

# PCI Compliance Testing

and

## Capturing PCI Bus transactions

---

*using the FuturePlus Systems FS2100/FS2101  
with HP logic analyzers*

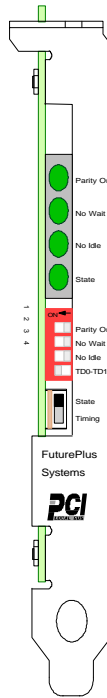
*This application note is designed to help FS2100/FS2101 PCI Analysis Probe users trigger their HP logic analyzers in order to capture PCI compliance violations and PCI bus transactions. These triggers have been tested but are **not guaranteed**. Please report any problems to [tech\\_sup@futureplus.com](mailto:tech_sup@futureplus.com). A good understanding of the PCI specification and the HP logic analyzer is assumed.*

<b>FILTERING DATA BEFORE IT IS ACQUIRED</b>	<b>4</b>
<b>TRIGGERING ON THE FIRST OCCURRENCE OF A TRANSACTION</b>	<b>6</b>
<b>SAVE A CERTAIN TRANSACTION TYPE (<i>NO TRIGGER</i>)</b>	<b>7</b>
<b>TRIGGER ON A PARTICULAR SEQUENCE OF TRANSACTIONS</b>	<b>7</b>
<b>USING TRIGGER MACRO'S</b>	<b>8</b>
<b>TRIGGERING HINTS</b>	<b>8</b>
<b>COMPLIANCE VIOLATION TRIGGERS</b>	<b>9</b>
Protocol Violation - LATENCY: Target first data	9
Protocol Violation - LATENCY: Target subsequent data phases	9
Protocol Violation - LATENCY: Master	10
Compliance Checklist - MP14 ( <i>IRDY/FRAME</i> )	10
Protocol Violation - MASTER REQ	11
Compliance Checklist - TP20 ( <i>Target Stop/DEVSEL/TRDY Release</i> )	11
Compliance Checklist - TP5 ( <i>Target TRDY early release</i> )	11
Compliance Checklist - TP29/TP6 ( <i>Early deassertion of DEVSEL</i> )	12
Protocol Violation - Late assertion of DEVSEL w.r.t TRDY.	13
Compliance Checklist - TP7 ( <i>STOP early release</i> )	13
Compliance Checklist - TP19 ( <i>Target No TRDY in turn around cycle</i> )	14
Protocol Violation - FRAME/STOP	14
Protocol Violation - Illegal back to back commands	15
Protocol Violation- FRAME/IRDY relationship	15
Protocol Violation - TRDY/DEVSEL	16

<b>Compliance Checklist MP32 (<i>Dual Address Cycle</i>)</b>	<b>16</b>
<b>Protocol Violation - IRDY deassertion</b>	<b>17</b>
<b>Protocol Rule - IRDY assertion</b>	<b>17</b>
<b>MP6 and MP7: FRAME and IRDY early release</b>	<b>17</b>
Case 1: Frame asserted, burst transactions	18
Case 2: FRAME released, single data transactions	19
<b>TP30: Special Cycles and Reserved Commands</b>	<b>19</b>
<b>Protocol Violation- Illegal AD0/1, C/BE combination</b>	<b>19</b>
<b>Protocol Violation - Improper Master Abort</b>	<b>20</b>
<b>Protocol Violation - Illegal Use of PERR</b>	<b>21</b>
<b>THE ACQUISITION CONTROL FIELD</b>	<b>21</b>
<b>The Acquisition Mode Field</b>	<b>22</b>
The Trigger Position Field	22
Branches Taken Stored/Not Stored	22
<b>USING SYMBOLS</b>	<b>22</b>
<b>The ADDR Label</b>	<b>22</b>
<b>The BUS_UT variable</b>	<b>22</b>
<b>The L_CMD variable</b>	<b>23</b>

## FILTERING DATA BEFORE IT IS ACQUIRED

The front panel switches on the FS2100/FS2101 can be used to filter out unwanted PCI WAIT and IDLE cycles before they are clocked to the analyzer. Use these switches to save analyzer memory thus capturing more transactions. However, when measuring performance or looking for certain compliance violations it is necessary to allow the WAIT and IDLE states to be clocked to the logic analyzer. In those cases throw the respective switches to the off position.



Each switch is labeled and explained in the chart below.

Switch	Setting	LED
Parity On	Parity Checking enabled	ON
Parity On	Parity checking disabled	OFF
No WAIT	No WAIT cycles acquired	ON
No WAIT	All WAIT cycles acquired	OFF
No IDLE	No IDLE cycles acquired	ON
No IDLE	All IDLE cycles acquired	YES
TDO/TDI	TDO connected to TDI	No LED, switch in rightmost position
TDO/TDI	TDO not connected to TDI	No LED, switch in leftmost position
State/Timing	State	ON
State/Timing	Timing	OFF

Several resource terms have also been defined in the FS2100/FS2101 software.

They are:

Resource Terms	L_CMD	AVALID	DVALID	MABORT	GRANT
TRANSACT	IO_READ	0	X	X	X
DATA	X	X	0	X	X
AVALID	X	0	X	X	X
GRANT	X	X	X	X	0
MABORT	X	X	X	0	X

*all other labels are DON'T CARE (X)*

TRANSACT can be defined to be any transaction with any address range or data pattern that you are looking for. When using the term TRANSACT as a store qualifier change AVALID to a DON'T CARE (x).

Other resource terms can be added to the trigger specification to take advantage of all the hardware assist contained on the FS2100/FS2101 module. The following is a list of all the cycle bits that are sent to the logic analyzer from the FS2100/FS2101 hardware.

Cycle bit name	Function
EOFT - End of Transaction	Low true for one clock cycle and indicates the last cycle of a transaction
CALPER - Calculated Parity Error	Low true for one clock cycle and indicates that the on board parity logic has detected a parity that is different than the parity transmitted on the bus. Please note that the Parity Checking switch must be in the ON position.
MABORT - Master Abort	Low true when a Master Abort condition has been detected. Five clock cycles on a single data transfer with no DEVSEL assertion and six clock cycles on a multi-beat transfer with no DEVSEL asserted. Remains true for one clock cycle.
PVALID - Parity Valid	Low true for the cycles in which parity is being transmitted on the PCI bus.
IDLE - IDLE cycle	Low True when the bus is IDLE. False when the bus is busy.
DVALID - Data Valid	Low true when data is being transferred on the PCI bus.
WINITI - Master Initiated WAIT State	Low true when a WAIT state is being initiated by the master
WTARGT - Target Initiated WAIT State	Low true when a WAIT state is being initiated by the target
RETRY - Retry	Low true when a retry condition has been detected on the PCI bus
TABORT - Target Abort	Low true when a Target Abort condition has been detected on the PCI bus. This signal is true for one clock bit.
WNODEV - WAIT state caused by no assertion of DEVSEL#	Low true when a WAIT state has been caused by no assertion of DEVSEL#.
GNT - The Grant signal for that slot	In State Mode this signal is latched and held until end of transaction. Useful as a store qualifier. In timing mode the GNT# from the PCI bus is passed through to the logic analyzer.
L_CMD - The latched command lines	The C/BE signals latched during the command /address phase and held until end of transaction.

AVALID - Address Valid	Low true on the first assertion of FRAME# and the rising edge of the PCI clock. True for one cycle except on Dual Address cycles when it is true for two cycles.
------------------------	--

All of these signals are available for use in the trigger specification.

## TRIGGERING ON THE FIRST OCCURRENCE OF A TRANSACTION

The following shows a STATE mode trigger specification that will trigger on the first occurrence of an interrupt acknowledge cycle.

- 1 While storing anystate  
TRIGGER on "TRANSACT" 1 time
- 2 Store anystate

Under TERMS, define TRANSACT as INTACK under the LABEL L\_CMD TRANSACT can be changed to reflect any transaction or group of transactions ( memory reads, IO writes, etc.) under the L\_CMD LABEL. To further qualify this trigger set the ADDR\_B label (demultiplex mode only) for the TERM TRANSACT to the address that you are looking for and DATA to the data you are looking for.

2M Sample LA B
Trigger 1
Cancel
Run

State Sequence Levels		Timer	1	2		
1	While storing "anystate" TRIGGER on "TRANSACT" occurring 1 time	-	-			
2	Store "anystate"	-	-			

Arming Control

Acquisition Control

Count Time

Modify Trigger

Label	L_CMD	ADDR	DVALID	AVALID	GNT
Terms	Symbol	Hex	Hex	Hex	Binary
TRANSACT	INTACK	XXXXXXXX	X	X	X
DATA	absolute XXXX	XXXXXXXX	0	X	X
AVALID	absolute XXXX	XXXXXXXX	X	0	X
GRANT	absolute XXXX	XXXXXXXX	X	X	0

## **SAVE A CERTAIN TRANSACTION TYPE (NO TRIGGER)**

---

The previous was used to trigger on a certain transaction and save all cycles before and after that transaction occurred. The need often arises to save only a certain transaction type or types and to not actually trigger on an event. The logic analyzer trigger menu will not allow you to delete the sequence level that has the word *trigger* embedded in it. This makes the task of fooling the logic analyzer a bit confusing. The key is to have the logic analyzer trigger on the transaction you are looking to store. It should be noted that the Acquisition Control Field will control where the trace starts and ends. Also, the analyzer will not stop and display the acquired data until the trigger condition is met or the acquisition memory is full. If the store conditions set do not fill the memory simply press STOP and the analyzer will display the captured data.

- 1      While storing TRANSACT  
         TRIGGER on "TRANSACT" 1 time
  
- 2      Store TRANSACT

## **TRIGGER ON A PARTICULAR SEQUENCE OF TRANSACTIONS**

---

By creating two terms TRANSACT and NEXT a trigger can be setup to look for a particular sequence of transactions. This can be useful in tracking down reads and writes to registers.

- 1      While storing anystate  
         Find TRANSACT\*EOFT 1 time
  
- 2      While storing anystate  
         TRIGGER on "NEXT" 1 time
  
- 3      Store anystate

Define TRANSACT and NEXT to be any transaction type (using L\_CMD) with any address(ADDR\_B) and data (DATA ) that is necessary.

This trigger will store all states and look for the end of TRANSACT before it moves on to level 2. Once in level 2 it will store all states and trigger on NEXT. If there are too many states between TRANSACT and NEXT such that the logic analyzer runs out of memory you can qualify the store statement in level 2.

## USING TRIGGER MACRO'S

---

The HP16500 series of logic analyzers contains *Trigger Macros* that can be invoked when editing the state sequence levels in the trigger menu. These macros can be customized using the cycle bits and other resource terms. Below is a list of these useful macros.

### Basic Macros

1. Find anystate n times
2. Find event n times
3. Find event n consecutive times
4. Find event 2 immediately after event 1

### Sequence Dependent Macros

1. Find event2 n times after event1 before event3 occurs
2. Find too few states between event1 and event2
3. Find too many states between event1 and event2
4. Find n-bit serial pattern

### Time Violations

1. Find event2 occurring too soon after event1
2. Find event2 occurring too late after event1

### Delay

WAIT n external clock states

The macros can be “broken down” for additional customization by accessing the trigger menu and touching MODIFY TRIGGER. This button can be found on the lower right hand corner of the TRIGGER menu.

## TRIGGERING HINTS

---

- In the trigger statement there is the term that is the object of the trigger, *ex. TRIGGER on “FRAME”, FRAME would be considered the object of the trigger*, and then there is the object of the ELSE statement. The logic analyzer will always evaluate the trigger statement first and if it is true will trigger. The ELSE statement is only evaluated if the object of the trigger is false.
- In some of the above trigger macros the analyzer restricts the trigger sequence level by not allowing “combination” to be selected. To get around this either “break down” the macros or create resource terms with more labels included (when possible).

## COMPLIANCE VIOLATION TRIGGERS

---

The FS2100/FS2101 and your HP logic analyzer make for a powerful combination when tracking down compliance violations. These triggers have been tested but are not guaranteed. Please send all comments to [tech\\_sup@futureplus.com](mailto:tech_sup@futureplus.com). Suggestions for additional triggers are welcome and encouraged. New versions of this document will be published and posted on our web site [www.futureplus.com](http://www.futureplus.com).

### Protocol Violation - LATENCY: Target first data

Operating Rule 25. All targets are required to complete the initial phase of a transaction (read or write) within 16 clocks from the assertion of FRAME.

WAIT states must be acquired by the FS2100/FS2101.

#### Resource terms needed:

WAIT\_T        WTARGET=0, all others don't care (x)

- 1        While storing "anystate"  
         TRIGGER on 16 consecutive occurrences  
         of "WAIT\_T"
- 2        Store "anystate"

### Protocol Violation - LATENCY: Target subsequent data phases

Operating Rule 26. The target is required to complete a subsequent data phase within 8 clocks from the completion of the previous data phase.

WAIT states must be acquired by the FS2100/FS2101.

#### Resource terms needed:

DATA         DVALID=0, all others don't care (x)

AVALID       AVALID=0, all others don't care (x)

EOFT         EOFT=0, all others don't care (x)

WAIT\_T       WTARGET=0, all others don't care (x)

- 1        While storing "anystate"  
         Find "DATA" 1 time
- 2        While storing "anystate"  
         TRIGGER on "WAIT\_T" 8 times  
         ELSE on "AVALID+EOFT" go to level 1
- 3        Store "anystate"

## Protocol Violation - LATENCY: Master

Operating Rule 27. A master is required to assert its IRDY# within 8 clocks for any given data phase.

WAIT states must be acquired by the FS2100/FS2101.

### Resource terms needed:

WAIT\_IWINITI=0, all others don't care (x)

- 1 While storing "anystate"  
TRIGGER on 8 consecutive occurrences  
of "WAIT\_I"
- 2 Store "anystate"

## Compliance Checklist - MP14 (IRDY/FRAME)

Operating Rule 8C. *FRAME# cannot be deasserted unless IRDY# is asserted. (IRDY# must always be asserted on the first clock edge that FRAME# is deasserted) This is also covered by Compliance Checklist MP14*

*MP14: IUT never deasserts FRAME unless IRDY is asserted or will be asserted (3.3.3.1).*

Use the trigger macro: **Find event 2 immediately after event 1**

For this trigger IDLE cycles must be acquired by the FS2100/FS2101. This trigger will catch the case where the bus goes from a state where both IRDY and FRAME are asserted to an IDLE state (both deasserted). This trigger can be varied for more complete coverage of the IRDY/FRAME relationship.

### Resource terms needed:

IDLE FRAME=1, IRDY=1 all others don't care (x)

FR/IRDY FRAME=0, IRDY=0 all others don't care (x)

- 1 While storing "anystate"  
TRIGGER on "IDLE" immediately  
after "FR/IRDY"
- 2 Store "anystate"

## Protocol Violation - MASTER REQ

Operating Rule 10. When the current transaction is terminated by the target either by retry or disconnect (with or without data) the master must deassert its REQ# signal before repeating the transaction. The master must deassert REQ# for a minimum of two clocks, one being when the bus goes to the IDLE state ( at the end of the transaction where STOP# was asserted) and either the clock before or the clock after the IDLE state.

To capture this violation the FS2100/FS2101 must be set to acquire both IDLE and WAIT states. Use the trigger macro, **Find too few states between event1 and event2** to set up the following trigger.

Resource terms needed:

REQ            REQ=0, all others don't care (x)  
STOP/GNT    STOP=0, GNT=0, all others don't care (x)

- 1        While storing "anystate"  
         TRIGGER on "REQ" after "STOP/GNT"  
         with < 2 states in between
- 2        Store "anystate"

## Compliance Checklist - TP20 (*Target Stop/DEVSEL/TRDY Release*)

From the PCI Compliance Checklist TP20: The IUT always deasserts TRDY#, STOP# and DEVSEL# the clock following the completion of the last data phase. This corresponds to the PCI Specification section 3.3.3.2. and 3.3.3.3.2.1

Try this 3 level trigger for capturing this violation. The FS2100/FS2101 must acquire IDLE and WAIT states.

**Resource terms needed:**

AVALID        AVALID=0, all others don't care (x)  
EOFT          EOFT=0, all others don't care (x)  
STOP          STOP=0, all others don't care (x)  
TRDY          TRDY=0, all others don't care (x)  
DEVSEL        DEVSEL=0, all others don't care (x)

- 1        While storing "anystate"  
         Find "EOFT" 1 time
- 2        While storing "anystate"  
         TRIGGER on "TRDY+STOP+DEVSEL" 1 time  
         ELSE on "AVALID" go to level 1
- 3        Store "anystate"

**Compliance Checklist  
- TP5  
(Target TRDY early release)**

TP5: Once the IUT has asserted TRDY it never changes TRDY until the data phase completes.

The FS2100/FS2101 needs to acquire WAIT states for this trigger.

**Resource terms needed:**

DATA DVALID=0, all others don't care (x)

EOFT EOFT=0, all others don't care (x)

TRDY TRDY=0, all others don't care (x)

- 1 While storing "anystate"  
Find "TRDY \* ≠DATA" 1 time
- 2 While storing "anystate"  
TRIGGER on "≠TRDY" 1 time  
ELSE on "DATA+EOFT" go to level 1
- 3 Store "anystate"

## **Compliance Checklist - TP29/TP6 (*Early deassertion of DEVSEL*)**

TP6: Once the IUT has asserted TRDY it never changes DEVSEL# until the data phase completes.

TP29: Once the IUT has asserted DEVSEL it never deasserts DEVSEL until the last data phase has completed except to signal target abort (3.7.1) ( Operating rule 15)

The FS2100/FS2101 needs to acquire WAIT states for this trigger

**Resource terms needed:**

EOFT EOFT=0, all others don't care (x)

TABORT TABORT=0, all others don't care (x)

DEVSEL DEVSEL=0, all others don't care (x)

- 1 While storing "anystate"  
Find "DEVSEL" 1 time
- 2 While storing "anystate"  
TRIGGER on "≠DEVSEL\*≠TABORT" 1 time  
ELSE on "EOFT" go to level 1
- 3 Store "anystate"

### Protocol Violation - Late assertion of DEVSEL w.r.t TRDY.

Operating rule 14 reads: DEVSEL must be asserted with or prior to the edge at which the target enables its outputs (in this case TRDY). This trigger will look for TRDY being asserted before DEVSEL and then trigger.

The FS2100/FS2101 must be set to acquire all IDLE and WAIT states.

AVALID AVALID=0, all others don't care (x)  
 TRDY TRDY=0, all others don't care (x)  
 DEVSEL DEVSEL=0, all others don't care (x)

- 1 While storing "anystate"  
Find "AVALID" 1 time
- 2 While storing "anystate"  
TRIGGER on "≠DEVSEL\*TRDY" 1 time  
ELSE on "DEVSEL" go to level 1
- 3 Store "anystate"

### Compliance Checklist - TP7 (*STOP early release*)

TP7: Once the IUT has asserted TRDY# it never changes STOP# until the data phase completes.

This trigger looks for STOP# being asserted and will trigger if STOP# deasserts prior to the data phase completing. This trigger can be modified to look for STOP released and then asserted prior to the data phase by complimenting the trigger term STOP.

The FS2100/FS2101 must be set to acquire all IDLE and WAIT states.

**Resource terms needed:**

DATA DVALID=0, all others don't care (x)  
 AVALID AVALID=0, all others don't care (x)

STOP STOP=0, all others don't care (x)

- 1 While storing "anystate"  
Find "STOP" 1 time
- 2 While storing "anystate"  
TRIGGER on "≠STOP\*≠AVALID" 1 time  
ELSE on "DATA\*≠AVALID" go to level 1
- 3 Store "anystate"

### **Compliance Checklist - TP19 (Target No TRDY in turn around cycle)**

TP19: The IUT never asserts TRDY# during the turnaround cycle on a read (PCI Specification section 3.3.1).

The FS2100/FS2101 must acquire all WAIT states.

TRDY TRDY=0, all others don't care (x)

RD\_CMD L\_CMD=xxx0, AVALID=0, all others don't care

- 1 While storing "anystate"  
Find "RD\_CMD" 1 time
- 2 While storing "anystate"  
TRIGGER on "TRDY" 1 time  
ELSE on "≠TRDY" go to level 1
- 3 Store "anystate"

### **Protocol Violation - FRAME/STOP**

The PCI Specification on page 42 states "Whenever STOP# is asserted, the master must deassert FRAME as soon as IRDY can be asserted."

This trigger will look for the case of STOP, FRAME and IRDY being all asserted as suggested by the above rule. If on the next clock tic FRAME is still asserted the analyzer will trigger.

The FS2100/FS2101 must acquire all WAIT states.

#### **Resource terms needed:**

STOP STOP=0, all others don't care (x)

IRDY IRDY=0, all others don't care (x)

FRAME FRAME=0, all others don't care (x)

- 1 While storing "anystate"  
Find "STOP\*FRAME\*IRDY" 1 time
- 2 While storing "anystate"  
TRIGGER on "FRAME" 1 time  
ELSE on "≠FRAME" go to level 1
- 3 Store "anystate"

## Protocol Violation - Illegal back to back commands

Page 59 section 3.4.2 of the PCI specification describes the conditions for compliant back to back transactions. The following trigger will look for a read command followed by a any transaction with no intervening IDLE cycles by the same master (the PCI add-in card in the extender card connector). This would be the violation case. This trigger also takes into consideration the case of a Master Abort. The transaction following a Master Abort can be a fast back to back transaction.

For this trigger IDLE cycles must be acquired by the FS2100/FS2101.

Use the trigger macro: **Find event 2 immediately after event 1**

### Resource terms needed:

AVALID AVALID=0, all others don't care (x)

RDCMD\_EN L\_CMD=xxx0, EOFT=0, MABORT=1, all others don't care

- 1 While storing "anystate"  
TRIGGER on "AVALID" immediately  
after "RDCMD\_EN"
- 2 Store "anystate"

## Protocol Violation- FRAME/IRDY relationship

The PCI Specification clearly defines the relationship between IRDY and FRAME. A PCI bus cannot go from FRAME asserted to an IDLE state. This would be an illegal operation. The below trigger will capture this violation.

Use the trigger macro: **Find event 2 immediately after event 1.**

IDLE and WAIT states must be acquired by the FS2100/FS2101 for this trigger to properly capture the violation.

### Resource terms needed:

IRDY IRDY=0, all others don't care (x)

FRAMEFRAME=0, all others don't care (x)

IDLE FRAME=1, IRDY=1 all others don't care (x)

- 1 While storing “anystate”  
TRIGGER on “IDLE” immediately  
after “FRAME\*IRDY”
- 2 Store “anystate”

## Protocol Violation - TRDY/DEVSEL

The PCI Specification is clear on the role of TRDY and DEVSEL. TRDY or DEVSEL should not be asserted during an address cycle nor should it be asserted during an IDLE cycle.

To create the combination that is the object of the trigger the resource terms have to be carefully laid out. Define IDLE as the first term, AVALID as the second term and TRDY as the third. DEVSEL can be defined as the eighth term, IDLE1 as the sixth and AVALID1 as the seventh. *IDLE1 is equivalent to IDLE and AVALID1 is equivalent to AVALID.*

For this trigger IDLE and WAIT cycles must be acquired by the FS2100/FS2101.

### Resource terms needed:

AVALID AVALID=0, all others don't care (x)  
 TRDY TRDY=0, all others don't care (x)  
 IDLE IRDY=1, FRAME=1, all others don't care (x)

- 1 While storing “anystate”  
TRIGGER on “(IDLE + AVALID)TRDY+  
(IDLE1 + AVALID1)DEVSEL” 1 time
- 2 Store “anystate”

## Compliance Checklist MP32 (Dual Address Cycle)

Per the PCI Specification section 3.10.1 and the PCI Compliance Checklist item MP32 FRAME shall remain asserted during the cycle following a Dual Address Command.

MP32: The IUT always holds FRAME asserted for the cycle following a DUAL command.

The following trigger will capture this violation. WAIT states must be acquired by the FS2100/FS2101.

### Resource terms needed:

FRAMEFRAME=0, all others don't care (x)  
 DAC L\_CMD=DAD\_CY, AVALID=0

- 1 While storing “anystate”  
TRIGGER on “≠FRAME” immediately  
after “DAC”
- 

16

## Protocol Violation - IRDY deassertion

IRDY must be deasserted after the last transfer OR when FRAME is deasserted and STOP was asserted. A portion of this rule is in checklist item MP15.

MP15: The IUT never deasserts IRDY until at least one clock after FRAME is deasserted (3.3.3.1).

IDLE and WAIT states must be acquired by the FS2100/FS2101.

Use the trigger macro: **Find event 2 immediately after event 1 and** the below trigger to capture this violation.

### Resource terms needed:

IRDY            IRDY=0, all others don't care (x)  
EOFT            EOFT=0, all others don't care (x)

- 1            While storing "anystate"  
             TRIGGER on "IRDY" immediately  
             after "EOFT"
- 2            Store "anystate"

## Protocol Rule - IRDY assertion

IRDY can not assert while FRAME is released.

Use the trigger macro: **Find event 2 immediately after event 1 and** the below trigger to capture this violation.

IDLE and WAIT states must be acquired by the FS2100/FS2101.

### Resource terms needed:

IRDY            IRDY=0, all others don't care (x)  
IDLE            IRDY=0, FRAME=0 all others don't care (x)

- 1            While storing "anystate"  
             TRIGGER on "IRDY" immediately  
             after "IDLE"
- 2            Store "anystate"

This trigger will find the first assertion of IRDY and if it occurs after a cycle with FRAME released then the logic analyzer will trigger. IDLE defines both IRDY and FRAME released.

## MP6 and MP7: FRAME and IRDY early release

---

MP7 : Once the IUT asserts IRDY it never changes IRDY until the current data phase completes.

MP6 : Once the IUT asserts IRDY it never changes FRAME until the current data phase completes.

**Resource terms needed:**

DATA DVALID=0, all others don't care (x)

EOFT EOFT=0, all others don't care (x)

TRDY TRDY=0, all others don't care (x)

IRDY IRDY=0, all others don't care (x)

FRAMEFRAME=0, all others don't care (x)

- 1 While storing "anystate"  
Find " $\neq$ TRDY \* IRDY $\neq$ EOFT" 1 time
- 2 While storing "anystate"  
TRIGGER on " $\neq$ IRDY" 1 time  
ELSE on "DATA+ EOFT" go to level 1
- 3 Store "anystate"

Since the analyzer does not have a "Find if toggled" feature we need two triggers to look for the two cases of FRAME (asserted with IRDY and deasserted with IRDY).

***Case 1: Frame asserted, burst transactions***

- 1 While storing "anystate"  
Find "FRAME\* IRDY" 1 time
- 2 While storing "anystate"  
TRIGGER on " $\neq$ FRAME $\neq$ DATA" 1 time  
ELSE on "DATA" go to level 1
- 3 Store "anystate"

## Case 2: FRAME released, single data transactions

- 1 While storing “anystate”  
Find “≠FRAME\* IRDY” 1 time
- 2 While storing “anystate”  
TRIGGER on “FRAME\*≠DATA” 1 time  
ELSE on “DATA” go to level 1
- 3 Store “anystate”

## TP30: Special Cycles and Reserved Commands

TP30: IUT never responds to special cycles (3.7.2). Per the PCI Specification the same is true of reserved commands. This one trigger will look for both these violations by looking for the assertion of either TRDY or DEVSEL to a Reserved command or Special Cycle.

WAIT states must be acquired by the FS2100/FS2101.

### Resource terms needed:

EOFT EOFT=0, all others don't care (x)  
 TRDY TRDY=0, all others don't care (x)  
 DEVSEL DEVSEL=0, all others don't care (x)  
 RESVAL\_CMD=010x, AVALID=0 all others don't care (x)  
 RESVB\_CMD=100x, AVALID=0 all others don't care (x)  
 SP\_CYC L\_CMD=0001, AVALID=0 all others don't care (x)

- 1 While storing “anystate”  
Find “RESVA+RESVB+SP\_CYC” 1 time
- 2 While storing “anystate”  
TRIGGER on “TRDY+DEVSEL” 1 time  
ELSE on “EOFT” go to level 1
- 3 Store “anystate”

## Protocol Violation- Illegal AD0/1, C/BE combination

Section 3.2.2 lists a table that indicates valid address/byte enable combinations that a master may use when addressing I/O space.

AD1	AD0	C/BE3	C/BE2	C/BE1	C/BE0
0	0	X	X	X	0
0	1	X	X	0	1
1	0	X	0	1	1

1	1	0	1	1	1
X	X	1	1	1	1

**Illegal combination of AD0,1 and C/BE lines and the necessary resource terms.  
Each of these resource terms should have DVALID=0**

Resource Terms	ADDR_B bit1	ADDR_B bit0	C/BE3	C/BE2	C/BE1	C/BE0
AD1	0	0	X	X	X	1
AD2	0	1	X	X	1	X
AD3	0	1	X	X	X	0
AD4	1	0	X	1	X	X
AD5	1	0	X	X	0	X
AD6	1	0	X	X	X	0
AD7	1	1	1	X	X	X
AD8	1	1	X	0	X	X
AD9	1	1	X	X	0	X
AD10	1	1	X	X	X	0

The following trigger will capture any combination listed in the above chart.

The FS2100/FS2101 must be in demultiplex mode (6 PODS for a 32 bit system, 9 PODS for a 64 bit system). All 10 resource terms will be used for this trigger.

- 1 While storing "anystate"  
TRIGGER on  
"IAD1+IAD2+IAD3+IAD4+IAD5+IAD6+IAD7+IAD8+IAD9+IAD10\*" 1 time
- 2 Store "anystate"

## Protocol Violation - Improper Master Abort

Section 3.3.3.1 of the PCI specification lists the proper response by the master for no DEVSEL assertion. This trigger will trigger if the master fails to terminate the transaction within 8 clocks.

The FS2100/FS2101 must acquire IDLE and WAIT states.

### Resource terms needed:

IRDY IRDY=0, all others don't care (x)

FRAMEFRAME=0, all others don't care (x)  
IDLE FRAME=1, IRDY=1, all others don't care (x)  
MABORT MABORT=0, all others don't care (x)

- 1 While storing "anystate"  
Find "MABORT" 1 time
- 2 While storing "anystate"  
TRIGGER on "FRAME+IRDY" 1 time  
ELSE on "IDLE" go to level 1
- 3 Store "anystate"

## Protocol Violation - Illegal Use of PERR

PERR is used to signal data parity errors only. It is to be asserted 2 cycles after the errored data is transmitted. If PERR is asserted 2 cycles after an address cycle this would be illegal.

The FS2100/FS2101 must acquire WAIT states for this trigger.

### Resource terms needed:

PERR PERR=0, all others don't care (x)  
AVALID AVALID=0, all others don't care (x)

- 1 While storing "anystate"  
Find "AVALID" 1 time
- 2 While storing "anystate" then WAIT 1 external  
clock state
- 3 While storing "anystate"  
TRIGGER on "PERR" 1 time  
Else on "anystate" go to level 1
- 4 Store "anystate"

## THE ACQUISITION CONTROL FIELD

---

In the HP16500 mainframe, the Acquisition Control field can be found on the right hand side of the trigger menu and can be used to access the Acquisition Control menu. The Acquisition Control menu is used to set the acquisition mode and trigger position within available memory.

## The Acquisition Mode Field

The Acquisition Mode field toggles between Manual and Automatic. When set to Automatic in STATE mode, the trigger position is computed based on the sequence specification. In TIMING mode, the trigger position and the sample period is computed based on the sec/Div and the delay settings in the Waveform menu.

When the Acquisition Mode field is set to Manual, additional configuration fields become available. Use these fields to further qualify what data is stored.

## The Trigger Position Field

The trigger Position field accesses a selection menu with the options of Start, Center, End or User Defined. When an option is selected, that point of the available memory is positioned relative to the trigger. In TIMING mode, the start of the acquisition can also be delayed.

## Branches Taken Stored/Not Stored

The Branches Taken field is a toggle field which sets the analyzer to store, or not to store, the resource term that sent the analyzer off on a branch. The trigger specifications shown in this application note assume that this field is set to Branches Taken Stored.

*For more detailed information please refer to the User's Reference of your Hewlett-Packard Logic Analyzer.*

## USING SYMBOLS

---

In the HP16500 mainframe, the SYMBOLS button can be found in the upper right hand corner of the FORMAT menu. Once selected, the user can set up symbols for any label listed in the FORMAT menu. These symbols can be added to or deleted. The user can create new labels in the format menu and these labels can be any combination of PCI signals that are acquired by the FS2100/FS2101 PCI Analysis Probe.

## The ADDR Label

Adding symbols via the FORMAT menu to the ADDR label can also help find problems when doing system testing. This feature allows you to replace the address in the PCI inverse assembly display with a symbol.

## The BUS\_UT variable

The Bus Utilization BUS\_UT variable is made up of the following cycle bits : WNODEV, AVALID, TABORT, DVALID, WTARGET, WINITI, RETRY, IDLE, MABORT. The list of symbols defined for this variable are the signal names themselves. This is a convenient grouping that helps make triggering and performance analysis easier.

## The L\_CMD variable

This variable is the C/BE[3:0] lines latched with the first rising edge of the PCI clock with FRAME asserted. These signals are held until the end of the transaction. They indicated the command that is being transmitted on the PCI Local bus. Below is the encoding of these signals and the symbols defined for the L\_CMD variable. These encodings can also be found in the PCI Specification.

Symbol	L_CMD encoding
INTACK	0000
SPEC_CYC	0001
I/O_RD	0010
I/O_WR	0011
RESRVD	0100
RESRVD	0101
MEM_RD	0110
MEM_WR	0111
RESRVD	1000
RESRVD	1001
CON_RD	1010
CON_WR	1011
MEMRDM	1100
DAC_CY	1101
MEMRDL	1110
MEMWRI	1111
I/O_XACTIONS	001X
MEM_XACTIONS	011X
CONFIG_XACTIONS	101X

Additional information of triggering can be found in Hewlett-Packard Application note 1223 *Logic Analyzer Triggering Applications*. This application note can be obtained from your local Hewlett-Packard sales representative.

For more information on the PCI Analysis Probe (FS2100/FS2101) please contact:

FuturePlus Systems Corporation  
2790 North Academy Blvd. Suite 307  
Colorado Springs, CO 80917-5329  
TEL:719-380-7321  
FAX:719-380-7362  
internet: [sales@futureplus.com](mailto:sales@futureplus.com)  
[www.futureplus.com](http://www.futureplus.com)