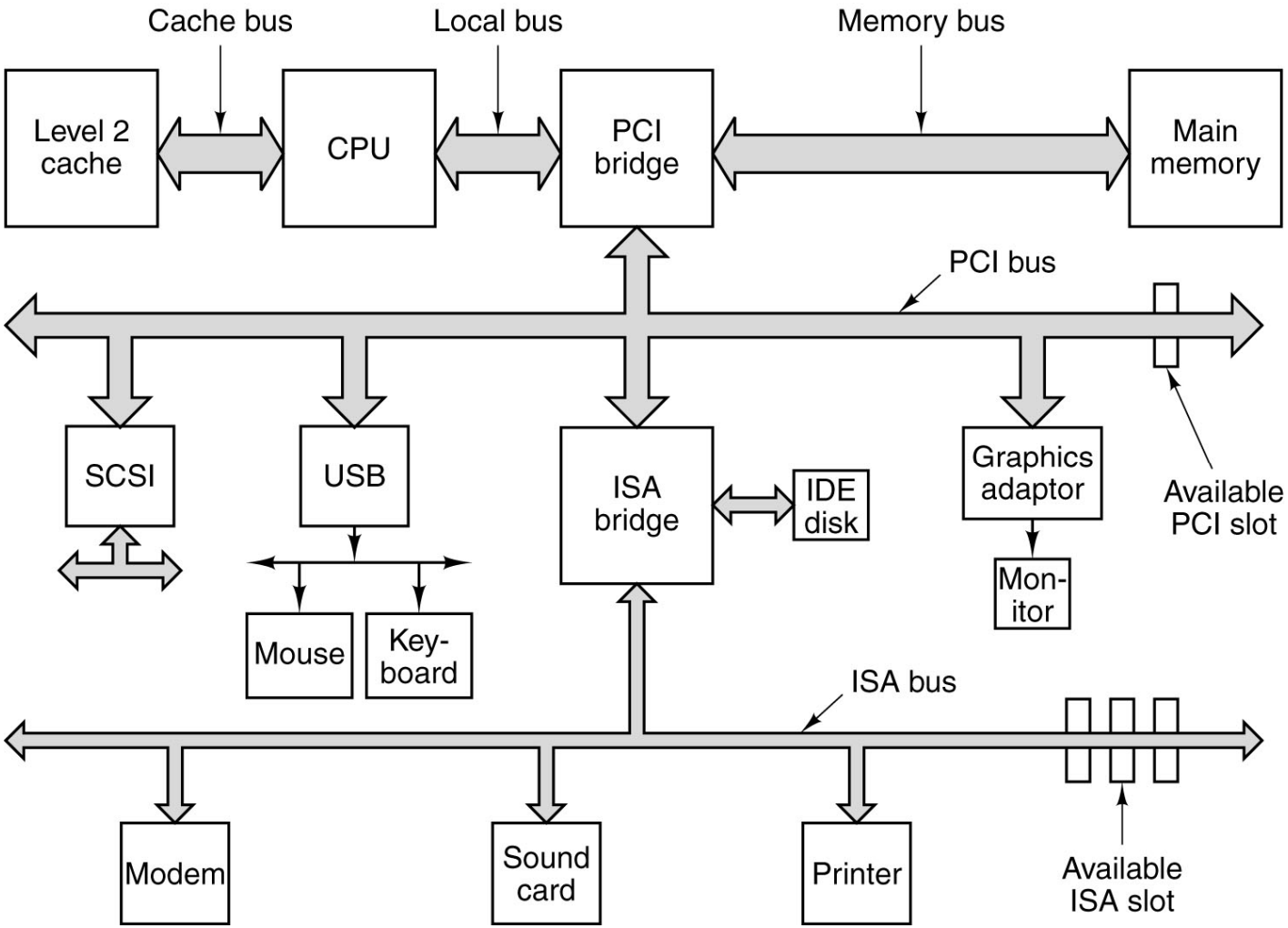
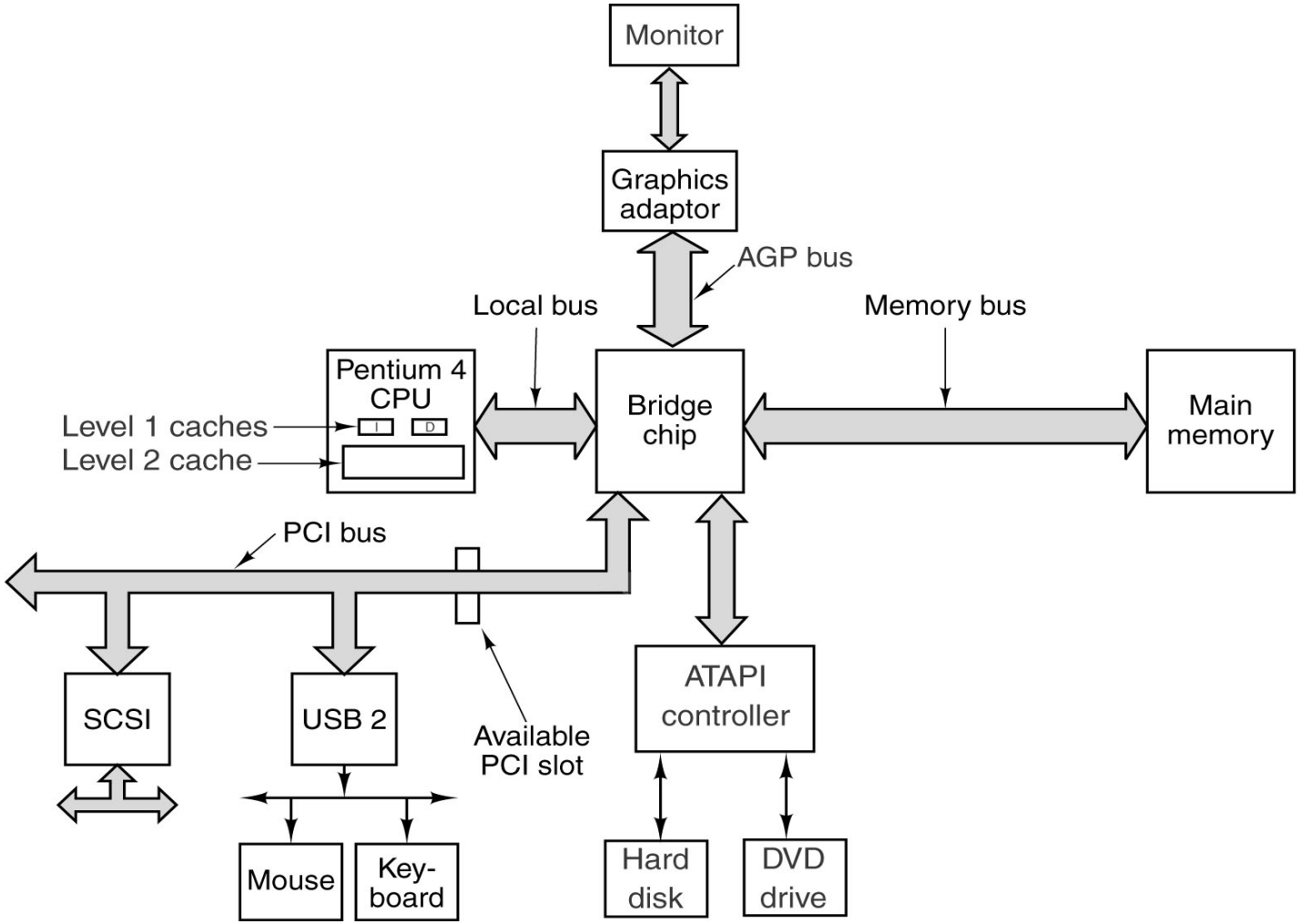
PCI

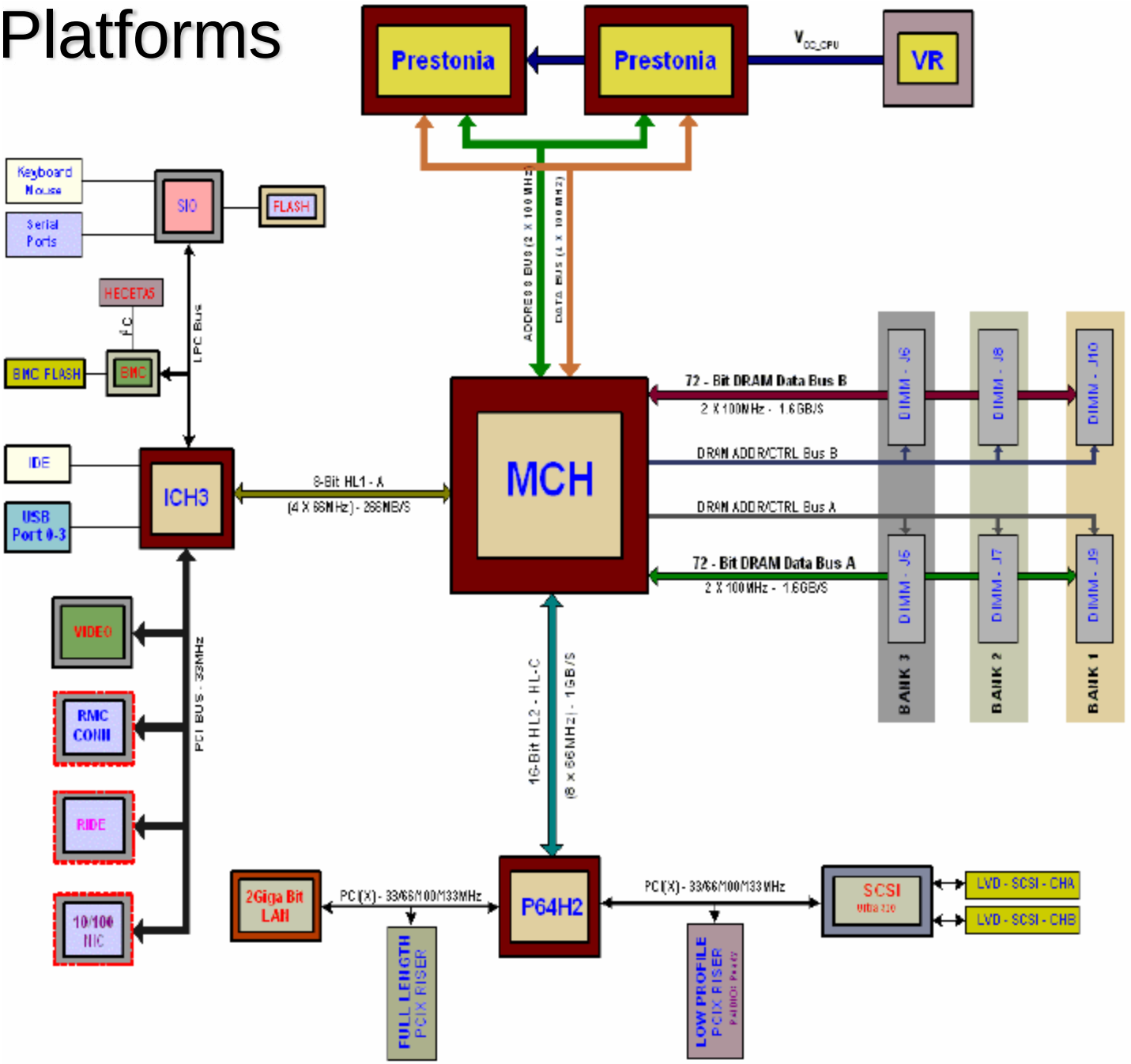
Older PC Implementations



Newer PC Implementations



Server Platforms



PCI Overview

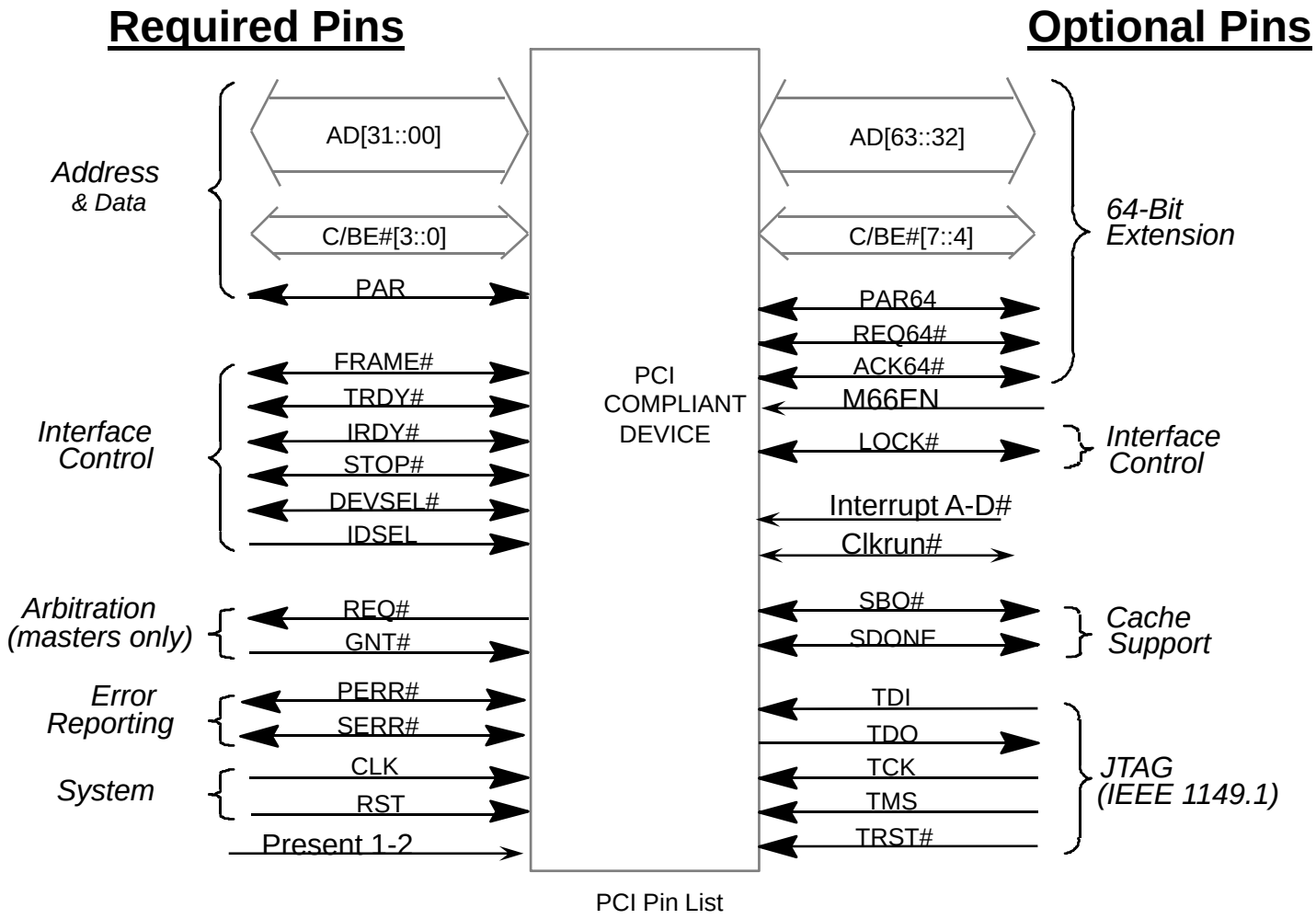
PCI Features

- **32-Bit or 64-Bit address and data**
- **66 or 33 down to 0 MHz synchronous operation**
- **Single or multiple bus masters**
- **Reflected bus signaling**
- **Stepped signaling**
- **Bus parity error reporting**
- **5 or 3.3 volt operation**
- **Cache support**
- **JTAG testing**

PCI Bus

- **Bus Signals**
- **Bus Commands**
- **Bus Transactions**
- **Arbitration**

PCI Bus Signals



PCI Pin List

PCI Bus Signals

Signal	Lines	Master	Slave	Description
CLK	1			Clock (33 MHz or 66 MHz)
AD	32	×	×	Multiplexed address and data lines
PAR	1	×		Address or data parity bit
C/BE	4	×		Bus command/bit map for bytes enabled
FRAME#	1	×		Indicates that AD and C/BE are asserted
IRDY#	1	×		Read: master will accept; write: data present
IDSEL	1	×		Select configuration space instead of memory
DEVSEL#	1		×	Slave has decoded its address and is listening
TRDY#	1		×	Read: data present; write: slave will accept
STOP#	1		×	Slave wants to stop transaction immediately
PERR#	1			Data parity error detected by receiver
SERR#	1			Address parity error or system error detected
REQ#	1			Bus arbitration: request for bus ownership
GNT#	1			Bus arbitration: grant of bus ownership
RST#	1			Reset the system and all devices

PCI Bus Signals (cont'd)

Signal	Lines	Master	Slave	Description
REQ64#	1	×		Request to run a 64-bit transaction
ACK64#	1		×	Permission is granted for a 64-bit transaction
AD	32	×		Additional 32 bits of address or data
PAR64	1	×		Parity for the extra 32 address/data bits
C/BE#	4	×		Additional 4 bits for byte enables
LOCK	1	×		Lock the bus to allow multiple transactions
SBO#	1			Hit on a remote cache (for a multiprocessor)
SDONE	1			Snooping done (for a multiprocessor)
INTx	4			Request an interrupt
JTAG	5			IEEE 1149.1 JTAG test signals
M66EN	1			Wired to power or ground (66 MHz or 33 MHz)

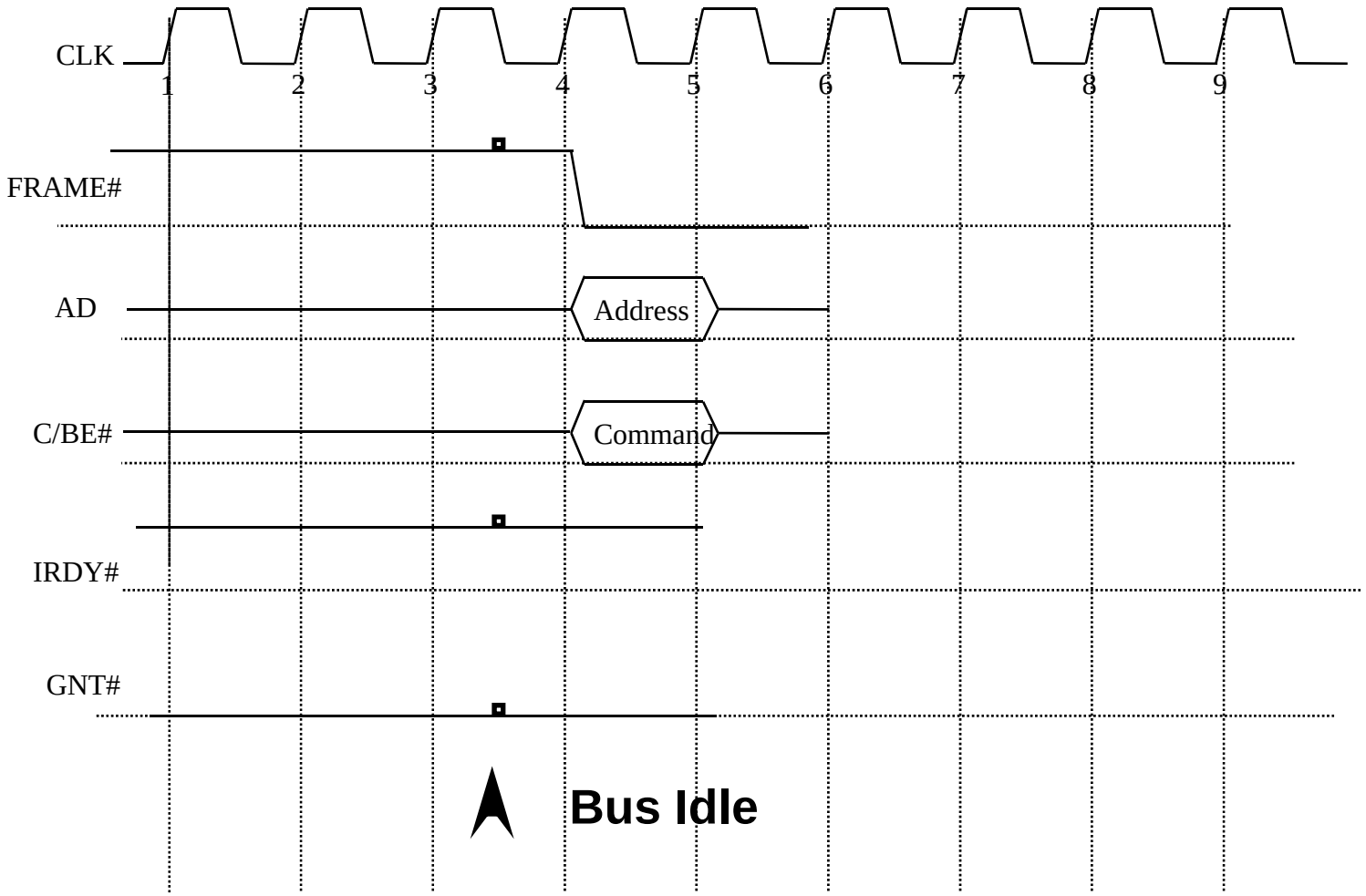
PCI bus access

- **PCI is a Multimaster Bus**
- **All transactions *initiated* by a master**
- **All transactions to/from a *target***

PCI Bus Control Signals

- **FRAME#**
  - **driven by master to indicate transfer start and end**
- **IRDY#**
  - **driven by master to indicate it is ready to transfer data**
- **TRDY#**
  - **driven by target to indicate it is ready to transfer data**

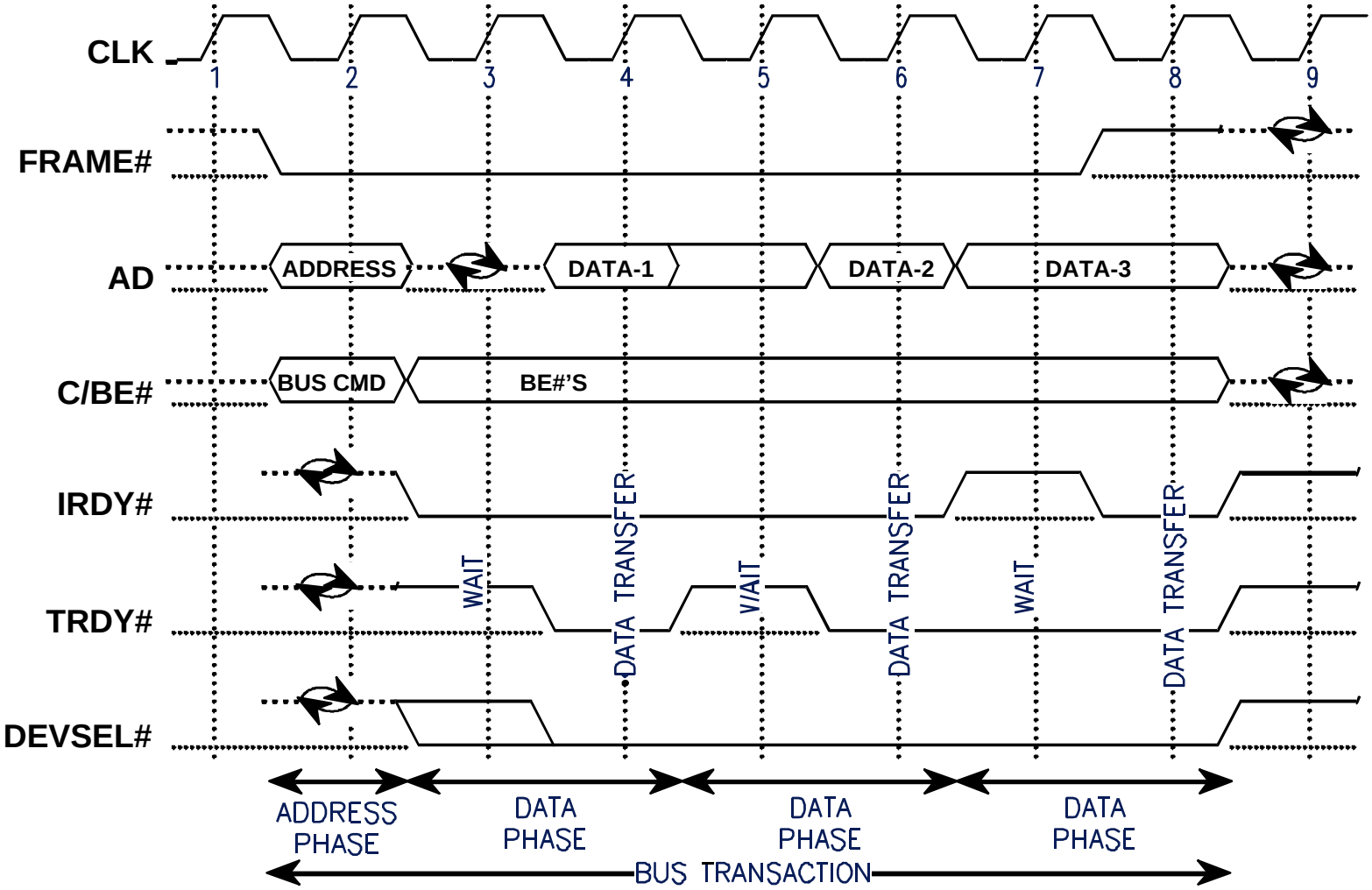
Bus transaction start



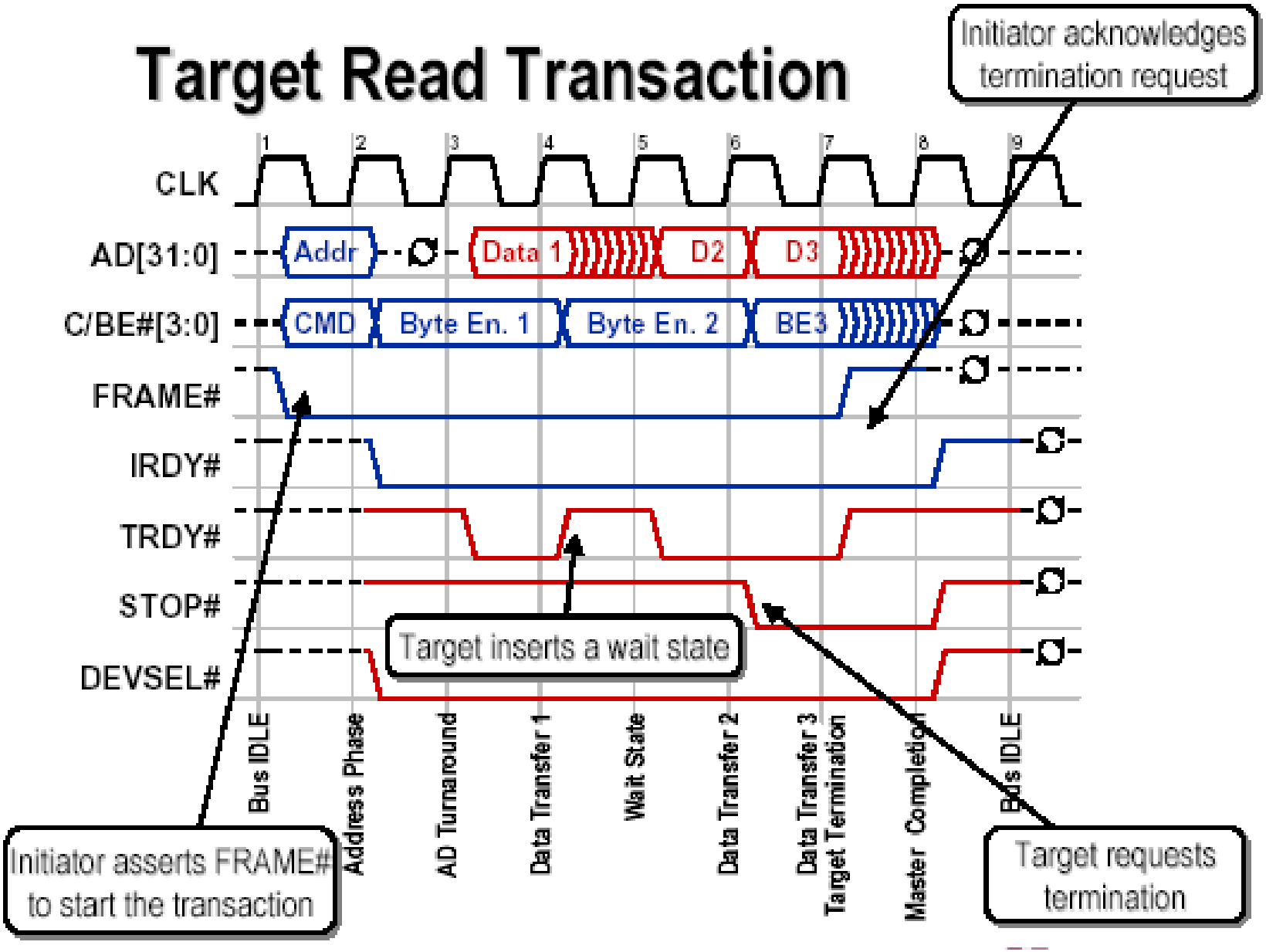
PCI Command Definition

C/BE [3::0]#	Command Type
0000	Interrupt acknowledge
0001	Special Cycle
0010	I/O Read
0011	I/O Write
0110	Memory Read
0111	Memory Write
1010	Configuration Read
1011	Configuration Write
1100	Memory Read Multiple
1101	Dual Address Cycle
1110	Memory Read Line
1111	Memory Write and Invalidate

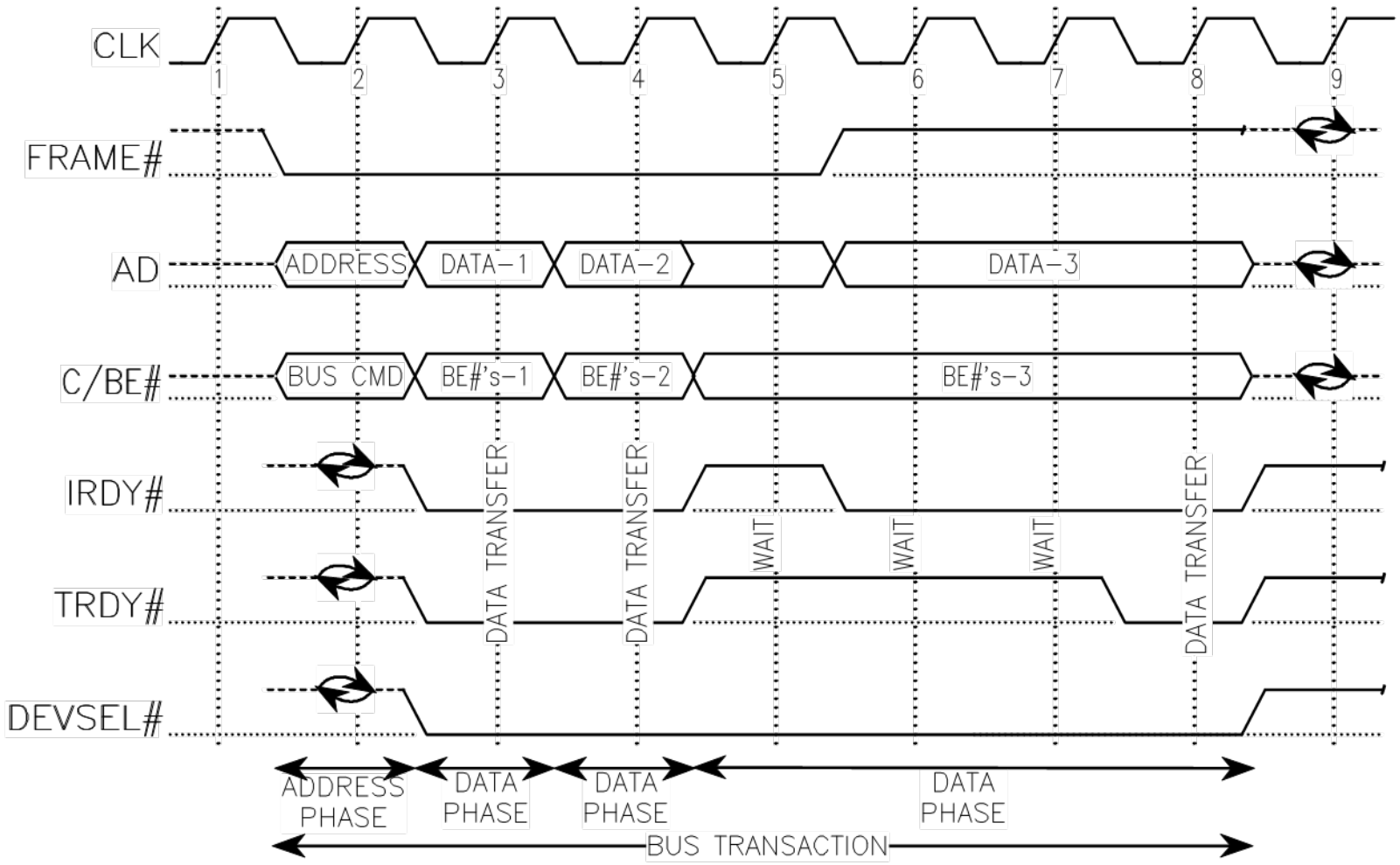
PCI Bus Read


CLK
1
2
3
4
5
6
7
8
9
FRAME#
AD
ADDRESS
DATA-1
DATA-2
DATA-3
C/BE#
BUS CMD
BE#'S
IRDY#
TRDY#
DEVSEL#
WAIT
DATA TRANSFER
WAIT
DATA TRANSFER
WAIT
DATA TRANSFER
ADDRESS PHASE
DATA PHASE
DATA PHASE
DATA PHASE
BUS TRANSACTION

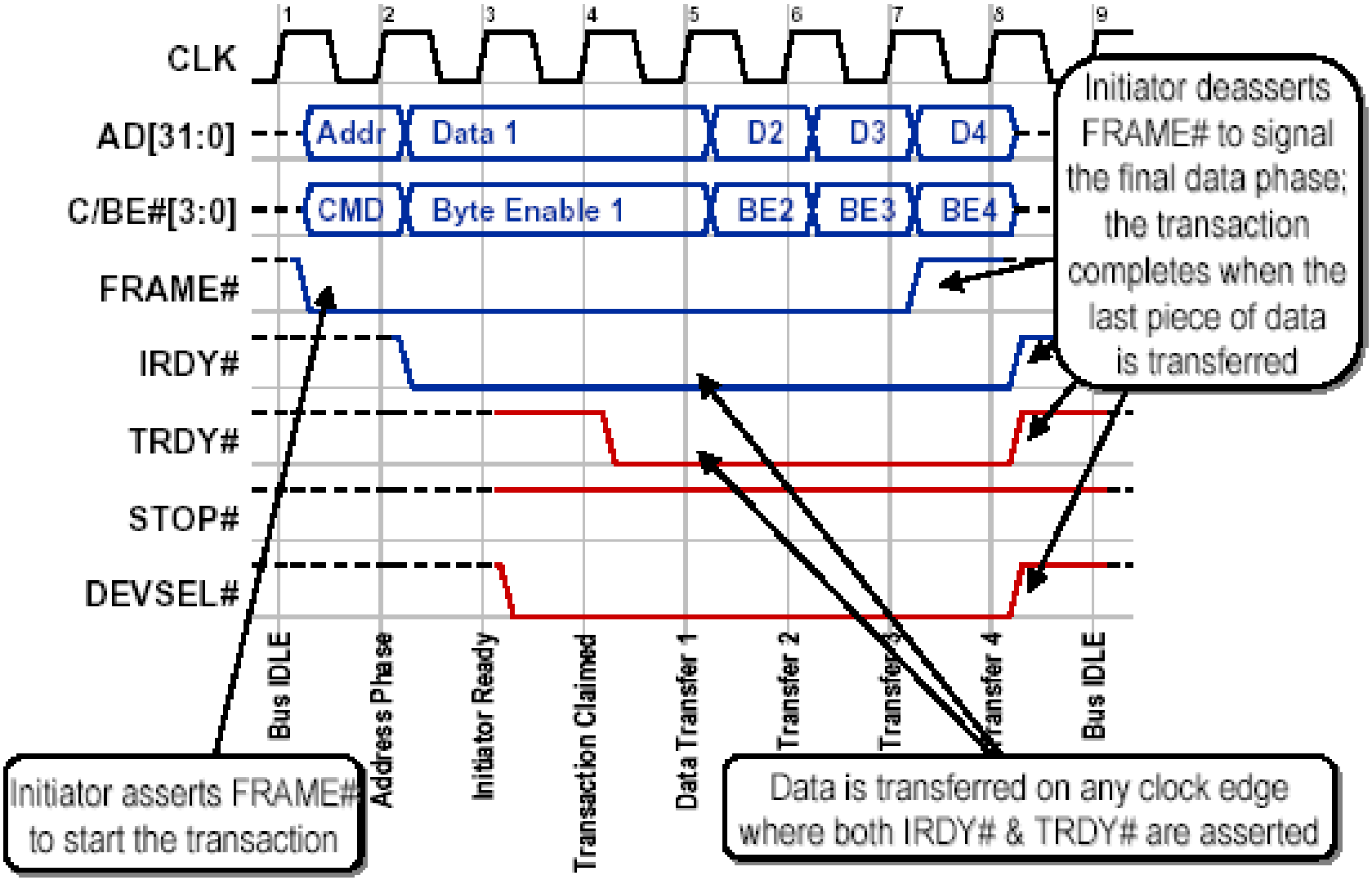
Target Read Transaction



PCI Bus Write

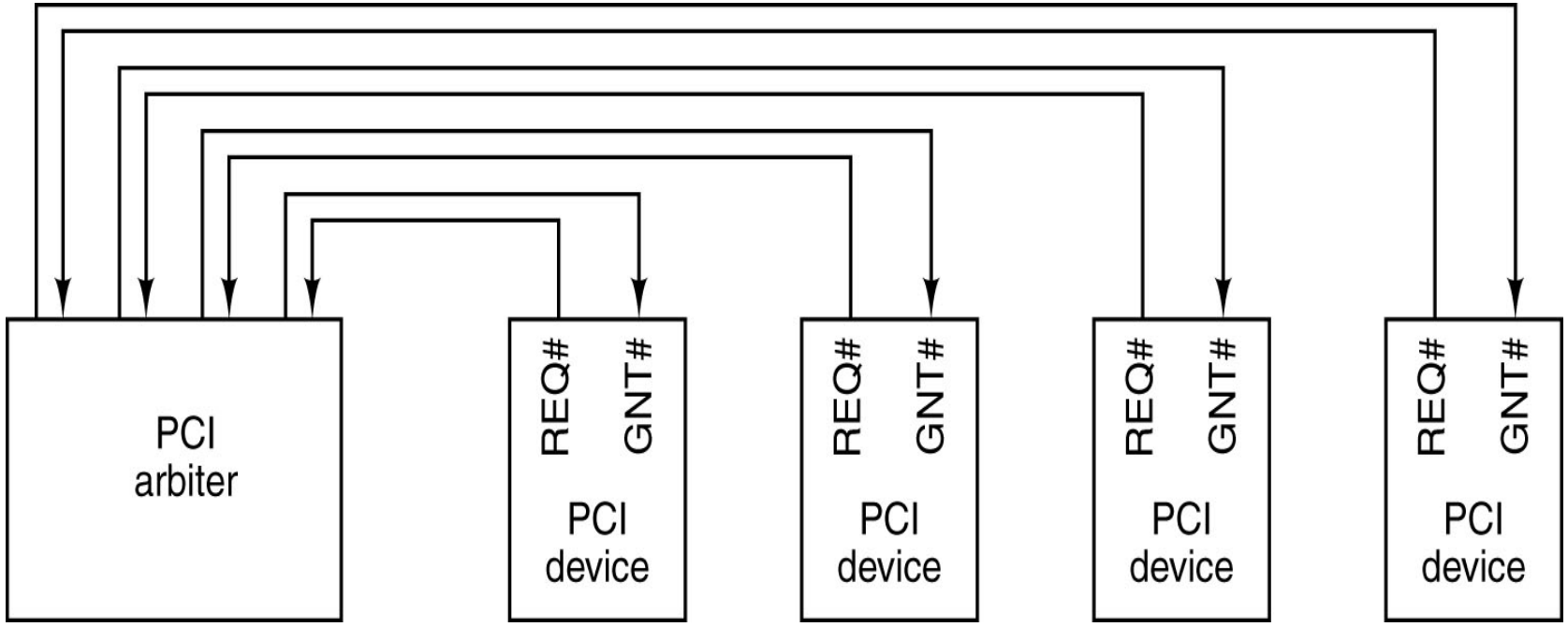


Basic Write Transaction

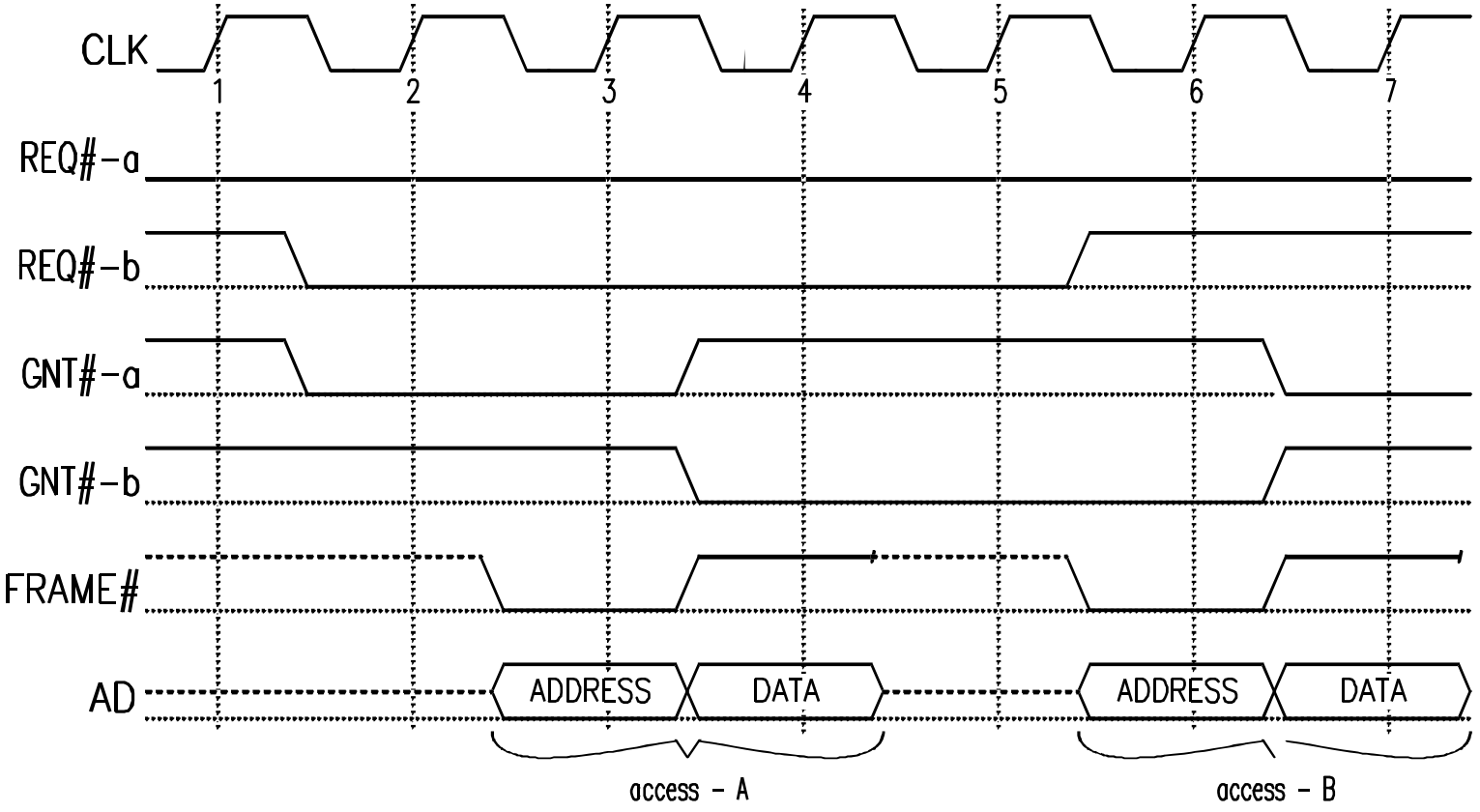


PCI Arbitration

Centralized Arbitration



PCI Bus Arbitration



Arbitration

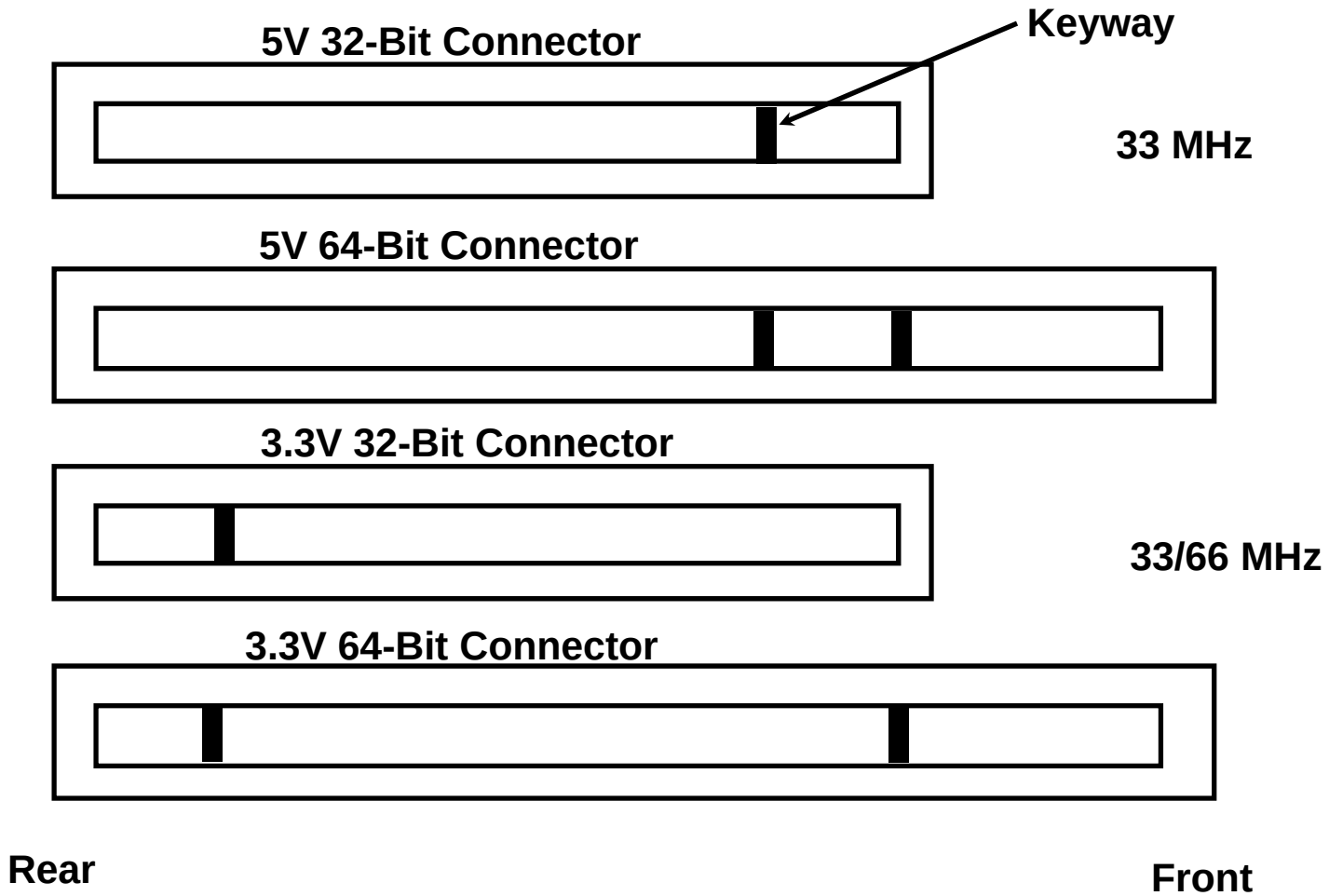
- Arbitration is access based
  - Master must arbitrate for each bus access
- Central arbitration scheme
  - Each master has a unique request and grant signal
- Arbitration is hidden
  - Occurs during previous bus cycle

Bus Parking

- Parking permits the arbiter to select an agent, by asserting its GNT#, when no other agent is using or requesting the bus
- The arbiter determines how this selection is made
  - Fixed, Last Used, …, or None

PCI Hardware

PCI Card Connectors

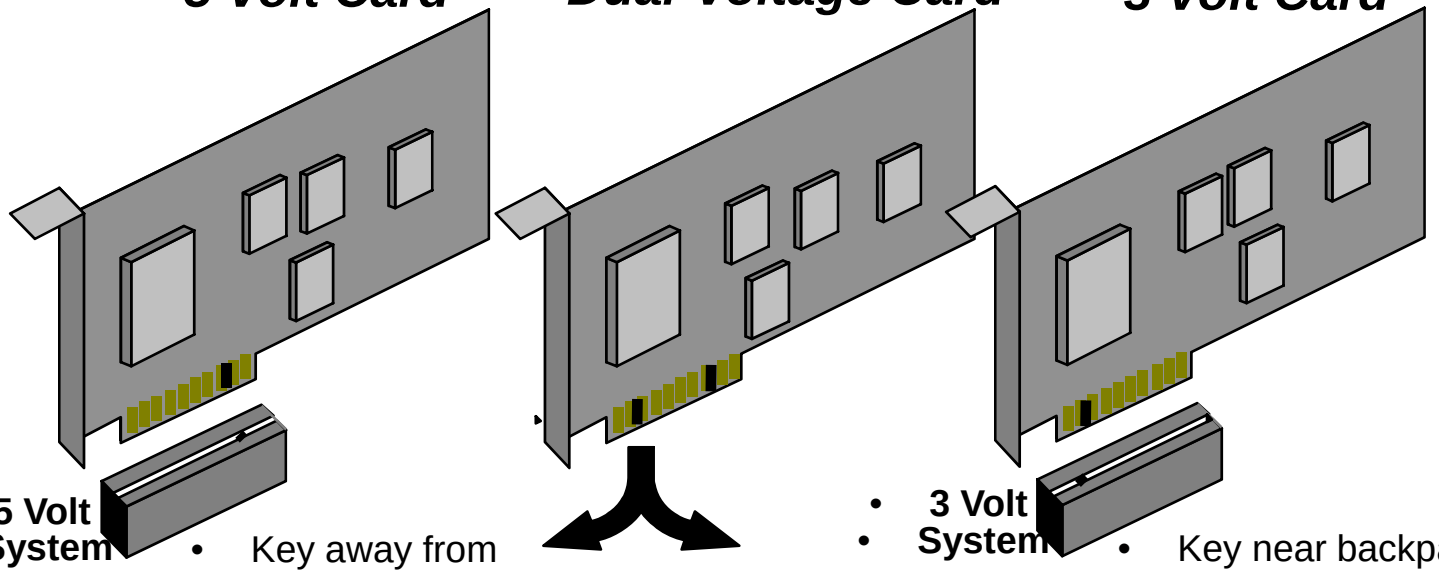


5 V To 3.3 V Migration Path

- ***5 Volt Card***
- **5 Volt System**

- ***Dual-Voltage Card***
- Key away from backpanel

- ***3 Volt Card***
- **3 Volt System**
- Key near backpanel



PCI Configuration

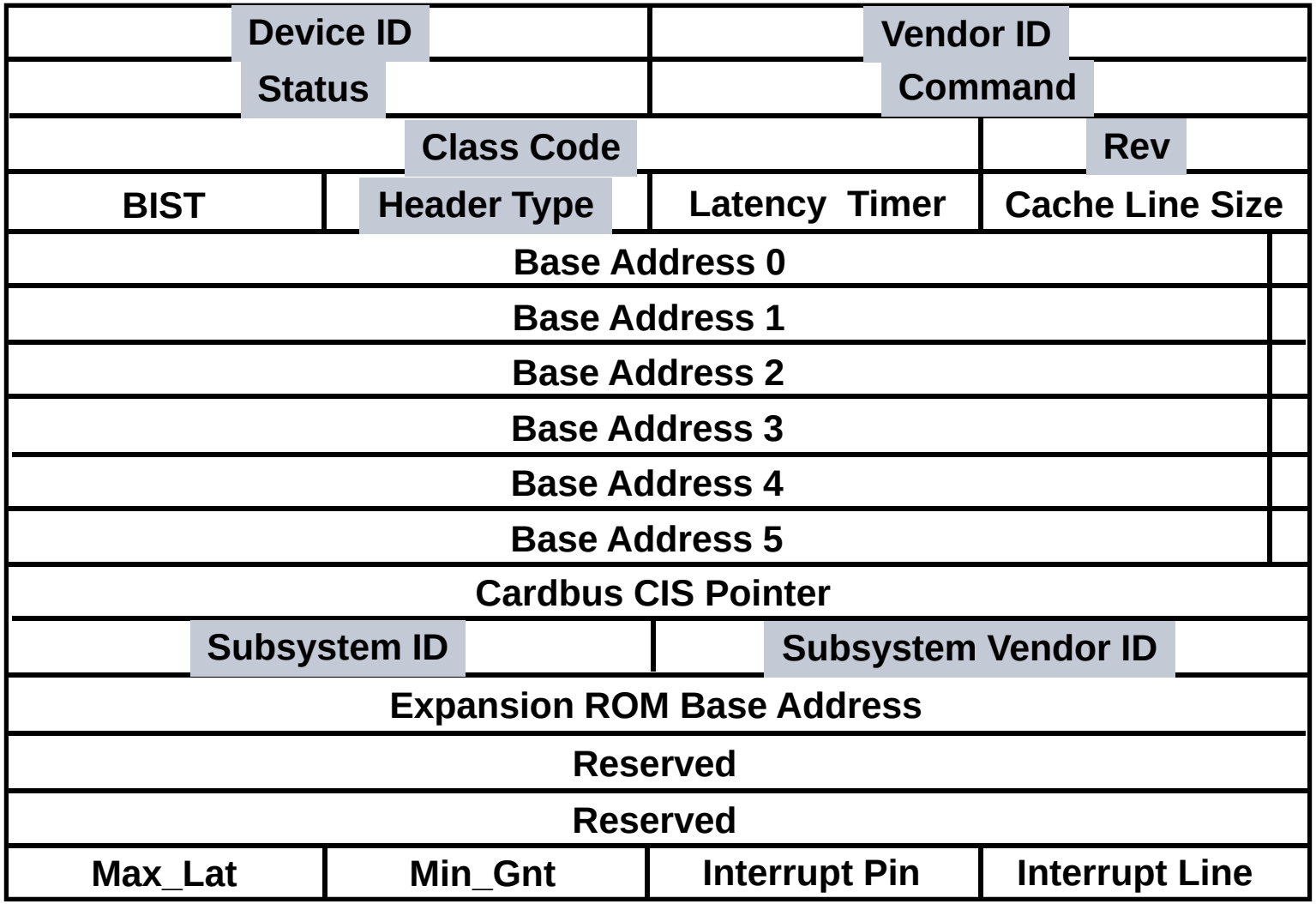
System Initialization

- Configuration allows software (BIOS) to initialize the system
- Each device has configuration registers
- At power up software scans bus(es)
- Software analyses system requirements
- Configuration registers are set to configure individual devices

Configuration Types

- Specific bus commands
  - configuration read (C/BE# = 1010)
  - configuration write (C/BE# = 1011)
- Type 0
  - local PCI bus
    - IDSEL line indicates device
    - address field indicates register
- Type 1
  - remote PCI bus (through bridge)
    - address field indicates bus, device and register

Configuration Space Header



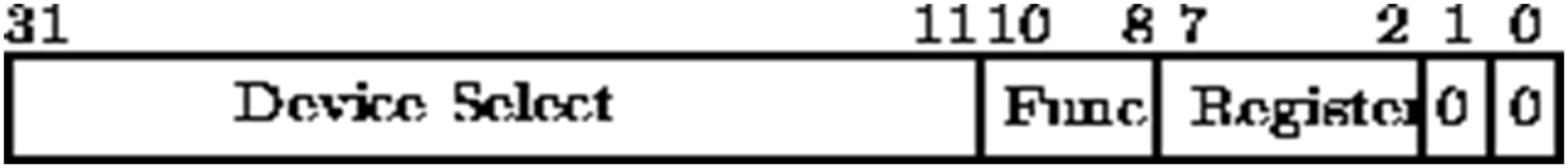


Figure: Type 0 PCI Configuration Cycle

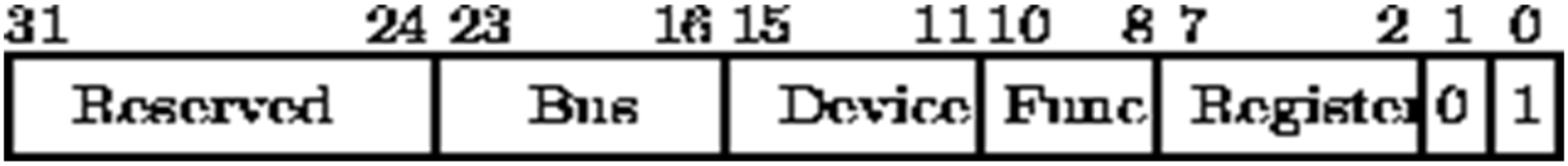
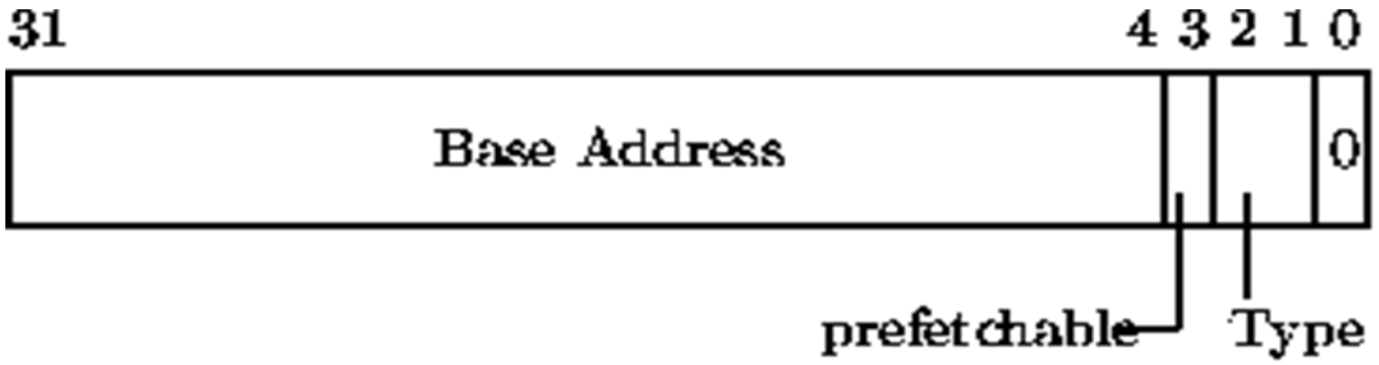


Figure: Type 1 PCI Configuration Cycle

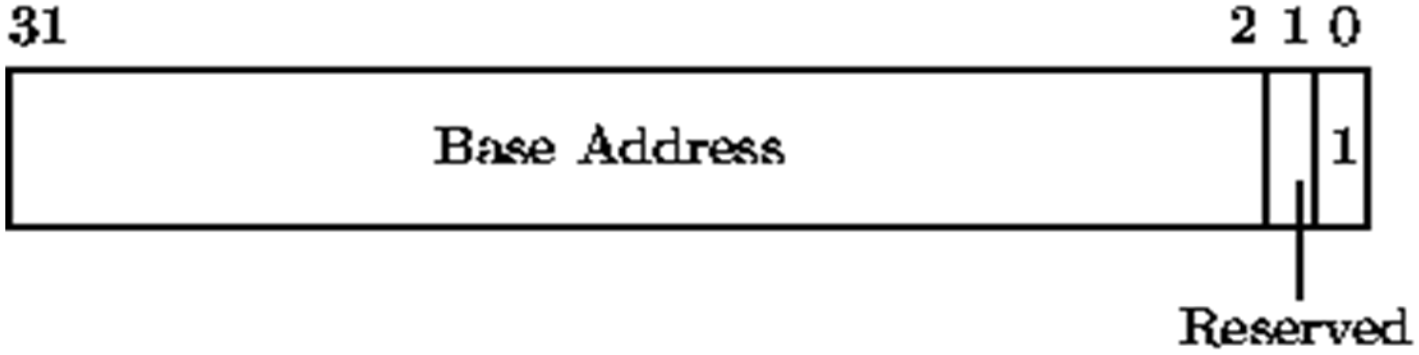
PCI Configuration Ports on PC

- **All x86 based PCs use the IO Bus to support an address and data port for PCI configuration**
  - **Address port is 4 bytes wide located at 0x0CF8**
  - **Data port is 4 bytes wide located at 0x0CFC**
- **Configuration cycles first write (IO IN) a canonical address to the address port**
  - **A read (IO OUT) from the data port will now return the 4 byte config register(s) addressed**
  - **A write (IO IN) to the data port will now set the addressed configuration register(s)**



Base Address for PCI Memory Space

TYPE 00 locate anywhere in lower 4GB
01 locate beyond 1MB
10 locate anywhere beyond 4GB
11 reserved



Base Address for PCI I/O Space

BAR Management

- **Write all 1s to a BAR and then read**
  - **If return value is all 0s then not used**
  - **If not zero, then check least sig bit**
    - **If 1 then IO assignment**
    - **If 0 then memory assignment**
    - **Bit position of least sig bit is used for size determination (i.e. if bit 6 is on decoder requires 2^6 or 64 bytes of space)**

BAR Management (cont'd)

- **If all 1s are written to a BAR and the return value is 0xFFFF0000**
  - **The BAR is a memory decoder**
  - **The BAR is not prefetchable**
  - **The BAR requires an address < 4GB**
  - **The BAR requires 2^{16} or 64 KB of space**
- **0xFFFF0008 as above but prefetchable**
- **0xFFFF000A prefetchable and 64 bit**

BAR Management (cont'd)

- **If all 1s are written to a BAR and the return value is 0xFFFFF001**
  - **The BAR is an IO decoder**
  - **The BAR requires an address < 64 KB**
  - **The BAR requires 2^{12} or 4 KB of space**
- **Minimum memory size is 16 bytes**
- **Minimum IO size is 4 bytes**