CPE 186 Computer Hardware Design

Configuration Registers

Dr. Pang

Intro to Configuration Header Region

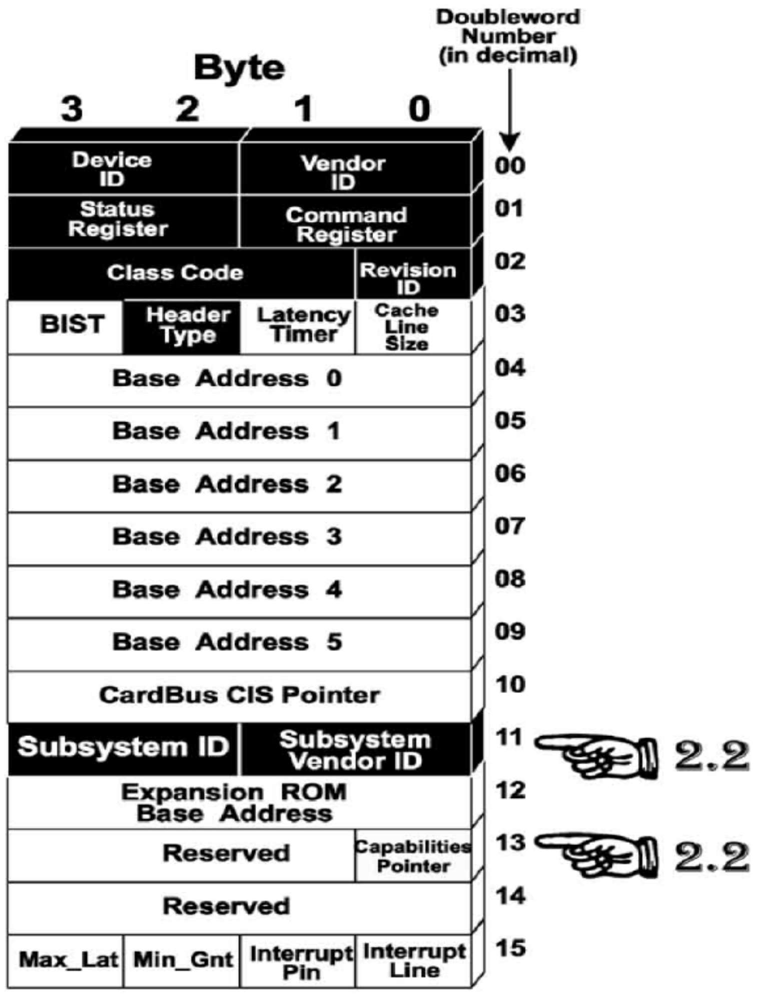
- Each PCI function possesses a block of 64 configuration dwords reserved for the implementation of its configuration registers. The format and usage of the first 16 dwords **is** predefined by the PCI specification. This area is referred to as the device's Configuration Header Region (or Header Space). The specification currently defines three Header formats, referred to as Header types Zero, One and Two.
- Header Type One is defined for PCI-to-PCI bridges. A full description of Header Type One can be found in "Configuration Registers" on page 552.
- Header Type Two is defined for PCI-to-CardBus bridges and is fully defined **in** the PC Card spec.
- Header Type Zero is used for all devices other than PCI-to-PCI and cardBus bridges. This chapter defines Header me Zero.

Registers Used to Identify Device's Driver

The OS uses some combination of the following mandatory registers to determine which driver to load for a device:

- Vendor ID.
- Device ID.
- Revision.
- Classcode.
- SubSystem Vendor ID.
- Subsystem ID.

Format of a PCI Function's Configuration Header

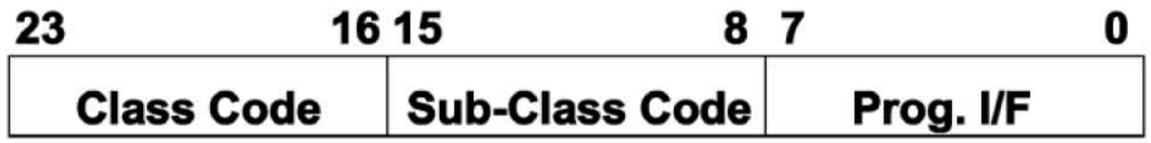


Byte 3	Byte 2	Byte 1	Byte 0	Doubleword Number (in decimal)	
Device ID		Vendor ID		00	
Status Register		Command Register		01	
Class Code			Revision ID	02	
BIST	Header Type	Latency Timer	Cache Line Size	03	
Base Address 0				04	
Base Address 1				05	
Base Address 2				06	
Base Address 3				07	
Base Address 4				08	
Base Address 5				09	
CardBus CIS Pointer				10	
Subsystem ID		Subsystem Vendor ID		11	2.2
Expansion ROM Base Address				12	
Reserved			Capabilities Pointer	13	2.2
Reserved				14	
Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	15	

Vendor ID Register

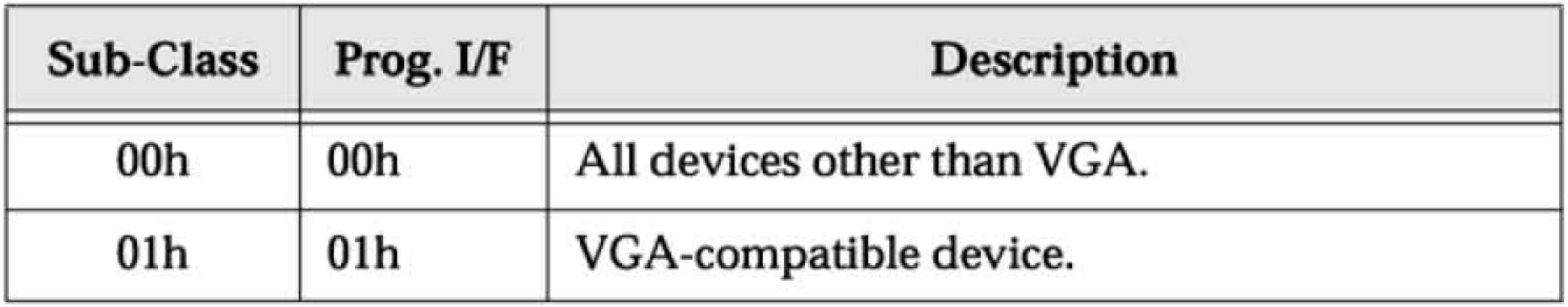
- The value FFFFh is reserved and must be returned by the host/PCI bridge when an attempt is made to perform a configuration read from a non-existent device's configuration register. The read attempt results in a Master Abort.
    - ❑ The Master Abort is not considered to be an error
    - ❑ The bridge must none the less set its Received Master Abort bit in its configuration Status register.

Class Code Register



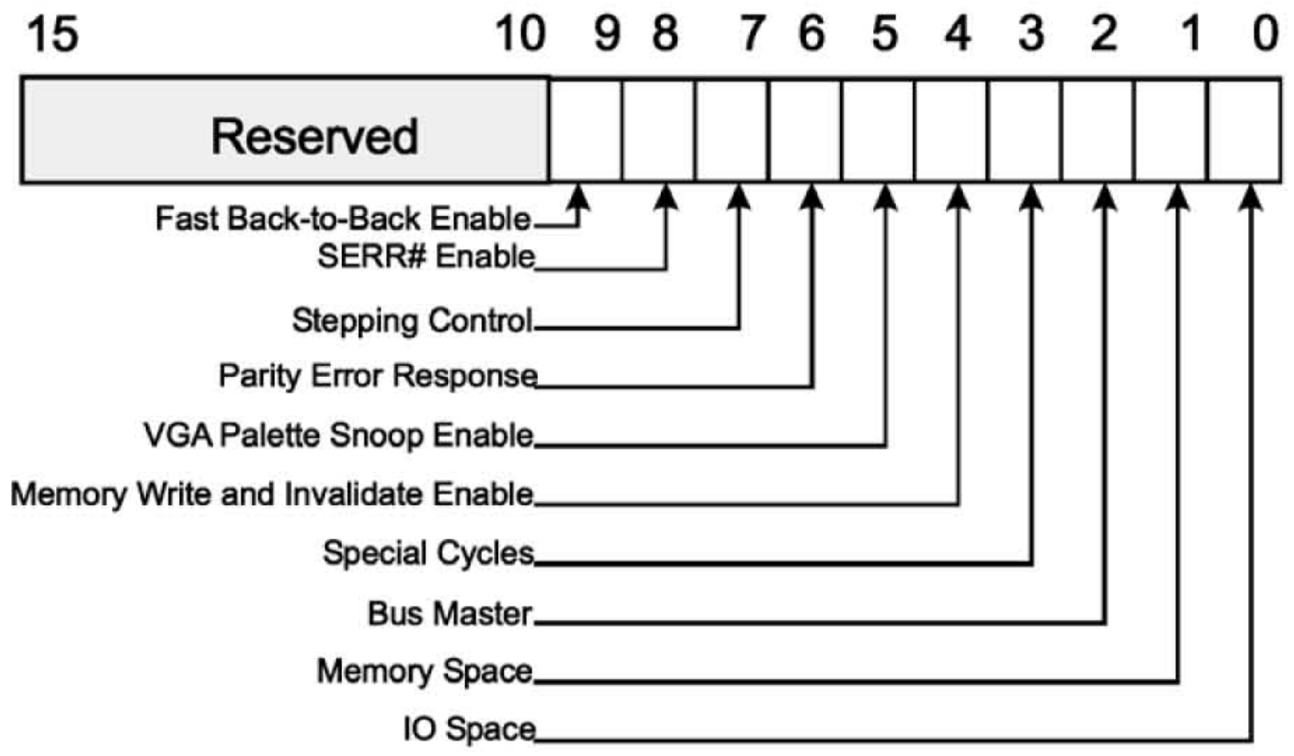
Prog. Interface Byte

Class Code 0 (rev 1.0)

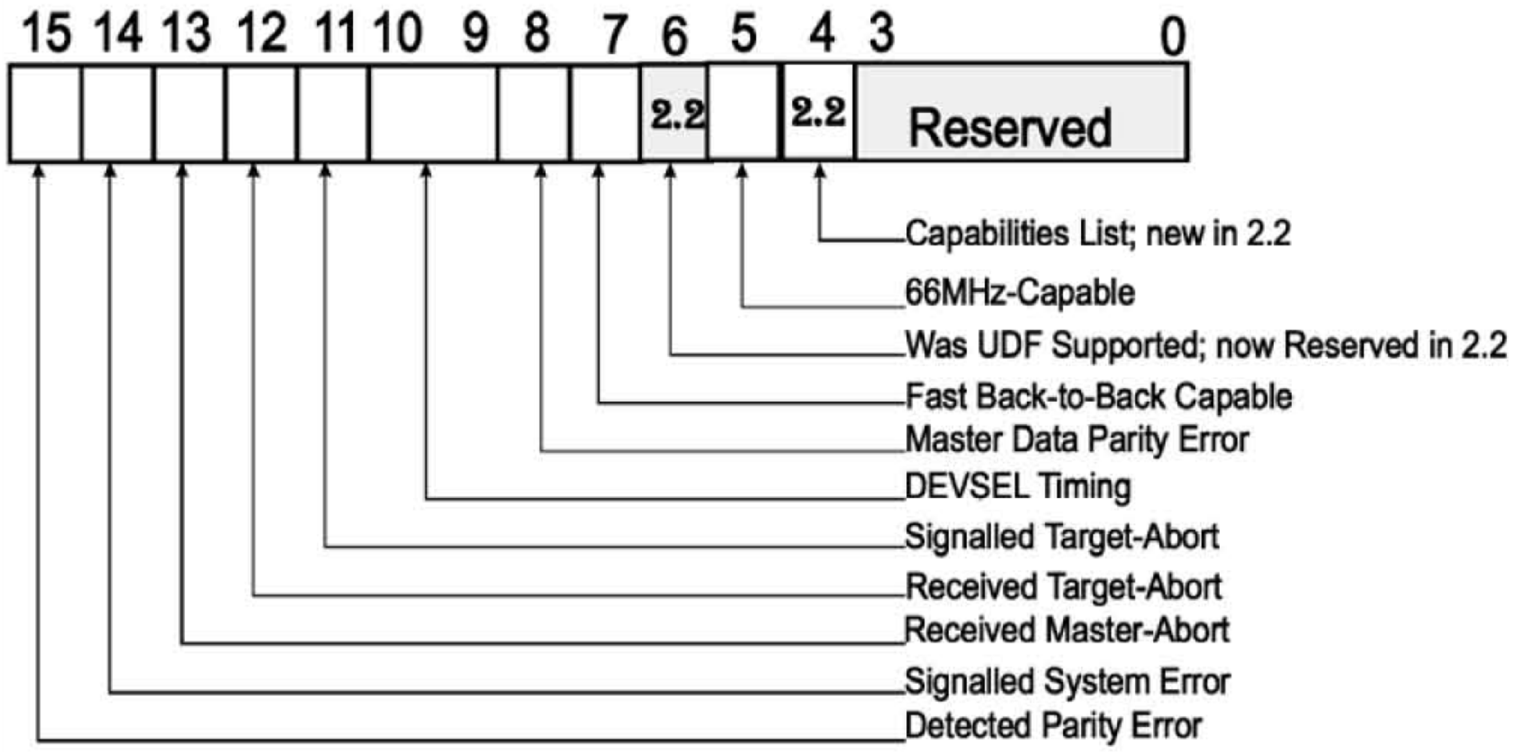


Sub-Class	Prog. I/F	Description
00h	00h	All devices other than VGA.
01h	01h	VGA-compatible device.

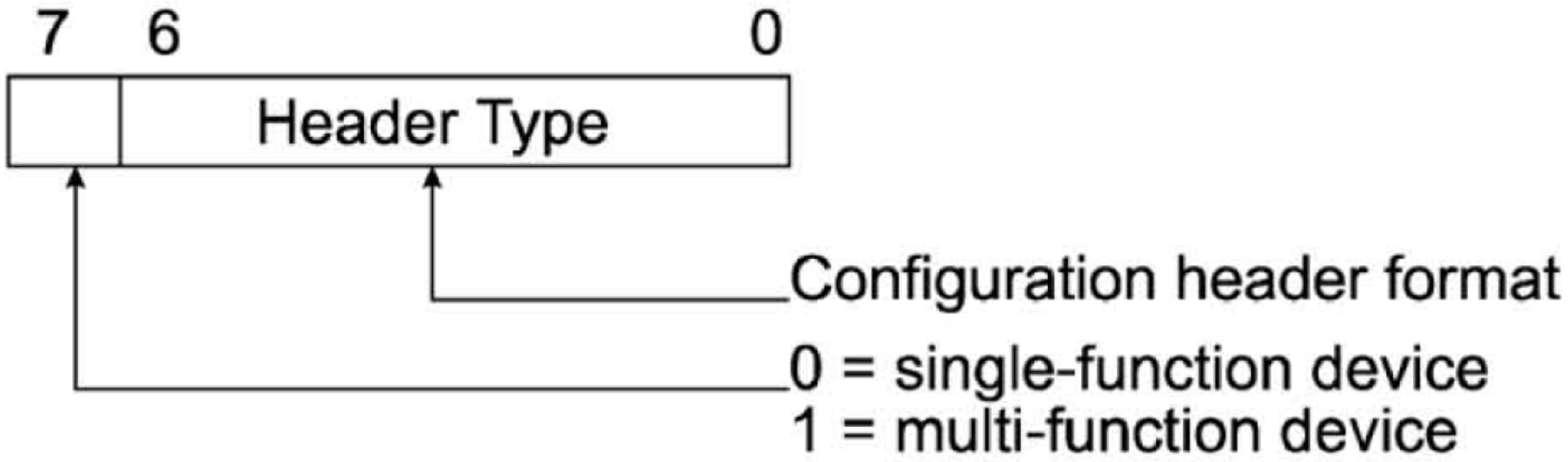
Command Register Bit Assignment



Status Register Bit Assignment



Header Type Register Bit Assignment


7
6
0
Header Type
Configuration header format
0 = single-function device
1 = multi-function device

PCI Capability

- Vendor specific

Cache Line Size Register

- This read/write configuration register specifies the system cache Line size in dword increments (e.g., a P6-based system would store the value O8h, indicating a cache line size of eight dwords, or 32 bytes).
- The register must be implemented by bus masters that implement the Memory Write-and-Invalidate command. Because it must know the cache line size in order to ensure that it starts transactions on a cache line boundary and keeps its promise to write an entire line into memory.
- The bus master may not use the Memory Write-and-Invalidate command when this register is set to zeros **(**which indicates that the configuration software hasn't yet told it the cache line size). In this case, the master should only use Memory Write transactions to update memory.
- A device may limit the number of cache line sizes that it supports. If an unsupported value is written to the register by the configuration software, the device behaves as if the value zero was written.

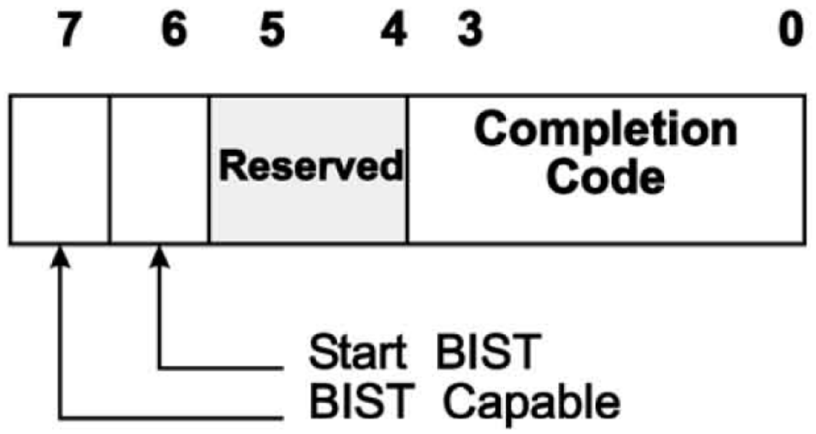
Latency Timer

- Mandatory (read/writable) for masters that perform burst transactions.
- It defines the minimum amount of time, in PCI dock cycles. that the bus master can retain ownership of the bus whenever it Initiates a new transaction.
- The bus master decrements its Latency Timer by one on each rising-edge of the clock after it initiates a transaction. It may continue its transaction until either:
  - ❑it has completed the overall burst transfer (if it doesn't lose its **grant).** or
  - ❑the target asserts STOP# to orernaturelv terminate the transaction. or
  - ❑it **has** exhausted its time slice **(LT** value) and it has been preempted (lost its GNT# to another **PCI** master).

  whichever comes first.

BIST Register

- Optional. This register may be implemented by both master and target devices.



Base Address Registers (BARS)

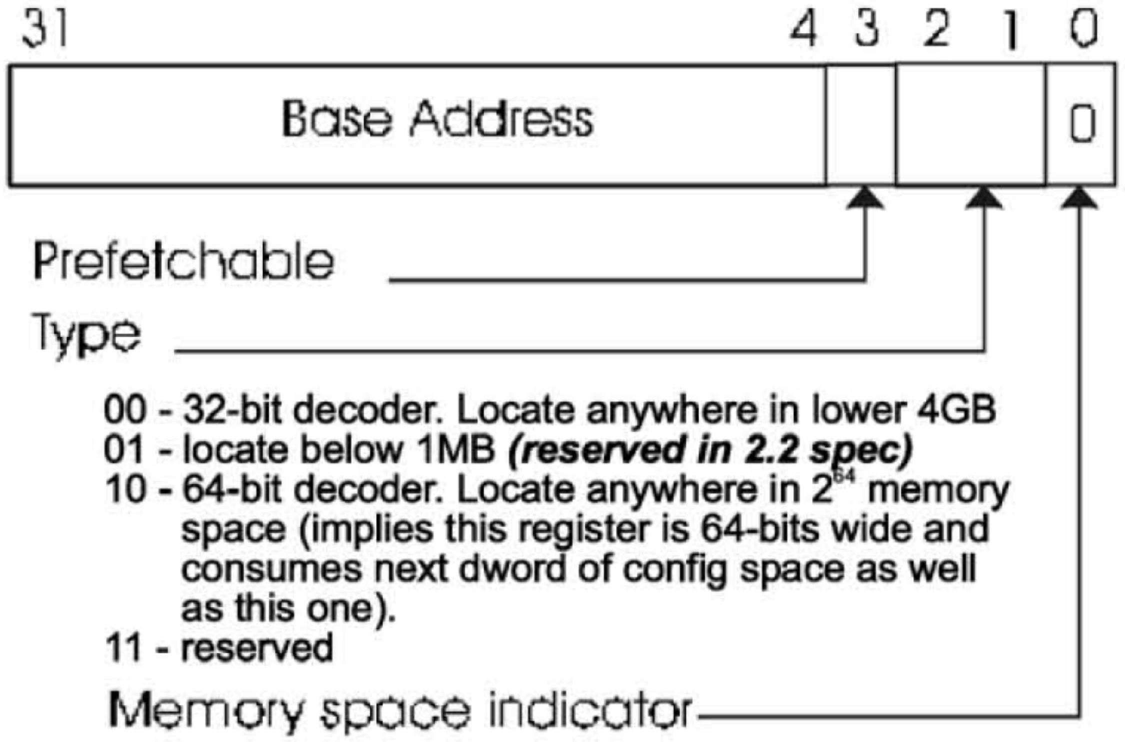
The Base Address Registers (BARS) located in dwords 4-through-9 of the device's configuration Header space, are used to implement a function's programmable memory and/or IO decoders.

- Each register is 32-bits wide (or 64-bits wide if it's a memory decoder and its associated memory block can be located above the 4GB address boundary)
- Required if a device implements memory and/or IO decoders.
- Bit 0 = 0, the register is a memory address decoder.
- Bit 0 = 1, the register is an IO address decoder.

Memory-Mapping Recommended

- In a PC environment, IO space is densely populated and will only become more so in the future.
- The specification strongly recommends that the device designer provide only a Memory Base Address Register that maps a device's register set into memory space. Optionally, an IO Base Address Register may also be included to map it into IO space, but this is not recommended.
- This gives the configuration software the flexibility to map the device's register set into memory space and, if an IO Base Address Register is also provided, into IO space as well.

Memory Base Address Register


31
4 3 2 1 0
Base Address
0
Prefetchable
Type
00 - 32-bit decoder. Locate anywhere in lower 4GB
01 - locate below 1MB *(reserved in 2.2 spec)*
10 - 64-bit decoder. Locate anywhere in 2^64 memory space (implies this register is 64-bits wide and consumes next dword of config space as well as this one).
11 - reserved
Memory space indicator

Base Address Field:
This field consists of bits [31:4] for a 32-bit memory decoder and bits [63:4] for a 64-bit memory decoder.

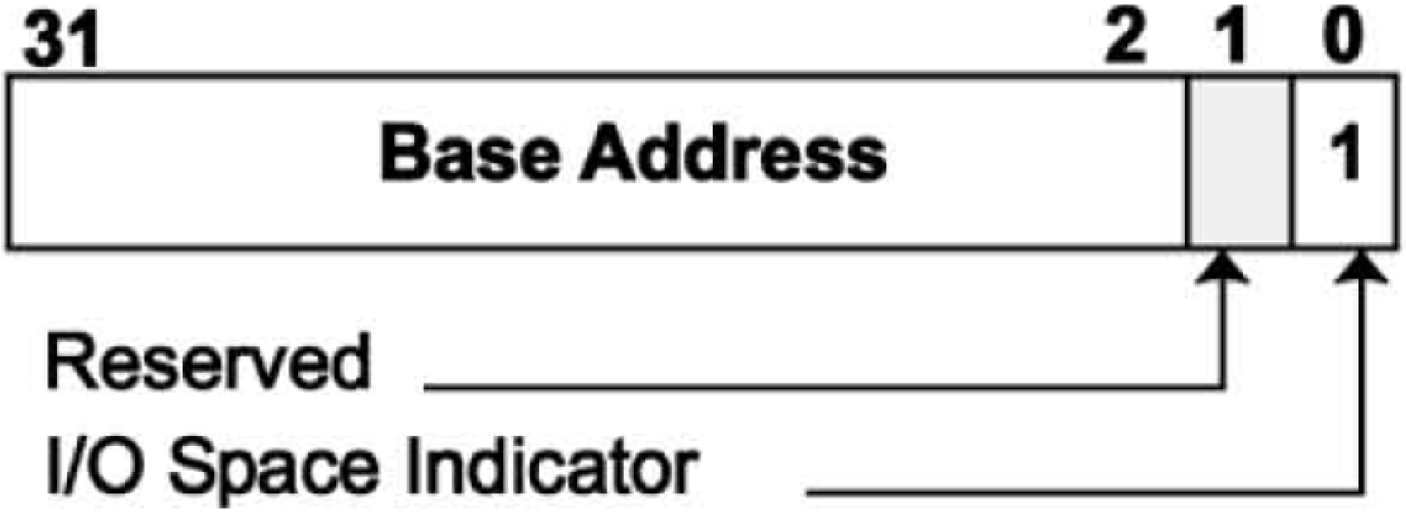
- **Base Address Field:** This field consists of bits [31:4] for a 32-bit memory decoder and bits [63:4] for a 64-bit memory decoder.

A Memory Example

As an example, assume that **FFFFFFFFh** is written to the Base Address Register at configuration dword 04d and the value read back is **FFF00000**h. The fact that any bits could be changed to one indicates that the Base Address Register is implemented.

- Bit 0 = 0, indicating that this is a memory address decoder.
- Bits [2:1] = 00b, indicating that it's a 32-bit memory decoder.
- Bit 3 = 0, indicating that it's not Prefetchable memory.
- Bit 20 is the first one bit found in the Base Address field. The binary weighted value of this bit is 1,048,576, indicating that this is an address decoder for 1MB of memory.
- The programmer then writes a 32-bit base address into the register. However, only bits [31:20] are writable. The decoder accepts bits [31:20] and assumes that bits [19:0] of the assigned base address are zero. This means that the base address is divisible by 1MB, the size of the requested memory range. It is a characteristic of PCI decoders that the assigned start address is always divisible by the size of requested range.

IO Base Address Register



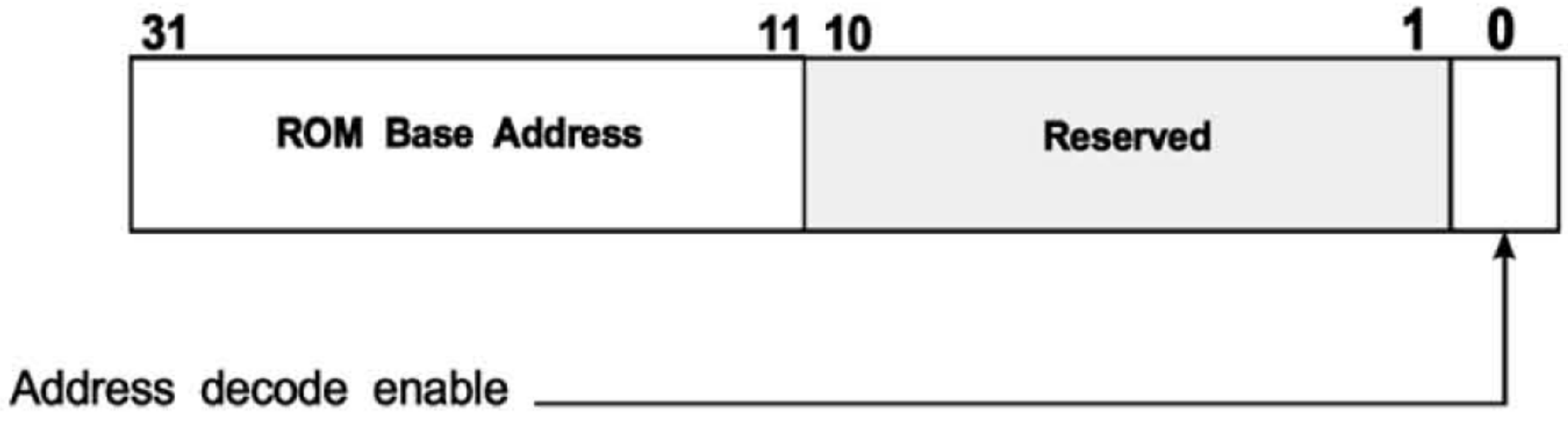
An IO Example

As a second example, assume that FFFFFFFFh is written to a function's Base Address Register at configuration dword address 05d and the value read back is FFFFFF01h. Bit 0 is a 1, indicating that this is an IO address decoder.

Scanning upwards starting at bit 2 (the least-significant bit of the Base Address field), bit 8 is the first bit that was successfully changed to one. The binary-weighted value of this bit is 256**,** indicating that this is IO address decoder is requesting 256 bytes of IO space.

- The programmer then writes a 32-bit base IO address into the register. However. only bits [31:8] are writable. The decoder accepts bits [31:8] and assumes that bits [7:0] of the assigned base address are zero. This means that the base address is divisible by 256, the size of the requested IO range.

Expansion ROM Base Address Register



Interrupt Pin Register

- The read-only Interrupt Pin register defines which of the four PCI interrupt request pins, INTA#-through-INTD#, a PCI function is connected to.

Interrupt Line Register

- The Interrupt Line register is used to identify which input on the interrupt controller the function's PCI interrupt request pin (as specified in its Interrupt Pin register) is routed to.
- For example, in a PC environment the values 00h-through-0Fh in this register (refer correspond to the IRQ0-through-IRQ15 inputs on the interrupt controller.

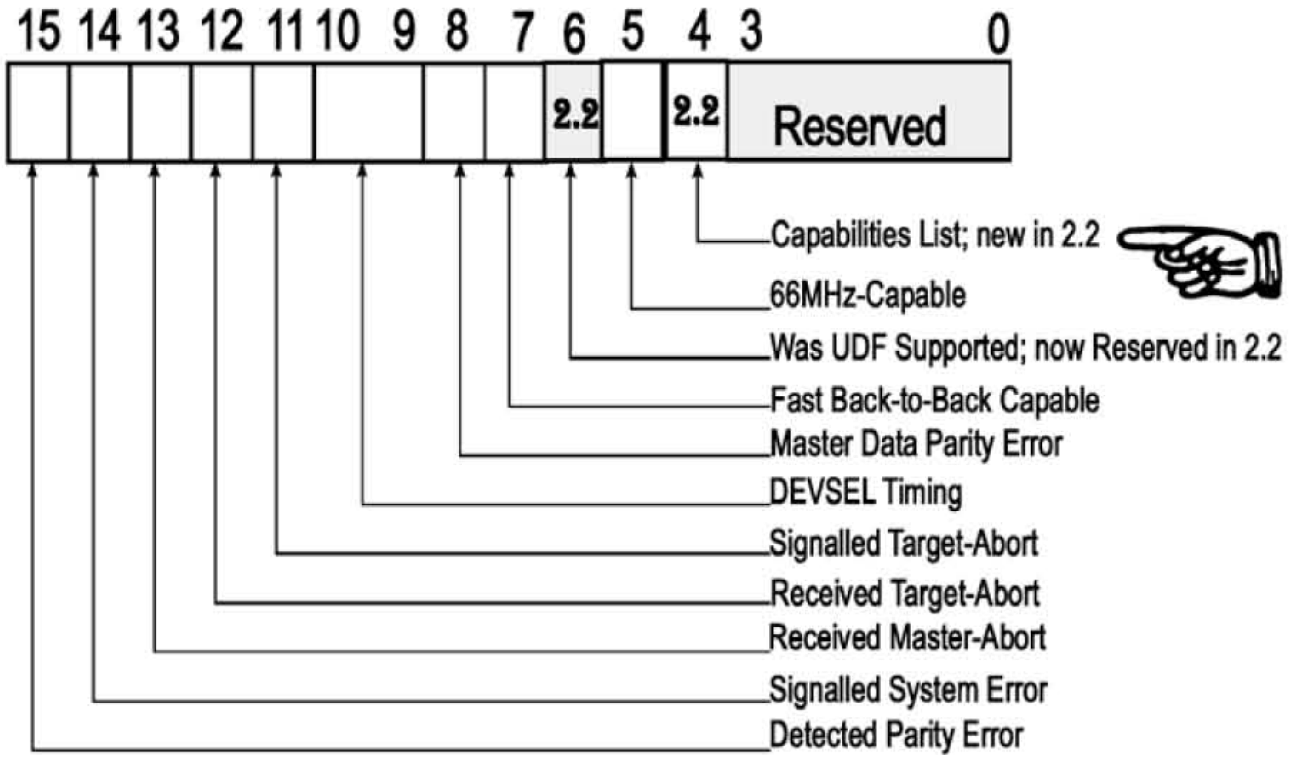
Min-Gnt Register: Timeslice Request

- Optional for a bus master and not applicable to non-master devices.
- The value indicates how long a burst period the device needs (in increments of 1/4 of a microsecond, or 250 us).
- The Max-Lat register and the Min-Gnt register and are registers used by the configuration software to determine:
  - how often a **bus** master typically requires access to the PCI bus and
  - the duration of a typical transfer when it does acquire the bus.

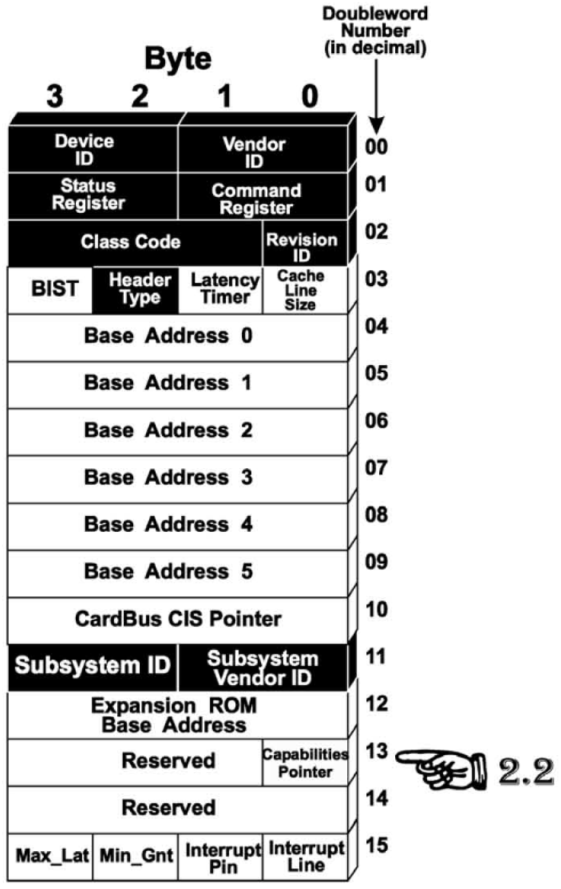
Max-Lat Register: Priority-Level Request

- Optional for a bus master and not applicable to non-master devices.
- The specification states that this read-only register specifies "how often" the device needs access to the PCI bus (in increments of 1/4 of a microsecond, or 250us).
- A value of zero indicates the device has no stringent requirement in this area.
- The Max_Lat register value indicates how often the master would like to have access to the bus (i.e., receive its **GNT#** from the arbiter).

PCI Status Register



New Capabilities


Doubleword Number (in decimal)
Byte
3
2
1
0
Device ID
Vendor ID
00
Status Register
Command Register
01
Class Code
Revision ID
02
BIST
Header Type
Latency Timer
Cache Line Size
03
Base Address 0
04
Base Address 1
05
Base Address 2
06
Base Address 3
07
Base Address 4
08
Base Address 5
09
CardBus CIS Pointer
10
Subsystem ID
Subsystem Vendor ID
11
Expansion ROM Base Address
12
Reserved
Capabilities Pointer
13
2.2
Reserved
14
Max_Lat
Min_Gnt
Interrupt Pin
Interrupt Line
15