PCI & PCIe Enumeration (Bus Scan) Process

by

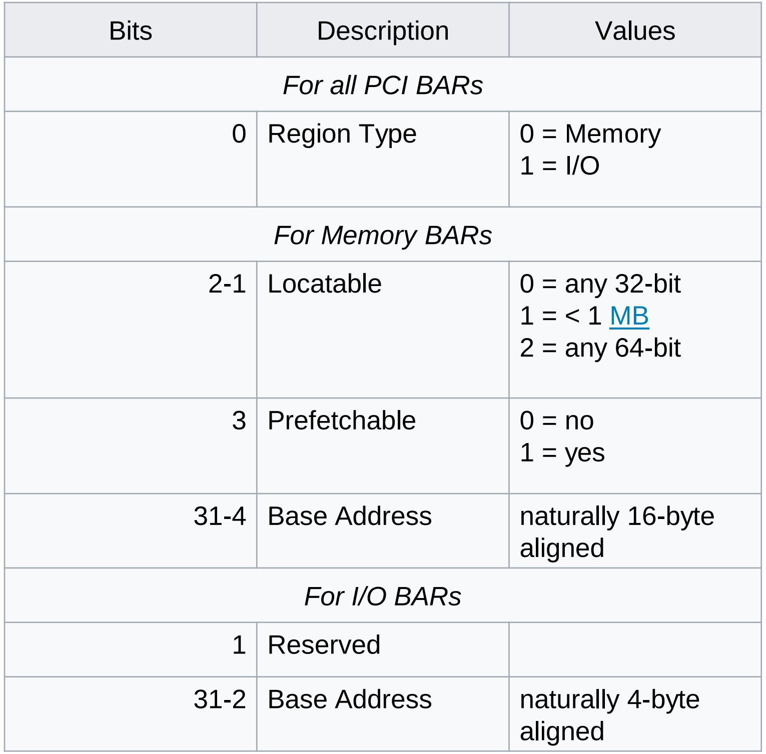
Jitesh Verma

altran

PCI & PCIe Enumeration/Scan Process

```
Uint32 vendor_id_device_id;
Uint8 config_header_type;
For (bus=0; bus < 256; bus++)
    For (device=0; device <32; device++)
              For (pci_function=0; pci_function < 8; pci_function++)
                        {
                                  vendor_id_device_id = read_vendor_id_device_id(bus, device, pci_function);
                                  if (vendor_id_device_id == 0xFFFFFFFF)
                                            break;
                                  config_header_type = Read_config_header_type();
                                  if (config_header_type == 0)        // PCI/PCIe Endpoint device
                                  {
                                            Allocate_system_resources_to_the_pci_function();      // Allocate memory/IO region & Intr
                                            Program_the_BARs_of_the_pci_function();
                                  }
                                  else if (config_header_type == 1) // PCI/PCIe Root-complex, switch or bridge device
                                  {
                                            Program_the_RC_or_Switch_or_Bridge_and_mark_a_new_bus();
                                            Scan_the_newly_detected_bus();
                                  }
                        }
```

PCI & PCIe Configuration – Base Address Register (BAR)



Bits	Description	Values
For all PCI BARs		
0	Region Type	0 = Memory 1 = I/O
For Memory BARs		
2-1	Locatable	0 = any 32-bit 1 = < 1 MB 2 = any 64-bit
3	Prefetchable	0 = no 1 = yes
31-4	Base Address	naturally 16-byte aligned
For I/O BARs		
1	Reserved	
31-2	Base Address	naturally 4-byte aligned

PCI & PCIe Configuration – Device & Register Addressing

Bit 31	Bits 30-24	Bits 23-16	Bits 15-11	Bits 10-8	Bits 7-0
Enable Bit	Reserved	Bus Number	Device Number	Function Number	Register Offset

PCI & PCIe Configuration Space – Register Offsets

Register	Byte Offset	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
0x0	0x0	Device Id		Vendor Id	
0x1	0x4	Status		Command	
0x2	0x8	Class Code	Sub-class	ProgIF	Revision Id
0x3	0xC	BIST	Header Type	Latency Timer	Cache Line Size
0x4	0x10	BAR0			
0x5	0x14	BAR1			
0x6	0x18	BAR2			
0x7	0x1C	BAR3			
0x8	0x20	BAR4			
0x9	0x24	BAR5			
0xA	0x28	CardBus CIS Pointer			
0xB	0x2C	Subsystem Device Id		Subsystem Vendor Id	
0xC	0x30	Expansion ROM BAR			
0xD	0x34	Reserved			Capability Ptr
0xE	0x38	Reserved			
0xF	0x3C	Max Latency	Min Grant	Interrupt Pin	Interrupt Line

PCI & PCIe Configuration - Resource Determination & Allocation

```
u32 bar0;
u32 base_address;
U32 initial_address = 0xABCDEF0X;          // Chosen by complex resource distribution algorithm
u32 info_bits_mask = 0xFFFFFFF0;           // 4 LSBits are read-only resource (Mem/IO) info bits
write_bar(BAR0, 0xFFFFFFFF);               // Write all ones
bar0 = read_bar(BAR0);                     // Read updated value (e.g. 0xFFFFF00X)
bar0 = bar0 & info_bits_mask;              // Zero out the 4 info bits (0xFFFFF000)
bar0 = ~bar0;                              // Invert the bits (0x00000FFF)
resource_length = bar0 + 1;            // Add 1 to get the resource (mem/IO) length (0x00001000 = 4KB)
base_address = initial_address | ~bar0;    // Base address of the resource (0xABCDE000)
write_bar(BAR0, base_address);             // Write the base address of the address range into BAR
```

Note: For 64-bit PCI/PCIe devices / system, pair of two BARs (e.g. BAR0 & BAR1) are combined to form 64-bit BARs.

PCI & PCIe Configuration References

- https://en.wikipedia.org/wiki/PCI_configuration_space
- https://stackoverflow.com/questions/19006632/how-is-a-pci-pcie-bar-size-determined
- https://pcisig.com/