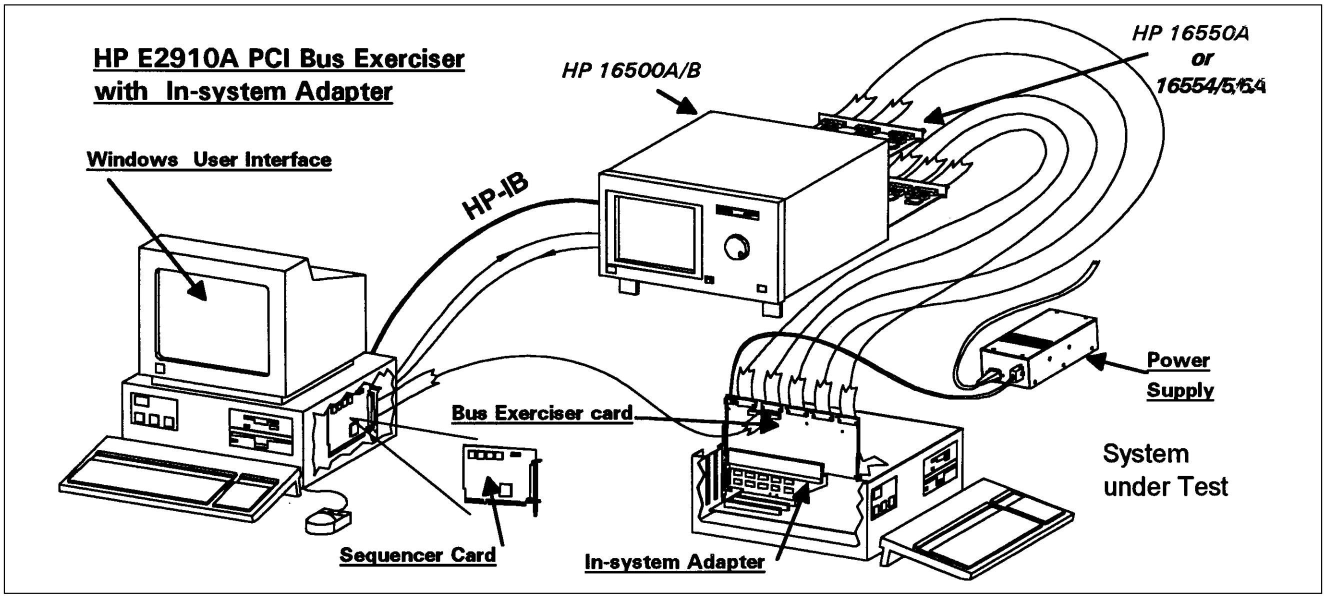
The HP E2910A PCI Exerciser

Technical Data

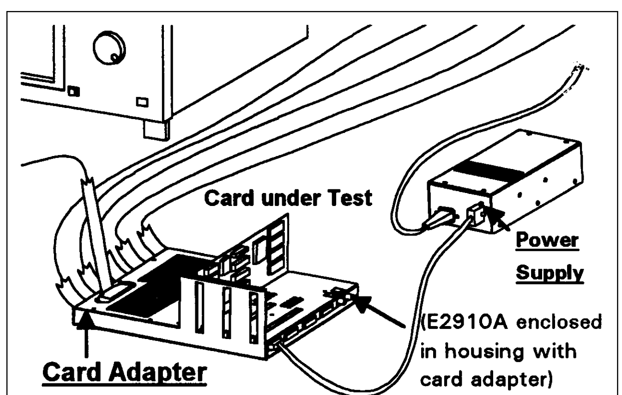
Product Description and Specifications



With in-system adapter for test at any PCI slot

Key functions

- generate all types of PCI transaction and protocol variations from 0 to 33 MHz
- continuously monitor for protocol violations
- capture and disassemble traffic for multi-level analysis via integrated logic analyzer
- replay and edit captured PCI transactions
- build and run suites of tests automatically
- test PCI protocol compliance



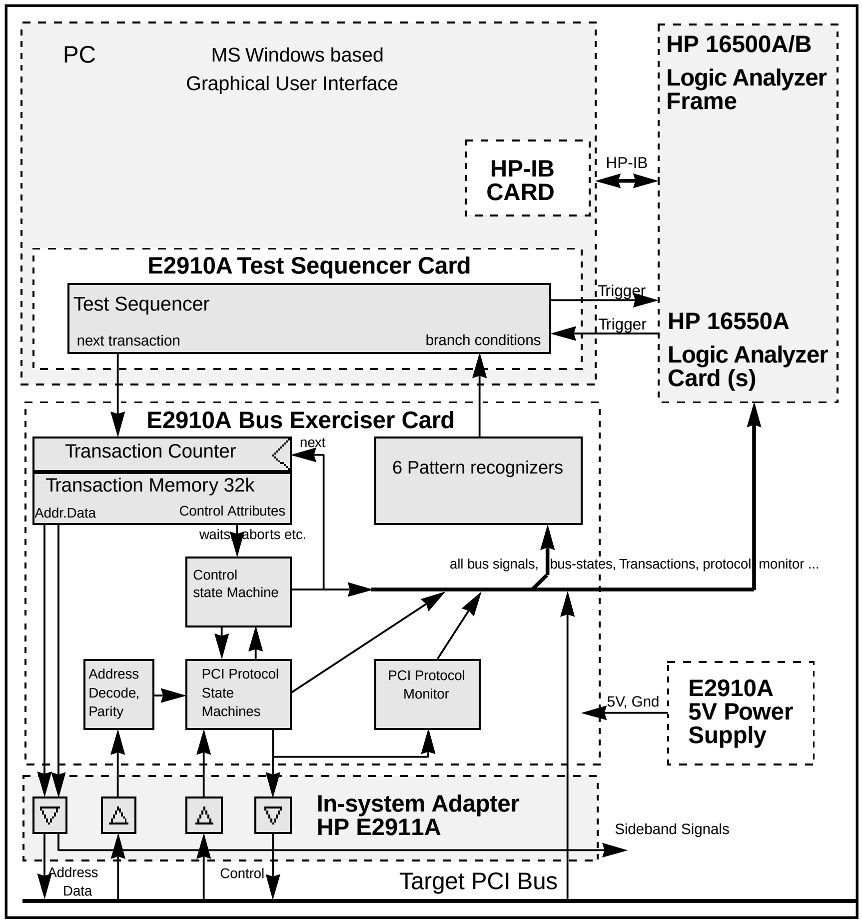
With card adapter
- 3 Slots
- PCI arbiter

Introduction

Accelerate development of dependable products

The HP E2910A PCI bus exerciser provides you with a powerful tool to accelerate the development of PCI devices and systems significantly and ensure exhaustive verification at the earliest opportunity. Together with a PC running MS Windows 3.1 and a Hewlett-Packard 16500B with HP 16550/4/5/6A logic analyzer for data capture, the HP E2910A brings sophisticated control and analysis capabilities to the PCI bus. From bring-up and debug of individual devices through to integration and troubleshooting in-system, use this independent, controllable agent on the bus to:

- emulate missing device traffic for concurrent development
- generate corner-cases and exceptions within the protocol to verify thoroughly
- record, replay and modify real traffic to reproduce critical cases and isolate the root cause of real problems faster
- identify the ”guilty party“ fast when anomalies occur
- automate functional and compliance testing to reduce retest time following silicon revisions



Product summary

Bus exerciser card

- generates user-defined transactions as master or target under sequencer control
- detects bus conditions using a pattern recognizor for triggering and branching the test sequencer
- continuously monitors a defined set of protocol rules
- generates sideband signals synchronized to bus transactions

Test sequencer card

- plugs into ISA slot in host PC to control bus exerciser with trigger or branch on event
- provides external trigger input/ output for cross-triggering with logic analyzer

Windows user interface provides

- intuitive set-up and control
- PCI bus transaction description language for defining master and target traffic with complete control
- bus traffic analysis at multiple levels in data lister windows using data from logic analyzer
- automatic comparison of data and protocol behavior with expected data and behavior
- record and replay of bus traffic at transaction level, including wait states

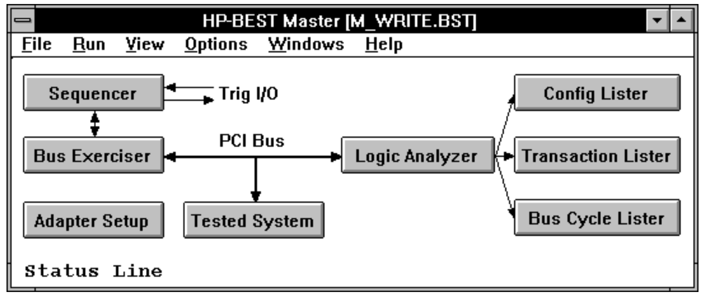
Windows test executive provides

- building and automated sequencing of test suites
- operator configuration of individual test parameters

Customizable compliance test suite

- tests to check PCI protocol compliance

User interface

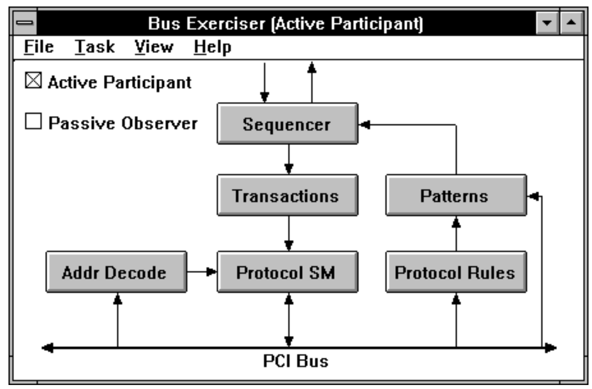


MS Windows based

The HP E2910A software runs under Windows, fitting seamlessly into the working environment on your PC.

Data-flow oriented

The user interface is oriented around the test-data flow, from sequence and bus exerciser set-up through to captured traffic analysis, making it intuitive and quick to learn and operate. Click on the system block diagram to set-up each part of the system.



ASCII format for data interchange

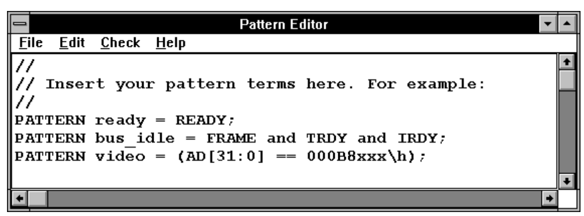
All set-up and data files are stored in ASCII format for conversion to or from other formats, or processing in other applications. You can easily use custom programs or applications such as spreadsheets to manipulate the files.

On-line help

Comprehensive on-line help is available at all times, covering all aspects of operating the bus exerciser and shortening the learning curve.

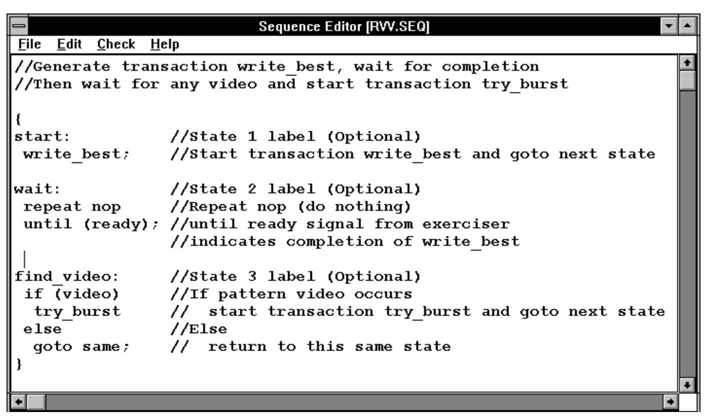
Pattern editor

Set up logical combinations of signals which the pattern recognizor should detect. The patterns can then be used in the test sequencer to make branching decisions.



Sequence editor

Define states, actions and branching conditions to set up a transaction sequence based on your defined transaction blocks and pattern events.



Transaction editor

Use the transaction editor to set up labeled blocks of one or more master and/or target transactions using the PCI bus transaction description language. Each block can then be called from the test sequencer. The PCI bus transaction description language (BTDL) gives you complete control over the protocol behavior of the exerciser.

Phase commands

Use the PCI BTDL to set up transactions on a per phase basis as master or target.

Transaction phase	MASTER COMMAND	TARGET COMMAND
Request	m_req	
Address	m_addr	
Address step	m_addr_step	
Data	m_data	t_data
Data step	m_data_step	t_data_step
Last data phase	m_last	
Dual address	m_dual_addr	

Parameters for protocol control

Within each phase command, use command parameters to control the protocol behavior of the exerciser as master or target.

Attribute	Action
ack64	Assert ACK64# in data phase
addr=<value>	AD[31::0] in address phase
hi_addr=<value>	AD[64::32] in Dual Address
byten=<value>	C/BE[7::0] in data phase
cmd=<value>	C/BE[7::0] in address phase
data=<value>	AD[31::0] in data phase
hi_data=<value>	AD[64::32] in data phase
interrupt(a..d)	Assert interrupt line(s)
lock	Exclusive access in address phase
marker=<string>	Mark phase to find in lister
no_req	Deassert REQ# in data phase
perr	Assert PERR#
req64	Assert REQ64# in address phase
serr	Assert SERR#
side0, side1, ...side7	Assert sideband signal
term = retry disconnect abort	Terminate transaction in target data phase
try_back	Try fast back-to-back in address phase
wait = <value>	Force wait states in data phase
wr_par	Assert wrong PAR#
wr_par64	Assert wrong PAR64#

```
Transaction Editor [TED.BCY]
File  Edit  Check  Search  Help
#define TARGET a000000\h
//Test target's burst capabilities
//Expect target to accept a burst memory write of 4 DWORD with fast or
//medium address decoding, 1-0-0-0 or 0-0-0-0 wait states.

burst_write:
{
  m_addr(addr = TARGET, cmd = m_write, exp_decode = fast| medium);
  m_data(data = 01010101\h, exp_wait = 0::1, exp_term = no);
  m_data(data = 02020202\h, exp_wait = 0, exp_term = no);
  m_data(data = 03030303\h, exp_wait = 0, exp_term = no);
  m_last(data = 04040404\h, exp_wait = 0, exp_term = no);
}

//Expect target to accept a burst memory read of 4 DWORD with fast or
//medium address decoding, 4-0-0-0, 3-0-0-0, 2-0-0-0, 1-0-0-0 or
//0-0-0-0 wait states and check expected data.

burst_read:
{
  m_addr(addr = TARGET, cmd = m_read, exp_decode = fast| medium);
  m_data(exp_data = 01010101\h, exp_wait = 0::4, exp_term = no);
  m_data(exp_data = 02020202\h, exp_wait = 0, exp_term = no);
  m_data(exp_data = 03030303\h, exp_wait = 0, exp_term = no);
  m_last(exp_data = 04040404\h, exp_wait = 0, exp_term = no);
}
```

Parameters for evaluating test results

Command parameters also let you define the expected behavior of devices under test during each phase of transactions. Define expected data values or protocol behavior to be automatically checked when the captured bus traffic is uploaded and analyzed.

Attribute	Expect
exp_ack64 = id	yes/no//dont_care
exp_addr=<value>	AD[31::0] (or implied in burst)
exp_hi_addr= <value>	AD[63::32] (or implied in burst)
exp_backtoback =id	yes/no/dont_care
exp_burst = id	first, middle , last or any phase
exp_byten= <value>	C/BE[7::0] of data phase
exp_cmd=<value>	C/BE[7::0] in address phase of current transfer
exp_decode = id	fast, medium, slow, subtractive or none
exp_data= <value>	AD[31::0] in data phase
exp_hi_data= <value>	AD[64::32] in data phase
exp_int = id	inta, intb, intc or intd
exp_lock = id	locked access yes/no/dont_care
exp_perr = id	yes/no/ don't care
exp_prot_err = error	suppress protocol error report
exp_req64 = id	yes/no/don't care
exp_serr	yes/no/don't care
exp_side = id	side1, side2., side 3, side4
exp_term =	Target termination retry, disconnectalblc, abort, accept, any
exp_toreach = id	Phase reached yes/no/dont_care
exp_wait = <exp>	Number or range of waits
exp_wr_par = id	yes/no/dont_ care
exp_wr_par64=id	yes/no/dont_ care

Protocol violations

The exerciser can generate violations such as:

- reserved commands
- illegal number of wait states
- invalid BE/AD[1:0] combinations
- undefined special cycles

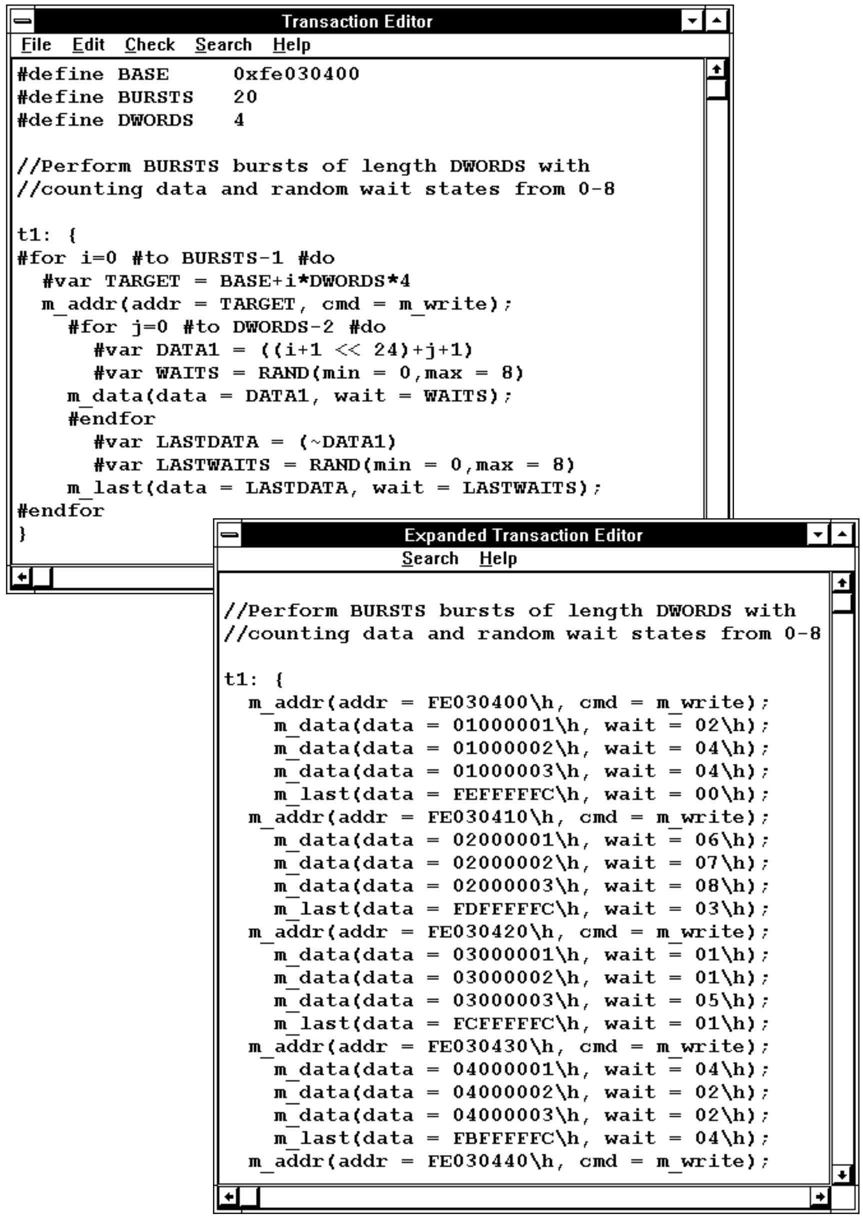
To prevent bus contention, signalling protocol cannot be violated.

Macro expansion

Use the extensive macro processing capabilities of the editors to

- simplify the creation of long or complex transactions by generating them algorithmically
- create random tests
- import test parameter data from data files

The macro commands and functions can be used in any of the editors and are expanded at compilation time before the test is carried out.



Macro functions

Command	Description
EOF (handle, remaining)	Returns true if less than remaining items in file
OPEN (name, delimits)	Opens file and returns handle
RAND (min, max)	Returns random integer between specified limits
READ (handle, step)	Returns next integer from file and increments pointer

Macro commands

Command	Description
#CALC	See #VAR
#CLOSE	Close a file or all files
#DEFINE or #REDEFINE	Define and evaluate new symbol
#DO , #TIMES	Expand/execute a specified number of times
#ERROR	Generate an error message
#FOR, #TO, #STEP, #DO	Expand/execute a specified number of times, using a stepped index
#IF ,#ELSE	Expand/execute on conditions
#IFDEF , IFNDEF #ELSE	Expand/execute if symbol defined/no defined
#INCLUDE	Include another filename
#MACRO	Define macro
#SEED	Seed or reseed the random number generator
#REPEAT #UNTIL	Repeat expansion/execution until a condition occurs
#UNDEFINE	Undefine a symbol for reuse
#VAR	Evaluate an expression and assign to symbol
#WHILE #DO	Expand/execute while a condition is true

Traffic analysis listers

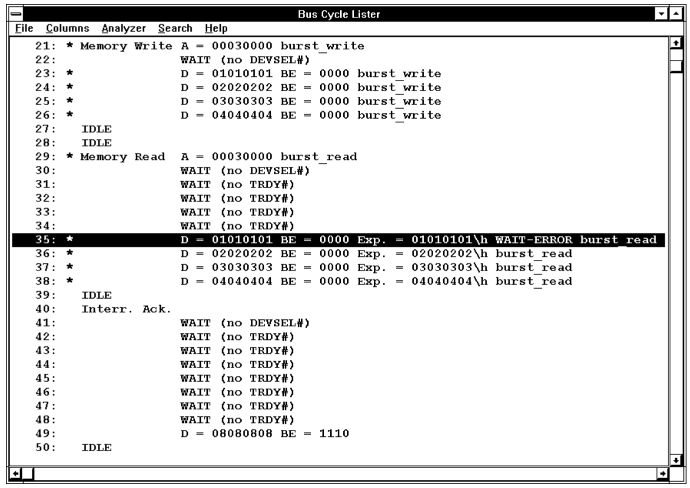
Data listers for traffic analysis

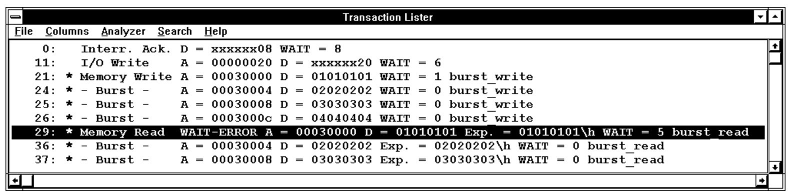
Use the hierarchical data listers to quickly and easily analyze up to 4k of bus state data captured by the logic analyzer. (Use the logic analyzer user interface to pre-filter the data, and for timing analysis). Note that the data listers accept data from state or timing mode.

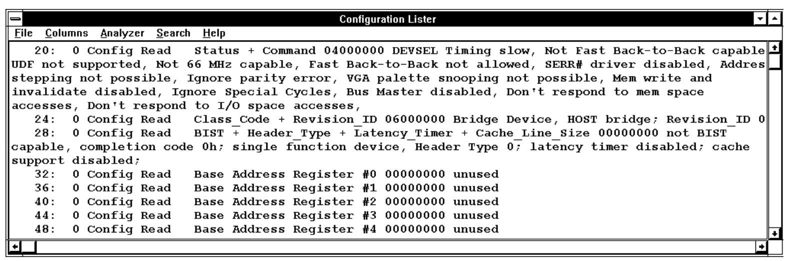
Bus cycle lister: disassembles the captured data to provide a complete clock-based listing of PCI cycles. Detected protocol violations are clearly indicated. If you define expected data values or protocol behavior in the bus transaction editor, the actual data transfers and protocol are checked and any unexpected results reported.

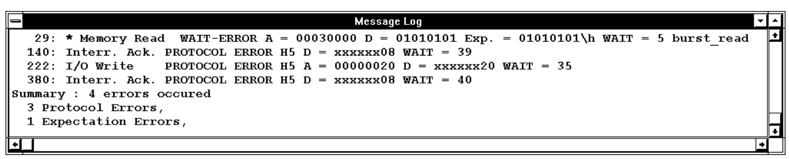
Bus transaction lister: further disassembles the data to provide a per transaction overview - for example transaction type, address and data content.

Configuration lister: decodes configuration transactions with additional information, interpreting the contents of the configuration data.









Lister correlation

Highlight a section of bus traffic in any lister and it is automatically also marked in the other listers, and in the source bus transaction editor (if the HP E2910A participated in the transaction). The logic analyzer markers can also be set to indicate the corresponding traffic in the waveform or listing displays.

Record & replay

Captured bus traffic can be moved from the bus transaction lister to the bus transaction editor for deterministic replay by the exerciser. The captured transactions are converted to the PCI bus transaction language, including recorded wait states, and can be edited as required.

Message log window

The message log window summarizes the results of a test. It lists sequentially all transaction markers and any associated error states of any type from the bus cycle lister, followed by a summary of the total number of protocol violations, data errors and expected protocol behavior errors.

Real-time performance

System performance is a crucial specification, and comparing long term bus-performance measurements under real operating conditions can help you

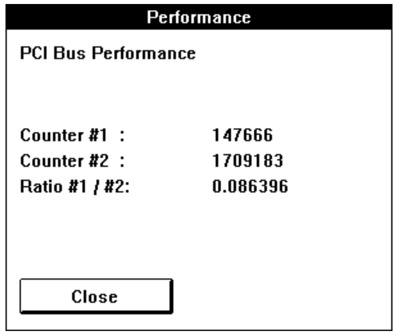
- identify bottlenecks
- optimize system settings
- compare system platforms and devices

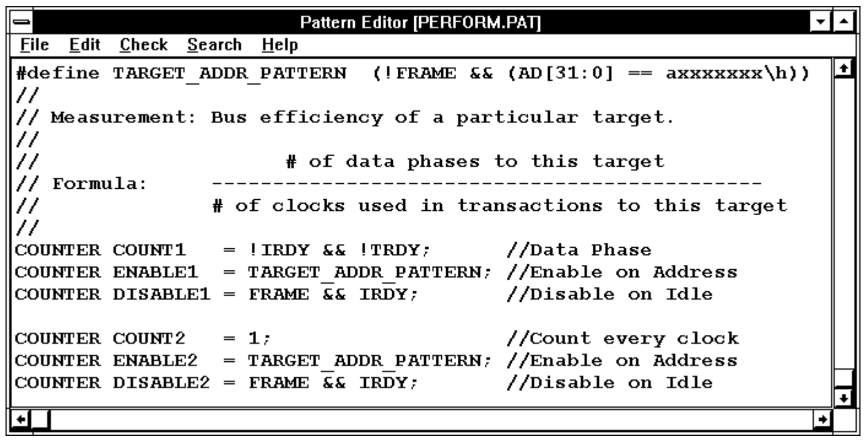
In passive observer mode, two 64 bit hardware counters are able to count defined bus events in real time to provide cumulative performance statistics over long periods of time.

The counters are controlled via the pattern recognizor and you can define three control patterns for each counter

- enable counter
- count
- disable counter

This allows you to count events within a specified window. For example, you can enable a counter on an address phase to a particular address range, count all data transfer phases and disable the counter on an idle phase in order to count all data transfers to the target address range. By using the second counter to count the wait states, you can measure the transfer efficiency, in terms of wait states per data transfer, to that target.





Automated test environment

Automated testing

Use the automatic testing capabilities of the HP E2910A to run suites of tests automatically, and reduce the time necessary for retesting of new silicon revisions or system configurations. A test suite is supplied to help test PCI protocol compliance of a device. The suite requires customization for a particular implementation under test (IUT), and can be expanded to include your own tests. The automated test environment consists of three elements:

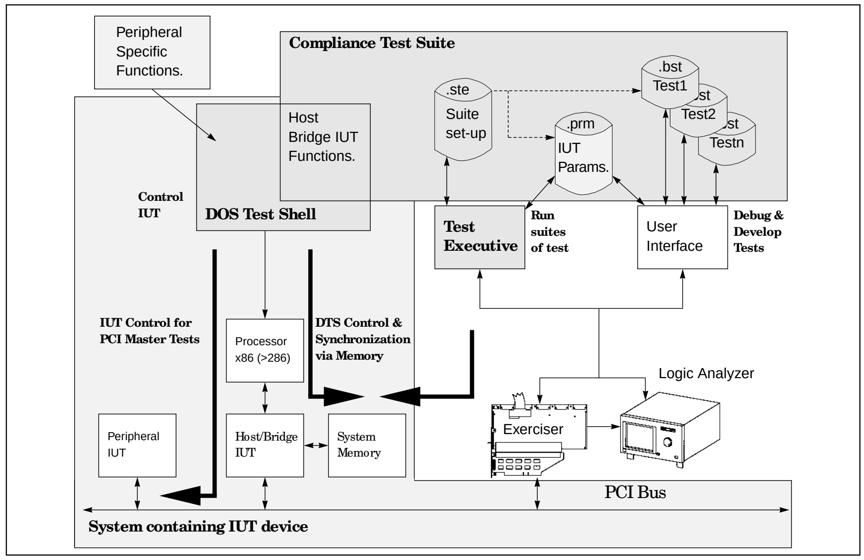
- test executive shell
- test suite for PCI protocol compliance
- DOS test shell

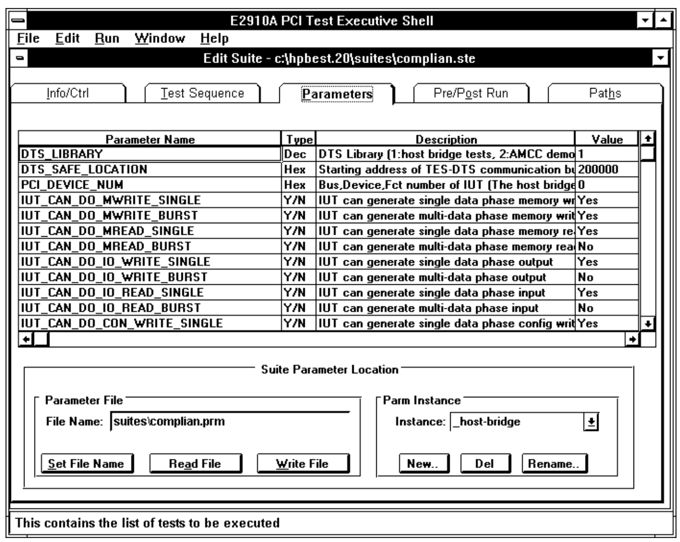
Test executive shell (TES)

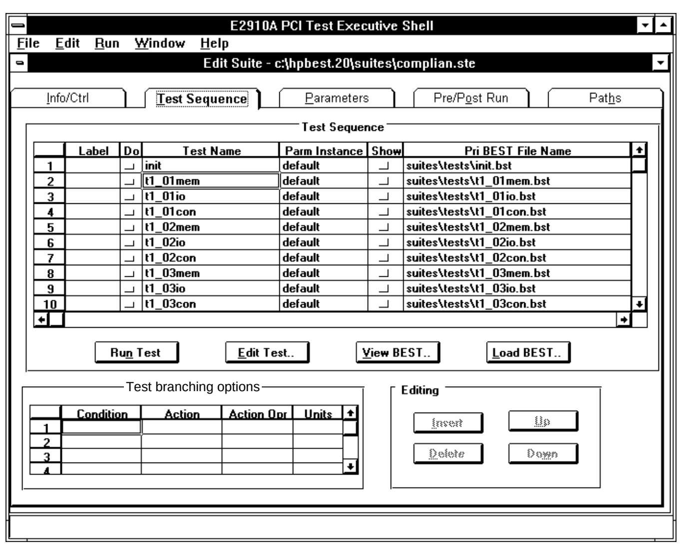
With the test executive shell (TES) you can set up, configure and run sequences of HP E2910A tests. New tests are developed and debugged using the normal HP E2910A user interface, then built into a test suite using the TES. When running a test suite under the TES you can:

- configure test suite parameters, such as IUT memory address, HP E2910A addresses, and IUT/system capabilities
- configure test sequence and branching conditions, such as stop/pause/repeat/skip on test pass or fail
- generate a test report file with test result summary, test log and test configuration record

The TES judges a test as failed if any protocol, expected data or expected IUT behavior errors occur during the test. When a test fails, you use the HP E2910A user interface to load and re-run the test and investigate the root cause of the failure in detail with the analysis listers.







Compliance test suite

The test suite provides more than forty HP E2910A tests to check most items in the PCI SIG component protocol checklist for master and target devices.

Scenario	
1.1	Device speed: Memory, I/O, configuration
1.2	Target abort: Memory, I/O, configuration
1.3	Target retry : Memory, I/O, configuration
1.4	Single data disconnect: Memory, I/O, configuration
1.5	Multi-data phase target abort: Memory, dual address, I/O, configuration, MRM, MRL, MW&I
1.6	Multi-data phase retry: Memory, I/O, configuration, MRM, MRL, MW&I
1.7	Multi-data phase disconnect: memory, I/O, configuration, MRM, MRL, MW&I
1.8	Multi-data phase & TRDY#: memory, dual address, I/O, Configuration, MRM, MRL, MW&I
1.9	Data parity error single data: memory, I/O, configuation
1.10	Data parity error multi-data phase: Memory, dual address, configuration, MRM, MRL, MW&I
1.11	Bus master timeout: MW&I
1.12	Target lock:
2.2	Target reception of special cycle
2.3	Target detection of A/D parity error: address, data
2.4	Target reception of I/O with legal and illegal byte enables
2.5	Target ignores reserved commands
2.6	Target receives configuration cycles
2.7	Target receives I/O waith A/D parity errors
2.8	Target gets configuration cycles with A/D parity errors
2.9	Target receives memory cycles
2.10	Target gets memory cycles with A/D parity errors
2.11	Target gets fast back-to-back cycles
2.12	Target performs exlusive access: accepts lock, releases lock
2.13	Target gets cycles with IRDY data stepping (waits)

For 1.x scenarios, the HP E2910A behaves as target to the IUT's master. For 2.x scenarios, the HP E2910A generates traffic as master to the IUT's target. The following cases within the scenarios are not covered by the suite:

- fast/subtractive decode as target of memory, memory read multiple, memory read line and memory read and invalidate
- fast/subtractive/slow decode as target of configuration cycles
- tests requiring control over the arbiter
- multiple master tests
- interrupt tests

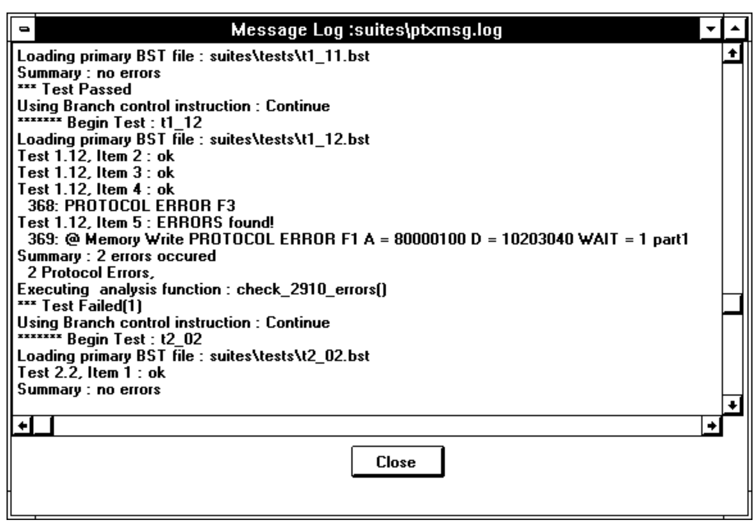
Each HP E2910A test contains comments explaining which test items are addressed by the test, as well as markers throughout the file which indicate the individual test items and allow you to correlate between the test report, the PCI compliance checklist and the test source file.

Using the compliance suite

To configure the test suite parameters you need to know your IUT and IUT system. For example, you need to set up parameters identifing the types of PCI capabilities supported by the IUT, and parameters indicating free system memory and PCI address spaces for the HP E2910A and DTS to use. Once configured, test suites can be run without detailed knowledge of PCI, HP E2910A syntax or the IUT.

The compliance tests require that the DTS is running on the system containing the IUT in order to control the IUT. The DTS already includes test functions for host bridge IUTs, but you may need to develop DTS functions for a peripheral IUT. Refer to the DOS test shell description.

To debug and identify the root cause of a failed test, use the HP E2910A user interface to correlate the traffic in the analysis listers with the HP E2910A test, and to rerun the test. Each test contains comments to identify the test cases covered and how it is checked, but you need to be familiar with the HP E2910A bus transaction language, the PCI protocol and checklist and your IUT specification to debug a problem effectively.



DOS test shell (DTS)
The DOS test shell (DTS) runs under DOS on the system containing the implementation Under test (IUT). An HP E2910A test controls the DTS through system memory, using a predefined set of handshaking macros. This allows an HP E2910A test to call associated DTS functions by library number and function number. DTS functions are used to set-up and control the IUT from the CPU in the IUT system. For example, to make the IUT generate master transactions for the 1.x scenarios, or to initialize the IUT if this is not practical using the HP E2910A.

The DTS includes a host bridge function library which is ready to use with the compliance test suite to test host bridge devices. These functions make use of PCI BIOS calls to control the host bridge.

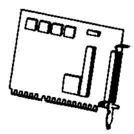
Function	Description
1	Nop
5	Initialize IUT for test
10	Clear IUT status register
20	Read IUT config register
30	Perform memory write
40	Perform memory read
50	Perform I/O read
60	Peform I/O write
70	Perform config read
80	Perform config write
90	Perform config read to empty slot
91	Perform config write to empty slot
92	Perform memory read multiple
93	Perform memory read line
94	Perform memory write & invalidate
95	Perform dual address memory write
96	Perform dual address memory read
100	Locate HP E2910A
141	Set PCI register bit
142	Save IUT command register
143	Restore IUT command register
150	End of test

To test peripheral devices you have to add a new library of DTS functions which force your IUT to perform the PCI master functions (30 to 96) required for 1.x scenarios, perhaps using an existing driver. The number of functions you need to create depends on the PCI master capabilities of your IUT. No functions are required for the 2.x scenarios (IUT target). To program new DTS routines for an IUT you need register-level knowledge of the IUT, or its driver and Microsoft Visual C++.

The DTS requires the IUT system to be IBM PC_AT compatible with 386 or later CPU, DOS 5.0 or later, at least 512k RAM, PCI BIOS and one free PCI slot for the HP E2910A with in-system adapter.

The entire DTS source code (C++ and assembler) is supplied, and may be used as a template for developing new functions or for porting to other platforms.

Test sequencer card



The test sequencer card plugs into an ISA-compatible slot in the controlling PC. Running at up to 33 MHz, the test sequencer state machine allows you to define complex sequences of transactions with triggering/branching on system events detected by the pattern recognizor or logic analyzer. This helps you reproduce problems which occur after a complex sequence of system events. The test sequence is defined as a series of test states during which actions, and the branching conditions for branching between states occur.

Number of states
60 states maximum.

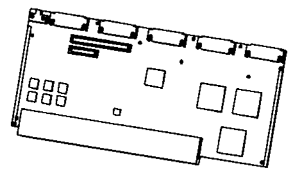
Branching inputs
Maximum of seven inputs selected from the available pattern recognizor terms and external trigger input.

Branching conditions
Conditional and multiway branching on a logical combination of the branching inputs with IF/ELSE, CASE, WHILE and REPEAT constructs unconditional branching on completion of state actions using GOTO construction.

State actions
Start a named block of bus transactions (defined in bus transaction editor).
Execute a named block of transactions with automatic retry after target disconnect/retry.
Assert external trigger output (SNB).

Bus exerciser card

Transaction memory



Transactions are defined using address and data phases up to a maximum burst length of 1,750 data phases for a single transaction. Multiple transactions can be defined up to the maximum transaction memory depth of 13,950 phases.

Pattern recognizor
Up to seven logical combinations of signal patterns can be defined. Occurrences of these patterns are available to the test sequencer for triggering/branching. The pattern recognizor has access to the following signals:
- all PCI Bus signals
- internal state signals from the bus exerciser's master/target state machines
- protocol violation signal from protocol monitor

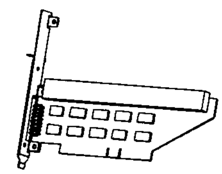
When programmed to retry terminated transactions, five of the pattern terms are predefined.

Protocol monitor
The protocol monitor permanently monitors the bus control signals and checks 25 PCI protocol rules in real-time. A violation signal is available to the pattern recognizor and logic analyzer for triggering or branching. The data listers also check the traffic data captured by the logic analyzer against the same protocol rules and indicate which rule was violated.

Sideband signals
In addition to the PCI bus signals, the bus exerciser provides sideband signals which can be used to drive or monitor and trigger from other signals in the system under test.

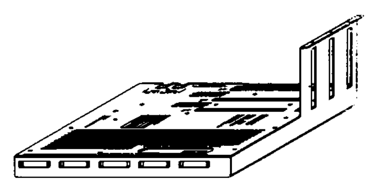
Fixturing for in-system and stand-alone test

Adapters for in-system test



Adapt theHP E2910A for in-system testing at PCI slots
- HP E2911A for 5 V signalling environment
- HP E2913A for 3.3 V signalling environment

Adapters for stand-alone card test



Adapt the HP E2910A to support stand-alone (without PCI mother-board) testing of up to three PCI cards. The HP logic analyzer is still required to observe bus traffic.
- three PCI slots
- PCI arbiter
- mechanical housing
- connector for timing analysis with HP 16517/18A
- **HP E2912A** for 5 V signalling environment.
- **HP E2914A** for 3.3 V signalling environment

Configuration guide

Logic analyzer support

The HP 16500B logic analyzer system mainframe is integrated into the system via HP-IB and HP 16550A or 16554/5/6A logic analyzer cards are used to record bus/system activity for analysis. Up to 4k state data is uploaded to the PC for disassembly and analysis in the HP E2910A data listers (Timing data may also be analyzed). Standard PCI set-ups for the logic analyzer cards are included, and any modified set-ups can be transferred to and from the PC, or stored locally on the analyzer. You have full control over the logic analyzer via its local user interface, allowing you to use the logic analyzer trigger capabilities to filter and capture the bus activity you are interested in, for example:

- filtering out idle states
- capturing configuration transactions only
- capture acceses to a particular address range

System configuration

To use the HP E2910A PCI bus exerciser, the following additional equipment is required:

Computer
PC: IBM-PC or 100% compatible with recommended minimum 66MHz 486 CPU and 3.5“ floppy drive.

Graphics: 640 x 480 VGA minimum. 1024 x 768 SVGA recommended.

O/S: MS-DOS version 5.0 with Windows 3.1 required.

Memory: 16 MB minimum, 24MB recommended.

Hard Disk: minimum 50MB available disk space required, 100MB recommended.

Expansion Slots: two ISA slots required for test sequencer card and HP-IB card.

I/O Interface: HP-IB interface card supplied with HP E2910A requires ISA slot in PC.

Logic analyzer: HP 16500B logic analyzer system mainframe with HP 16550A or HP 16554/5/6A logic analyzer card(s):Up to 4k of logic analyzer state data, or equivalentiming data, can be analyzed by the HP E2910A. 16554/5/6A memory depth must be set to 4k for upload into HP E2910A software.

	Number of LA Cards required:	
LA Card	**32 Bit PCI**	**64 Bit PCI**
16550A	1	2
16554A 16555A 16556A	2	3

Adapters

At least one of the following adapters is required to adapt the HP E2910A PCI bus exerciser for in-system or card test:
HP E2911A: 5V in-system
HP E2912A: 5V card
HP E2913A: 3.3V in-system
HP E2914A: 3.3V card

General specifications

Operating temperature range: +20°C to +30°C
Safety: IEC1010, CSA1010

E2910A PCI bus exerciser:
Product includes:
- software for MS-Windows 3.1
- PCI bus exerciser card
- test sequencer card (ISA) for PC
- power supply HP 15291A
- cables for logic analyzer
- HP-IB card and cable

HP 15291A power supply power requirements: 100-240V+/-10%, 50-60Hz, 300VA max.

Test sequencer card: requires 8-bit or 16-bit ISA slot. Occupies I/O-space only (DIP-Switch). No interrupts required.

Power consumption: 4 A max. @ +5V

Length: 175mm

HP E2910A PCI specification

The following specifications use the outline of the PCI Rev 2.1 specification. Specifications applicable to a particular adapter are indicated by the adapter product number:
HP E2911A: 5 V adapter for in-system test
HP E2912A: 5 V adapter for stand-alone card test
HP E2913A: 3.3 V adapter for in-system test
HP E2914A: 3.3 V adapter for stand-alone card test

2. Signal definition	All PCI bus signals (except JTAG), eight sideband signals (HP E2911A, E2913A only) and some internal state information can be observed by the logic analyzer and used by the pattern recognizor to change the test sequence in real time.
2.2.6. Interrupts pins (optional)	Any of the four interrupts can be asserted at any time.
2.2.7. Cache support pins (optional)	Not implemented/supported. **HP E2912A, HP E2914A:** SBO# and SDONE pulled up.
2.2.8 Additional signals	PRSNT# Pins: see section 4.4.1 CLKRUN#: Not supported since not defined for connector.
2.2.9. 64-bit bus extension pins (optional)	Master and target support 64-bit data transfers. A second HP 16550A required analyzer card to observe 64 Bit signals.
2.2.9. JTAG/boundary scan pins (optional)	Not supported. **HP E2911A, HP E2913A:** TDI is hardwired to TDO. **HP E2912A, HP E2914A:** not connected.
2.3. Sideband signals	**HP E2911A, HP E2913A:** eight input/output sideband signals are observable at the logic analyzer (2nd HP 16550A card required) and pattern recognizor to influence the test sequence. four open-drain and four totem-pole sideband outputs can be programmed on a single clock cycle basis as part of the intended test. **HP E2912A, HP E2914A:** no sideband signals.
Reserved signals	Observable at separate connector to detect unexpected usage.
3.1. Bus commands	Master: all command types, including ”reserved“, can be generated. Target: can be programmed to react on any set of command types. All read/write cycles are aliased to memory read/memory write commands.
3.2.2. Addressing	Master: supports 32-bit addressing, configuration addressing (see 3.6.4) and 64 Bit addressing (dual address cycle). Target: can detect up to two independent 32 Bit address ranges (”01X“ patterns on AD[31::0]) or IDSEL and qualify with any set of command types C/BE[3::0]. Can detect one 64 Bit address range (01x pattern, see 3.9.1).
3.2.3. Byte alignment	Master: any combination of AD[1::0] and C/BE[3::0] can be programmed. Target: always returns all data bytes.
3.3 Bus transactions	Any command can be programmed as burst or single transfer. 0 to 30 wait states can be generated in any master or target data phase. More waits can be inserted using data_step commands.
3.3.3.1. Master initiated termination	Completion: the length of a burst is programmable. Time-out: master terminates as part of the intended test, not as a reaction to the deassertion of GNT#. Master abort: six cycles after FRAME# is asserted, if target does not respond.

3.3.3.2. Target initiated termination

Master: can be programmed to retry a target-terminated access, starting at the address of the next untransferred data.
Target: terminations can be programmed in any target data phase. The assertion of STOP# can be placed on a clock cycle basis using the "wait" parameter.
Disconnect A/B: terminates after data transfer.
Retry/disconnect C: terminates without data transfer.
Target abort: terminates without data transfer and deasserts DEVSEL#.

3.4. Arbitration

Master: can request the bus independent of or in combination with transactions.
HP E2912A, HP E2914A: contain arbiter with fixed priorities for slot 1 (highest), HP E2910A, slot 2, slot 3 (lowest). This provides a more predictable test environment than a fairness-based algorithm.

3.4.2. Fast back-to-back transactions

Master: fast back-to-back can be selected as part of intended test.
Target: supports fast back-to-back.

3.4.3. Arbitration parking

HP E2912A, HP E2914A: the central arbiter uses HP E2910A as a parking master which drives AD[63::0] and C/BE[7::0], PAR and PAR64 to a stable value.

3.5 Latency on PCI

Arbitration latency: 1 clock cycle fixed (**HP E2912A, HP E2914A**, see 3.4).
Bus acquisition latency: one or more clock cycles programmable.
Data transfer latency (waits): For both master and target any number of wait cycles between 0 and 30 can be programmable individually per data phase. Additional waits can be inserted using data_step commands.
Master latency timer: none, see 3.3.3.1.

3.6. Exclusive access

Master: can start a locked access and complete it at the end of the next transaction or later.
Target: locks its whole address range.

3.6.1 Starting an exclusive access

Master: releases LOCK# after master or target abort.

3.6.3 Accessing a locked agent

Master: withholds a locked access while another master has LOCK# asserted.
Target: signals retry when locked and accessed with a non-exclusive access.

3.6.4 Completing an exclusive access

Master: LOCK# is released one clock cycle after the last data in a burst.

3.6.6. Complete bus lock

HP E2912A, HP E2914A: not supported

3.7.1. Device selection

Target: can detect up to two independent 32 Bit address ranges ("01X" patterns on AD[31::0]) or IDSEL and qualify it with any set of commands types C/BE[3::0].
Can detect one 64 Bit address range (01x pattern, see 3.9.1).

3.7.2. Special cycle

Master: can initiate a special cycle with programmable message. Bursted special cycle is not supported.

3.7.3. Address/data stepping

Any number of discrete steps, with programmable values for AD[63::0], C/BE[7::0], PAR, and PAR64 for each step, can be inserted in the address phase (master only) or data phase (master or target). Continuous stepping is not implemented.

3.7.4. (Accepting) configuration cycles

Target: can react on IDSEL qualified by any set of command types and present any data (see 6).

3.7.4.1. Generating configuration cycles

Master can generate all types of configuration cycle (HP E2911A: only if the motherboard connects IDSELx to address lines).

3.7.5. Interrupt acknowledge

Master: can generate an interrupt acknowledge cycle.
Target: can respond to an interrupt acknowledge cycle as part of the intended test.

3.8.1. Parity

Address or data phases can individually be programmed to generate correct or wrong PAR/ PAR64 bits.

3.8.2.1 PERR#

Master and target can assert PERR# 2 clock cycles after any read/write data transfer as part of the intended test. Although parity errors can be detected they are not used to assert PERR#.

3.8.2.2 SERR#

Master and target can assert SERR# at any time as part of the intended test.

3.9. Cache support

Not implemented/supported.

3.10. 64-Bit bus extension

Master: can initiate 64-bit data transfers.
Target: each transaction can individually accept or refuse 64-bit data accesses as part of the intended test.

3.10.1. 64-bit addressing on PCI

Master: can initiate a dual address cycle. Optionally AD[63::32] and C/BE[7::4] can be driven as well
Target: can decode one 64-bit address (01x pattern). HI-ADDR and command are decoded from AD[31::0], C/BE[3::0].

4.2.1. 5V signaling environment

The following adapters supported the 5 V signalling environment:
HP E2911A: 5 V adapter for in-system test.
HP E2912A: 5 V adapter for stand-alone card test.
HP E2911A, HP E2912A: data outputs are driven by 74FCT16823ET registered buffers, and sensed by 74ABT16245 buffers. Control outputs are driven by 74ABT25245 incident wave switching buffers, and sensed by 74FCT16245ET buffers.
HP E2912A: 3.3V is supplied to the 3.3V power pins, allowing 5 V signalling cards which have some devices powered from 3.3 V to be tested.
HP E2911A: all buffers are powered from the PCI connector.

4.2.2. 3.3V signaling environment

The following adapters support the 3.3 V signalling environment:
HP E2913A: 3.3 V adapter for in-system test.
HP E2914A: 3.3 V adapter for stand-alone card test.
HP E2913A, HP E2914A: data outputs are driven by 74LVT16952 registered buffers, and sensed by 74LVT16245 buffers. Control outputs are driven by 74LVT244 buffers (selected as incident wave switching devices), and sensed by 74LVT16245 buffers.
HP E2913A: all buffers are powered from the PCI connector.

4.2.3.1. Clock specification

Operates at any frequency between DC and 33 MHz. The frequency must be known to be in one of the ranges: <5 MHz, 5 - 20 MHz, or >20 MHz. In the two highest ranges the frequency must be stable.
The range < 5 MHz is intended for very slow devices such as ASIC emulators and expects a large hold-time.
HP E2911A, HP E2913A: clocked from the PCI connector. HP E2912A, E2914A: each slot can be clocked from internal oscillator (33.3 MHz or 16.6 MHz) or from one of two BNC-type external clock inputs.

4.2.3.2. Timing parameters

Above 5 MHz

CLOCK FREQUENCY > 5 MHz

Symbol	Min	Max	Notes
tva (data)l	2 ns	18 ns	1, 6
tval (control)	2 ns	18 ns	2, 5, 6
tval (PAR)	2 ns	20 ns	3, 5, 6
tval (ptp)	2 ns	18 ns	4, 5, 6
ton	2 ns		
toff		28 ns	
tsu	7 ns		
tsu (ptp)	10 ns		
th	0 ns		

Notes:
All transitions measured at 1.5 V.
1. AD[63::0], C/BE#[7::0], SERR#, PERR#, INTA-D#
2. FRAME#, DEVSEL#, IDSEL, TRDY#, IRDY#, LOCK#, STOP#, REQ64#, ACK64#
3. PAR, PAR64
4. GNT#, REQ#
5. Driven by incident wave switching drivers (**HP E2911/12A:** 74ABT25245, **HP E2913/14A:** selected 74LVT244)
6. Exceeds limit set by PCI Spec. Rev 2.1

Below 5 MHz

CLOCK FREQUENCY < 5 MHz

Symbol	Min	Max
tval		60 ns
tsu	0 ns	
th	40 ns	

4.3 System (motherboard) specification

Section 4.3 applies to add-in card adapters HP E2912A and HP E2914A only

4.3.1 Clock skew (motherboard)

2 ns max.

4.3.2. Reset

RST# is under SW control.

4.3.4.1. Power requirements

HP E2912A: provides 6A @ 5V/3.3V, 1.5A @ +12V, and 0.5A @ -12V for 3 PCI slots together.
HP E2914A: provides 6A @ 3.3V/5V, 1.5A @ +12V, and 0.5A @ -12V for 3 PCI slots together.

4.3.6.2 Motherboard impedance

45 Ohm typical.

4.3.7 Connector pin assignment

PRSNTx# signals of all three slots can be observed at separate connector.

4.4 Expansion board specification

Section 4.4 applies to in-system adapters HP E2911A and HP E2913A only

4.4.1. Board pin assignment

PRSNTx# signals are programmable.

4.4.2 Power requirements

HP E2911A: consumes 0.5A @ 5V from PCI slot for I/O buffers
HP 2913A: consumes 0.5A @ 3.3V from PCI slot for I/O buffers.

4.4.3.1. Trace length limits

All PCI signal traces are max. 1.5 inches (32-bit), 2 inches (64-bit). CLK signal trace is 2.5 inches.

4.4.3.3. Impedance

80 Ohm typical.

4.4.3.4. Signal loading

HP E2911A: AD[63::0] and C/BE[7::0] 14 pF typical.
Control signals 20 pF typical.
HP E2913A: AD[63::0] and C/BE[7::0] 18 pF typical.
Control signals 20 pF typical.
All signals are connected to one input buffer and one output buffer on opposite sides of the PC board.

4.4.x Reset

RST# tristates all output buffers and resets all bus statemachines on HP E2910A board.

5.2 Expansion card physical dimensions

Length: like a long card. Close to the PCI connector shorter than a short card (160 mm).
Height: 265 mm including HP E2910A and cables to logic analyzer.
Connector bevel: 20 degrees.

5.2.1 Connector physical description

HP E2911A: 5 V 64-bit card edge connector.
HP E2913A: universal 64-bit card edge connector.
HP E2912A: 5 V 64-bit connector.
HP E2914A: 3.3V/64-bit connector.

6. Configuration space	HP E2910A does not interactively participate in the configuration phase. However, the intended test can be programmed to emulate any device's configuration space, (e.g. device with multiple memory spaces, bus bridge, multi-function device.) by presenting a set of answers to expected queries. Even strange, inconsistent or wrong data can be programmed to stress the BIOS.
7. 66 MHz PCI	Not supported.

For more information on Hewlett-Packard Test & Measurement products, applications or services please call your local Hewlett-Packard salesoffices. A current listing is available via Web through AccessHP at http://www.hp.com.

If you do not have access to the internet please contact one of the HP centers listed below and they will direct you to your nearest HP representative.

United States:
Hewlett-Packard Company
Test and Measurement Organization
5301 Stevens Creek Blvd.
Bldg. 51L-SC
Santa Clara, CA 95052-8059
1 800 452 4844

Canada:
Hewlett-Packard Canada Ltd.
5150 Spectrum Way
Mississauga, Ontario
L4W 5G1
(905) 206 4725

Europe:
Hewlett-Packard
European Marketing Centre
P.O. Box 999
1180 AZ Amstelveen
The Netherlands

Japan:
Hewlett-Packard Japan Ltd.
Measurement Assistance Center
9-1, Takakura-Cho, Hachioji-Shi,
Tokyo 192, Japan
Tel: (81-426) 48-0722
Fax: (81-426) 48-1073

Latin America:
Hewlett-Packard
Latin American Region Headquarters
5200 Blue Lagoon Drive
9th Floor
Miami, Florida 33126
U.S.A.
(305) 267 4245/4220

Australia/New Zealand:
Hewlett-Packard Australia Ltd.
31-41 Joseph Street
Blackburn, Victoria 3130
Australia
131 347 ext. 2902

Asia Pacific:
Hewlett-Packard Asia Pacific Ltd
17-21/F Shell Tower, Times Square,
1 Matheson Street, Causeway Bay,
Hong Kong
(852) 2599 7070

Data subject to change

Hewlett-Packard Company
Printed in USA 07/96 (CT)
5965-1438E