CPE 186 Computer Hardware Design

Configuration Transactions (Part 2)

Dr. Pang

Type 0 Configuration Transaction

Address Phase

- During any PCI transaction, all PCI devices on the bus latch the following information at the end of the address phase:
- The contents of the AD bus.
    - ❖In a configuration transaction, this consists of the target function, Configuration dword and 00b on the least-significant two bits if it's a Type 0 configuration transaction, or the target bus number, device number, function number, dword number and 01b on the least-significant two bits if it's a Type 1 configuration transaction.
- The state of the FRAME# signal (asserted, indicating that a valid address and command have been latched from the bus).
- The state of the IDSEL# signal **(**only has meaning if this is a Type 0 configuration transaction).
    - ❖The PCI device (i.e., package) that samples its IDSEL asserted is the target PCI device. The bridge implements a separate IDSEL signal for each PCI device implemented on its secondary bus (see Figure 18-5 on page 337).

Type 0 Configuration Transaction

Address Phase

During any PCI transaction, all PCI devices on the bus latch the following information at the end of the address phase:

- The command on C/BE#[3:0]. In this case, it indicates that this is a configuration read or write transaction. The command type is derived from the type of access the processor is performing with the host/PCI bridge's Configuration Data Port. An IO read converts to a configuration read and an IO write converts to a configuration write.
- 00b on AD[1:0] indicates that this is a Type 0 configuration transaction targeting one of the devices on this PCI bus. In essence, the 00b is a shorthand way of indicating that the bridge has already done the target bus comparison and established that the transaction is targeting a device on this bus.

Type 0 Configuration Transaction

- The PCI device that samples its IDSEL asserted is the target device.
- AD[1:0] are 00b, ‘Type 0’ transaction targeting one of the devices on this bus.
- AD[7:2] : the target configuration dword**.**
- AD[10:8]: the target function within the physical device.
- AD[31:11]: reserved.

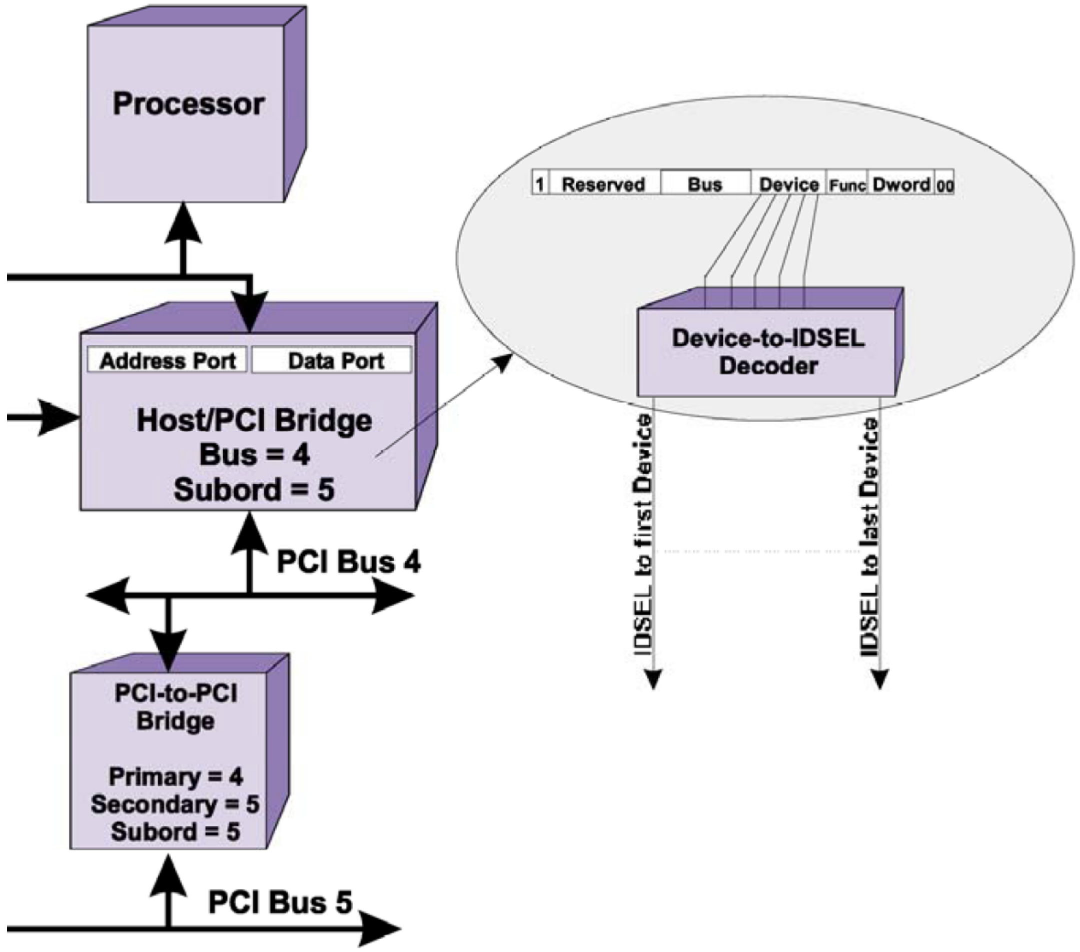
Type 0 Configuration Transaction

- The target device number specified in bits [15:11] in the Configuration Address Port are decoded within the bridge) and the decoder asserts the appropriate IDSEL output signal during the transaction's address phase.
- If the bridge's device decoder determines that there is no device (or card slot) implemented at the target device position on its secondary bus, it will not assert any IDSEL outputs during the transaction's address phase. The transaction will end up experiencing a Master Abort because no target will assert DEVSEL# to claim the transaction.

Type 0 Configuration Transaction

- Device Number field within the Configuration Address Port: a 5-bit field (device number from 0-through-31d as the target device). Obviously, a PCI bus with 32 devices implemented on it (each one presenting a load to the bus) **is** not going to function correctly.
- In reality, a 33MHz PCI bus typically supports no more than 10 devices on the bus.
- The bottom line is that most of the 32d possible device positions on any PCI bus are going to be unoccupied.
- There is no rule. however, that dictates which device position each device must occupy.
- They don't even have to be implemented as contiguous device number, but that wouldn't make a lot of sense.
- As some examples, the system board designer could attach devices to the following IDSEL signals (which correlate to device numbers):
  - 0-through-8.
  - 4-through-9.
  - 16-through-25.
  - 19-through-26.

Bridge’s Device Decode



Implementation of IDSEL

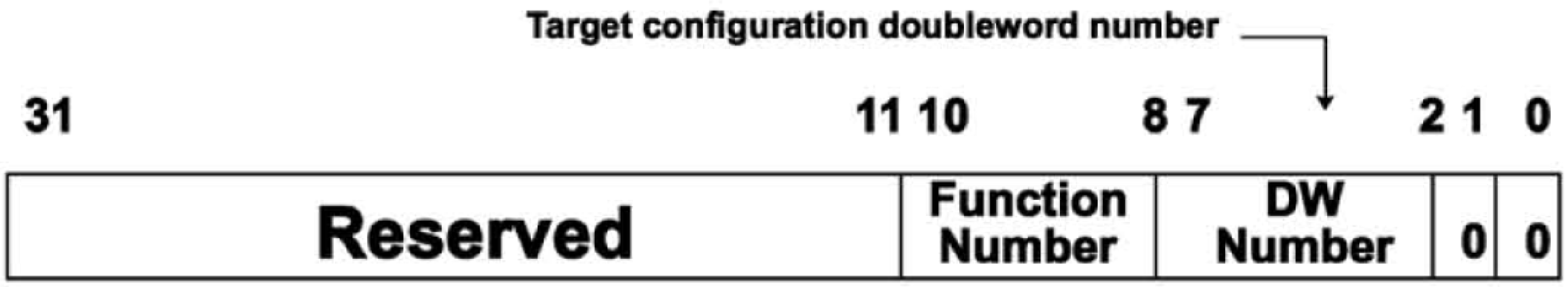
The IDSEL outputs can be Implemented in one of two ways by the host/PCI bridge designer.

- Be aware that PCI-to-PCI bridges must use the first method.

Method One - lDSELs Routed Over Unused AD Lines

- This is the method used by most host/PCI bridge and system board designers.
- The upper twenty-one address lines AD[31:11], are not used during the address phase of a Type 0 configuration access. The system board designer is therefore free to use these signal Lines as IDSEL signals to the various physical PCI packages (up to twenty-one of them).
- Internally, the bridge decodes the target Device number contained in Configuration Address Port bits [15:11] to select which AD line to set to one. Each of these address lines is connected to the IDSEL input of a separate PCI device.

Contents of the AD Bus During Address Phase of a Type 0 Configuration Access



Method One - lDSELs Routed Over Unused AD Lines (continued)

- This approach places an additional load on the AD line, however, and it's a rule that each PCI device is only permitted to place one electrical load on each PCI bus signal.

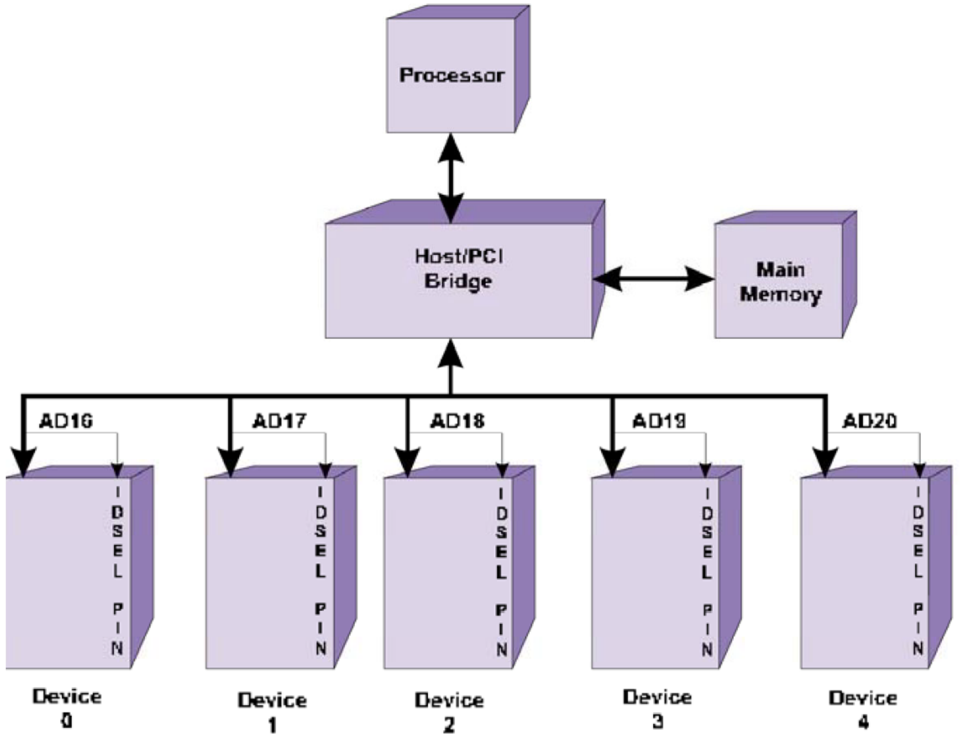
Method One - lDSELs Routed Over Unused AD Lines (continued)

- As an example of upper AD line-to-IDSEL mapping, AD11 could be set to one if the target device is device zero (the first PCI device); AD12 set to one If the target device is device one, etc. The specification suggests (it's only a suggestion) the following mapping. The bridge internally decodes the device number field in the Configuration Address Port and selects an IDSEL signal to assert**.** Rather than implementing IDSEL output pins, the IDSEL signals internal to the bridge are directed to AD[31:16] in the following manner:
  - ❑ The IDSEL associated with device 0 is connected to AD16.
  - ❑ The lDSEL associated with device i is connected to AD17.
  - ❑ The IDSEL associated with device 2 is connected to AD18, etc.
  - ❑ The IDSEL associated with device 15 is connected to AD31.
  - ❑ For devices 16 through 31, none of the upper AD lines should be asserted when the Type 0 configuration transaction is performed. Since no device detects its IDSEL asserted, none respond, resulting in a Master Abort by the bridge.

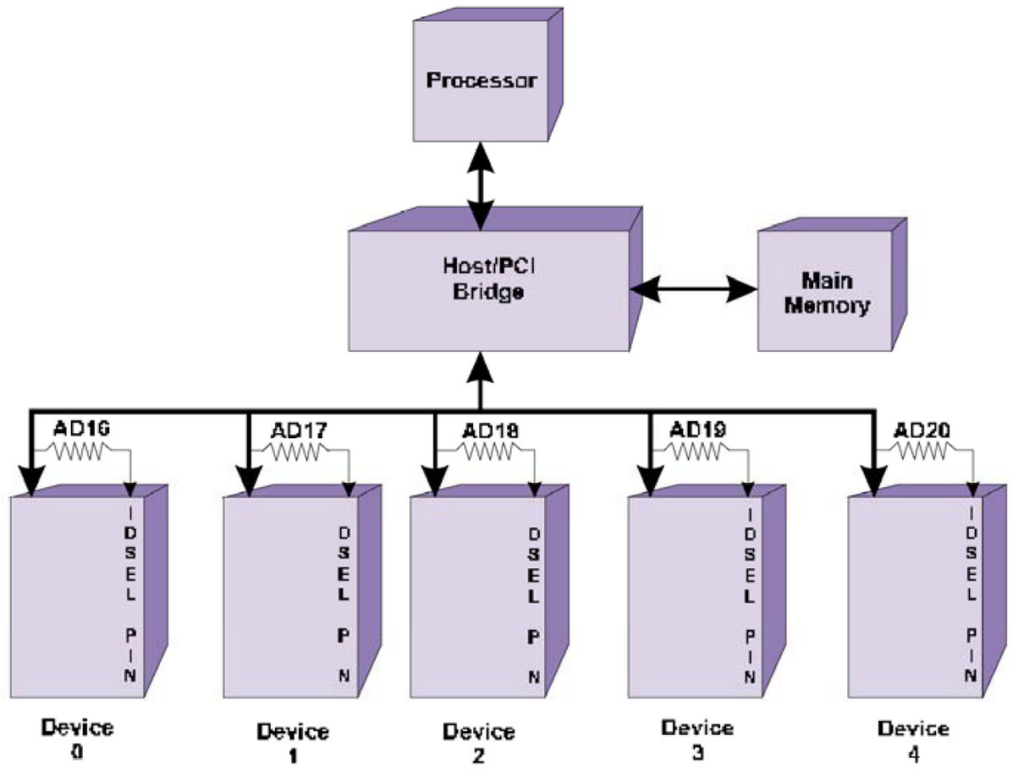
Method One - lDSELs Routed Over Unused AD Lines (continued)

- This approach supports the implementation of 16 devices on a PCI bus (from a configuration standpoint; from a realistic electrical loading standpoint, you're only going to place 10 to 11 loads on a 33MHz bus and have it work correctly).

Direct Connection of Device IDSEL Pins to Upper AD Lines



Resistive-Coupling Device IDSEL Pins to Upper AD Lines



Method Two IDSEL Output Pin/Traces

The host/PCI bridge designer can decode bits [15:11] (target physical Device number) in the Configuration Address Port and assert the target physical device's IDSEL output signal line. This method requires the implementation of a separate IDSEL output pin on the bridge for each physical package on the PCI bus and a separate point-to-point IDSEL trace on the system board between the bridge and each physical PCI device or connector. Most host/PCI bridge designs don't implement IDSEL output pins because it is a pin- and trace-intensive solution.

Resistive Coupling Means Stepping In Type 0 Transactions

- When the IDSELs are resistively-coupled to upper AD lines, it takes some time for the ones or zeros on the upper AD lines to traverse the resistors and arrive at the correct value at each device's IDSEL input pin. For this reason, the host/PCI bridge must use address stepping (see 'Designer May Step Address, Data, PAR (and PAR64) and IDSEL" on page 164). but only for type 0 configuration transactions.
- The host/PCI bridge initiates the Type **0** configuration transaction by driving out the target function and dword number on the AD bus with AD[1:0] set to 00b to indicate that this is a Type 0 configuration transaction. It also outputs its internal device decoder's IDSEL output signals onto the upper AD lines. The configuration read or write command is driven onto C/BE#[3:0].
- However, the bridge doesn't assert FRAME# yet. It delays a sufficient number of clocks to let the bits on the upper AD lines propagate through the resistors to the IDSEL pins at the devices and settle to the correct state and then asserts FRAME#.
- No devices will pay any attention to the transaction until FRAME# is asserted.

Data Phase Entered, Decode Begins

As the data phase is entered, the bridge sets C/BE#[3:0] to indicate which bytes it wishes to transfer within the currently-addressed configuration dword. It gets this information from the processor's access to the bridge's Configuration Data Port which is a one-byte **(**so just assert C/BE0#), two-byte (assert C/BE#[1:0] or four-byte (assert all four C/BE lines) read or write.

As the data phase is entered, the PCI devices are performing the address decode to determine which of them is the target of the transaction (00b on AD[1:0] indicates it is for one of them). Devices that sampled their IDSEL inputs deasserted at the end of the address phase ignore the transaction. When a device detects its IDSEL pin was asserted at the end of the address phase. It must determine whether or not to claim the transaction. How it does this depends on whether it is a single- or multi-function device:

Data Phase Entered, Decode Begins

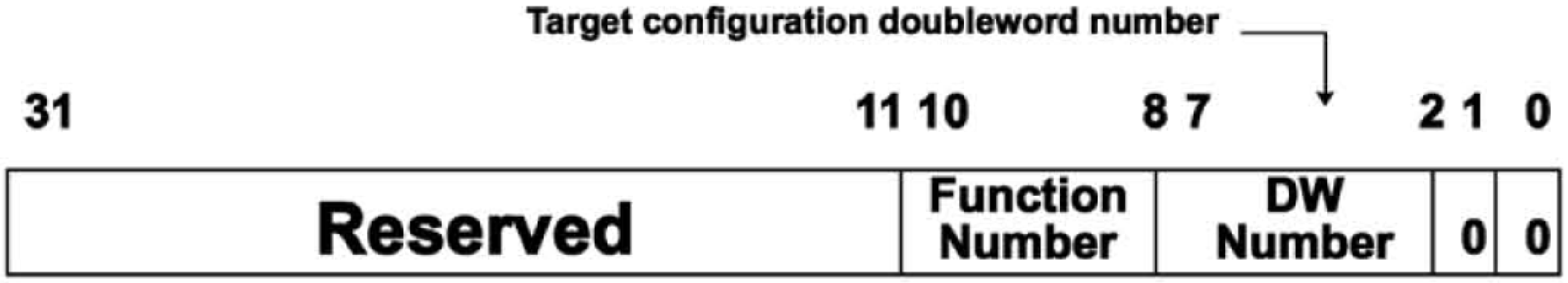
2.2 SPEC states that a single function device can either:

- Decode the function number and only assert DEVSEL# for function zero.
- Or may respond to all FUNCTION numbers by asserting DEVSEL#.

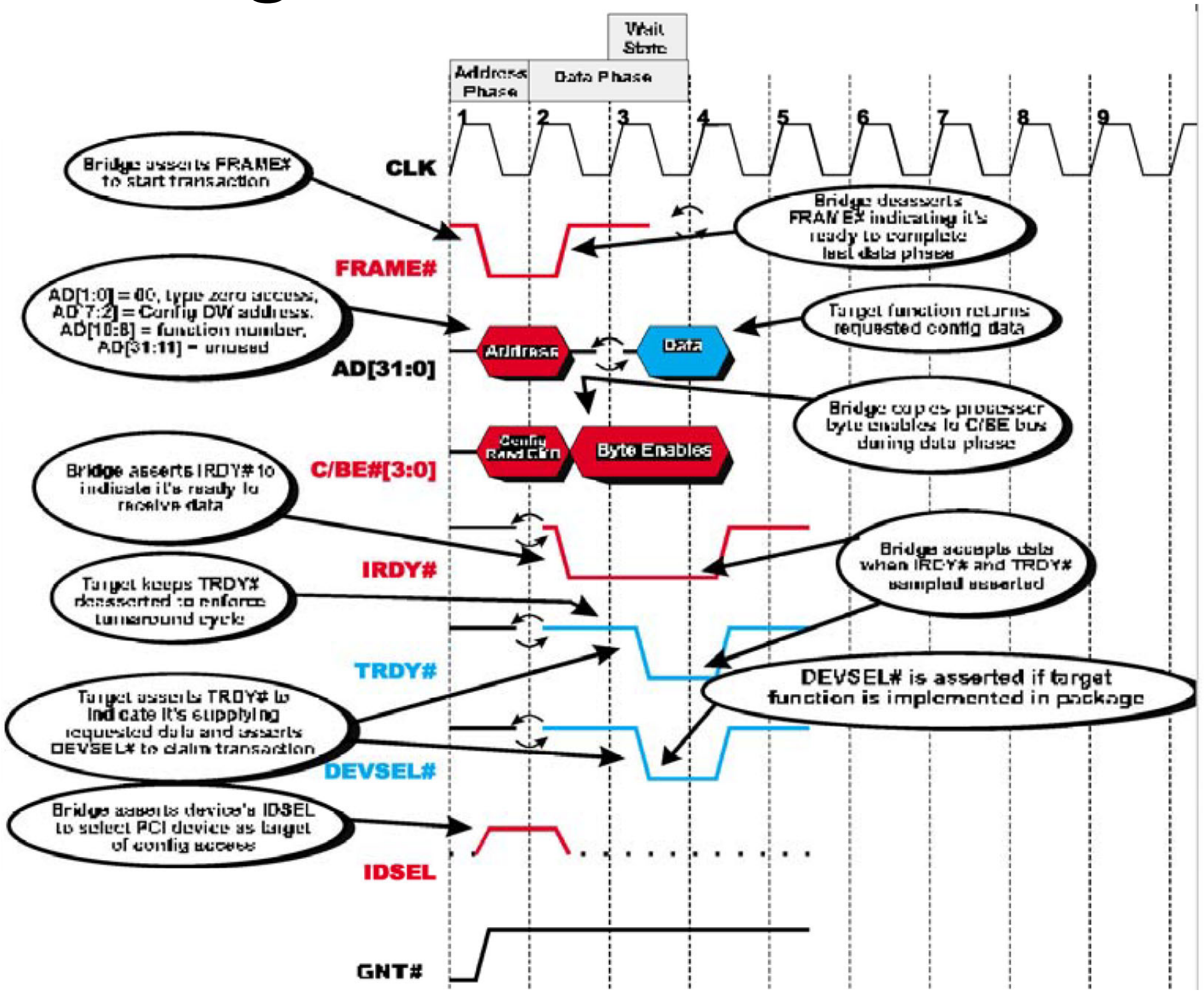
Data Phase Entered, Decode Begins

- If it’s a multi-function device, it must implement a function decoder and it must decode the function number delivered on AD[10:8] during the address phase, if the target function is implemented, the device asserts DEVSEL# and claims the transaction. Otherwise, it ignores the transaction (because the target function is not implemented in this device.)

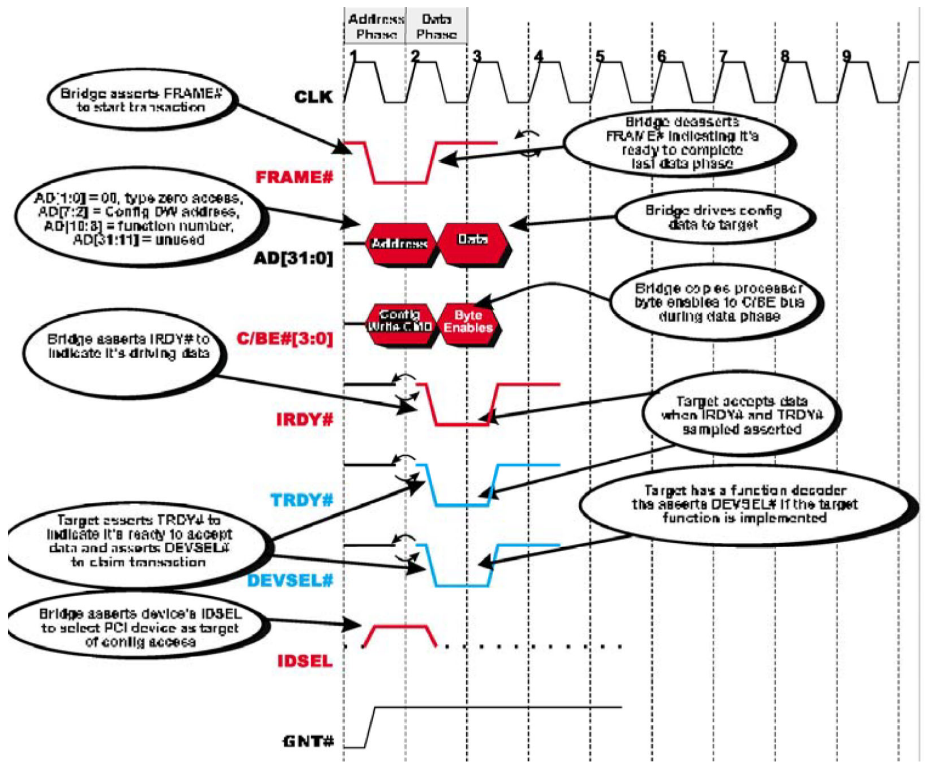
Contents of the AD Bus During Address Phase of a Type 0 Configuration Access



Type 0 Configuration Read Access

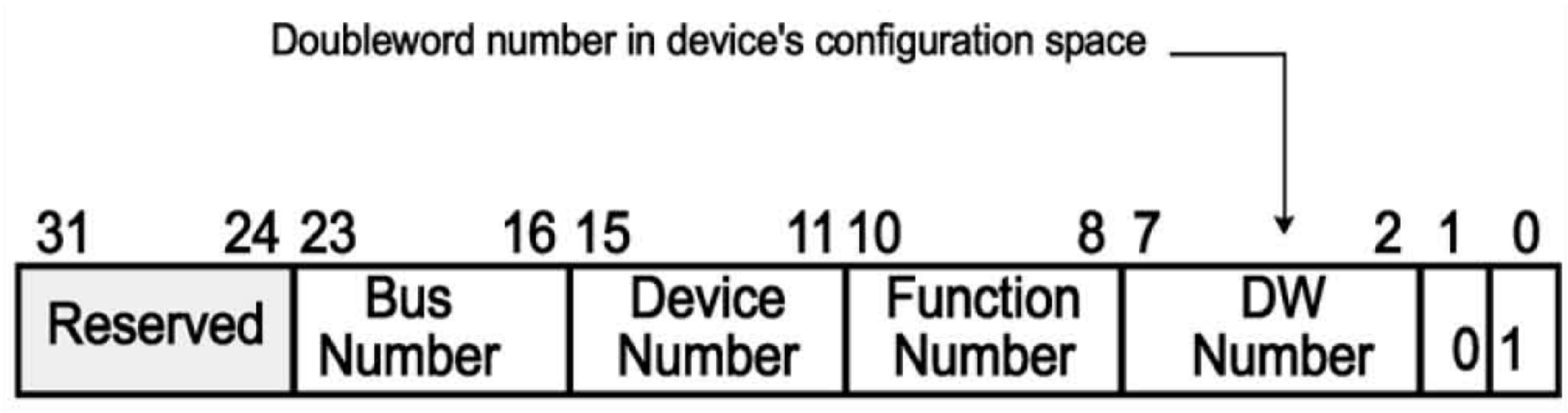

Wait State
Address Phase
Data Phase
1
2
3
4
5
6
7
8
9
CLK
Bridge asserts FRAME# to start transaction
Bridge deasserts FRAME# indicating it's ready to complete last data phase
FRAME#
AD[1:0] = 00, type zero access, AD[7:2] = Config DW address, AD[10:8] = function number, AD[31:11] = unused
Address
Data
Target function returns requested config data
AD[31:0]
Bridge copies processor byte enables to C/BE bus during data phase
Config Read Cmd
Byte Enables
C/BE#[3:0]
Bridge asserts IRDY# to indicate it's ready to receive data
IRDY#
Bridge accepts data when IRDY# and TRDY# sampled asserted
Target keeps TRDY# deasserted to enforce turnaround cycle
TRDY#
DEVSEL# is asserted if target function is implemented in package
Target asserts TRDY# to indicate it's supplying requested data and asserts DEVSEL# to claim transaction
DEVSEL#
Bridge asserts device's IDSEL to select PCI device as target of config access
IDSEL
GNT#

Type 0 Configuration Write Access



Address Phase
Data Phase
1
2
3
4
5
6
7
8
9
CLK
Bridge asserts FRAME# to start transaction
Bridge deasserts FRAME# indicating it's ready to complete last data phase
FRAME#
AD[1:0] = 00, type zero access, AD[7:2] = Config DW address, AD[10:8] = function number, AD[31:11] = unused
Bridge drives config data to target
Address
Data
AD[31:0]
Bridge copies processor byte enables to C/BE bus during data phase
Config Write CMD
Byte Enables
C/BE#[3:0]
Bridge asserts IRDY# to indicate it's driving data
IRDY#
Target accepts data when IRDY# and TRDY# sampled asserted
TRDY#
Target has a function decoder the asserts DEVSEL# if the target function is implemented
Target asserts TRDY# to indicate it's ready to accept data and asserts DEVSEL# to claim transaction
DEVSEL#
Bridge asserts device's IDSEL to select PCI device as target of config access
IDSEL
GNT#

Contents of the AD Bus During Address Phase of a Type 1 Configuration Access



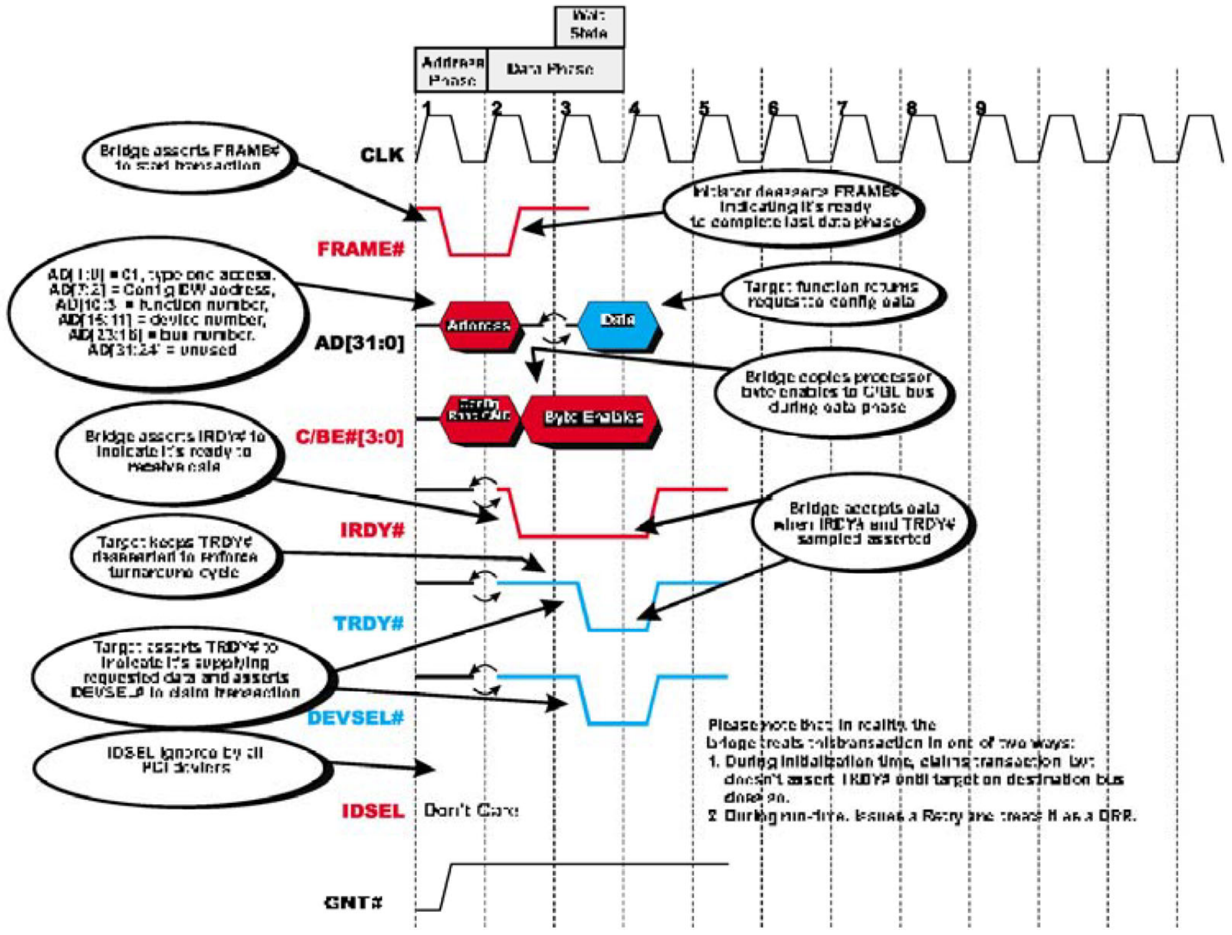
Type 1 Configuration Access Example (continued)

- When any PCI-to-PCI bridge latches a Type 1 configuration access (command = configuration read or write and AD[I:0] = 01 b) on its primary side, it must determine which of the following actions to take:
- **ACTION 1.** If the bus number field on the AD bus doesn't match the number of its secondary bus and isn't within the range of its subordinate buses, the bridge should ignore the access.
- **ACTION** 2. If the bus number field matches the bus number of its secondary bus, it should claim and pass the configuration access onto its secondary bus as a Type **0** configuration access. AD[1:0] on the secondary bus are set to 00**b** (Indicating a Type 0 access). AD[10:2] (target function and dword) are passed as is to its secondary AD bus. The device number field is decoded within the bridge to select one of the IDSEL lines to assert on the secondary bus. The configuration command is passed from the primary to the secondary C/BE bus.

Type 1 Configuration Access Example (continued)

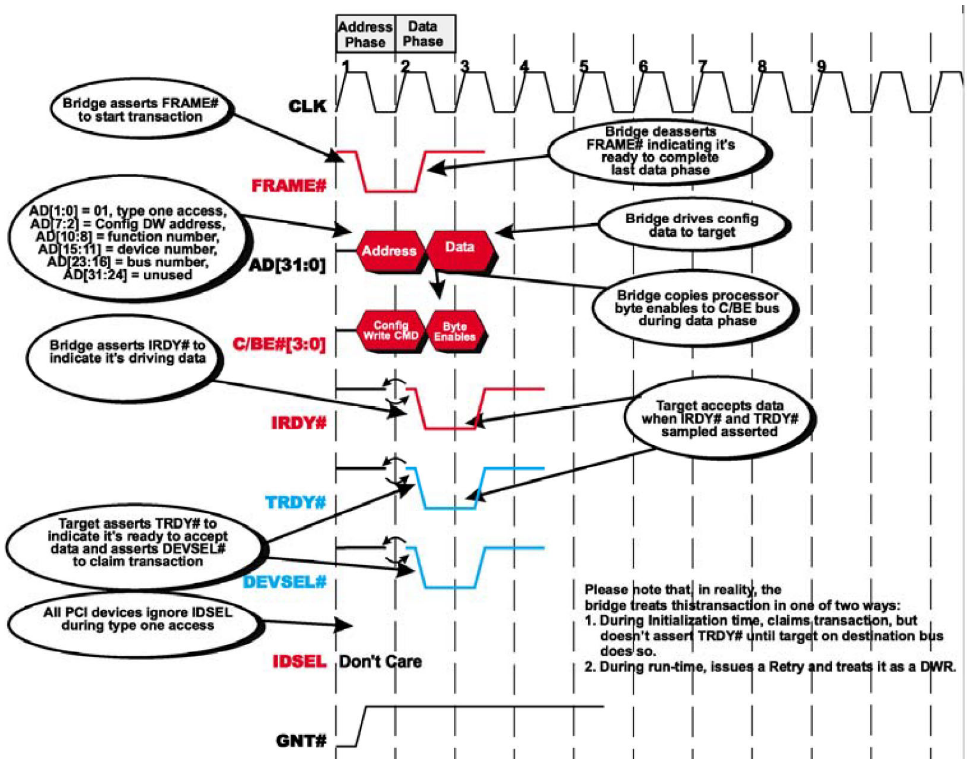
- **ACTION** ***3.*** If the bus number field isn't equal to its secondary bus, but is within the range of buses that are subordinate to the bridge, the bridge claims and passes the access through as a Type 1 access**.** AD[31:0] (target bus, device, function and dword) are passed to the secondary AD bus as is. AD[1:0] are set to 01b, indicating that a Type 1 access is in progress on the secondary bus. The configuration command is passed from the primary to the secondary C/BE bus.
- Each PCI-to-PCI bridge is required to implement a Primary Bus Number register, a Secondary Bus Number register, and a Subordinate Bus Number register.
- PCI-to-PCI bridges on PCI bus 0 are discovered and numbered during the configuration process.

Type 1 Configuration Read ACESS


Wait State
Address Phase
Data Phase
1
2
3
4
5
6
7
8
9
CLK
FRAME#
AD[31:0]
C/BE#[3:0]
IRDY#
TRDY#
DEVSEL#
IDSEL
GNT#
Address
Data
Config Read
Byte Enables
Don't Care
Bridge asserts FRAME# to start transaction
AD[1:0] = 01, type one access. AD[7:2] = Config DW address, AD[10:8] = function number, AD[15:11] = device number, AD[23:16] = bus number. AD[31:24] = unused
Bridge asserts IRDY# to indicate it's ready to receive data
Target keeps TRDY# deasserted to enforce turnaround cycle
Target asserts TRDY# to indicate it's supplying requested data and asserts DEVSEL# to claim transaction
IDSEL ignored by all PCI devices
Initiator deasserts FRAME# indicating it's ready to complete last data phase
Target function returns requested config data
Bridge copies processor byte enables to C/BE bus during data phase
Bridge accepts data when IRDY# and TRDY# sampled asserted

Please note that in reality, the bridge treats this transaction in one of two ways:
1. During initialization time, claims transaction but doesn't assert TRDY# until target on destination bus done so.
2. During run-time, issues a Retry and treats it as a DRR.

Type 1 Configuration Write ACESS

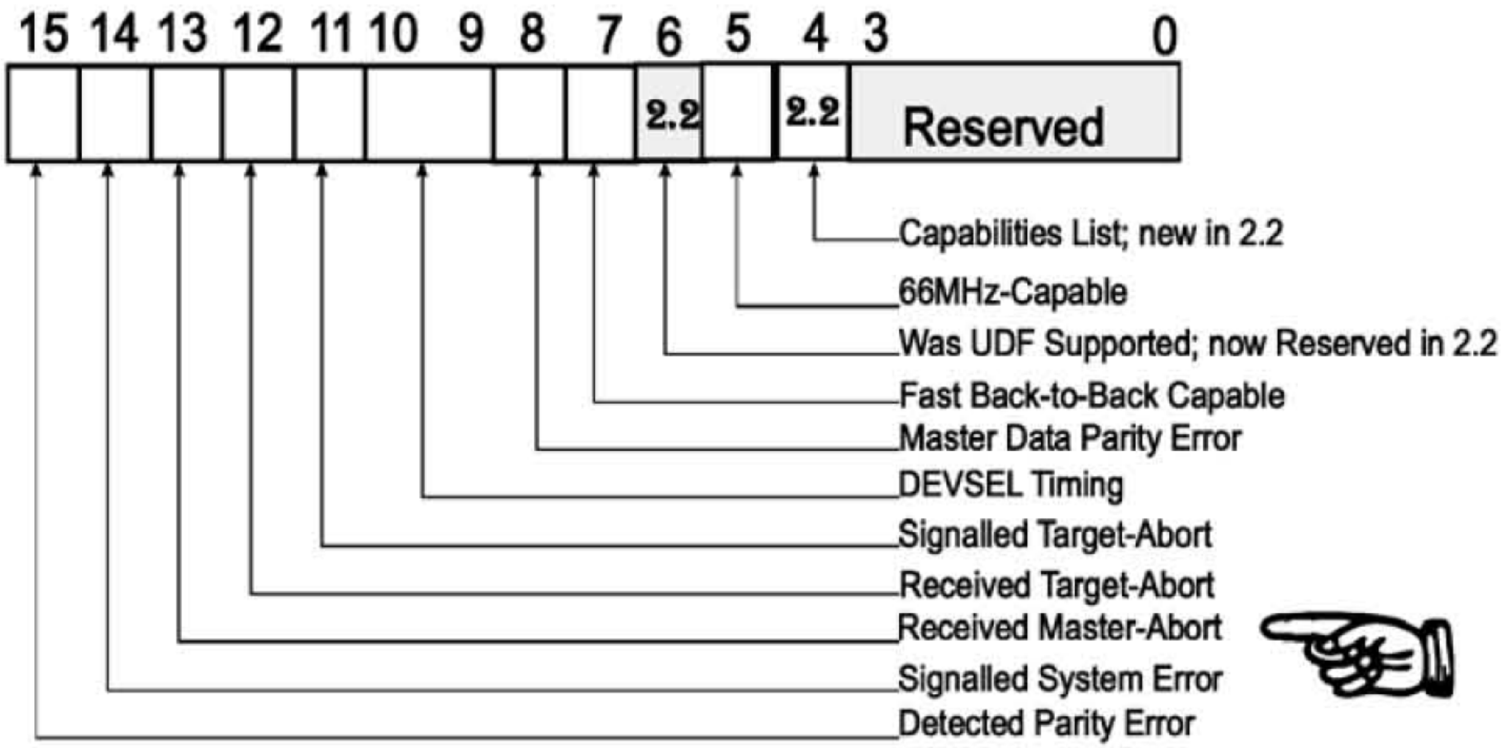

Address Phase
Data Phase
1
2
3
4
5
6
7
8
9
CLK
FRAME#
AD[31:0]
C/BE#[3:0]
IRDY#
TRDY#
DEVSEL#
IDSEL
GNT#
Bridge asserts FRAME# to start transaction
Bridge deasserts FRAME# indicating it's ready to complete last data phase
AD[1:0] = 01, type one access, AD[7:2] = Config DW address, AD[10:8] = function number, AD[15:11] = device number, AD[23:16] = bus number, AD[31:24] = unused
Address
Data
Bridge drives config data to target
Bridge copies processor byte enables to C/BE bus during data phase
Config Write CMD
Byte Enables
Bridge asserts IRDY# to indicate it's driving data
Target accepts data when IRDY# and TRDY# sampled asserted
Target asserts TRDY# to indicate it's ready to accept data and asserts DEVSEL# to claim transaction
All PCI devices ignore IDSEL during type one access
Don't Care

Please note that, in reality, the bridge treats thistransaction in one of two ways:
1. During Initialization time, claims transaction, but doesn't assert TRDY# until target on destination bus does so.
2. During run-time, issues a Retry and treats it as a DWR.

Target Device Doesn't Exist

- If the target of a configuration read transaction doesn't exist, DEVSEL# isn't asserted by any PCI function. Note that the subtractive decoder in the expansion bus bridge (i.e., the ISA bridge) does not claim unclaimed PCI configuration transactions.
- When DEVSEL# remains deasserted, the host/PCI bridge Master Aborts the transaction and sets the Received Master Abort bit in its configuration Status register.
- On a read that receives a Master Abort, the bridge returns all ones to the processor as the read data.
- On a write that experiences a Master Abort, the bridge acts as if the write completed OK.

Status Register



Configuration Burst Transactions Permitted

- Although virtually all configuration transactions today are single data phase transactions, the specification permits burst configuration transactions.
- Linear addressing is implied in a multiple data phase configuration burst transaction.
- If the target doesn't support burst configuration transactions, it will issue a Disconnect.

64-Bit Configuration Transactions Not Permitted

- **The specification states:**

The bandwidth requirements for IO and configuration commands cannot justify the added complexity.