

## Errata

**Document Title:** Logic Analyzer Triggering Applications for the 16550A 100 MHz State / 500 MHz Timing Module (AN 1223)

**Part Number:** 5091-4121E

**Revision Date:** March 1992

---

### HP References in this Application Note

This application note may contain references to HP or Hewlett-Packard. Please note that Hewlett-Packard's former test and measurement, semiconductor products and chemical analysis businesses are now part of Agilent Technologies. We have made no changes to this application note copy. The HP XXXX referred to in this document is now the Agilent XXXX. For example, model number HP8648A is now model number Agilent 8648A.

### About this Application Note

We've added this application note to the Agilent website in an effort to help you support your product. This manual provides the best information we could find. It may be incomplete or contain dated information, and the scan quality may not be ideal. If we find a better copy in the future, we will add it to the Agilent website.

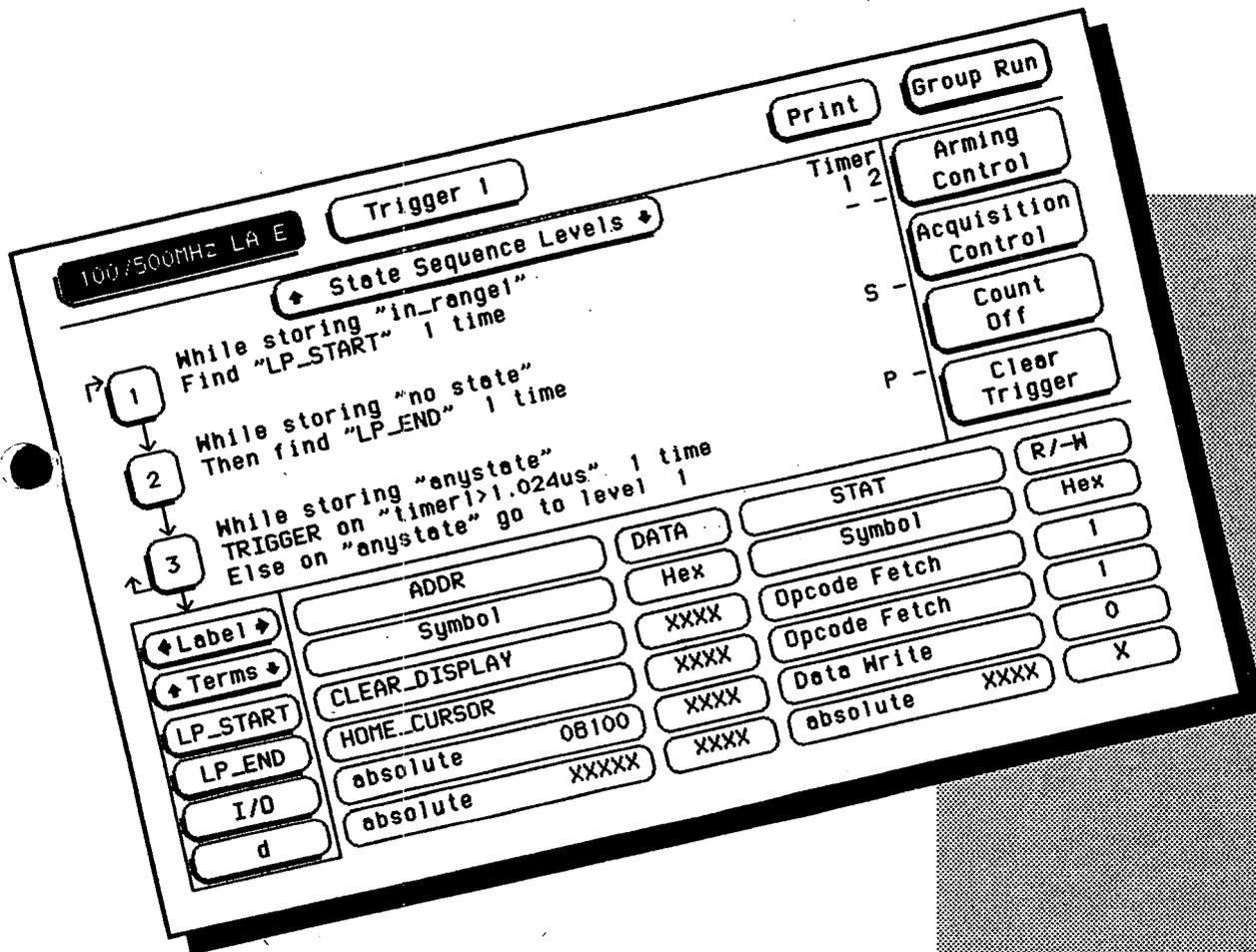
### Support for Your Product

Agilent no longer sells or supports this product. You will find any other available product information on the Agilent website:

[www.agilent.com](http://www.agilent.com)

Search for the model number of this product, and the resulting product page will guide you to any available information. Our service centers may be able to perform calibration if no repair parts are needed, but no other support from Agilent is available.

# Logic Analyzer Triggering Applications



For the HP 16550A  
100-MHz State/500-MHz  
Timing Module

Application Note 1223

April 1992

## Table Of Contents °

Application Note Overview .....	1
The HP 16550A Trigger Sequencer.....	1
Timing Analyzer Triggering Applications .....	4
State Analyzer Triggering Applications .....	5
Timing Trigger Application #1: Trigger on Stable Pattern .....	8
Timing Trigger Application #2: Trigger on nth Edge Occurrence.....	10
Timing Trigger Application #3: Find Pattern, Wait t Sec, then Trigger .....	12
Timing Trigger Application #4: Trigger on Edge while Pattern Is Valid.....	14
Timing Trigger Application #5: Trigger if Pattern Does Not Occur within t Sec of Falling Edge .....	16
Timing Trigger Application #6: Trigger if Falling Edge Is Not within Specified Time Interval after Rising Edge .....	18
Timing Trigger Application #7: Trigger on Violation of Edge Sequence .....	20
Timing Trigger Application #8: Trigger when Edge 1 and Edge 2 Are Asserted Simultaneously .....	22
Timing Trigger Application #9: Trigger on Pattern 1 Followed by Edge 1 Followed by Pattern 2 .....	24
Timing Trigger Application #10: Trigger on 8-bit Serial Pattern .....	26
Timing Trigger Application #11: Trigger Timing Analyzer with State Analyzer.....	28
State Trigger Application #1: Trigger on nth Event Occurrence.....	30
State Trigger Application #2: Store n Samples .....	32
State Trigger Application #3: Trigger on System Crash.....	34
State Trigger Application #4: Trigger on a Series of Non-Consecutive Events.....	36
State Trigger Application #5: Trigger on a Series of Consecutive Events.....	38
State Trigger Application #6: Trigger upon Exiting a Loop.....	40
State Trigger Application #7: Store Block of Events, Time Block of Events .....	42
State Trigger Application #8: Trigger on Event 2 Only After Event 1 Has Been Executed n Times .....	44
State Trigger Application #9: Trigger if n Cycles Do Not Occur between Event 1 and Event 2.....	46
State Trigger Application #10: Trigger if $\geq$ n Cycles Occur between Event 1 and Event 2.....	48
State Trigger Application #11: Trigger if t Seconds Do Not Occur between Event 1 and Event 2.....	50
State Trigger Application #12: Trigger if Event 3 Occurs $\geq$ n Times between Events 1 and 2 .....	52
State Trigger Application #13: Count Occurrences of Event 3 between Events 1 and 2.....	54

State Trigger Application #14: Trigger when Event 2 Follows  
Event 1, But Only if Event 3 Does Not Occur In-between ..... 56  
State Trigger Application #15: Trigger on Execution of a  
Specific Sequence of Program Flow ..... 58  
State Trigger Application #16: Trigger State Analyzer with  
Timing Analyzer ..... 60

## Application Note Overview

This application note contains 27 triggering situations, each with an explanation on how to set up the HP 16550A logic analyzer trigger sequencer to trigger on the situation's occurrences. These triggering applications are just a very few examples of the many different triggering scenarios that can be composed with the HP 16550A sequencer and state/timing cross-trigger controls. The descriptions are intended to get you started setting up your own specific triggering sequences for the individual problems you encounter while debugging, testing, and characterizing your design or system. In this introduction we will give you an overview of the HP 16550A sequencer and cross-triggering controls. Then in the triggering applications you will see how they are used. Pages 3 through 6 contain a listing with a brief description of each application so that you can quickly reference those examples which most interest you.

Note: the specific examples shown in each application can be run using the HP Logic Analyzer Training Kit board as a target system. The training kit is supplied with an HP 16500A mainframe, or it can be purchased as an option with the HP 16550A module. It can also be ordered as part number E2433-60003.

## The HP 16550A Trigger Sequencer

The HP 16550A sequencer and cross-triggering capabilities are controlled in the Trigger menu, shown here with its state analyzer default settings. The sequencer, the large area in the upper half of the screen, can have up to 10 levels in the timing analyzer and 12 levels in the state analyzer. It runs as fast as 125 MHz. Every 8 ns it evaluates the data coming into the timing analyzer and takes action if a pattern, edge/glitch, range, or timer term has been matched or satisfied. Every time it receives a synchronous clock it evaluates the data coming into the state analyzer. A given sequence level is programmed by touching the specific sequence level number, which pops up that level's setup menu. Sequence levels are either added or deleted from within the sequence level setup menu also.

100/500MHz LA E		Trigger 1		Print	Run
State Sequence Levels				Timer	Arming Control
1	While storing "angstate"	1	2	-	Acquisition Control
	TRIGGER on "a" 1 time			-	Count Off
2	Store "angstate"			-	Clear Trigger
Label	Label				
Terms	Hex				
a	XXXX				
b	XXXX				
c	XXXX				
d	XXXX				

The sequencer has 10 pattern terms, 2 range terms, 2 edge/glitch terms (in the timing analyzer only), and 2 timer terms. These terms are specified by the user in the bottom portion of the screen. They can all be accessed by touching the field labeled **Terms** so that it becomes light blue, and then rolling the knob up and down to the desired resource.

The HP 16550A sequencer and cross-triggering controls were designed to give you the ability to either generally or specifically define when the logic analyzer triggers. When you have a nebulous problem that you want to begin investigating by observing the activity in your system, you can use the default trigger in the HP 16550A sequencer to control data acquisition. Then, as you start to focus in on a specific portion of circuitry or section of code, you can add sequence levels, increase your usage of terms, and even cross-trigger state and timing analyzers to manage your data acquisition and trigger on exactly what you want to measure.

The following sections give brief explanations on controlling where the trigger event is stored in acquisition memory, qualifying states stored, and cross-triggering. Acquisition memory control and cross-triggering are applicable to every state and timing scenario in this application note, while storage qualification will be applicable to the state triggering examples only.

#### **HP 16550A Acquisition Memory Setup:**

You can place the trigger event you set up in the logic analyzer at the start, center, end, or a user-defined position in the acquisition memory by using the Acquisition Control menu. To access this menu, touch the **Acquisition Control** field. Touch **Acquisition Mode** to toggle it from Automatic to Manual. Touch **Trigger Position** to get the pop-up. From that pop-up you can select **Start**, **Center**, **End**, or **User-Defined** (where the trigger is placed according to a user-specified amount of post-store memory). You will be able to see the trigger position in the menu's memory diagram.

#### **HP 16550A Storage Qualification:**

Storage qualification can be used to "filter out" background code execution so you can focus on things such as a suspect subroutine, a specific range of memory accesses, or a certain type of read/write cycle.

In the HP 16550A state analyzer you are allowed to qualify the states that are stored in each level of the sequencer. If you look at any of the timelines shown in the state triggering applications, the states represented by the darker patterned areas would all be states that could be qualified. A different storage qualifier could be used in each of the areas between the vertical event bars. The qualification is done on the basis of the term or combination of terms designated in the 'While storing' line of each sequence level. If you specify 'While storing "anystate"' in each of the sequence levels, it indicates that you want all states stored. If you set each level to read 'While storing "no state"', only the occurrences designated in the 'Find', 'Then find', and

'TRIGGER' lines of the sequence levels will be stored by the state analyzer. As a final note, if you want to store the occurrences you have set up in the 'While storing' lines and not store the occurrences specified in the 'Find', 'Then find', and 'TRIGGER' lines, you can change the 'Branches Taken Stored' option in the Acquisition Control menu to 'Branches Taken Not Stored'.

#### **HP 16550A Cross-Triggering:**

Cross-triggering between a state machine and a timing machine, or between two state machines, is accomplished in the Arming Control field on the right side of the screen. Within the Arming Control menu you can also set the timing or state analyzer to send an arming signal to, or receive an arming signal from, another HP 16500A module, such as the HP 16532A, 1-GSa/s digitizing oscilloscope. (HP 16500A cross-domain analysis and cross-module triggering will be the subject of another, future application note.)

## Timing Analyzer Triggering Applications

App. #	Description	Possible Uses	Page #
1	Trigger on Stable Pattern	<ul style="list-style-type: none"> <li>* To wait for all status bus lines to finish transitioning before triggering.</li> <li>* To filter out spurious triggers because of transitions occurring during periods when the target system's state machine is indeterminate.</li> </ul>	8
2	Trigger on nth Edge Occurrence	<ul style="list-style-type: none"> <li>* To find the 3rd occurrence of the start of a data transfer.</li> <li>* To find the 1000th occurrence of a chip select line being asserted.</li> </ul>	10
3	Find Pattern, Wait t Sec, then Trigger	<ul style="list-style-type: none"> <li>* To hold off the trigger and look at control signals later in time than when the address bus pattern becomes invalid.</li> <li>* To look for a receiver's response which is supposed to occur 3 sec after a transmission.</li> </ul>	12
4	Trigger on Edge while Pattern Is Valid	<ul style="list-style-type: none"> <li>* To verify that a memory chip's select line is strobed when the chip's address on the address bus is stable.</li> <li>* To make sure the write signal to a peripheral is not violating data setup specifications.</li> </ul>	14
5	Trigger if Pattern Does Not Occur within t Sec of Falling Edge	<ul style="list-style-type: none"> <li>* To measure interrupt response time.</li> <li>* To trigger when expected data does not appear on the data bus from a remote device when requested.</li> </ul>	16
6	Trigger if Falling Edge Is Not within Specified Time Interval after Rising Edge	<ul style="list-style-type: none"> <li>* To test minimum and maximum pulse limits.</li> <li>* To verify that all pulses controlling a mechanical device fall within specifications.</li> </ul>	18
7	Trigger on Violation of Edge Sequence	<ul style="list-style-type: none"> <li>* To detect a handshake violation.</li> <li>* To trigger on incorrect control signal generation from a PLD.</li> </ul>	20
8	Trigger when Edge 1 and Edge 2 Are Asserted Simultaneously	<ul style="list-style-type: none"> <li>* To detect bus contention.</li> <li>* To view system activity when two entities are trying to seize a digital communications channel at once.</li> </ul>	22
9	Trigger on Pattern 1 Followed by Edge 1 Followed by Pattern 2	<ul style="list-style-type: none"> <li>* To verify a correct address bus, control signal, data bus sequence.</li> <li>* To check whether a data packet was sent, a handshake signal followed, and an acknowledgement was returned.</li> </ul>	24
10	Trigger on 8-bit Serial Pattern	<ul style="list-style-type: none"> <li>* To view system activity after pattern transmission.</li> <li>* To look at system status when an error pattern is detected.</li> </ul>	26
11	Trigger Timing Analyzer with State Analyzer	<ul style="list-style-type: none"> <li>* To look at control and status signals with finer resolution than once per bus cycle during execution of a specific subroutine.</li> <li>* To make timing measurements on external cache when code that continually accesses the cache is running.</li> </ul>	28

## State Analyzer Triggering Applications

App. #	Description	Possible Uses	Page #
1	Trigger on nth Event Occurrence	<ul style="list-style-type: none"> <li>* To find the 50th occurrence of a DSP subroutine.</li> <li>* To trigger on the 3rd write to a specific memory address.</li> </ul>	30
2	Store n Samples	<ul style="list-style-type: none"> <li>* To view the first 200 reads and writes to a FIFO.</li> <li>* To look at 75 pushes onto the stack.</li> </ul>	32
3	Trigger on System Crash	<ul style="list-style-type: none"> <li>* To store and display all activity leading up to a system crash.</li> <li>* To run the logic analyzer indefinitely until Stop is pressed, so that the user can observe system activity at his/her own discretion.</li> </ul>	34
4	Trigger on a Series of Non-Consecutive Events	<ul style="list-style-type: none"> <li>* To trigger on the occurrence of a calculation subroutine <u>after</u> 2 initialization subroutines have executed.</li> <li>* To trigger on the access to an I/O port after its 2 I/O registers have been set.</li> </ul>	36
5	Trigger on a Series of Consecutive Events	<ul style="list-style-type: none"> <li>* To trigger on the occurrence of a subroutine <u>only</u> when it has been called from a specific branch of the main program.</li> <li>* To look for data writes to 4 consecutive memory locations with no reads in-between.</li> </ul>	38
6	Trigger upon Exiting a Loop	<ul style="list-style-type: none"> <li>* To filter out a background monitor loop that runs until a control key is pressed.</li> <li>* To verify that all stacks and registers are restored correctly before exiting a subroutine.</li> </ul>	40
7	Store Block of Events, Time Block of Events	<ul style="list-style-type: none"> <li>* To store and time the execution of a memory management subroutine.</li> <li>* To store and time an access to a disk drive.</li> </ul>	42
8	Trigger on Event 2 Only After Event 1 Has Been Executed n Times	<ul style="list-style-type: none"> <li>* To trigger on the 1st write after the 20th read.</li> <li>* To trigger on an error code only after error recovery has been attempted twice.</li> </ul>	44
9	Trigger if n Cycles Do Not Occur between Event 1 and Event 2	<ul style="list-style-type: none"> <li>* To detect when a subroutine is exited prematurely from any of a number of exit points.</li> <li>* To find a protocol violation in sending control messages to a peripheral.</li> </ul>	46
10	Trigger if $\geq n$ Cycles Occur between Event 1 and Event 2	<ul style="list-style-type: none"> <li>* To trigger when secondary cache must be accessed between 2 consecutive memory reads, producing extra cycles.</li> <li>* To detect when an interrupt routine is executing for an excessive number of cycles.</li> </ul>	48
11	Trigger if t Sec Do Not Occur between Event 1 and Event 2	<ul style="list-style-type: none"> <li>* To trigger when a wait loop executed between 2 instructions does not produce the desired delay.</li> <li>* To trigger on a data overrun.</li> </ul>	50

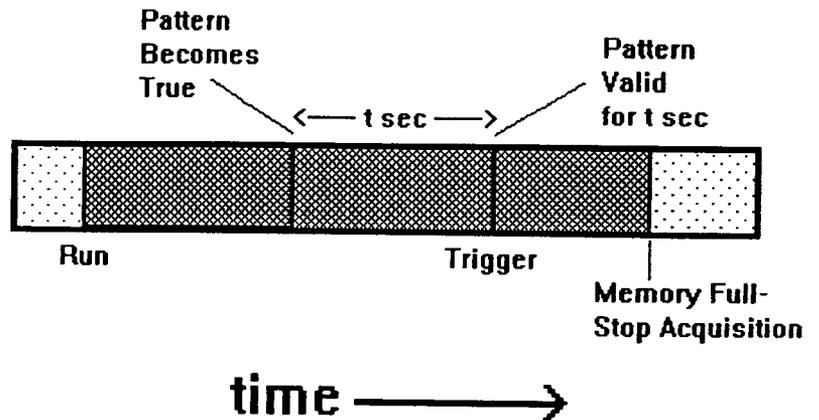
12	Trigger if Event 3 Occurs >= n Times between Events 1 and 2	<ul style="list-style-type: none"> <li>* To examine code execution when an ASIC issues a data request interrupt more than 5 times during the execution of a time-critical subroutine.</li> <li>* To trigger if a loop is executed more than 10 times between 2 non-consecutive routines.</li> </ul>	52
13	Count Occurrences of Event 3 between Events 1 and 2	<ul style="list-style-type: none"> <li>* To verify that a memory refresh routine is executing the number of times expected.</li> <li>* To count the number of memory write cycles within a segment of code.</li> </ul>	54
14	Trigger when Event 2 Follows Event 1, but Only if Event 3 Does Not Occur In-Between	<ul style="list-style-type: none"> <li>* To trigger when routine 2 follows routine 1, but only if an interrupt does not occur between them.</li> <li>* To verify that data transmit follows data receive without an error occurring as a result of the transmission.</li> </ul>	56
15	Trigger on Execution of a Specific Sequence of Program Flow	<ul style="list-style-type: none"> <li>* To trigger when procedure 3 displays an error message <u>only</u> when called by procedure 2, which is called by procedure 1.</li> <li>* To trigger on the 3rd nested occurrence of a recursive subroutine.</li> </ul>	58
16	Trigger State Analyzer with Timing Analyzer	<ul style="list-style-type: none"> <li>* To examine software execution when a timing violation occurs.</li> <li>* To determine whether an incorrectly timed pulse is the result of a hardware defect or an incorrectly programmed counter.</li> </ul>	60

# **Triggering Applications Examples**

---

**Timing Trigger Application #1:  
Trigger on Stable Pattern**

Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, looking for the specified patterns and their duration.

**HP 16550A Sequencer Setup:**

In the following example, indicative of this triggering application, you will look for a pattern of 00 hex, and you want it to be present for more than 40 ns before you trigger. The Trigger menu showing the sequencer setup for this example is shown opposite.

100/500MHz LA E		Trigger 1		Print	Run
Timing Sequence Levels				Timer 1 2 - -	Arming Control
1	TRIGGER on "PATTERN" > 40 ns			Acquisition Control	Clear Trigger
Label	Lab1				
Terms	Hex				
PATTERN	00				
b	XX				
c	XX				
d	XX				

#### HP 16550A Term Setup:

The only term needed in this instance is a pattern term. In the menu above the pattern term name is changed from 'a' to 'PATTERN' in the bottom part of the screen so that the trigger sequence in the upper part of the screen will be clearer. The 00 hex pattern for the 'PATTERN' term is also entered next to the term name.

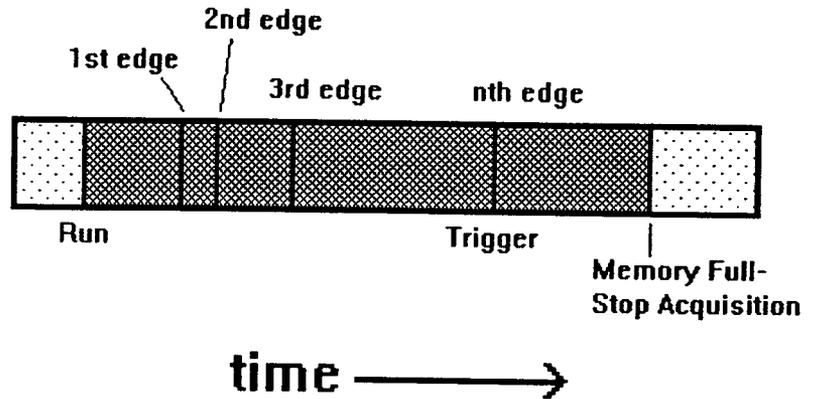
#### Possible Applications for This Triggering Model:

- \* To wait for all status bus lines to finish transitioning before triggering.
- \* To filter out spurious triggers because of transitions occurring during periods when the target system's state machine is indeterminate.

---

## Timing Trigger Application #2: Trigger on nth Edge Occurrence

Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, looking for the specified edges.

### HP 16550A Sequencer Setup:

In the following example, indicative of this triggering application, you will use  $n=10$  and find the 10th occurrence of a falling edge on any of the 8 channels associated with Label 1. The Trigger menu showing the sequencer setup for this example is pictured on the next page.

100/500MHz LA E		Trigger 1		Print	Run
Timing Sequence Levels				Timer	Arming Control
1 TRIGGER on "glitch/edge1" 10 times				1 2	Acquisition Control
				- -	Clear Trigger
◀ Label ▶	Lab1				
◀ Terms ▶	Binary				
Glitch/Edge 1	↓↓↓↓↓↓↓				
Glitch/Edge 2	.....				

**HP 16550A Term Setup:**

The only term needed in this instance is Glitch/Edge 1. It designates the falling edge of all 8 channels of Label 1, as shown in the bottom part of the screen on the previous page. (Rising edge, falling edge, either edge, or no edge can be specified for each channel of each label by the Glitch/Edge term.) By setting up a falling edge on all 8 channels, the edge indicators are in effect ORed together, so that if a falling edge occurs on any of the channels, the edge term is met.

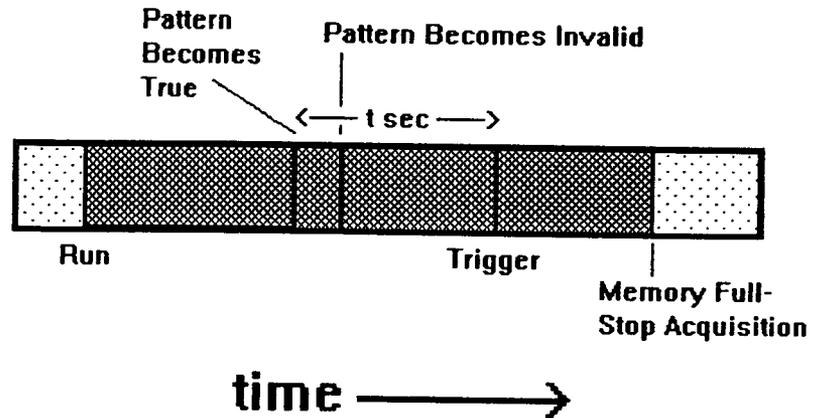
**Possible Applications for This Triggering Model:**

- \* To find the 3rd occurrence of the start of a data transfer.
- \* To find the 1000th occurrence of a chip select line being asserted.

---

**Timing Trigger Application #3:  
Find Pattern, Wait t Sec, then  
Trigger**

Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, looking for the specified patterns and their duration.

**HP 16550A Sequencer Setup:**

In the following example, indicative of this triggering application, you will find a pattern value of 50 hex, then wait 800 ns before triggering. The Trigger menu showing the sequencer setup for this example is included on the next page. As shown on the right-hand side of the sequencer field, timer 1 is started as the sequencer enters level 2, right after the pattern is found.

100/500MHz LA E		Trigger 1		Print	Run
Timing Sequence Levels				Timer	Arming Control
1	Find "PATTERN" 1 time	1	-	-	Acquisition Control
2	TRIGGER on "timer1>800ns" 1 time	S	-	-	Clear Trigger
Label	Lab1				
Terms	Hex				
PATTERN	50				
b	XX				
c	XX				
d	XX				

### HP 16550A Term Setup:

A pattern term and a timer term are needed in this example. As shown above, the pattern term name is changed from 'a' to 'PATTERN' in the bottom part of the screen so that the trigger sequence in the upper part of the screen will be clearer. The 50 hex pattern for the 'PATTERN' term is also entered next to the term name. Not shown in the picture is the timer 1 term, which is set to 800 ns. The timer 1 term is found by activating the **Terms** field and rolling the knob as explained in the application note overview on page 1.

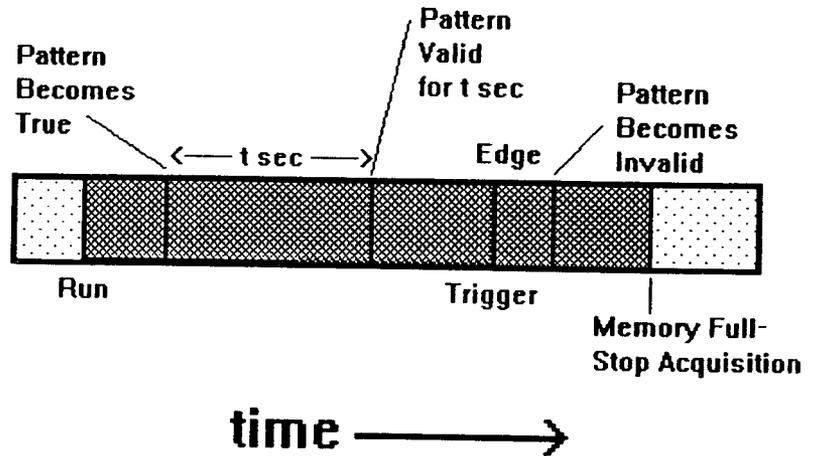
### Possible Applications for This Triggering Model:

- \* To hold off the trigger and look at control signals later in time than when the address bus pattern becomes invalid.
- \* To look for a receiver's response which is supposed to occur 3 sec after a transmission.

---

**Timing Trigger Application #4:  
Trigger on Edge while Pattern Is  
Valid**

Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, looking for the specified patterns, pattern durations, and edges.

**HP 16550A Sequencer Setup:**

In the following example, indicative of this triggering application, you will look for an 89 hex pattern to be stable for 80 ns, then trigger on the falling edge of the least significant of the 8 bits being used. In effect this setup allows you to trigger on the end of the 89 hex bus cycle and confirm that the least significant bit is the first bit to change. The Trigger menu showing the sequencer setup for this example is pictured on the next page.

100/500MHz LA E		Trigger 1		Print	Run
Timing Sequence Levels				Timer	Arming Control
1	Find "PATTERN" > 80 ns	1		--	Acquisition Control
2	TRIGGER on "glitch/edge1" 1 time Else on "PATTERN" go to level 1	2		--	Clear Trigger
Label	Label				
Terms	Hex				
PATTERN	89				
b	XX				
c	XX				
d	XX				

### HP 16550A Term Setup:

A pattern term and a glitch/edge term are needed in this example. As shown above, the pattern term name is changed from 'a' to 'PATTERN' in the bottom part of the screen so that the trigger sequence in the upper part of the screen will be clearer. The 89 hex pattern for the 'PATTERN' term is also entered next to the term name. Not shown in the picture is the glitch/edge 1 term, which is set to a falling edge on the least significant bit of the 8 channels assigned to Label 1. The glitch/edge 1 term is found by activating the Terms field and rolling the knob as explained in the application note overview on page 1.

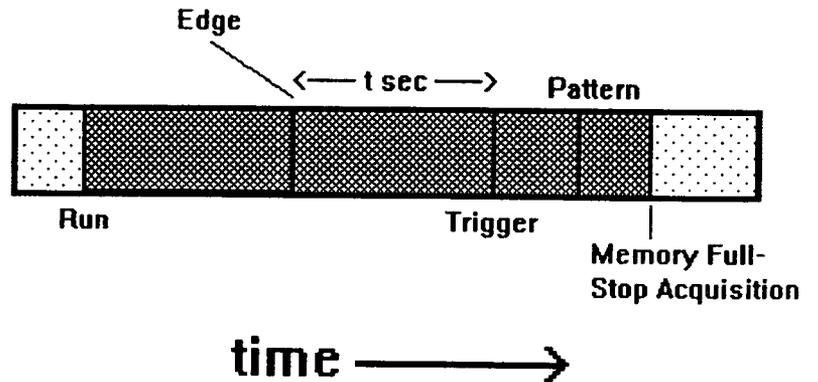
### Possible Applications for This Triggering Model:

- \* To verify that a memory chip's select line is strobed when the chip's address on the address bus is stable.
- \* To make sure the write signal to a peripheral is not violating data setup specifications.

---

**Timing Trigger Application #5:  
Trigger if Pattern Does Not  
Occur within t Sec of Falling  
Edge**

Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, looking for the specified edges, time intervals, and patterns.

**HP 16550A Sequencer Setup:**

In the sequencer setup example shown on the next page you will first find the rising edge of the most significant of the 8 bits to which you are connected. Then you will trigger if more than 13 microseconds elapse between that rising edge and a 00 hex pattern on the 8 bits. The timer is started as the sequencer enters level 2, right after the edge is found (as indicated by the 'S' under timer 1 in the right part of the sequencer field). The arrows coming out of level 2 and going into level 1 indicate the branching that occurs back to level 1 if the 00 hex pattern is detected and no timing violation occurs.

100/500MHz LA E		Trigger 1	Print	Run
Timing Sequence Levels			Timer	Arming Control
1	Find "glitch/edge1" 1 time		1 2	Acquisition Control
2	TRIGGER on "timer1>13.01us" 1 time Else on "PATTERN" go to level 1		S -	Clear Trigger
◀Label▶				
▶Terms▶				
Timer1	13.01 us			
Timer2	400 ns			

### HP 16550A Term Setup:

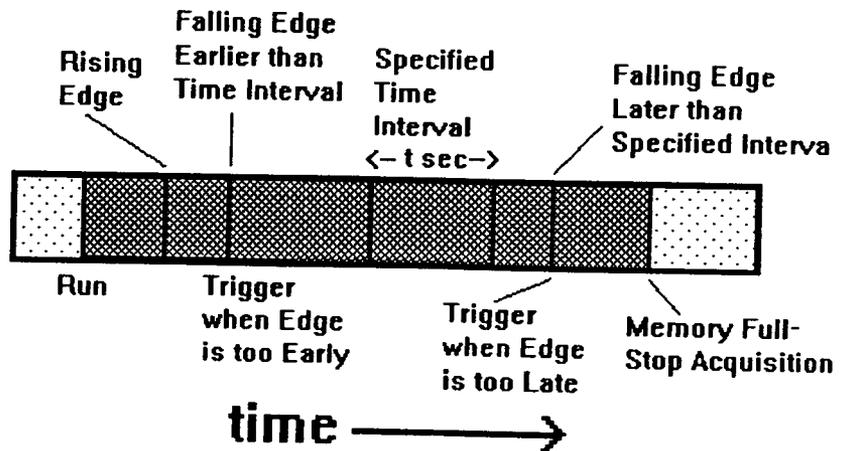
A pattern term, a glitch/edge term, and a timer term are needed in this example. In the picture above, the timer term setting of 13 microseconds is shown in the bottom part of the screen. In this part of the screen, even though it is not visible in this picture, the pattern term name is changed from 'a' to 'PATTERN' so that the trigger sequence in the upper part of the screen will be clearer. The 00 hex pattern for the 'PATTERN' term is also entered next to the term name. Also not shown in the picture is the glitch/edge 1 term, which is set to a rising edge on the most significant bit of the 8 channels assigned to Label 1. The glitch/edge 1 term and 'PATTERN' term are found by activating the Terms field and rolling the knob as explained in the application note overview on page 1.

### Possible Applications for This Triggering Model:

- \* To measure interrupt response time.
- \* To trigger when expected data does not appear on the data bus from a remote device when requested.

**Timing Trigger Application #6:  
Trigger if Falling Edge Is Not  
within Specified Time Interval  
after Rising Edge**

Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, looking for the specified edges and time intervals.

**HP 16550A Sequencer Setup:**

In the following example, indicative of this triggering application, you will look for a timing violation on the most significant bit of the 8 channels associated with Label 1. The Trigger menu showing the sequencer setup for this example is pictured opposite. In level 1 you look for the rising edge. In level 2 you look for the falling edge. The 2 timers are started as this level is entered, right after the rising edge is found, as displayed in the right-hand portion of the sequencer field. In level 3 you pause the timers to check them, and trigger the analyzer if the pulse was shorter than 496 nanoseconds or longer than 10 microseconds. Note the logical combination of the two timer terms in level 3 to define the trigger condition. (Terms can be logically combined in any of the term fields in any of the sequence levels.) Also, the arrow going out of level 3 and into level 1 indicates that the sequencer is programmed to reset if no violation was found.

100/500MHz LA E		Trigger 1		Print	Run
Timing Sequence Levels				Timer	Arming Control
1	Find "glitch/edge1" 1 time	1	2	-	-
2	Then find "glitch/edge2" 1 time	S	S	S	S
3	TRIGGER on "timer1<496ns+timer2>1..." 1 time Else on "anystate" go to level 1	P	P	P	P
↓Label ↓ ↓Terms ↓ Glitch/ Edge 1 Glitch/ Edge 2	Leb1 Binary ↑..... ↓.....	Acquisition Control Clear Trigger			

**HP 16550A Term Setup:**

Both glitch/edge terms and both timers are needed in this example. As shown above, glitch/edge 1 is set to find the rising edge of the most significant bit, and glitch/edge 2 is set to find the falling edge of that bit. Not displayed are the timer terms; timer 1 is programmed for 496 nanoseconds (the timer setting closest to 500 ns) and timer 2 for 10 microseconds. The timer terms are found by activating the Terms field and rolling the knob as explained in the application note overview on page 1.

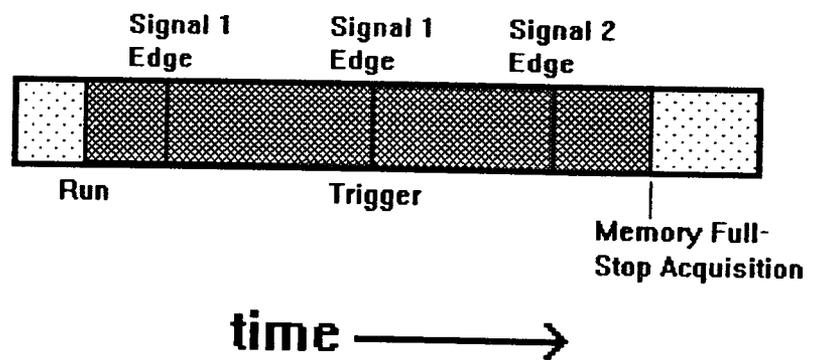
**Possible Applications of This Triggering Model:**

- \* To test minimum and maximum pulse limits.
- \* To verify that all pulses controlling a mechanical device fall within specifications.

---

**Timing Trigger Application #7:  
Trigger on Violation of Edge  
Sequence**

Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, looking for the specified edges.

**HP 16550A Sequencer Setup:**

In the following example, indicative of this triggering application, you will look for an edge on bit 0 of the 8 bits assigned to Label 1. Then you will trigger if the analyzer sees another edge on bit 0 before an edge on bit 1 is detected. The Trigger menu showing the sequencer setup for this example is included on the next page. Notice the arrows, coming out of level 2 and going into level 1. These arrows indicate that the sequencer is programmed to reset if the proper edge sequence is found.

100/500MHz LA E		Trigger 1		Print	Run
Timing Sequence Levels				Timer	Arming Control
				1	Acquisition Control
1	Find "glitch/edge1" 1 time			--	
2	TRIGGER on "glitch/edge1" 1 time Else on "glitch/edge2" go to level 1			--	
					Clear Trigger
Label	Label				
Terms	Binary				
Glitch/Edge 1	.....:				
Glitch/Edge 2	.....\$:				

#### HP 16550A Term Setup:

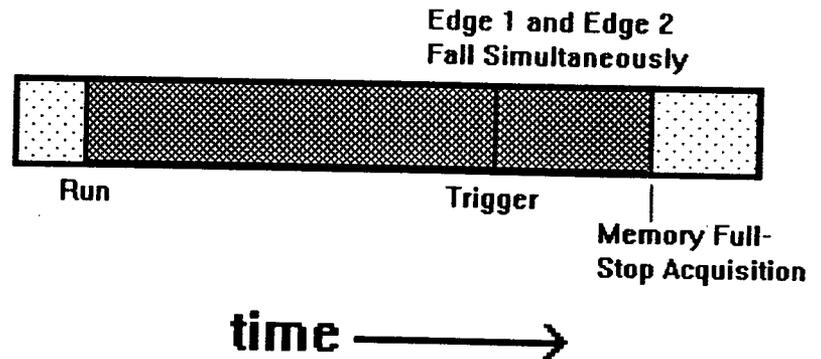
Only the glitch/edge terms are needed in this example. As shown above in the lower part of the menu, glitch/edge 1 is set to find any edge (either rising or falling) on bit 0, and glitch/edge 2 is set to find any edge on bit 1.

#### Possible Applications of This Triggering Model:

- \* To detect a handshake violation.
- \* To trigger on incorrect control signal generation from a PLD.

**Timing Trigger Application #8:  
Trigger when Edge 1 and Edge 2  
Are Asserted Simultaneously**

Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, looking for the specified edges.

**HP 16550A Sequencer Setup:**

In the following example, indicative of this triggering application, you trigger if a falling edge on bit 0 and a falling edge on bit 1 are detected at the same time. The Trigger menu showing the sequencer setup for this example is pictured on the next page. Note that a logical combination of terms is used to define the trigger condition. Terms can be logically combined in any of the term fields in any of the sequence levels.

100/500MHz LA E		Trigger 1		Print	Run
Timing Sequence Levels				Timer	Arming Control
1 TRIGGER on "glitch/edge1-glitch/e..." 1 time				1 2	Acquisition Control
				- -	Clear Trigger
◀Label▶	Leb1				
◀Terms▶	Binary				
Glitch/Edge 1	.....↓				
Glitch/Edge 2	.....↓				

#### HP 16550A Term Setup:

Only the glitch/edge terms are needed in this example. As shown above in the lower part of the menu, glitch/edge 1 is set to find a falling edge on bit 0, and glitch/edge 2 is set to find a falling edge on bit 1.

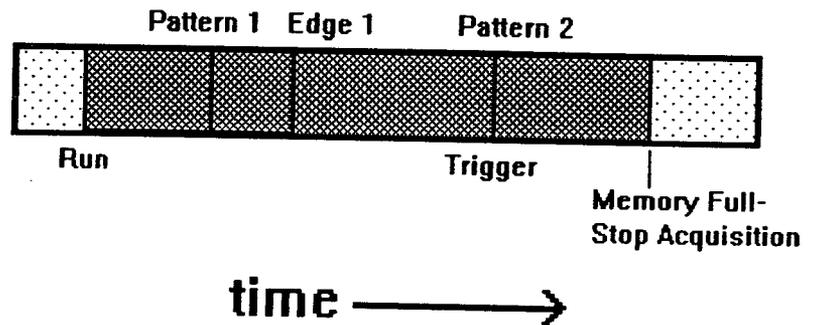
#### Possible Applications of This Triggering Model:

- \* To detect bus contention.
- \* To view system activity when two entities are trying to seize a digital communications channel at once.

---

**Timing Trigger Application #9:  
Trigger on Pattern 1 Followed  
by Edge 1 Followed by Pattern 2**

Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, looking for the specified patterns and edges.

**HP 16550A Sequencer Setup:**

In the following example, indicative of this triggering application, you will look for a 00 hex pattern on the 8 channels associated with Label 1, followed by a rising edge on bit 0, followed by a 02 hex pattern. The Trigger menu with the sequencer setup for this example is shown opposite. The analyzer is triggered only when all three events have been detected.

100/500MHz LA E		Trigger 1		Print	Run
Timing Sequence Levels				Timer	Arming Control
1	Find "PATTERN1" 1 time	1	--	Acquisition Control	
2	Then find "glitch/edge1" 1 time	2	--		
3	TRIGGER on "PATTERN2" 1 time	3	--	Clear Trigger	
Label	Label				
Terms	Hex				
PATTERN1	00				
PATTERN2	02				
c	XX				
d	XX				

### HP 16550A Term Setup:

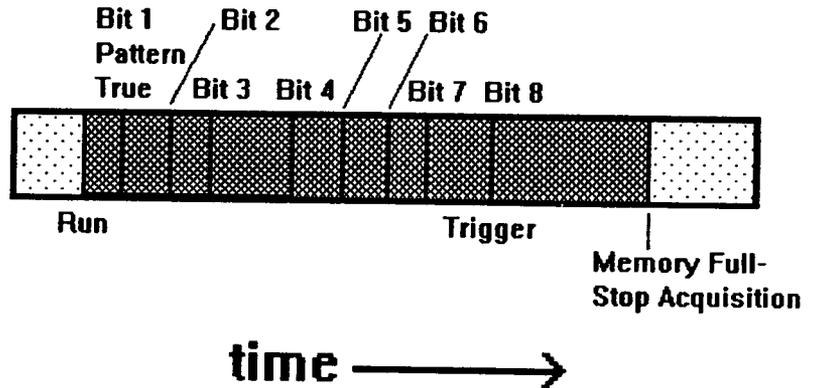
Two pattern terms and a glitch/edge term are needed in this example. As pictured above, the pattern term names are changed from 'a' to 'PATTERN1' and from 'b' to 'PATTERN2' in the bottom part of the screen so that the trigger sequence in the upper part of the screen will be clearer. The 00 hex pattern for the 'PATTERN1' term and the 02 hex pattern for the 'PATTERN2' term are also entered next to the term names. Not shown in the picture is the glitch/edge 1 term, which is set to a rising edge on the least significant bit of the 8 channels assigned to Label 1. The glitch/edge 1 term is found by activating the Terms field and rolling the knob as explained in the application note overview on page 1.

### Possible Applications for This Triggering Model:

- \* To verify a correct address bus, control signal, data bus sequence.
- \* To check whether a data packet was sent, a handshake signal followed, and an acknowledgement was returned.

**Timing Trigger Application #10:  
Trigger on 8-bit Serial Pattern**

Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, looking for each of the specified bits.

**HP 16550A Sequencer Setup:**

In this example you will find a sequence of 1's on the 8 channels associated with Label 1. The Trigger menu showing the sequencer setup for this example is included below.

100/500MHz LA E    Trigger 1    Print    Run

Timing Sequence Levels

6	Then find "BIT 6" 1 time Else on "¬BIT 6" go to level 1	Timer 1 2	Arming Control
7	Then find "BIT 7" 1 time Else on "¬BIT 7" go to level 1	--	Acquisition Control
8	TRIGGER on "BIT 8" 1 time Else on "¬BIT 8" go to level 1	--	Clear Trigger

Label: Lab1

Terms: Binary

BIT 1: XXXXXX1

BIT 2: XXXXXX1X

BIT 3: XXXXX1XX

BIT 4: XXXX1XXX

The entire trigger sequence uses 8 sequence levels (the picture shows levels 6 through 8), and it reads as follows:

- 1) Find "BIT 1" 1 time
- 2) Then find "BIT 2" 1 time  
Else on "not BIT 2" go to level 1
- 3) Then find "BIT 3" 1 time  
Else on "not BIT 3" go to level 1
- 4) Then find "BIT 4" 1 time  
Else on "not BIT 4" go to level 1
- 5) Then find "BIT 5" 1 time  
Else on "not BIT 5" go to level 1
- 6) Then find "BIT 6" 1 time  
Else on "not BIT 6" go to level 1
- 7) Then find "BIT 7" 1 time  
Else on "not BIT 7" go to level 1
- 8) TRIGGER on "BIT 8" 1 time  
Else on "not BIT 8" go to level 1

#### **HP 16550A Term Setup:**

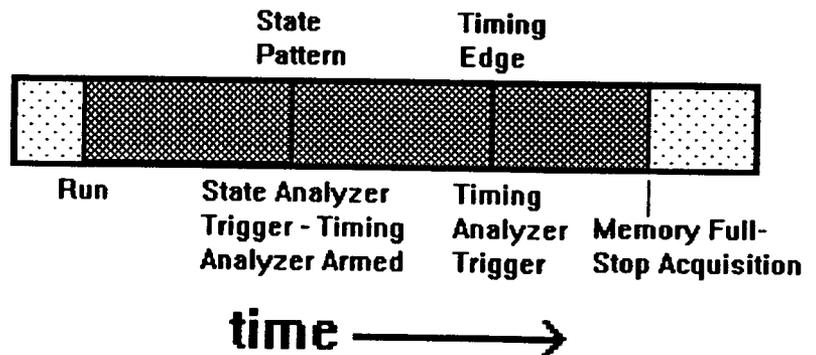
Eight pattern terms are needed in this example. As shown above, the pattern term names from 'a' through 'h' are changed to 'BIT 1' through 'BIT 8' in the bottom part of the screen so that the trigger sequence in the upper part of the screen will be clearer. The 1's patterns for the bit terms are also entered next to the term names. Not shown in the picture are the 'BIT 5' through 'BIT 8' terms. They are found by activating the **Terms** field and rolling the knob as explained in the application note overview on page 1.

#### **Possible Applications for This Triggering Model:**

- \* To view system activity after pattern transmission.
- \* To look at system status when an error pattern is detected.

**Timing Trigger Application #11:  
Trigger Timing Analyzer with  
State Analyzer**

**Events Timeline:**



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, looking for the state pattern and timing edge.

**HP 16550A Sequencer Setup:**

In the following example, indicative of this triggering application, you use both the timing analyzer (Machine 1) Trigger menu and the state analyzer (Machine 2) Trigger menu. These menus, showing the sequencer setups for this example, are pictured on the next page. The state sequencer, in the Trigger 2 menu, is set up to trigger when it detects a F5 hex pattern on the 8 bits associated with the ST\_LAB label. It is also configured to arm the timing analyzer when it triggers. This cross-arming is set up in the Arming Control menu, accessible by touching the Arming Control field at the right edge of the Trigger 2 menu. Also, to time-correlate state and timing analyzers, state time tags are turned on by touching the Count field at the right edge of the Trigger 2 menu and changing Off to Time. The timing sequencer, in the Trigger 1 menu, is triggered by the arm signal from the state analyzer ANDed with a rising edge on bit 0 of the 8 channels labeled TI\_LAB.

100/500MHz LA E		Trigger 1		Print	Run
Timing Sequence Levels				Timer 1 2 --	Arming Control
1	TRIGGER on "arm-glitch/edge1" 1 time			Acquisition Control	Clear Trigger
Label	TI_LAB				
Terms	Binary				
Glitch/Edge 1	.....↑				
Glitch/Edge 2	.....				

100/500MHz LA E		Trigger 2		Print	Run
State Sequence Levels				Arming Control	Acquisition Control
1	While storing "anystate" TRIGGER on "PATTERN" 1 time			Count Time	
2	Store "anystate"			Clear Trigger	
Label	ST_LAB				
Terms	Hex				
PATTERN	F5				
b	XX				
c	XX				
d	XX				

#### HP 16550A Term Setup:

In this example, a pattern term for the state analyzer and a glitch/edge term for the timing analyzer are needed. As shown above, the pattern term name in the Trigger 2 menu is changed from 'a' to 'PATTERN' in the bottom part of the screen so that the trigger sequence in the upper part of the screen will be clearer. The F5 hex pattern is also entered next to the term name. In the lower part of the Trigger 1 menu the glitch/edge 1 term is set to detect a rising edge on bit 0.

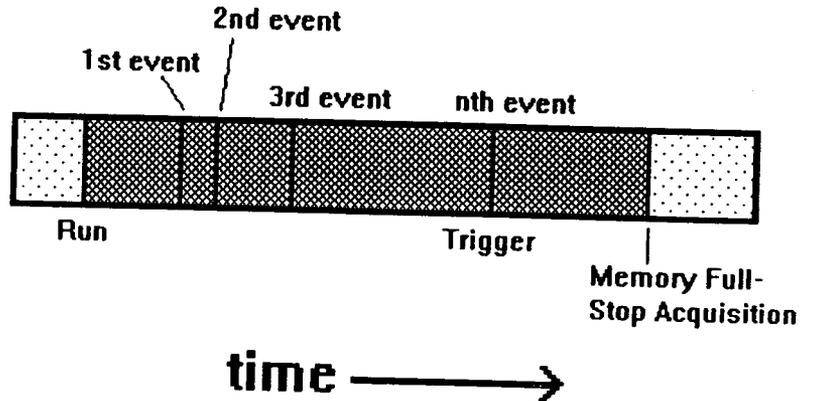
#### Possible Applications for This Triggering Model:

- \* To look at control and status signals with finer resolution than once per bus cycle during execution of a specific subroutine.
- \* To make timing measurements on external cache when code that continually accesses the cache is running.

---

**State Trigger Application #1:  
Trigger on nth Event  
Occurrence**

Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, storing qualified states and looking for the specified events.

**HP 16550A Sequencer Setup:**

In the following example, indicative of this triggering application, you will use  $n=50$  and trigger on the 50th occurrence of 15 hex on the 8 channels associated with Label 1. The Trigger menu showing the sequencer setup for this example is pictured opposite.

100/500MHz LA E		Trigger 1		Print	Run
State Sequence Levels				Timer	Arming Control
1	While storing "anystate"		TRIGGER on "EVENT" 50 times	1 2	Acquisition Control
2	Store "anystate"			--	Count Off
					Clear Trigger
Label	Label				
Terms	Hex				
EVENT	15				
b	XX				
c	XX				
d	XX				

**HP 16550A Term Setup:**

In this example you need only 1 pattern term for the state analyzer. As shown above, the pattern term is labeled with the name 'EVENT'. The 15 hex pattern is also entered next to the term name.

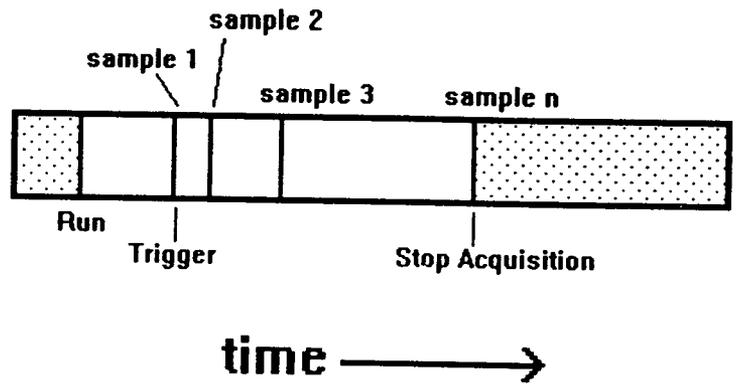
**Possible Applications for This Triggering Model:**

- \* To find the 50th occurrence of a DSP subroutine.
- \* To trigger on the 3rd write to a specific memory address.

---

**State Trigger Application #2:  
Store n Samples**

**Events Timeline:**



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The white areas between samples depicts the portion of time where the logic analyzer's sequencer is running, but no states are being stored except the sample states.

**HP 16550A Sequencer Setup:**

In this example you will use  $n=20$  and store 20 occurrences of EF hex on the 8 channels associated with Label 1. The Trigger menu showing the sequencer setup for this example is included on the next page.

100/500MHz LA E		Trigger 1		Print	Run
State Sequence Levels				Timer	Arming Control
1	While storing "SAMPLE" TRIGGER on "SAMPLE" 20 times			1 2	Acquisition Control
↓				--	Count Off
2	Store "no state"			--	Clear Trigger
Label	Labi				
Terms	Hex				
SAMPLE	EF				
b	XX				
c	XX				
d	XX				

### HP 16550A Term Setup:

In this example only 1 pattern term for the state analyzer is needed. In this case the pattern term is labeled with the name 'SAMPLE'. The EF hex pattern is entered next to the term name.

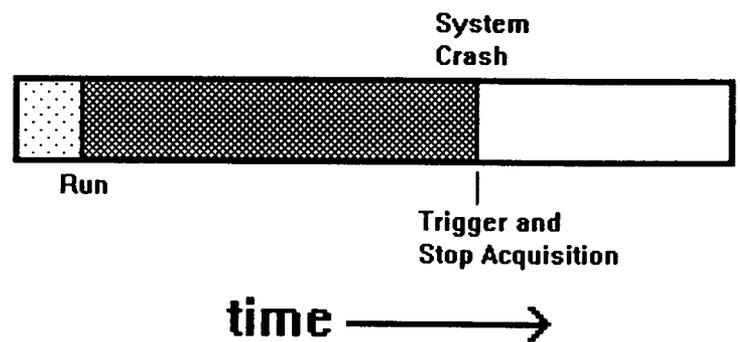
### Possible Applications for This Triggering Model:

- \* To view the first 200 reads and writes to a FIFO.
- \* To look at 75 pushes onto the stack.

---

### State Trigger Application #3: Trigger on System Crash

#### Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, storing qualified states.

#### HP 16550A Sequencer Setup:

In this application you trigger the state analyzer on 'no state', or in other words, no system activity. After the system crash there is no further data to store, so after the trigger you will also store 'no state'. Storing 'no state' after the trigger also signals the analyzer to place the trigger at the end of acquisition memory. The Trigger menu showing the sequencer setup for this application is pictured on the next page. In this specific triggering scenario the analyzer simply stops trying to acquire data if it sees no state clock, and by touching the Stop field you can see all acquired data leading up to the crash.

100/500MHz LA E		Trigger 1		Print	Run
State Sequence Levels				Timer	Arming Control
1	While storing "anystate"		1	-	Acquisition Control
	TRIGGER on "no state" 1 time		2	-	Count Off
2	Store "no state"			-	Clear Trigger
Label	Label				
Terms	Hex				
a	XX				
b	XX				
c	XX				
d	XX				

**HP 16550A Term Setup:**

In this application, no resource terms are needed. The global terms 'anystate' and 'no state' are used. All resource terms shown in the lower part of the screen are left in their default state.

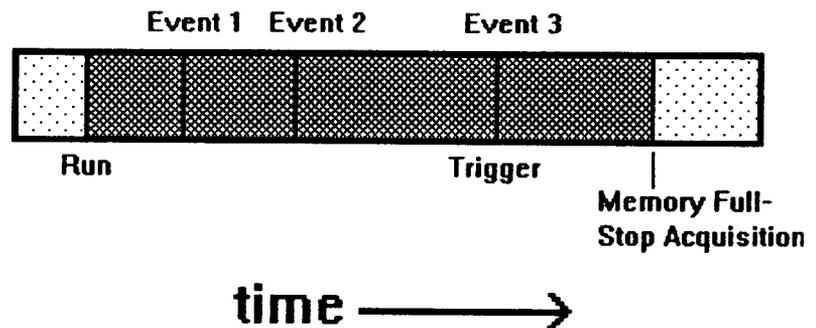
**Possible Applications for This Triggering Model:**

- \* To store and display all activity leading up to a system crash.
- \* To run the logic analyzer indefinitely until Stop is pressed, so that the user can observe system activity at his/her own discretion.

---

**State Trigger Application #4:  
Trigger on a Series of Non-  
Consecutive Events**

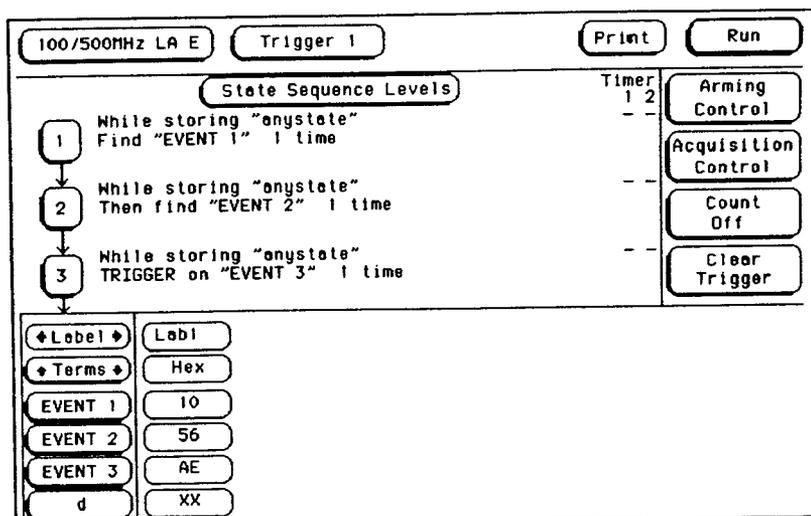
**Events Timeline:**



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, storing qualified states and looking for the specified events.

**HP 16550A Sequencer Setup:**

In the following example, indicative of this triggering application, you will tell the logic analyzer to look for a 10 hex, followed by a 56 hex, followed by an AE hex. The sequencer will trigger upon finding the 3rd event. The Trigger menu showing the sequencer setup for this example is pictured on the next page. You want to trigger in this case when the events occur in order relative to one another but regardless of the time or states that have elapsed between them.



This entire trigger sequence uses 4 sequence levels (the picture shows levels 1 through 3), and it reads as follows:

- 1) While storing "anystate"  
Find "EVENT 1" 1 time
- 2) While storing "anystate"  
Then find "EVENT 2" 1 time
- 3) While storing "anystate"  
TRIGGER on "EVENT 3" 1 time
- 4) Store "anystate"

This trigger sequence assumes that Event 1 does not ever occur on the next state directly after Event 2.

#### HP 16550A Term Setup:

The terms used in this example are pattern terms a, b, and c, which have been renamed 'EVENT 1', 'EVENT 2', and 'EVENT 3', as shown in the lower left part of the screen. Here, three 8-bit hex values (10 hex, 56 hex, and AE hex) have been entered as the respective events.

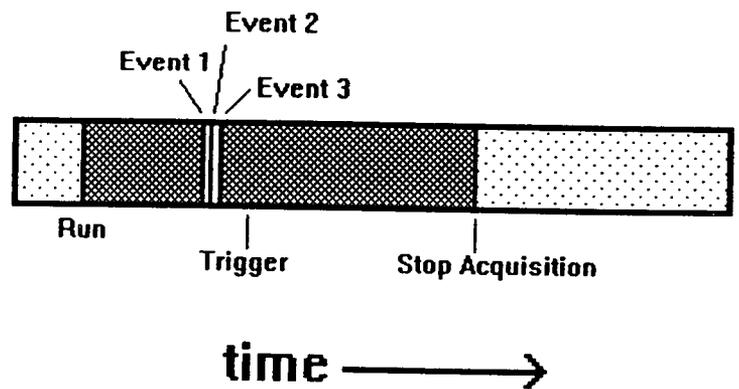
#### Possible Applications for This Triggering Model:

- \* To trigger on the occurrence of a calculation subroutine after 2 initialization subroutines have executed.
- \* To trigger on the access to an I/O port after its 2 I/O registers have been set.

---

## State Trigger Application #5: Trigger on a Series of Consecutive Events

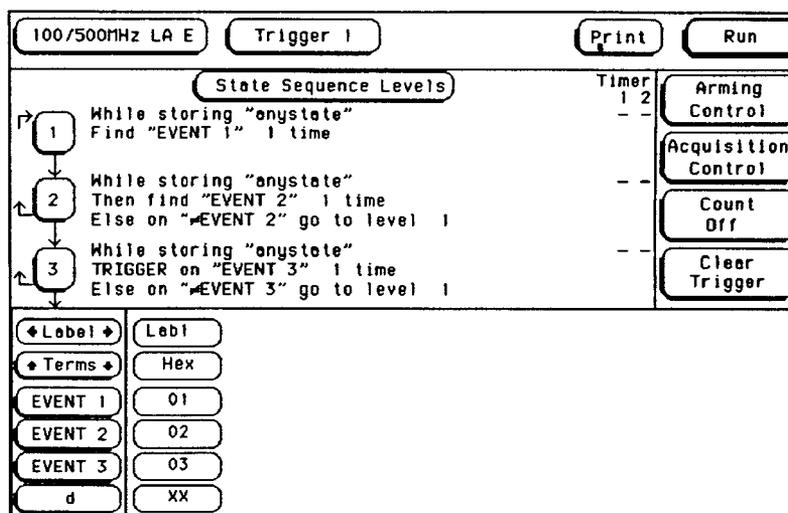
Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, storing qualified states and looking for the specified events. The white area between events is included to show that in this case all specified events are to occur consecutively, with no states in between.

### HP 16550A Sequencer Setup:

In the following example, indicative of this triggering application, you will tell the logic analyzer to look for a 01 hex, followed immediately by a 02 hex, followed immediately by an 03 hex. The sequencer will trigger upon finding the 3rd event. The Trigger menu showing the sequencer setup for this example is pictured opposite. You want to trigger in this case when the events occur in order, with each event being the very next state after the previous event. Notice the arrows coming out of levels 2 and 3 and going into level 1. They indicate that the sequencer will be reset to level 1 if 02 hex is not found immediately after 01 hex, or if 03 hex is not found immediately after 02 hex.



The entire trigger sequence uses 4 sequence levels (the picture shows levels 1 through 3), and it reads as follows:

- 1) While storing "anystate"  
Find "EVENT 1" 1 time
- 2) While storing "anystate"  
Then find "EVENT 2" 1 time  
Else on "not EVENT 2" go to level 1
- 3) While storing "anystate"  
TRIGGER on "EVENT 3" 1 time  
Else on "not EVENT 3" go to level 1
- 4) Store "anystate"

#### HP 16550A Term Setup:

The terms used in this example are pattern terms a, b, and c, which have been renamed 'EVENT 1', 'EVENT 2', and 'EVENT 3', as shown in the lower left part of the screen. Here, three 8-bit hex values (01 hex, 02 hex, and 03 hex) have been entered as the state values wanted for the respective events.

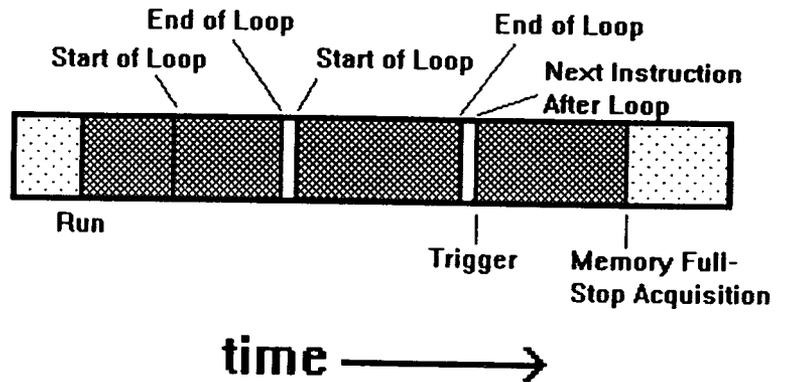
#### Possible Applications for This Triggering Model:

- \* To trigger on the occurrence of a subroutine only when it has been called from a specific branch of the main program.
- \* To look for data writes to 4 consecutive memory locations with no reads in-between.

---

## State Trigger Application #6: Trigger upon Exiting a Loop

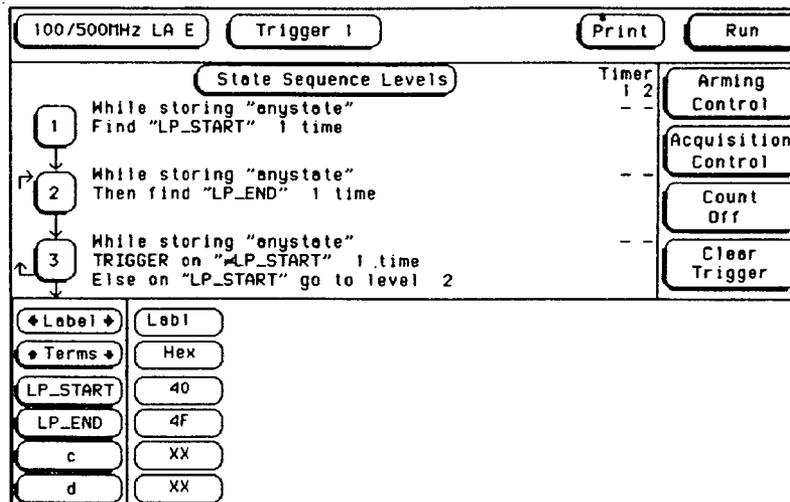
### Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, storing qualified states and looking for the specified events. The white area between events is included to show that between these specified events no states should occur.

### HP 16550A Sequencer Setup:

In the following example, indicative of this triggering application, the start of the loop is at 40 hex, and the end of the loop is at 4F hex. You want to trigger on the next instruction after the loop completes. The Trigger menu showing the sequencer setup for this example is pictured on the next page. In the first level the start of the loop is found, and in the second level the end of the loop is found. The third level is the decision level. If the next instruction is the loop again, the sequencer also loops back to level 2, as shown by the arrow coming out of level 3 and going into level 2. However, if the next instruction is not the loop, the analyzer is triggered.



The complete trigger sequence uses 4 sequence levels (the picture shows levels 1 through 3), and it reads:

- 1) While storing "anystate"  
Find "LP\_START" 1 time
- 2) While storing "anystate"  
Then find "LP\_END" 1 time
- 3) While storing "anystate"  
TRIGGER on "not LP\_START" 1 time  
Else on "LP\_START" go to level 2
- 4) Store "anystate"

#### HP 16550A Term Setup:

The terms used in this example are pattern terms a and b, which have been renamed 'LP\_START' and 'LP\_END', as shown in the lower left part of the screen. The 40 hex (beginning of the loop) value for the 8 channels of Label 1 and the 4F hex (end of the loop) value are both entered here to the right of their respective pattern terms.

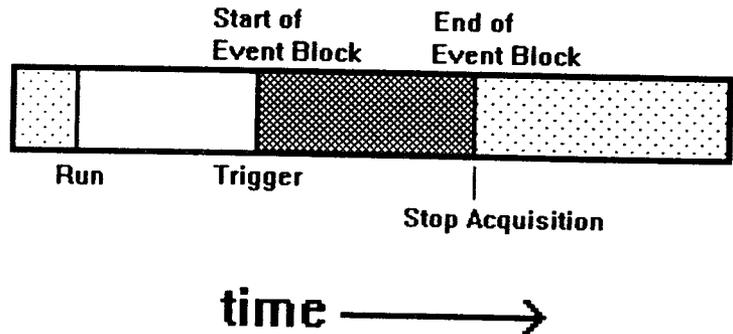
#### Possible Applications for This Triggering Model:

- \* To filter out a background monitor loop that runs until a control key is pressed.
- \* To verify that all stacks and registers are restored correctly before exiting a subroutine.

---

**State Trigger Application #7:  
Store Block of Events, Time  
Block of Events**

Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is storing the event block. The white area shows the segment of time where the sequencer is running, but no states are being stored.

**HP 16550A Sequencer Setup:**

In the example shown here, the start of the block is at A0 hex, and the end of the block is at BF hex. The Trigger menu showing the sequencer setup for this example is included on the next page. Level 1 is used to find and trigger the analyzer when the block, or range, is entered. All states within the range are then stored in level 2 until the block is exited (states are out of range). Because you are only interested in a specific block, 'no state' is stored in level 3 after the block is exited. When you store 'no state' in the last sequence level the analyzer stops acquisition and displays the data stored when the sequencer reaches this level. Also, to time the block of events, time tags are turned on as indicated in the Count Time field on the right side of the screen.

100/500MHz LA E		Trigger 1		Print	Run
State Sequence Levels				Timer	Arming Control
1	While storing "no state"	TRIGGER on "in_range1" 1 time		1	Acquisition Control
2	While storing "in_range1"	Then find "out_range1" 1 time		2	Count Time
3	Store "no state"				Clear Trigger
Label	Label				
Terms	Hex				
Range1					
upper	BF				
lower	A0				

#### HP 16550A Term Setup:

In this example, a range term is used to delimit the block you are interested in. The lower part of the menu above shows the Range 1 term used in the sequencer. The lower pattern field is set to the start of the block and the upper pattern field to the end of the block.

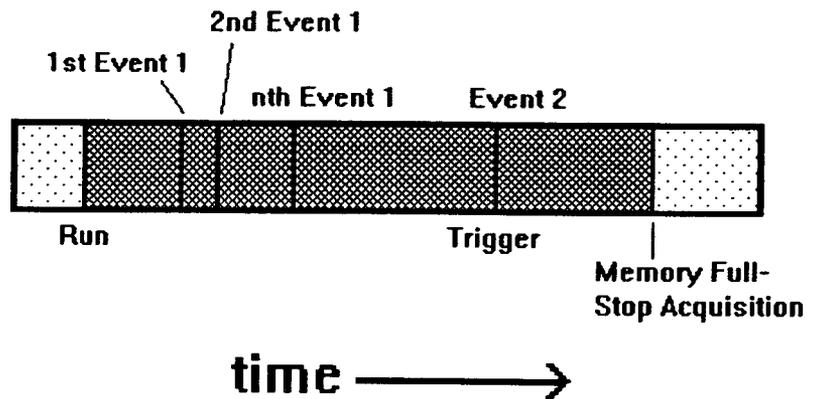
#### Possible Applications for This Triggering Model:

- \* To store and time the execution of a memory management subroutine.
- \* To store and time an access to a disk drive.

---

**State Trigger Application #8:  
Trigger on Event 2 Only After  
Event 1 Has Been Executed n  
Times**

Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, storing qualified states and looking for the specified events.

**HP 16550A Sequencer Setup:**

In the following example, indicative of this triggering application, you will use  $n=10$  and find the 10th occurrence of 67 hex on the 8 channels associated with Label 1, followed by a 77 hex on the channels. The Trigger menu with the sequencer setup for this example is shown opposite. In the first level the sequencer is set to find the 10 occurrences of 67 hex ('EVENT 1'). Once the second level is entered, the sequencer triggers on the first 77 hex ('EVENT 2') that it sees.

100/500MHz LA E		Trigger 1		Print	Run
State Sequence Levels				Timer	Arming Control
1	While storing "anystate"		Find "EVENT 1" 10 times	--	Acquisition Control
2	While storing "anystate"		TRIGGER on "EVENT 2" 1 time	--	Count Off
3	Store "anystate"			--	Clear Trigger
Label	Lobl				
Terms	Hex				
EVENT 1	67				
EVENT 2	77				
c	XX				
d	XX				

### HP 16550A Term Setup:

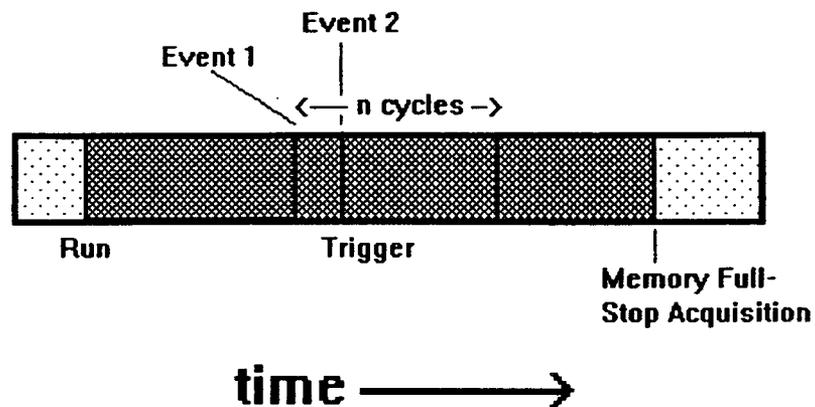
In this example, two pattern terms are needed. The a and b terms, which have been renamed 'EVENT 1' and 'EVENT 2', are used as shown in the lower left part of the screen. The 67 hex and 77 hex are entered next to these pattern terms.

### Possible Applications for This Triggering Model:

- \* To trigger on the 1st write after the 20th read.
- \* To trigger on an error code only after error recovery has been attempted twice.

**State Trigger Application #9:  
Trigger if n Cycles Do Not  
Occur between Event 1 and  
Event 2**

Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, storing qualified states and looking for the specified events and cycles.

**HP 16550A Sequencer Setup:**

In the following example, indicative of this triggering application, you will use  $n=20$  and trigger the analyzer if 20 cycles do not occur between 00 hex ('EVENT 1') and 05 hex ('EVENT 2'). The Trigger menu with the sequencer setup for this example is shown on the next page. In level 1 the sequencer is set to find 00 hex ('EVENT 1'). In level 2 you look for the 20 cycles you want to occur after Event 1. If 05 hex ('EVENT 2') is detected the sequencer is told to jump to level 4. Level 4 is the trigger level; you must jump to level 4 because you cannot tell the sequencer to trigger off of an 'Else on' event. The trigger condition has already been found if level 4 is reached, however, so you just direct the sequencer to trigger on 'anystate'. Because you cannot loop back to another level from a 'Then find' event, level 3 is used to loop back to level 1 and reset the sequencer if the 20 cycles are found. The arrows coming out of and into the sequence level numbers indicate which way a given branch condition will cause the sequencer to jump.

100/500MHz LA E		Trigger 1		Print	Run
State Sequence Levels				Timer	1 2
1	While storing "anystate" Find "EVENT 1" 1 time				Arming Control
2	While storing "anystate" Then find "EVENT 2" 20 times Else on "EVENT 2" go to level 4				Acquisition Control
3	While storing "anystate" Then find "no state" 1 time Else on "anystate" go to level 1				Count Off
					Clear Trigger
Label	Lab1				
Terms	Hex				
EVENT 1	00				
EVENT 2	05				
c	XX				
d	XX				

The entire trigger sequence for this scenario cannot fit on the screen shown above; it takes 5 levels. The full sequence reads as follows:

- 1) While storing "anystate"  
Find "EVENT 1" 1 time
- 2) While storing "anystate"  
Then find "not EVENT 2" 20 times  
Else on "EVENT 2" go to level 4
- 3) While storing "anystate"  
Then find "no state" 1 time  
Else on "anystate" go to level 1
- 4) While storing "anystate"  
TRIGGER on "anystate"
- 5) Store "anystate"

#### HP 16550A Term Setup:

In this example, two pattern terms are needed. Terms a and b, which have been renamed 'EVENT 1' and 'EVENT 2', are used as shown in the lower left part of the screen. The 00 hex and 05 hex are entered next to these pattern terms.

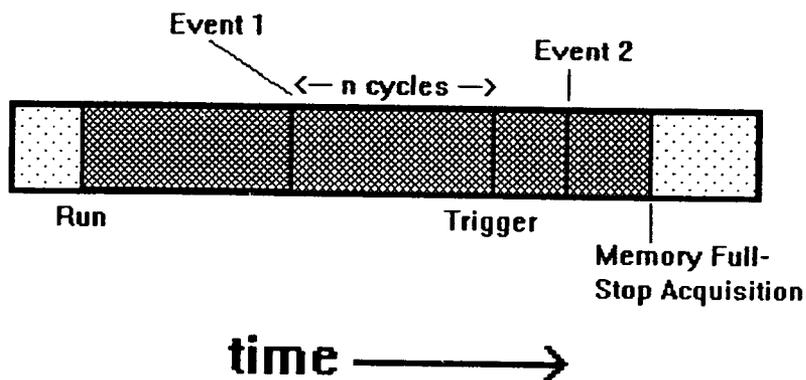
#### Possible Applications for This Triggering Model:

- \* To detect when a subroutine is exited prematurely from any of a number of exit points.
- \* To find a protocol violation in sending control messages to a peripheral.

---

**State Trigger Application #10:  
Trigger if  $\geq n$  Cycles Occur  
between Event 1 and Event 2**

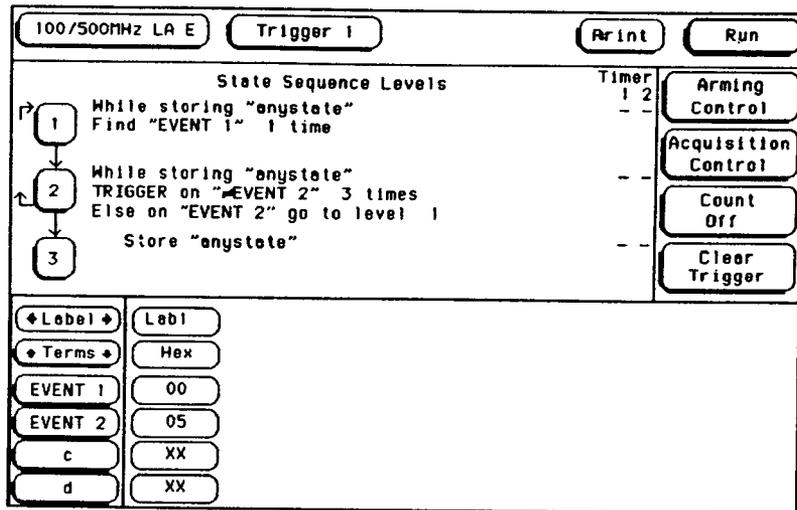
Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, storing qualified states and looking for the specified events and cycles.

**HP 16550A Sequencer Setup:**

In the following example, indicative of this triggering application, you will use  $n=3$  and trigger the analyzer if more than 3 cycles occur between 00 hex ('EVENT 1') and 05 hex ('EVENT 2'). The Trigger menu showing the sequencer setup for this example is included on the next page. In level 1 the sequencer is set to find 00 hex ('EVENT 1'). In level 2 the analyzer is triggered on the 3rd cycle that is not 05 hex ('EVENT 2'), but if the 05 hex is found before the 3rd cycle the sequencer is directed to reset to level 1. The arrows coming out of level 2 and going into level 1 indicate the reset branching that will occur if no violation is found.



#### HP 16550A Term Setup:

In this example, two pattern terms are needed. Terms a and b, which have been renamed 'EVENT 1' and 'EVENT 2', are used as shown in the lower left part of the screen. The 00 hex and 05 hex are entered next to these pattern terms.

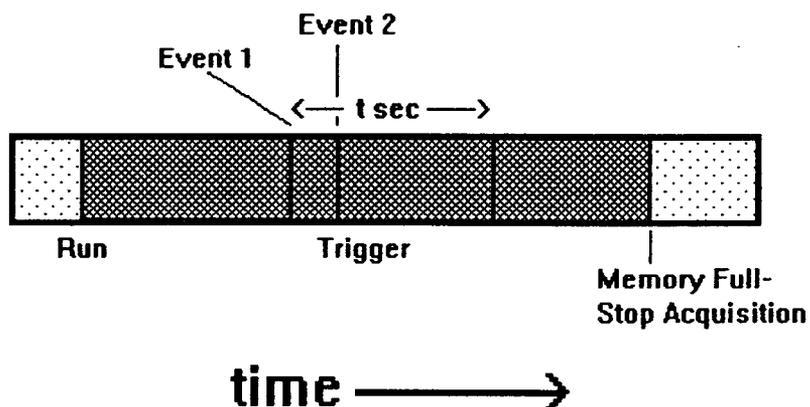
#### Possible Applications for This Triggering Model:

- \* To trigger when secondary cache must be accessed between 2 consecutive memory reads, producing extra cycles.
- \* To detect when an interrupt routine is executing for an excessive number of cycles.

---

**State Trigger Application #11:  
Trigger if t Seconds Do Not  
Occur between Event 1 and  
Event 2**

**Events Timeline:**



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, storing qualified states and looking for the specified events and time intervals.

**HP 16550A Sequencer Setup:**

In the following example, indicative of this triggering application, you will find a pattern of 00 hex on the 8 channels associated with Label 1, then trigger the analyzer if the pattern 0A hex occurs before 1 microsecond has elapsed. The Trigger menu with the sequencer setup for this example is shown opposite. As shown on the right-hand side of the sequencer field, timer 1 is started as level 2 is entered, right after 00 hex is found. The arrows coming out of level 2 and going into level 1 indicate that the sequencer is directed to reset to level 1 if 1 microsecond has passed without Event 2 being seen.

100/500MHz LA E		Trigger 1		Print	Run
State Sequence Levels				Timer	Arming Control
1	While storing "anystate"	Find "EVENT 1" 1 time	1	2	Acquisition Control
2	While storing "anystate"	TRIGGER on "EVENT 2" 1 time	S	-	Count Off
		Else on "timer1>1.000us" go to level 1			Clear Trigger
3	Store "anystate"				
Label	Lab1				
Terms	Hex				
EVENT 1	00				
EVENT 2	0A				
c	XX				
d	XX				

### HP 16550A Term Setup:

In this example, two pattern terms and one timer term are needed. Terms a and b, which have been renamed 'EVENT 1' and 'EVENT 2', are used as shown in the lower left part of the screen. The 00 hex and 0A hex are entered next to these pattern terms. Not shown in the picture is the timer 1 term, which is set to 1 microsecond. The timer 1 term is found by activating the Terms field and rolling the knob as explained in the application note overview on page 1.

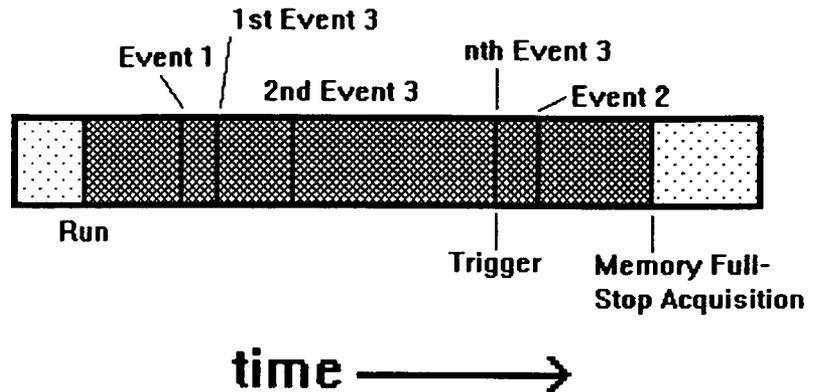
### Possible Applications for This Triggering Model:

- \* To trigger when a wait loop executed between 2 instructions does not produce the desired delay.
- \* To trigger on a data overrun.

---

**State Trigger Application #12:  
Trigger if Event 3 Occurs  $\geq n$   
Times between Events 1 and 2**

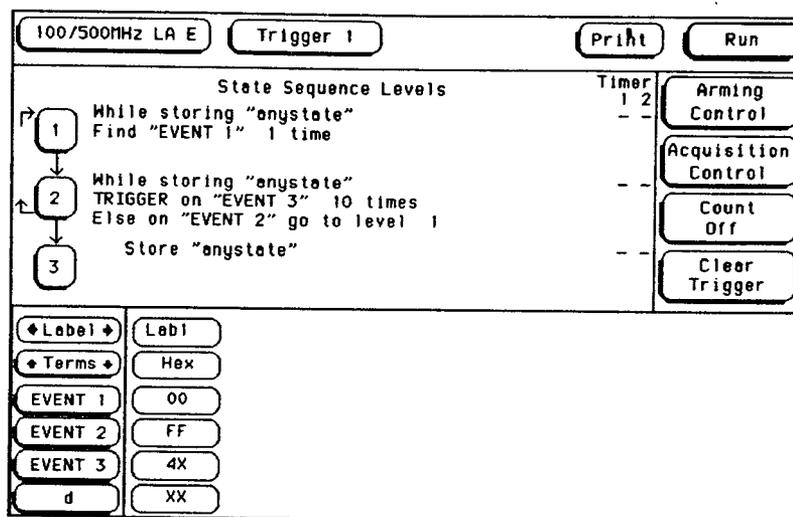
Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, storing qualified states and looking for the specified events.

**HP 16550A Sequencer Setup:**

In the example shown here, you will use  $n=10$  and trigger if 10 or more 4X hex states ('EVENT 3' - actually any of 16 possible values because of the don't care term for the second byte) occur between 00 hex ('EVENT 1') and FF hex ('EVENT 2'). The Trigger menu showing the sequencer setup for this example is pictured on the next page. The arrows going out of level 2 and coming into level 1 show how the sequencer will reset in this example if Event 2 is found before the 10 occurrences of Event 3.



### HP 16550A Term Setup:

The terms used in this example are pattern terms a, b, and c, which have been renamed 'EVENT 1', 'EVENT 2', and 'EVENT 3', as shown in the lower left part of the screen. Here, three 8-bit hex values (00 hex, FF hex, and 4X hex) have been entered as the state values wanted for the respective events.

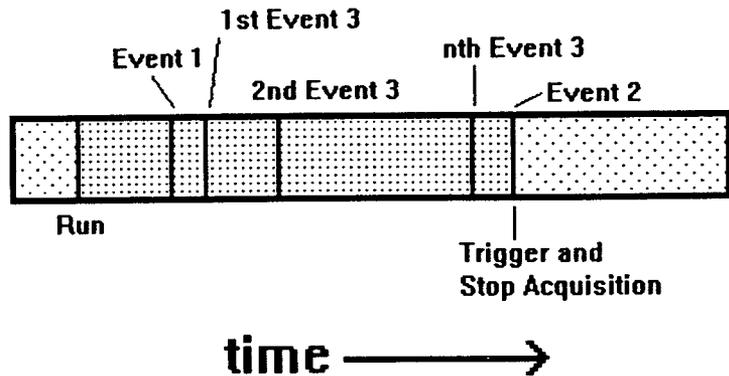
### Possible Applications for This Triggering Model:

- \* To examine code execution when an ASIC issues a data request interrupt more than 5 times during the execution of a time-critical subroutine.
- \* To trigger if a loop is executed more than 10 times between 2 non-consecutive routines.

---

**State Trigger Application #13:  
Count Occurrences of Event 3  
between Events 1 and 2**

**Events Timeline:**



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The heavier dotted area depicts the portion of time where the logic analyzer's sequencer is running, storing qualified states, looking for the specified events, and counting event occurrences.

**HP 16550A Sequencer Setup:**

In the example detailed here you will count how many 4X hex states ('EVENT 3' - actually any of 16 possible values because of the don't care term for the second byte) occur between 00 hex ('EVENT 1') and FF hex ('EVENT 2'). The Trigger menu showing the sequencer setup for this example is pictured on the next page. The reason that 'EVENT 3' does not appear in the trigger sequence is that it is not being stored or used as a 'Find' or 'TRIGGER' term; instead the 'EVENT 3' term is just counted. The counting function is set up in the Count menu, which can be accessed by touching the Count field on the right side of the Trigger menu. When Count States is designated as shown in this menu you can designate any term or combination of terms to be counted, like 'EVENT 3'. The count will be displayed under the Count label in the Listing menu. In the example, since the sequencer is set up to store 'no state' after the trigger, the state listing will contain only 2 states ('EVENT 1' and 'EVENT 2') with a count next to 'EVENT 2' showing the number of occurrences of 'EVENT 3'.

100/500MHz LA E      Trigger 1      Print      Run

State Sequence Levels		Timer	
		1	2
1	While storing "no state" Find "EVENT 1" 1 time	-	-
2	While storing "no state" TRIGGER on "EVENT 2" 1 time	-	-
3	Store "no state"	-	-

Arming Control
Acquisition Control
Count States
Clear Trigger

Label	Lab1
Terms	Hex
EVENT 1	00
EVENT 2	FF
EVENT 3	4X
d	XX

### HP 1650A Term Setup:

The terms used in this scenario are pattern terms a, b, and c, which have been renamed 'EVENT 1', 'EVENT 2', and 'EVENT 3', as shown in the lower left part of the screen. Here, three 8-bit hex values (00 hex, FF hex, and 4X hex) have been entered as the state values wanted for the respective events.

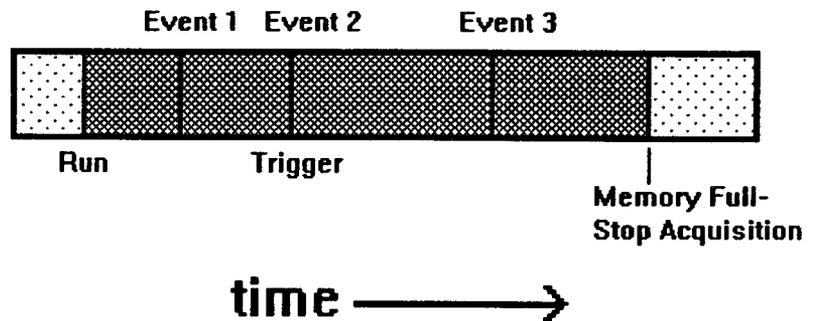
### Possible Applications for This Triggering Model:

- \* To verify that a memory refresh routine is executing the number of times expected.
- \* To count the number of memory write cycles within a segment of code.

---

**State Trigger Application #14:  
Trigger when Event 2 Follows  
Event 1, But Only if Event 3  
Does Not Occur In-between**

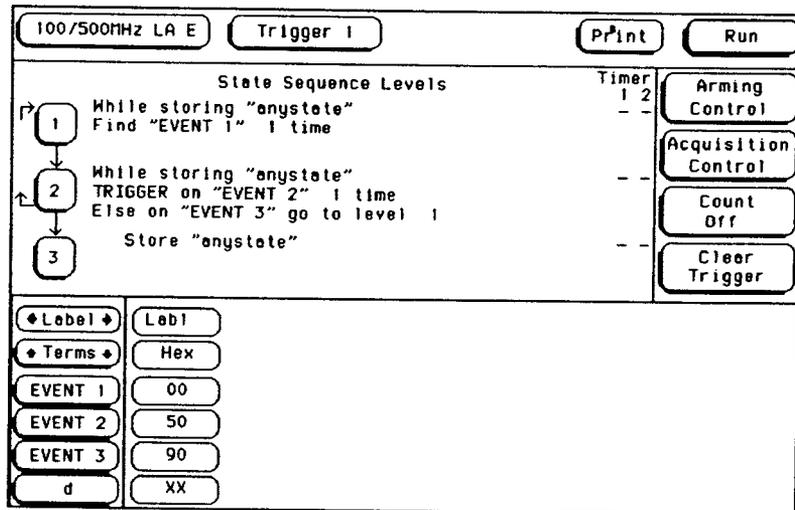
Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, storing qualified states and looking for the specified events.

**HP 16550A Sequencer Setup:**

In the following example, indicative of this triggering application, you want to first find Event 1 (00 hex), then trigger the analyzer if Event 2 (50 hex) is found without Event 3 (90 hex) occurring in-between. The Trigger menu showing the sequencer setup for this example is pictured on the next page. The arrows coming out of level 2 and going into level 1 indicate that the sequencer will be reset if no violation is detected (Event 3 occurs between Event 1 and Event 2).



### HP 16550A Term Setup:

The terms used in this scenario are pattern terms a, b, and c, which have been renamed 'EVENT 1', 'EVENT 2', and 'EVENT 3', as shown in the lower left part of the screen. Here, three 8-bit hex values (00 hex, 50 hex, and 90 hex) have been entered as the state values wanted for the respective events.

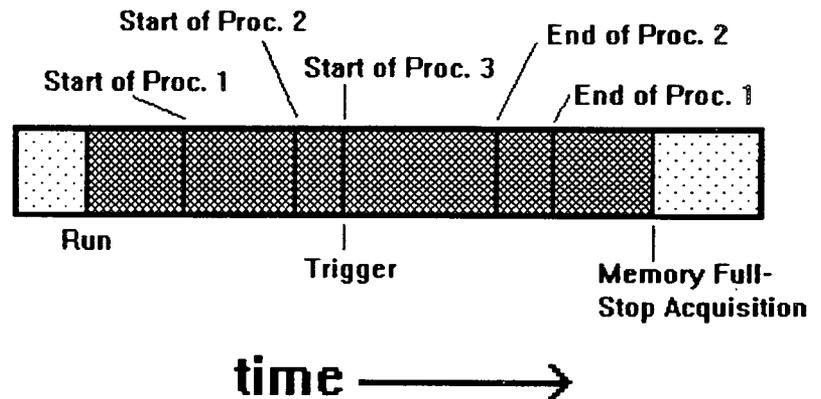
### Possible Applications for This Triggering Model:

- \* To trigger when routine 2 follows routine 1, but only if an interrupt does not occur between them.
- \* To verify that data transmit follows data receive without an error occurring as a result of the transmission.

---

**State Trigger Application #15:  
Trigger on Execution of a  
Specific Sequence of Program  
Flow**

Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, storing qualified states and looking for the specified procedures.

**HP 16550A Sequencer Setup:**

In the example shown here you want to trigger the state analyzer when Procedure 3 is started while Procedure 2 and Procedure 1 are still being executed. In other words, you want to look for the start of Procedure 1, then look for the start of Procedure 2 provided that Procedure 1 is still accessed. Then you will trigger when Procedure 3 is found provided that Procedure 2 is still accessed. The sequencer setup that accomplishes this triggering is shown in the Trigger menu on the next page. Here the start of each procedure is a pattern term, or event. The end of procedures 1 and 2 are also events that you want the sequencer to search for. For the example, the start of Procedure 1 ('P1\_START') is at 00 hex, the end of Procedure 1 ('P1\_END') is at FF hex, the start of Procedure 2 ('P2\_START') is at 10 hex, the end of Procedure 2 ('P2\_END') is at 6F hex, and the start of Procedure 3 ('P3\_START') is at 25 hex.

In level 1 you program the sequencer to find the start of Procedure 1. In level 2 the sequencer looks for the start of Procedure 2 and the end of Procedure 1. If it finds the start of Procedure 2 first, it proceeds to level 3. If it finds the end of Procedure 1 first, it loops back to level 1, as indicated by the arrows coming out of level 2 and going into level 1. In level 3 the sequencer looks for the start of Procedure 3 and the end of Procedure 2. If it finds the start of Procedure 3 first, it triggers the

analyzer. If it finds the end of Procedure 2 first (assuming that the end of Procedure 2 always occurs before the end of Procedure 1), it loops back to level 2, again indicated by the arrows.

100/500MHz LA E		Trigger 1	Print	Run
State Sequence Levels			Timer	Arming Control
1	While storing "anystate" Find "P1_START" 1 time		1	Acquisition Control
2	While storing "anystate" Then find "P2_START" 1 time Else on "P1_END" go to level 1		2	Count Off
3	While storing "anystate" TRIGGER on "P3_START" 1 time Else on "P2_END" go to level 2			Clear Trigger
◀Label▶	Label			
◆Terms◆	Hex			
P1_START	00			
P1_END	FF			
P2_START	10			
P2_END	6F			

The entire trigger sequence for this scenario cannot fit on the screen shown above; it takes 4 levels. The full sequence reads as follows:

- 1) While storing "anystate"  
Find "P1\_START" 1 time
- 2) While storing "anystate"  
Then find "P2\_START" 1 time  
Else on "P1\_END" go to level 1
- 3) While storing "anystate"  
TRIGGER on "P3\_START" 1 time  
Else on "P2\_END" go to level 2
- 4) Store "anystate"

#### HP 16550A Term Setup:

In this example, 5 pattern terms are needed. The a, b, c, d, and e terms are used, and renamed 'P1\_START', 'P1\_END', 'P2\_START', 'P2\_END', and 'P3\_START' respectively, as shown in the lower part of the menu above. (The 'P3\_START' term cannot be seen in the picture, but it is found by activating the Terms field and rolling the knob as explained in the application note overview on page 1.) The 00 hex, FF hex, 10 hex, 6F hex, and 25 hex patterns are also entered next to their term names.

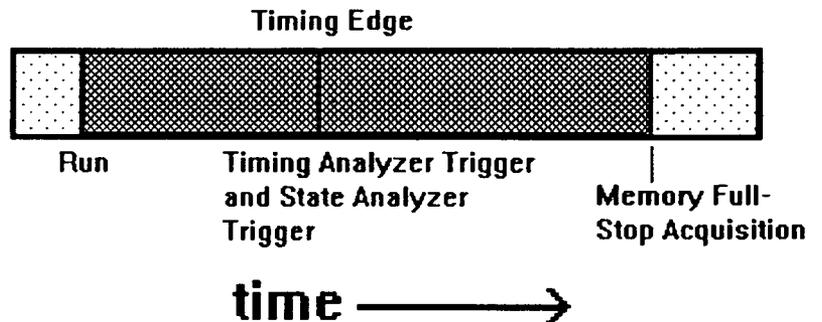
#### Possible Applications for This Triggering Model:

- \* To trigger when procedure 3 displays an error message only when called by procedure 2, which is called by procedure 1.
- \* To trigger on the 3rd nested occurrence of a recursive subroutine.

---

**State Trigger Application #16:  
Trigger State Analyzer with  
Timing Analyzer**

Events Timeline:



This timeline depicts both the events the HP 16550A sequencer will look for from the target system, shown above the timeline, and the actions of the logic analyzer, shown below the timeline. The darkly shaded area depicts the portion of time where the logic analyzer's sequencer is running, storing qualified states in the state analyzer, and looking for the specified timing edge.

**HP 16550A Sequencer Setup:**

In the following example, indicative of this triggering application, you use both the state analyzer (Machine 1) Trigger menu and the timing analyzer (Machine 2) Trigger menu. These menus, showing the sequencer setups for this example, are pictured on the next page. The timing sequencer, in the Trigger 2 menu, is set up to trigger when it detects a falling edge on the most significant bit of the 8 bits associated with the TI\_LAB label. It is also configured to arm the state analyzer when it triggers. This cross-arming is set up in the Arming Control menu, accessible by touching the Arming Control field at the right edge of the Trigger 2 menu. The state sequencer, in the Trigger 1 menu, is triggered by the arm signal from the state analyzer ANDed with pattern term 'a', left at its default, don't care value. With a don't care trigger term the state analyzer will trigger immediately after it is armed by the timing analyzer. Also, in order to time-correlate state and timing analyzers, state time tags are turned on by touching the Count field at the right edge of the Trigger 1 menu and changing Off to Time.

100/500MHz LA E		Trigger 1	Print	Run
State Sequence Levels			Timer	Arming Control
1	While storing "onystate" TRIGGER on "arm-a" 1 time		1	Acquisition Control
2	Store "onystate"		2	Count Time
				Clear Trigger
Label	ST_LAB			
Terms	Hex			
a	XX			
b	XX			
c	XX			
d	XX			

100/500MHz LA E		Trigger 2	Print	Run
Timing Sequence Levels			Arming Control	Acquisition Control
1	TRIGGER on "glitch/edge1" 1 time			Clear Trigger
Label	TI_LAB			
Terms	Binary			
Glitch/Edge 1	↓.....			
Glitch/Edge 2	.....			

### HP 16550A Term Setup:

In this example, just a glitch/edge term for the timing analyzer is needed. In the lower part of the Trigger 2 menu the glitch/edge 1 term is set to detect a falling edge on bit 7. Only the default terms are used in the state analyzer, as shown in the lower part of the Trigger 1 menu, because, in this example, you want to trigger the state analyzer immediately after the timing analyzer arms it.

### Possible Applications for This Triggering Model:

- \* To examine software execution when a timing violation occurs.
- \* To determine whether an incorrectly timed pulse is the result of a hardware defect or an incorrectly programmed counter.



**Technical information in this  
document is subject to change  
without notice.**

**Printed in U.S.A. 3/92  
5091-4121 E**