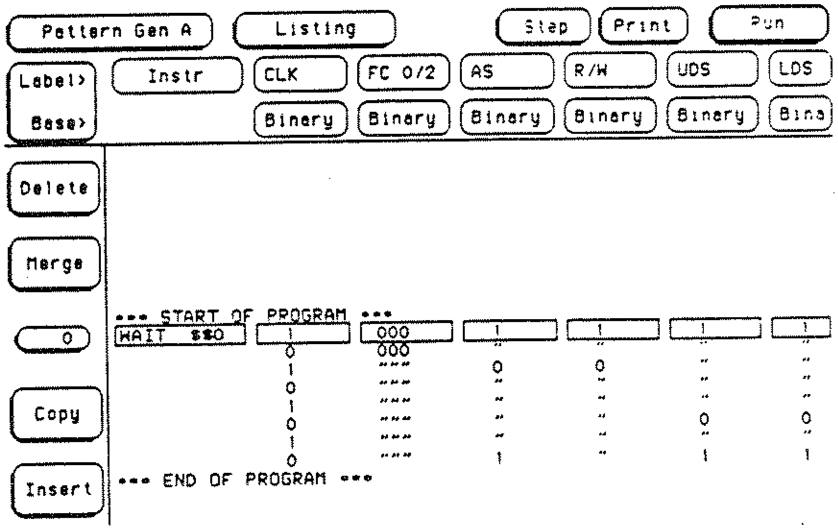


Wait External

External Inputs			
2	1	0	Result
0	0	0	Cont
0	0	1	Cont
0	1	0	Wait
0	1	1	Cont
1	0	0	Cont
1	0	1	Cont
1	1	0	Cont
1	1	1	Cont

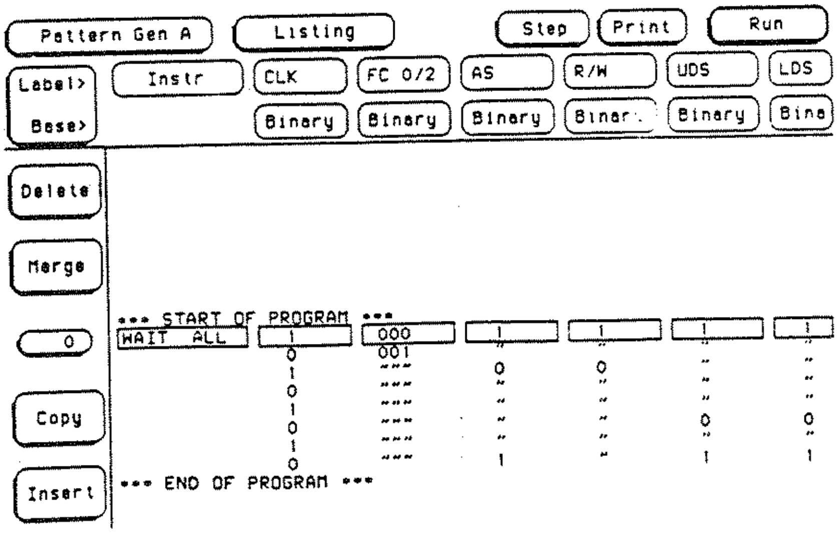
After you have set wait conditions and closed the **Wait External** pop-up, the **Instruction** field will display bit combinations that the pattern generator will look for to continue program execution. If you set a single wait condition, the **Instruction** field will display **WAIT $$$**. If you set multiple wait conditions, the **Instruction** field will display the continue conditions in an abrieviated form. For example, if you set wait conditions on 001, 011, 101, and 111 the **Instruction** field will display **WAIT XX0**, which means that the pattern generator will continue whenever the external input bit zero is a logic low. Remember, the **Wait External** table allows you to specify wait and continue conditions and the **Intruction** field shows you those conditions on which the pattern generator will continue.

If the **Instruction** field displays a **$**, it simply means that it cannot logically show all the bit combinations that were set. As an example, if **WAIT $$0** is displayed, it means that some wait conditions were set for external input bit zero, but not all. In other words, of the four possible wait conditions in which bit 0 is high, only two or three are set. Whenever a **$** appears in the **Instruction** field, it means that you'll need to go back to the **Wait External** table to see all the combinations.




```
*** START OF PROGRAM ***
WAIT  $$0   1     000    1     1     1     1
            0     000    "     "     "     "
            1     """    0     0     "     "
            0     """    "     "     "     "
            1     """    "     "     "     "
            0     """    "     "     0     0
            1     """    "     "     "     "
            0     """    1     "     1     1
*** END OF PROGRAM ***
```

If all the conditions are set to wait, the **Instruction** field will display **WAIT ALL.** If all the conditions are set to continue, the **Instruction** field will display **WAIT XXX**, meaning any combination will cause the pattern generator to continue.




```
*** START OF PROGRAM ***
WAIT  ALL   1     000    1     1     1     1
            0     001    "     "     "     "
            1     """    0     0     "     "
            0     """    "     "     "     "
            1     """    "     "     "     "
            0     """    "     "     0     0
            1     """    "     "     "     "
            0     """    1     "     1     1
*** END OF PROGRAM ***
```

The external wait inputs are sampled before the beginning of each data output cycle. If a **Wait** or **Cont** condition is met from 30 ns to 0 ns before the output data clock edge, the condition will be decoded immediately and there is no latency. The wait or continue condition will be active on the upcoming output cycle.

If the input qualifiers do not meet the 0 ns to 30 ns set-up time, but change after the positive clock edge, the condition will be active on the next clock cycle.

If a **Wait** instruction is placed in the first line of a program, the wait will be at least two data cycles long.

WAIT IMB

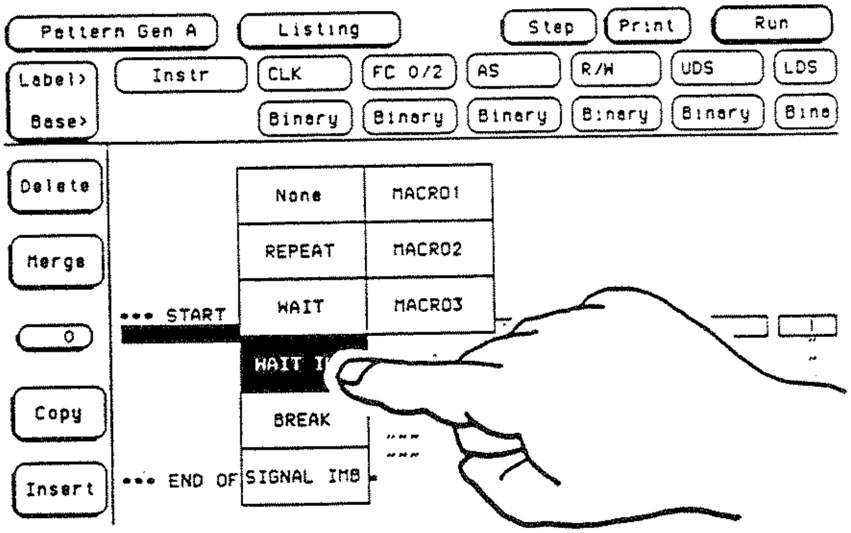
Menu: Listing
Field: Instruction (14)

Any module in the HP 16500A can signal the others through the Intermodule Bus (IMB). This is particularly useful if you need one module to tell another when to start.

If the pattern generator encounters a **WAIT IMB** instruction in the program, it will hold the data outputs at their current state, while the output data clock and the strobes continue to run. The pattern generator will not continue to the next program line until it sees a signal on the IMB. In other words, the pattern generator will wait until another module tells it to continue. This allows you to run part of a pattern generator program and then wait for an event captured by another module to occur before continuing.

The IMB can be armed only once per run of any given module. The IMB signal is latched by a receiving module and is not reset until the measurement is restarted. Therefore, it is not recommended that more than one **WAIT IMB** instruction be used in a pattern generator program. Since the IMB signal latch is not cleared until a new run is begun, multiple **WAIT IMB** instructions will result in the pattern generator sampling a previously set condition. Any **WAIT IMB** instruction after the first will thus be satisfied immediately.

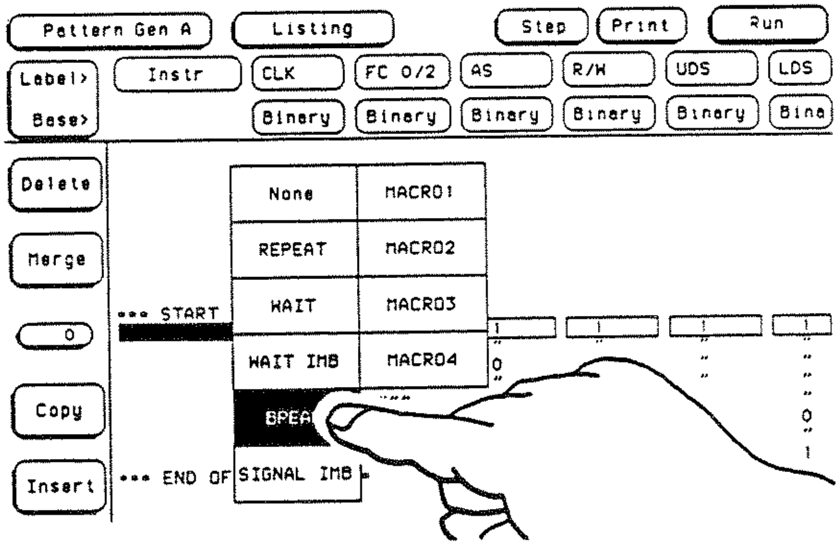
As an example, suppose you have a logic analyzer card in the HP 16500A in addition to your pattern generator. You have the logic analyzer "watching" for a service request from your system. When the logic analyzer sees the request, it can signal the pattern generator, through the IMB, to run a program you have written that deals with the interrupt.



BREAK

Menu: Listing
Field: Instruction (14)

Another feature of the pattern generator is its ability to run in single-step mode (see Chapter 7, "Running and Stopping a Program"). You may run the pattern generator until you get to a particular section, then single step the program. The **BREAK** instruction allows you to assume control of the program for single-step operation. When you insert a **BREAK** instruction, the module will halt and wait for you to tell it to resume from the **Step** menu. The **Step** menu is found in the **Listing** menu.



In **Independent Run Single** or **Group Run Single**, a **BREAK** instruction halts all pattern generator output. Data, strobes, and the output data clock will remain in the last state that occurred before the break.

In **Repetitive Run Indepedent**, a **BREAK** instruction will stop the pattern generation output. When the run control software detects that pattern generation output has stopped, it will clear the break and run the pattern generation program starting at the state immediately following the break. This means that the program will halt only briefly, until the run control software detects the stop and can start the program at the next state.

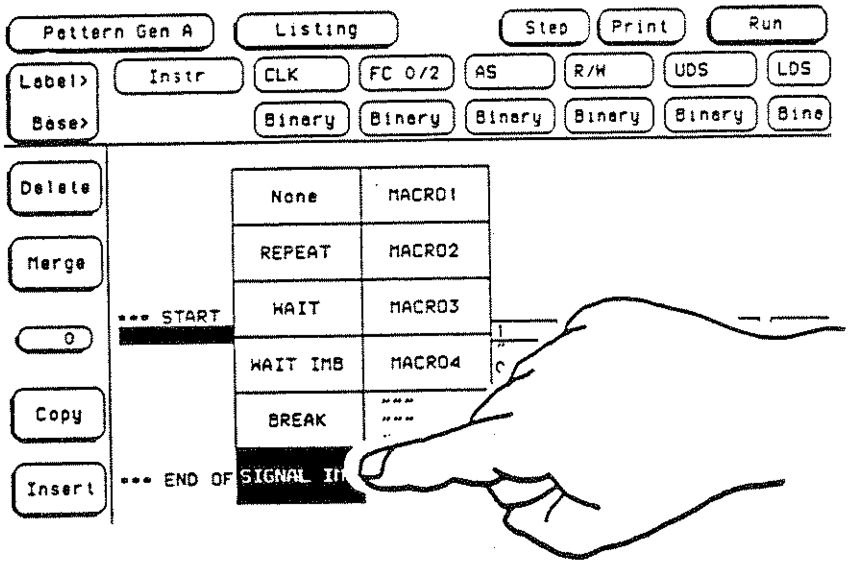
In **Repetitive Run Group**, the pattern generator will run until it encounters a **BREAK** instruction, and will wait for the other modules listed in the group run to complete their measurements. The pattern generator will then begin execution at the state immediately following the break. When the pattern generator gets to the end of its program, it will stop and wait for the other modules to complete their measurements before continuing from line 0.

For more information on independent and group run modes, see Chapter 7 of this manual and the section entitled "Run Modes."

SIGNAL IMB

Menu: Listing
Field: Instruction (14)

The complement of the **WAIT IMB** is the **SIGNAL IMB** instruction. When the pattern generator encounters a **SIGNAL IMB** instruction in a program, it will output a signal to the Intermodule Bus (IMB). This signal can be used to signal another module(s) to start running. The signal can also be used to trigger another HP 16500A mainframe through the IMB Out port.

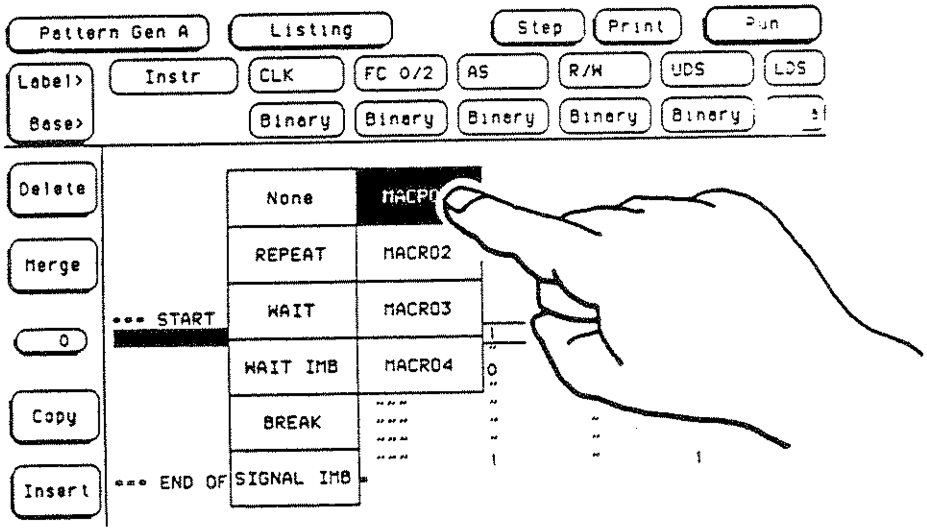


The IMB signal is latched on the rising edge by the receiving module(s). The latch is not cleared until the start of a new measurement. Because of this, any **SIGNAL IMB** instruction after the first, will have no effect on other modules that are monitoring the IMB. However, each **SIGNAL IMB** instruction can cause a signal to be sent out over the IMB Out port, by specifying it in the IMB menu.

MACROS

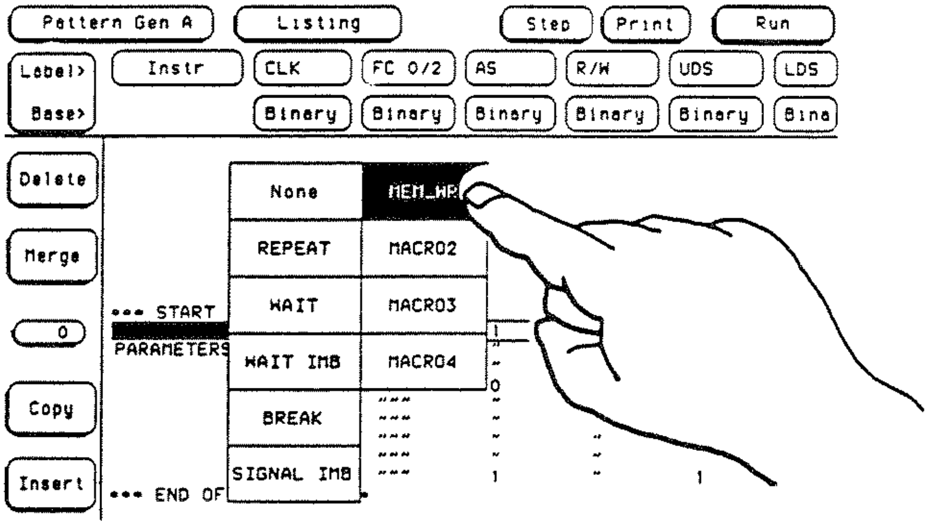
Menu: Listing
Field: Instruction (14)

The instruction field lets you call macros into your main program. You can define and call up to four macros in a program. If you have given the macros specific names, those names will appear in the pop-up. If you haven't renamed the macros, they will appear as **MACRO1, MACRO2, MACRO3,** and **MACRO4** in the pop-up.



To call a macro into your program, touch one of the macro fields from the pop-up. The pop up will close and the macro name will appear in the **Instruction** field. When the main program encounters the macro call, it will start running the specified macro.

For more information on writing and using macros, see Chapter 6 of this manual, "Creating and Using Macros."



11 Using a Printer

Setting Printer Configuration

All printer parameters are set in the **System Configuration** menu. If you have just connected your printer and are unsure of how to set the configuration, refer to the *HP 16500A Reference Manual* chapter entitled "Connecting a Printer."

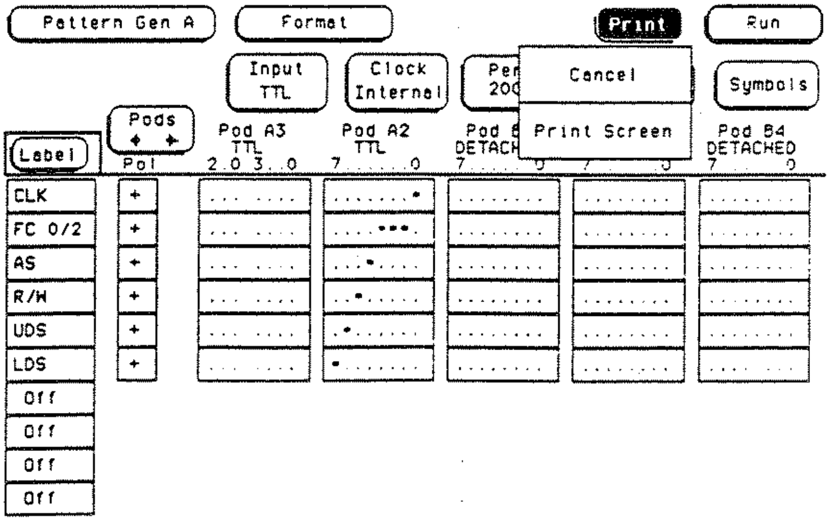
The HP 16500A supports HP-IB and selected RS-232C printers.

All the pictures in this manual were taken from an HP 16500A with one HP 16520A master card and one HP 16521A expansion card. If the screens on your instrument differ from the pictures in this manual, it simply means that you have a different card configuration. All other functions will work the same except where noted.

Printing Options

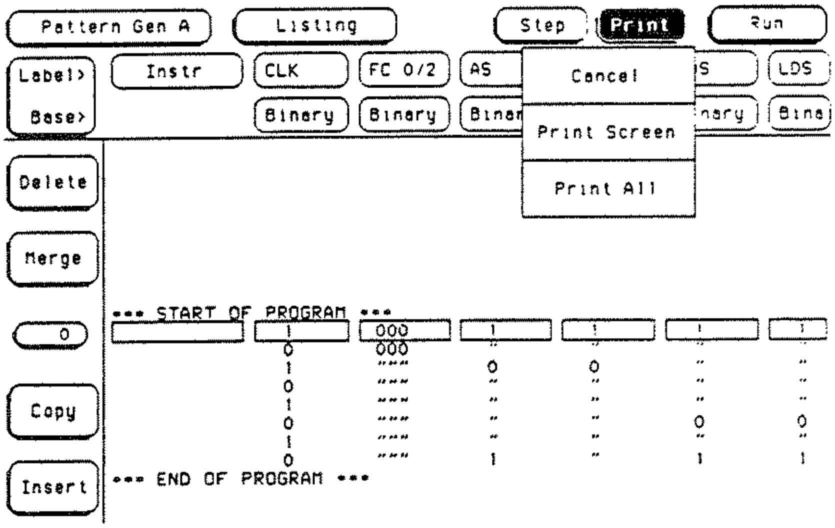
Menu: Any
Field: Print

All pattern generator menus include a **Print** field in the upper right of the screen. If you are in the **Format** menu and touch the **Print** field, a pop-up like the one shown below appears.



There are two fields in the pop-up, **Cancel** and **Print Screen**.

If you are in the **Listing** or any of the **MACRO List** menus, a slightly different pop-up will appear, like the one shown below.

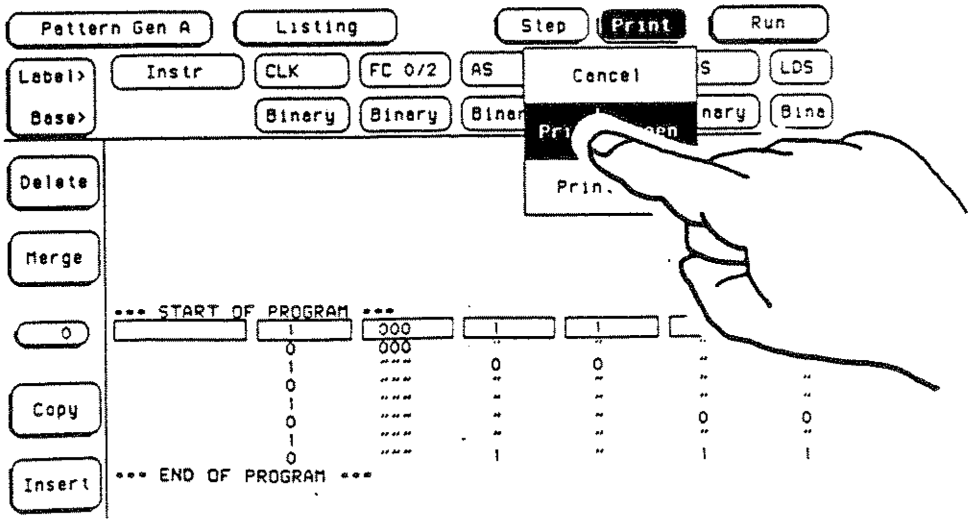


The pop-up contains three fields, **Cancel**, **Print Screen**, and **Print All**.

Printing On-Screen Data

Menu: Any
Field: Print

If you want a hardcopy record of the screen, touch the **Print** field and then the **Print Screen** field from the pop-up. This will send a copy of the screen to the printer in graphics mode.

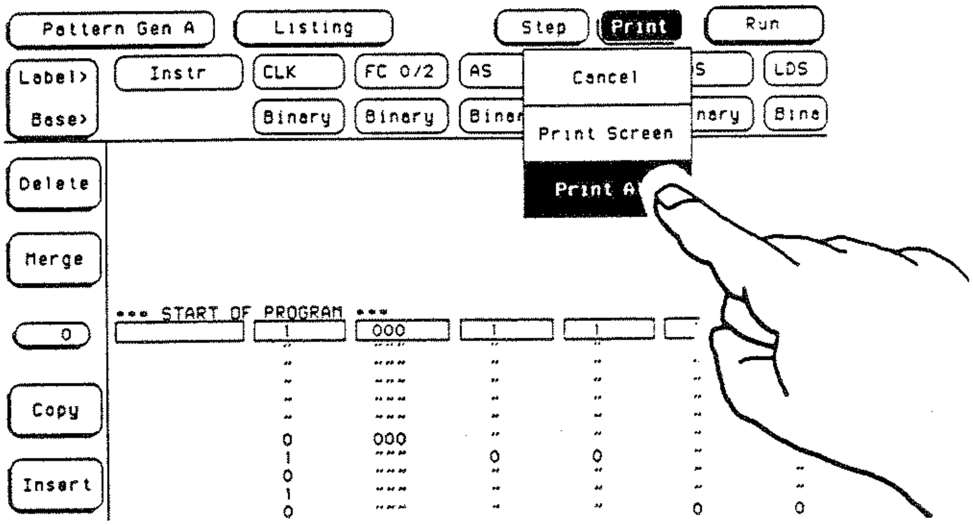


If you want to print part of a menu in graphics mode that is off screen, you must roll the screen vertically or horizontally to place the part on screen. When the desired part is on screen, touch the **Print Screen** field.

Printing Entire Program Lists

Menu: List, MACRO List
Field: Print

If you need a hardcopy record of an entire program or macro, touch the **Print** field and then the **Print All** field from the pop-up. The **Print All** field causes all the list and label data to be sent to the printer, but not in graphics mode like the **Print Screen** field. The data is sent in text mode to speed printing of long data lists. **Print All** prints the line in the line number field and all those following. This lets you print from the current line to the end of the program. If you want a hardcopy of the entire program list, move line zero to the line number field before touching **Print**.



A Installing Pattern Generator Cards

Installation Considerations

- You do not need to remove cards or filler panels that are below where the pattern generator cards will go.

- Only one intercard connecting cable is needed for any multiple card configuration.

- If other modules in the mainframe prevent you from installing the pattern generator according to the chart on following page, those modules will need to be moved to other slots.

- To maintain channel-to-channel skew and intercard signal fidelity, the shortest intercard connecting cable should be used.

- Expansion cards should be no more than two slots away from the master card.

Use the chart on the following page as a guide to selecting the correct intercard connecting cable. The chart also shows where expansion cards should be located in relation to the master card.

Installing Cards

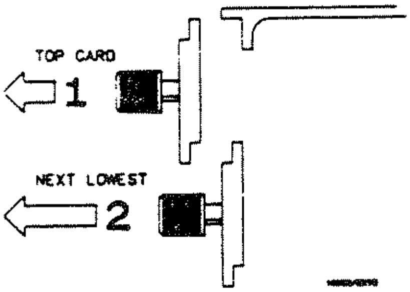
CAUTION

The effects of ELECTROSTATIC DISCHARGE can damage electronic components. Use grounded wriststraps and mats when performing the following installation procedure.

1. Turn the instrument power switch, located on the rear panel, to **Off**. Disconnect the power cord and any input or output connections.

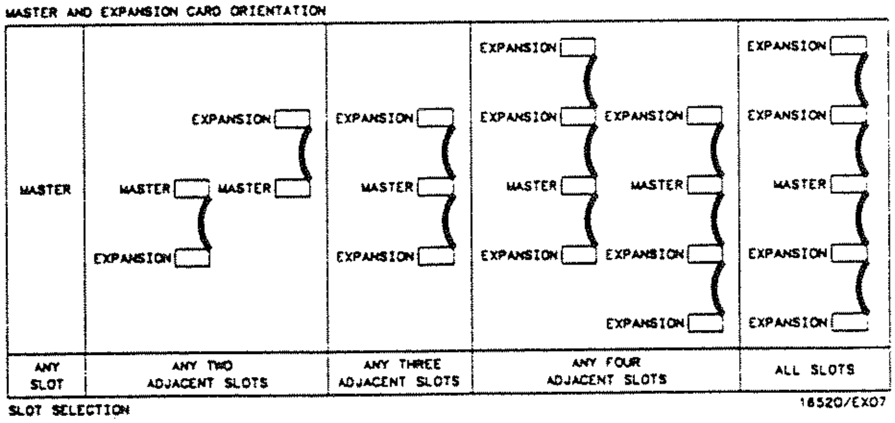
2. Starting from the top, loosen the thumb screws on any filler panels and cards already installed in the mainframe.

3. Starting from the top, begin pulling cards and filler panels out half way.



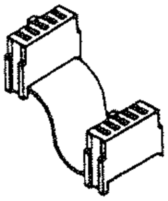
CAUTION

All multi-card modules will be cabled together. Care should be taken to pull these cards out together.

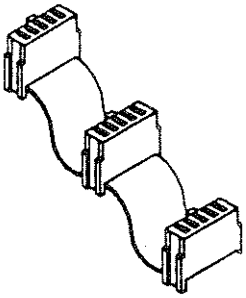


4. If you are installing only a pattern generator master card, it can be installed in any available slot. If you are installing a master card and expansion card(s), use the chart above to plan your card configuration.

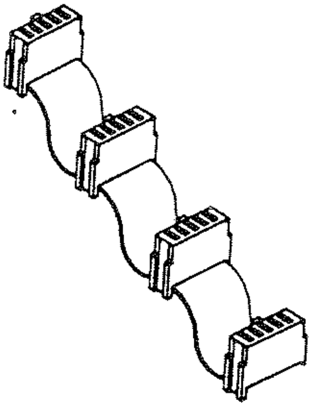
If you have a two card pattern-generator configuration, that is, one master card and one expansion, use the following intercard connector cable.



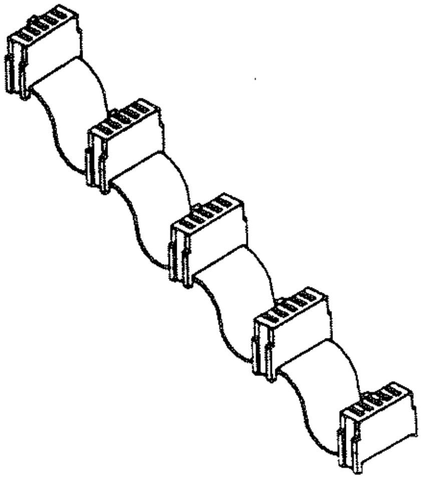
If you have a three card pattern-generator configuration, that is, one master card and two expansions, use the following intercard connector cable.



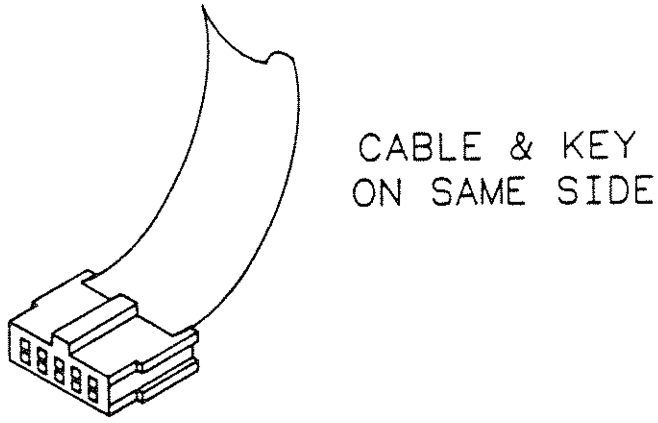
If you have a four card pattern-generator configuration, that is, one master card and three expansions, use the following intercard connector cable.



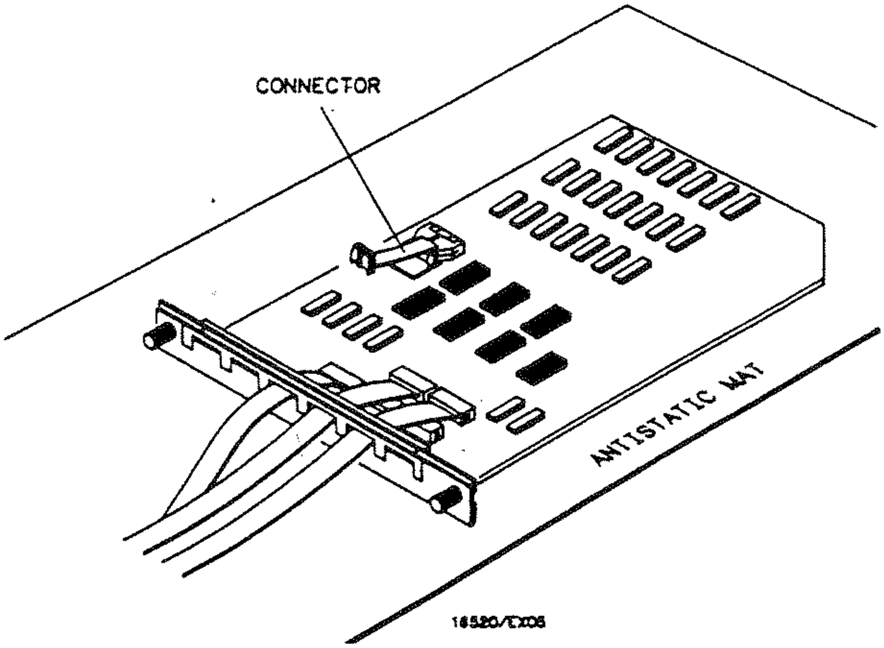
If you have a five card pattern-generator configuration, that is, one master card and four expansions, use the following intercard connector cable.



5. Insert the correct end of the intercard connector cable into the connector on the bottom card of the configuration.



6. Lay the cable of the intercard connector flat and pointing out to the rear of the card.



7. Slide the bottom card approximately half way into the lowest slot that you are going to use for the pattern generator.

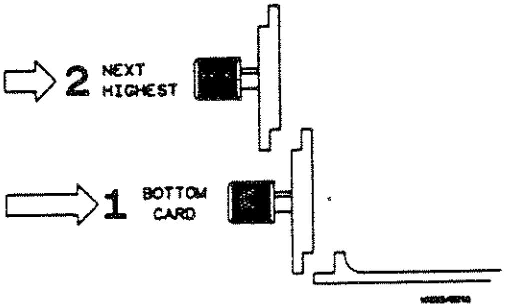
8. Slide the next card half way into the next highest slot, feeding the intercard connector cable up through the hole in the card.

9. Insert the intercard connector cable into the connector on the card.

10. If you have more than two cards to install, repeat the previous two steps until you have all the cards in the mainframe.

11. Push the bottom card all the way in and seat it into the backplane connector of the mainframe. Keep applying presure to the center of the card endplate while tightening the thumb screws finger tight.

12. Working your way up, push the rest of the cards in one at a time and seat them into the backplane connector.



Any filler panels that are not used should be kept for future use. Filler panels must be installed in all unused card slots to maintain proper air circulation within the mainframe.

If adding expansion cards to an already installed set, you'll need to pull all the installed pattern generator cards completely out of the mainframe. Remove the intercard connector cable, and use one that will connect all the cards (installed and new) with one cable. Then follow steps 5 through 12 to reinstall all the cards.

B Specifications and Characteristics

Specifications

Clock Sources (HP 16520A Only)	**Internal Clock**	
	Clock Period	programmable from 20 ns to 200 µs in a one-two-five sequence.
	Data Period Accuracy	± 2% (of period) ±1 ns
	External Clock (provided by user)	
	Input Clock Period	1 Hz to 50 MHz (20 ns min period) ECL or TTL, internal frequency divide (/1, /5, or /10) provided
	Duty Cycle	10 ns minimum high time 10 ns minimum low time
Strobes (HP 16520A Only)	**Number of Strobes**	3 (ECL or TTL)
	Bits/Channel	4095
	Maximum Bit Rate	20 MBit/s (50 ns period)
	Edge Placement	≤ 10 MBit/s: tenths of period > 10 MBit/s to 20 MBits/s: fifths of period (DELAY + WIDTH ≤ PERIOD)
	Minimum Delay	0/10 (0/5), maximum delay is 9/10 (4/5) data period
	Minimum Width	1/10 (1/5) of data period, maximum width is the data period (values in parentheses apply to 10 MBit/s timbase setting). If strobes are desired while operating with external clock, the data rate will be divided to 1/5 or 1/10 the external clock rate.

Characteristics

Eight channel pods can be assigned as either standard ECL or TTL levels. All characteristics are valid at the probe tip.

Output

	ECL	TTL
V_{OH} (steady state)	-0.98 V	2.7 V
V_{OL} (steady state)	-1.55 V (into 10kΩ, 10 pF)	0.6 V (into 10 kΩ, 10 pF)
Risetime/ falltime (typ)	2.3 ns (-0.98 V to -1.55V)	2.5 ms (0.6 V to 2.7V)
Channel-to-channel skew* (same card)	≤5 ns	≤5 ns
Channel-to-channel skew* (card-to-card)	< 10 ns	< 10 ns
Number of std loads	3 (10 KH ECL, @ V_{nh} = 150 mV)	3 (LS, @ V_{nl} = 250 mV)

(Output measurements made into a load consisting of 10 kΩ in series shunted with 10 pF to ground.)

(*) Skew measured at (+1.6 V) TTL and (-1.3 V) ECL levels.

Data Capacity

	16520A	16521A
Number of channels	12	48
Bits per channel	4095	4095
Maximum bit rate	50 MBit/s NRZ (20 ns period)	50 MBit/s NRZ (20 ns period)

Input		ECL	TTL
	V_{ih} (min)	-0.91 V	2.08 V
	V_{il} (max)	-1.69 V	1.12 V
	Maximum input voltage	±40 V	
	Input impedance	100 kΩ, 8 pF	
	External clock-in to clock-out delay	50 ns	
Editing Functions	Program Listing	DELETE, MERGE, COPY, INSERT	
Listing Bases		Binary, octal, decimal, hexadecimal, and symbol	
Step Mode		Single-step program execution in 1 to 999 program line steps, from a break.	
Data Instruction Set	Break	Stops program execution, last data vector is held at output.	
	Repeat	Repeats vector up to 256 times.	
	Wait IMB	Wait for intermodule trigger	
	Wait External	Wait for user-defined 3-bit patern on external input pod to become true. No data cycle latency when pattern is true between 30 ns and 0 ns before next clock edge.	
	Signal IMB	Arms other measurement cards.	
	Macro	Four different macros may be defined and inserted as needed. A six character name may be defined for each macro. Macros may contain	

REPEAT, WAIT EXTERNAL, WAIT IMB, BREAK AND SIGNAL IMB instructions.

Operating Environment	**Temperature**	Instrument, 0° to 55° C (+32° to 131 ° F). Probe lead sets and cables, 0° to 65° C (+32° to 149° F).
	Humidity	Instrument, up to 95% relative humidity at 40° C (104° F).
	Altitude	To 4600 m (15,000 ft).
	Vibration	
	Operating	Random vibration 5-500 Hz, 10 minutes per axis, ~0.3 g (rms)
	Non-operating	Random vibration 5-500 Hz, 10 minutes per axis, ~2.41 g (rms); and swept sine resonant search, 5-500 Hz, 0.75 g (0-peak), 5 minute resonant dwell @ 4 resonances per axis.

Index

This index covers the *HP 16520A/16521A Reference Manual*. It does not cover the *Getting Started Guide*. Sequences of pages are represented with a slash (for example, 6-9/6-17).

DIRECTORY

Worldwide Sales and Support

Hewlett-Packard products are sold and supported worldwide through HP offices.

- To contact one of the more than 240 HP Sales and Service Offices worldwide, check your telephone directory or contact one of the major HP offices listed.

Argentina
■ Martinez
Phone: 798-6086

■ **Australia**
North Ryde, N.S.W.
Phone: 02/887-1611

■ Blackburn, Victoria
Phone: 03/890-6351

Austria
■ Vienna
Phone: 222/3516210

Belgium
■ Brussels
Phone: 02/762-3200

Brazil
■ Alphaville, Barueri
Phone: 011/421-1311

Canada
■ Edmonton, Alberta
Phone: 403/452-3670

■ Richmond, British
Columbia
Phone: 604/270-2277

■ Mississauga, Ontario
Phone: 416/678-9430

Denmark
■ Birkeroed
Phone: 2/81-66-40

Finland
■ Espoo
Phone: 90/455-0211

■ Jyvaskyla
Phone: 41/216318

■ Oulu
Phone: 81/338785

France
■ Les Ulis
Phone: 1/9077825

Hong Kong
Wanchai
Phone: 5/832-3211

Italy
Cernusco Sul Naviglio
Phone: 2/903691

Japan
Yokogawa-Hewlett
Packard

■ Osaka
Phone: 6/304-6021

■ Sagamihara
Phone: 427/59-1311

■ Tokyo
Phone: 3/331-6111

Malaysia
■ Kuala Lumpur
Phone: 03/943-022

Mexico
■ Xochimilco, Mexico City
Phone: 905/676-8895

The Netherlands
■ Amstelveen
Phone: 020/472021

New Zealand
Pakuranga
Phone: 9/68-7159

■ Full Field Repair Center capabilities.

Norway
■ Oesteraas
Phone: 2/17-11-80

Puerto Rico
Carolina
Phone: 809/762-725

Saudi Arabia
■ HP Distributor;
Modern Electronics
Riyadh
Phone: 01/4919715

Singapore
■ Singapore
Phone: 631788

South Africa
■ Sandton, Transvaal
Phone: 11/802-511

Spain
■ Mirasierra, Madrid
Phone: 91/734-1162

Sweden
■ Spanga
Phone: 8/750-2000

Switzerland
■ Widen
Phone: 57/31-21-11

Taiwan
■ Taipei
Phone: 2/712-0404

United Kingdom
■ Altrincham
Phone: 61/928-6422

■ Winnersh, Wokingham
Phone: 734/784-774

United States
■ Santa Clara, California
Phone: 408/988-7000

■ Los Angeles, California
Phone: 213/970-7500

■ Englewood, Colorado
Phone: 303/771-3455

■ Atlanta, Georgia
Phone: 404/955-1500

■ Rolling Meadows, Illinois
Phone: 312/255-9800

■ Rockville, Maryland
Phone: 301/948-6370

■ Andover, Massachusetts
Phone: 617/861-8960

■ Novi, Michigan
Phone: 313/349-9200

■ Paramus, New Jersey
Phone: 201/265-5000

■ King of Prussia, Pennsylvania
Phone: 215/265-7000

■ Richardson, Texas
Phone: 214/231-6101

■ Bellevue, Washington
Phone: 206/643-4000

Venezuela
■ Caracus
Phone: 2/239-4133

West Germany
■ Boeblingen
Phone: 7031/667750

■ Full field Repair Center capabilities.

Other European Locations

For information on European locations other than those listed, contact:

Hewlett-Packard, S.A.
150, Route du Nant-d'Avril
P.O. Box
1217 Meyrin 2
Geneva, Switzerland

Other International Locations

For information on locations outside the U.S. and Europe that are not listed, contact:

Hewlett-Packard Intercontinental
3495 Deer Creek Road
Palo Alto, CA 94304 U.S.A.

HEWLETT PACKARD

8-Channel TTL Tri-State Buffer Pod

Model
10346A

Operating Note/August 1987

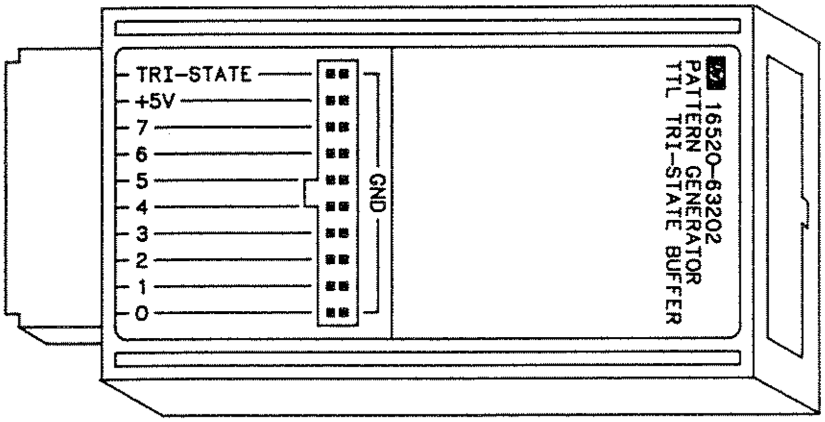


Figure 1. HP 10346A Tri-State Buffer Pod

Introduction

This operating note contains information on connecting the HP 10346A TTL Tri-State Buffer Pod (HP part number 16520-63202) to your test system. It also includes a schematic of the pod and a complete description of the pinouts.

The HP 10346A 8-Channel TTL Tri-State Buffer Pod buffers the TTL outputs of the HP 16520A and HP 16521A Pattern Generator and provides an external TTL tri-state control input.

Operating Note Part Number 10346-90901
Microfiche Part Number 10346-90801

Connecting the HP 10346A to the Pattern Generator

To connect the HP 10346A pod to the pattern generator:

1. Select the pattern generator cable for the output you want buffered and remove any probe adapters already connected to that cable.

2. Connect the HP 10346A pod to the pattern generator by aligning the key on the connector from the pattern generator cable with the slot on the pod connector and pushing them together (see figure 2).

3. If a probe adapter is required, connect the HP 10346A pod to the probe adapter of the pattern generator by aligning the key on the connector of HP 10346A pod with the slot on the probe adapter connector and pushing them together (see figure 2).

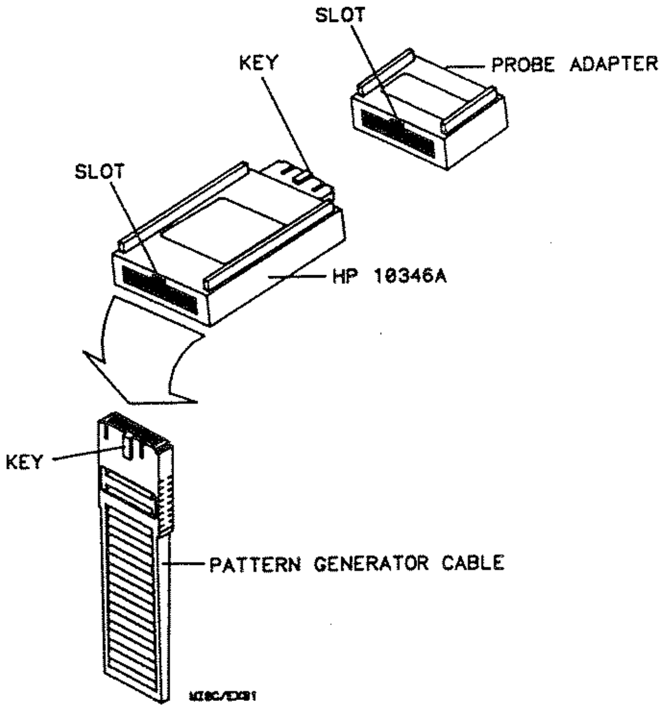


Figure 2. Connecting the HP 10346A Pod to the Cable Connector

Connecting to the Target System

Use the probes supplied with the pattern generator to connect the HP 10346A pod to the target system. The output pins for the pod are marked on the pod body (see figure 3). To connect to the target system:

1. Connect the ground probe of the HP 10346A pod to a ground pin on the target system or external supply.

2. Connect the +5 V input of the HP 10346A pod to a +5 V supply on the target system or other external source.

3. Connect the output pins 0 through 7 of the pod to the target system.

4. In order to control the pod, you must connect an input to the TRI-STATE pin of the HP 10346A pod (see figure 3). A 2.0 V high level input tri-states the HP 10346A outputs, while a 0.8 V low level input enables the pod to buffer out the input signals.

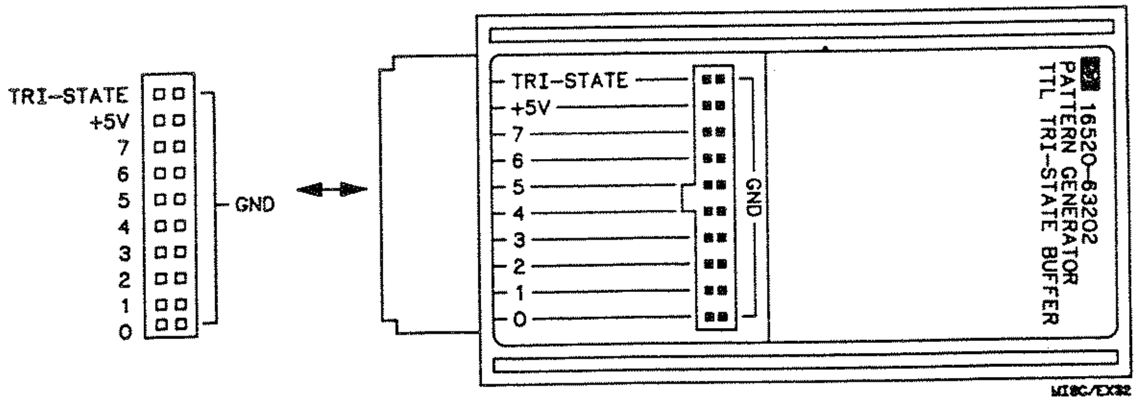


Figure 3. Labeling of Output Pins

Operating Characteristics

Low-Level Output (V_{OL} max):	+0.5 V
High Level Output (V_{OH} min):	+2.0 V
Typical Enable/Disable Time:	18 ns
Typical Propagation Delay:	12 ns
Maximum Low Output Sink Current (I_{OL}):	+24 mA
Maximum High Output Source Current (I_{OH}):	-15 mA
Typical Power Dissipation:	135 mW
Supply Voltage:	+5.0 V ±5%

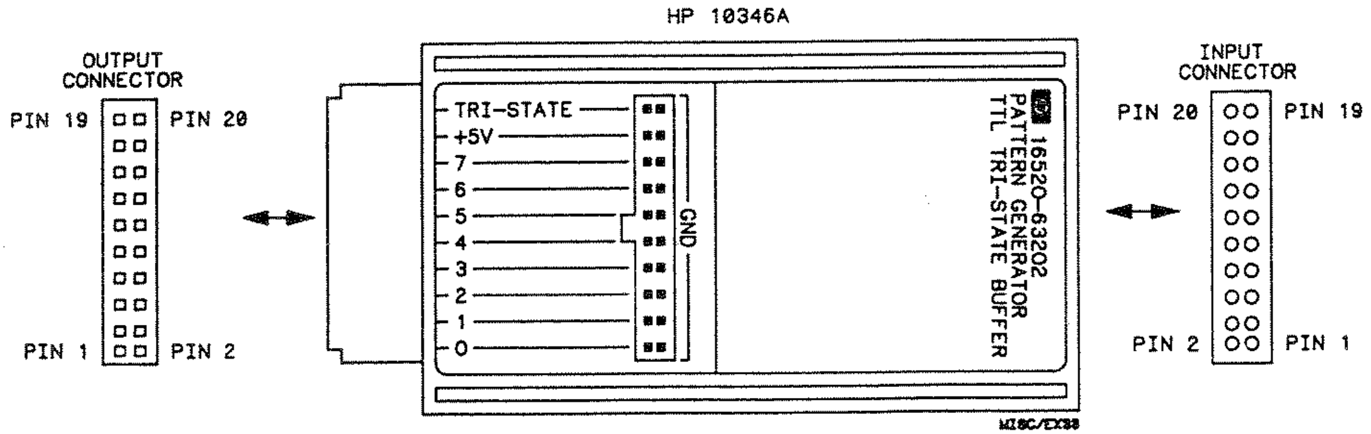


Figure 4. Pinouts for the HP 10346A Pod

Table 1. HP 10346A Signal Distribution

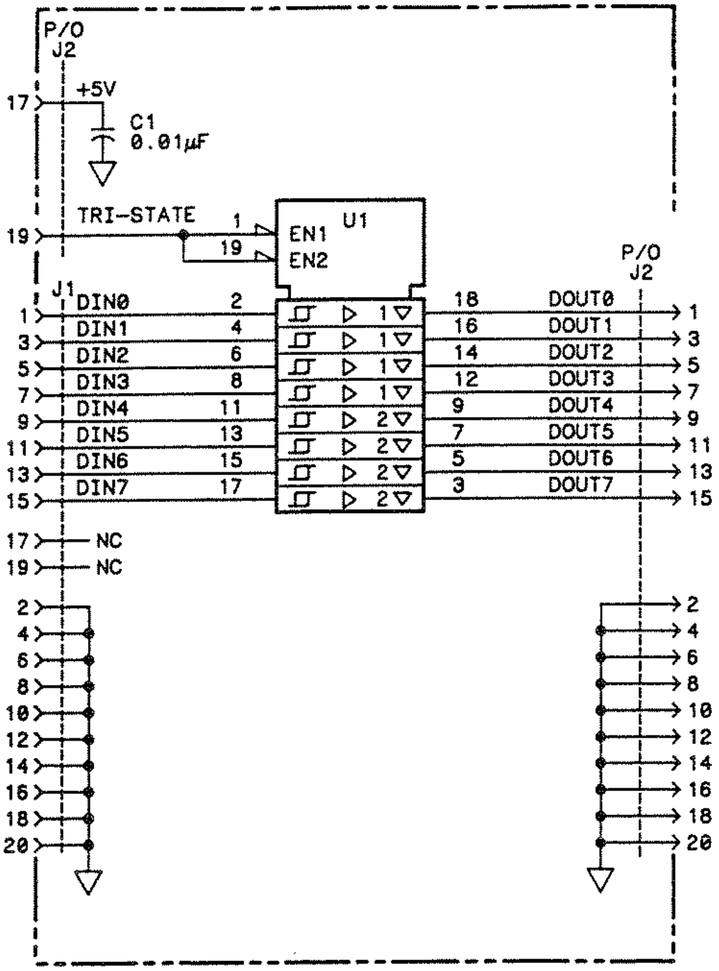
Input Pin	Description	Output Pin	Description
1	DIN0	1	DOUT0
2	GROUND	2	GROUND
3	DIN1	3	DOUT1
4	GROUND	4	GROUND
5	DIN2	5	DOUT2
6	GROUND	6	GROUND
7	DIN3	7	DOUT3
8	GROUND	8	GROUND
9	DIN4	9	DOUT4
10	GROUND	10	GROUND
11	DIN5	11	DOUT5
12	GROUND	12	GROUND
13	DIN6	13	DOUT6
14	GROUND	14	GROUND
15	DIN7	15	DOUT7
16	GROUND	16	GROUND
17	N/C	17	+5.0 V (Input)
18	GROUND	18	GROUND (Input)
19	N/C	19	TRI-STATE (Input)
20	GROUND	20	GROUND

N/C = No Connection

Note: See figure 4 for the pinouts of the HP 10346A pod.

Troubleshooting and Servicing

If a failure is suspected in the HP 10346A TTL Tri-State Buffer Pod, contact your nearest Hewlett-Packard Sales/Service Office for information on servicing the pod.



IC DEVICE POWER CONNECTIONS

SUPPLY	PIN NO.	IC GROUP
+5V GND	20 10	U1

NOTES:

1. GATES ARE SYMBOLIZED ACCORDING TO CIRCUIT FUNCTION.
2. UNLESS OTHERWISE NOTED:
   RESISTANCE IN OHMS
   CAPACITANCE IN MICROFARADS
   INDUCTANCE IN MICROHENRIES
3. UNLESS OTHERWISE NOTED:
   LOGIC LEVELS ARE TTL:
   +2.0V TO +5.0V=LOGIC"1"=H
   0V TO +0.8V=LOGIC"0"=L

PARTS ON THIS SCHEMATIC
C1 J1,2 U1

MISC/SC07/7-87

Figure 5. Schematic for the HP 10346A Pod

Product Warranty

This Hewlett-Packard product has a warranty against defects in material and workmanship for a period of 1 year from date of shipment. During warranty period, Hewlett-Packard Company will, at its option, either repair or replace products that prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by Hewlett-Packard. However, warranty service for products installed by Hewlett-Packard and certain other products designated by Hewlett-Packard will be performed at Buyer's facility at no charge within the Hewlett-Packard service travel area. Outside Hewlett-Packard service travel areas, warranty service will be performed at Buyer's facility only upon Hewlett-Packard's prior agreement and Buyer shall pay Hewlett-Packard's round trip travel expenses.

For products returned to Hewlett-Packard for warranty service, the Buyer shall prepay shipping charges to Hewlett-Packard and Hewlett-Packard shall pay shipping charges to return the product to the Buyer. However, the Buyer shall pay all shipping charges, duties, and taxes for products returned to Hewlett-Packard from another country.

Hewlett-Packard warrants that its software and firmware designated by Hewlett-Packard for use with an instrument will execute its programming instructions when properly installed on that instrument. Hewlett-Packard does not warrant that the operation of the instrument, software, or firmware will be uninterrupted or error-free.

Limitation of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by the Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. HEWLETT-PACKARD SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Exclusive Remedies

THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES. HEWLETT-PACKARD SHALL NOT BE LIABLE FOR ANY DIRECT. INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

Assistance

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For assistance, contact your nearest Hewlett-Packard Sales and Service Office. Addresses are provided at the back of this operating note.

Certification

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Bureau of Standards, to the extent allowed by the Bureau's calibration facility, and to the calibration facilities of other International Standards Organization members.

Safety

This product has been designed and tested according to International Safety Requirements. To ensure safe operation and to keep the product safe, the information, cautions, and warnings in this operating note must be heeded.

HP Model 10346A

(c) COPYRIGHT HEWLETT-PACKARD COMPANY 1987

ALL RIGHTS RESERVED

HEWLETT PACKARD

PRINTED IN U.S.A.

For more information, call your local HP sales office listed in the telephone directory white pages. Ask for the Electronic Instruments Department. Or write to Hewlett-Packard U.S.A. - P.O. Box 10301, Palo Alto, CA 94303-0890. Europe - P.O. Box 999, 1180 AZ Amstelveen, The Netherlands. Canada - 6877 Goreway Drive, Mississauga, L4V1M8, Ontario. Japan - Yokogawa-Hewlett-Packard Ltd., 3-29-21, Takaido-Higashi, Suginami-Ku, Tokyo 168. Elsewhere in the world, write to Hewlett-Packard Intercontinental, 3495 Deer Creek Road, Palo Alto, CA 94304.

HEWLETT
PACKARD

PROPERTY OF SSR

PRINTED IN U.S.A.