OVERVIEW OF PCI(e) SUBSYSTEM
KISHON VIJAY ABRAHAM I, VIGNESH R
Introduction

• PCIe Vs PCI

• Usage model and software interface same as PCI

• Serial communication interface like USB, SATA

• Performance: Higher throughput, multi-lane

• Add peripherals to the system (USB, Ethernet, SATA, DCAN etc..)
Terminology

- Root Complex - PCIe host
- Endpoint - PCIe device
- cpu_addr - CPU physical address (proc/iomem)
- pci_addr - PCI bus address
- Switches - allow more devices to be connected
- Enumeration - discovering devices
Topology
Topology
Address Space
Configuration Space Header

1) read BAR = 0xFFFF0008
2) Clear last 4 bits = 0xFFFF0000
3) Invert the bits = 0x00000000
4) Add '1' = 0x1000000
   1MB

00h 04h 08h 0Ch 10h 14h 18h 1Ch 20h 24h 28h 2Ch 30h 34h 38h 3Ch
Device Id  Vendor Id
Status  Command

Class Code  Revision Id
BIST  Header Type  Lat. Timer  Cache Line S.  1000

Base Address Registers

Cardbus CIS Pointer
Subsystem ID  Subsystem Vendor ID
Expansion ROM Base Address
Reserved  Cap. Pointer
Reserved

Max. Lat  Min. Gnt  Interrupt Pin  Interrupt Line

Notes:
- 0 = Memory Request
- 1 = IO Request
- 00 = 32-bit Address Decoding
- 10 = 64-bit Address Decoding
- 1 = pre-fetchable
- 0 = non pre-fetchable
Address Space
Enhanced Configuration Access Mechanism (ECAM)

- Memory address (PCIe address space) determines configuration register accessed
- Function of bus number, device number, function and register number
Address Space

SoC

Root Complex
- Configuration Space
- IP Registers
- Controller Physical Address
- Memory Address

CPU

Memory

1GB

1GB MEM ADDR

PCle Address Space

4KB

100000h

200000h

400000h

PCle Endpoint
- Configuration Space
- Memory Space

PCle Bridge
- Configuration Space
- Memory Space

Bus:1 Device:0 Function:0

Bus:2 Device:0 Function:0

Bus:4 Device:0 Function:0
Address Space

SoC

CPU

Memory

1GB

Root Complex

Configuration Space

4Kb

IP Registers

Source Address | Destination Address | Size  | Type
---|---|---|---
A | <ECAM> | 4Kb | CFG0
B | B | 256MB | MEM

Controller Physical Address

X

Y

Z

2^{32/64} - 1

1GB MEM ADDR

PCIe Address Space

4Kb

4Kb

4Kb
### Configuration Space Header

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>Device ID</td>
<td>Vendor ID</td>
</tr>
<tr>
<td>04h</td>
<td>Status</td>
<td>Command</td>
</tr>
<tr>
<td>08h</td>
<td>Class Code</td>
<td>Revision ID</td>
</tr>
<tr>
<td>0Ch</td>
<td>BIST</td>
<td>Header Type</td>
</tr>
<tr>
<td>10h</td>
<td>Lat. Timer</td>
<td>Cache Line S.</td>
</tr>
<tr>
<td>14h</td>
<td>Base Address Registers</td>
<td></td>
</tr>
<tr>
<td>18h</td>
<td>Cardbus CIS Pointer</td>
<td></td>
</tr>
<tr>
<td>1Ch</td>
<td>Subsystem ID</td>
<td>Subsystem Vendor ID</td>
</tr>
<tr>
<td>20h</td>
<td>Expansion ROM Base Address</td>
<td>Cap. Pointer</td>
</tr>
<tr>
<td>24h</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>28h</td>
<td>Max. Lat</td>
<td>Min. Gnt</td>
</tr>
<tr>
<td>30h</td>
<td>Reserved</td>
<td>Interrupt Pin</td>
</tr>
<tr>
<td>34h</td>
<td>Reserved</td>
<td>Interrupt Line</td>
</tr>
</tbody>
</table>

**Notes:**
- **BAR Calculation:**
  - 1) read BAR = 0xFFF00008
  - 2) Clear last 4 bits = 0xFFF00000
  - 3) Invert the bits = 0x000FFFFFF
  - 4) Add '1' = 0x1000000

- **Address:** 1MB

- **Flags:**
  - 0 = Memory Request
  - 1 = IO Request
  - 00 = 32-bit Address Decoding
  - 10 = 64-bit Address Decoding
  - 1 = pre-fetchable
  - 0 = non pre-fetchable
Address Space

SoC

Root Complex

Configuration Space
4Kb

IP Registers

Controller Physical Address

CPU

Memory

1GB

PCle Address Space

1GB MEM ADDR

$2^{32/64} - 1$
Configuration Space Header

1) read BAR = 0xFFF00008
2) Clear last 4 bits = 0xFFF00000
3) Invert the bits = 0x000FFFFFF
4) Add '1' = 0x100000

1MB
Address Space

SoC

Root Complex

Configuration Space

IP Registers

Controller
Physical Address

1GB MEM ADDR

PCIe Address Space

Bus:1 Device:0 Function:0

PCIe Endpoint

Configuration Space

Memory Space

Bus:2 Device:0 Function:0

PCIe Bridge

Configuration Space

Memory Space

Bus:4 Device:0 Function:0
Interrupts
# Configuration Space Header

<table>
<thead>
<tr>
<th>Address</th>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h-01h</td>
<td>Device Id</td>
<td>Vendor Id</td>
</tr>
<tr>
<td>04h</td>
<td>Status</td>
<td>Command</td>
</tr>
<tr>
<td>08h</td>
<td>Class Code</td>
<td>Revision Id</td>
</tr>
<tr>
<td>0Ch</td>
<td>BIST</td>
<td>Header Type</td>
</tr>
<tr>
<td>10h</td>
<td>Lat. Timer</td>
<td>Cache Line S.</td>
</tr>
<tr>
<td>14h</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Base Address Registers</td>
</tr>
<tr>
<td>1Ch-1Fh</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cardbus CIS Pointer</td>
</tr>
<tr>
<td>20h-21h</td>
<td>Subsystem ID</td>
<td>Subsystem Vendor ID</td>
</tr>
<tr>
<td>24h-25h</td>
<td>Expansion ROM Base Address</td>
<td></td>
</tr>
<tr>
<td>28h-29h</td>
<td>Reserved</td>
<td>Cap. Pointer</td>
</tr>
<tr>
<td>30h-31h</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34h-35h</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>38h-39h</td>
<td>Max. Lat</td>
<td>Min. Gnt</td>
</tr>
<tr>
<td>3Ch-3Dh</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

**Message Control**
- Next Pointer
- Capability ID

**Message Address**
- Message Data

- 0 = INTx Not used
- 1 = INTA#
- 2 = INTB#
- 3 = INTC#
- 4 = INTD#
Interrupts
Configuration Space Header

<table>
<thead>
<tr>
<th>Device Id</th>
<th>Vendor Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Command</td>
</tr>
<tr>
<td>Class Code</td>
<td>Revision Id</td>
</tr>
<tr>
<td>BIST</td>
<td>Header Type</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Base Address Registers

Cardbus CIS Pointer

<table>
<thead>
<tr>
<th>Subsystem ID</th>
<th>Subsystem Vendor ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expansion ROM Base Address</td>
<td>Cap. Pointer</td>
</tr>
</tbody>
</table>

Reserved

<table>
<thead>
<tr>
<th>Max. Lat</th>
<th>Min. Gnt</th>
<th>Interrupt Line</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Message Control

<table>
<thead>
<tr>
<th>Next Pointer</th>
<th>Capability ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Address</td>
<td></td>
</tr>
<tr>
<td>Message Data</td>
<td></td>
</tr>
</tbody>
</table>
Interrupts

SoC

Interrupt Controller → Root Complex

CPU → Root Complex

Memory → Root Complex

Legacy
- Assert_INTx
- Deassert_INTx

MSI
- Write Message Data to Message Address

PCIe Endpoint

Configuration Space

Memory Space
Device Tree

pcie1: pcie@51000000 {
    compatible = "ti,dra7-pcie";
    reg = <0x51000000 0x2000>, <0x51002000 0x14c>, <A 0x2000>;
    reg-names = "rc_dbics", "ti_conf", "config";
    interrupts = <0 232 0x4>, <0 233 0x4>;
    device_type = "pci";
    #address-cells = <3>;
    #size-cells = <2>;
    ranges = <0x82000000 0 B B 0 0xffffd000>;
    interrupt-map-mask = <0 0 0 7>;
    interrupt-map = <0 0 0 1 &pcie1_intc 1>,
                     <0 0 0 2 &pcie1_intc 2>,
                     <0 0 0 3 &pcie1_intc 3>,
                     <0 0 0 4 &pcie1_intc 4>;

    pcie1_intc: interrupt-controller {
        interrupt-controller;
        #address-cells = <0>;
        #interrupt-cells = <1>;
    };
};
'ranges' Property

#address.cells = <3>
#size.cells = <2>
ranges = <0x82000000 0 B B 0 0xffffd000>;

PCI Address

CPU Address

00: Configuration Space
01: I/O Space
10: 32 bit Memory Space
11: 64 bit Memory Space

prefetchable (cacheable)
relocatable

<table>
<thead>
<tr>
<th>CPU</th>
<th>PCIE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Address</td>
<td>Destination Address</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
</tr>
</tbody>
</table>

Controller Physical Address

PCIe Address Space
Device Tree

pcie1: pcie@51000000 {
    compatible = "ti,dra7-pcie";
    reg = <0x51000000 0x2000>, <0x51002000 0x14c>, <A 0x2000>;
    reg-names = "rc_dbics", "ti_conf", "config";
    interrupts = <0 232 0x4>, <0 233 0x4>;
    device_type = "pci";
    #address-cells = <3>;
    #size-cells = <2>;
    ranges = <0x82000000 0 B B 0 0xffffd000>;
    interrupt-map-mask = <0 0 0 7>;
    interrupt-map = <0 0 0 1 &pcie1_intc 1>,
        <0 0 0 2 &pcie1_intc 2>,
        <0 0 0 3 &pcie1_intc 3>,
        <0 0 0 4 &pcie1_intc 4>;

    pcie1_intc: interrupt-controller {
        interrupt-controller;
        #address-cells = <0>;
        #interrupt-cells = <1>;
    };
};
Linux PCI(e) Subsystem

DOMAIN STACK

tg3  r8169  xhci-pci  ahci  sdhci-pci

PCI CORE

Other Arch  

Other Plat

pci-exynos  pci-imx6  pci-mvebu  pci-dra7xx

PCI BIOS

(drivers/pci/)

(arch/arm/kernel/bios32.c)

PCIE Driver
(drivers/pci/host/)
Interrupt Handling

(VIRT)IRQ 6 -> tg3 -> request_irq(6)
(VIRT)IRQ 7 -> xhci-pci -> request_irq(7)
(VIRT)IRQ 8 -> ahci -> request_irq(8)

Root Complex Driver

Generic_handle_irq(6)
Generic_handle_irq(7)
Generic_handle_irq(8)

request_irq(3)

Interrupt Controller

1 2 3
4 5

Root Complex

TG3
USB-PCI
SATA-PCI

SW
HW
lspci

- Displays all PCI buses, devices in a system

```
root@am57xx-evm:~# lspci
00:00.0 PCI bridge: Texas Instruments Device 8888 (rev 01)
01:00.0 PCI bridge: Pericom Semiconductor Device 2304 (rev 05)
02:01.0 PCI bridge: Pericom Semiconductor Device 2304 (rev 05)
02:02.0 PCI bridge: Pericom Semiconductor Device 2304 (rev 05)
03:00.0 SATA controller: ASMedia Technology Inc. ASM1062 Serial ATA Controller (rev 01)
04:00.0 USB controller: Etron Technology, Inc. EJ168 USB 3.0 Host Controller (rev 01)
```

Acknowledgements

- Jingoo Han, Pratyush Anand, Bjorn Helgaas
- Linux Community
- Texas Instruments
- Linux Foundation
References

- PCI Local Bus Specification 3.0
- PCI Express Base Specification 3.0
- PCI Express System Architecture (by Mindshare)
Happy Hacking!

Feedback:
kishon@ti.com
kishonvijayabraham@gmail.com

vigneshr@ti.com
vignesh.r.blr@gmail.com