

Homework 7

钟赟

2016K8009915009

1. 什么情况下需要对 Cache 初始化？Cache 初始化过程中使用的 Cache 指令 Index Store Tag（如 7.1.4 节示例代码）的作用是什么？列举一种 MIPS 下其它 Cache 指令并解释。

系统复位后，Cache 处于未初始化状态，里面可能包含着残留的或者随机的数据，如果不经初始化，对 Cache 空间的访问也可能导致错误的命中，此时需要对 Cache 进行初始化。

Cache 指令 Index Store Tag 的作用是对数据 Cache 进行写 TAG 操作，目的是将 Cache 块的状态设置为无效，从而实现 Cache 的初始化。

MIPS 下其它 Cache 指令：Index Invalidate D 指令：将指定索引的 DCache 块设置为无效，常用于作废指令缓存。

2. Cache 初始化和内存初始化的目的有什么不同？系统如何识别内存的更换？

Cache 初始化的目的：初始化 Cache 结构之后，处理器可以使用缓存结构，可以提高处理器的整体运行效率。初始化 Cache 时将 Cache 块内的数据均设为无效，防止使用 Cache 时使用其内遗留的随机数据儿出错。

内存初始化的目的：内存的读写性能大大提高，因此初始化内存可以提高数据传输效率和访问性能。内存初始化主要初始化内存控制器，初始化内存访问链路，而非内存的数据。

系统识别内存更换的方式：处理器通过 I2C 总线对外部内存条的 SPD 芯片进行读操作，以获取内存的配置信息，包括内存条的类型、容量、行列地址数、运行频率等，与上一次的记录历史进行对比，可以识别内存的更换。

3. 从 HyperTransport 配置地址空间的划分上，计算整个系统能够支持的总线数量、设备数量及功能数量。

以 Type 0 为例，配置访问的地址为：

<u>FDEEh</u>	总线号	设备号	功能	偏移	
39	24 23	16 15	11 10	8 7	0
总线数量: $2^8 = 256$ 设备数量: $256^5 = 2^{13}$ 功能数量: $2^{(13+3)} = 2^{16}$					

4. 根据 PCI 地址空间命中方法及 BAR 的配置方式，给出地址空间命中公式。

31	n	n-1	4	3	2	1	0
writeable bits		read only 0s		prefetchable	64bit	IO	

根据 BAR 寄存器定义，地址空间命中即访问的地址落在 BAR 配置的地址中，设访问地址为 ADDR，则命中公式为：

$$\text{HIT} = \text{ADDR}[31:n] == \text{BAR}[31:n] ? 1 : 0;$$

5. 多核唤醒时，如果采用核间中断方式，从核的唤醒流程是怎样的？

从核将中断使能置为有效；

主核将从核需要运行的程序指针、参数等信息准备好后，向从核发出中断请求；

从核进入中断入口，判断中断源，若为核间中断，则关中断，完成从核的唤醒，开中断；

主核收到从核被唤醒的消息后，继续下一个从核的唤醒或执行其他任务。

6. 在一台 Linux 机器上，通过“lspci -v”命令查看该机器的设备列表，并列举其中三个设备的总线号、设备号和功能号，通过其地址空间信息写出该设备 BAR 的实际内容。

基本信息由“lspci -v”命令查看，BAR 寄存器内容由“lspci -x”命令查看。

显卡 VGA compatible controller:

```
00:02.0 VGA compatible controller: InnoTek Systemberatung GmbH VirtualBox Graphics Adapter (prog-if 00 [VGA controller])
    Flags: fast devsel, IRQ 10
    Memory at e0000000 (32-bit, prefetchable) [size=16M]
    Expansion ROM at <unassigned> [disabled]
```

总线号: 00 设备号: 02 功能号: 0

```
00:02.0 VGA compatible controller: InnoTek Systemberatung GmbH VirtualBox Graphics Adapter
00: ee 80 ef be 03 00 00 00 00 00 00 03 00 00 00 00
10: 08 00 00 e0 00 00 00 00 00 00 00 00 00 00 00
20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30: 00 00 00 00 00 00 00 00 00 00 00 0a 01 00 00
```

BAR: 11100000 00000000 00000000 00000100

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

网卡 Ethernet controller:

```
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
    Subsystem: Intel Corporation PRO/1000 MT Desktop Adapter
    Flags: bus master, 66MHz, medium devsel, latency 64, IRQ 9
    Memory at f0000000 (32-bit, non-prefetchable) [size=128K]
    I/O ports at d010 [size=8]
    Capabilities: <access denied>
    Kernel driver in use: e1000
    Kernel modules: e1000
```

总线号: 00 设备号: 03 功能号: 0

```
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
00: 86 80 0e 10 07 00 30 02 02 00 00 02 00 40 00 00
10: 00 00 00 f0 00 00 00 00 11 d0 00 00 00 00 00 00
20: 00 00 00 00 00 00 00 00 00 00 00 00 86 80 1e 00
30: 00 00 00 00 dc 00 00 00 00 00 00 09 01 ff 00
```

BAR: 11110000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 11010000 00010001 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

USB 控制器 USB controller:

```
00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB (prog-if 10 [OHCI])
    Flags: bus master, fast devsel, latency 64, IRQ 10
    Memory at f0804000 (32-bit, non-prefetchable) [size=4K]
    Kernel driver in use: ohci-pci
```

总线号: 00 设备号: 06 功能号: 0

```
00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB
00: 6b 10 3f 00 06 00 10 00 00 10 03 0c 00 40 00 00
10: 00 40 80 f0 00 00 00 00 00 00 00 00 00 00 00 00
20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30: 00 00 00 00 00 00 00 00 00 00 00 00 0a 01 00 00
```

BAR: 11110000 01000000 00100000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000