# PowerPC的PCI总线的dts配置

powerpc使用称为FDT扁平设备描述树的机制传递给内核硬件配置参数，从而引导内核。这样的优势是PowerPC在Linux上的移植基本上都是对dts文件的修改，而升级内核的工作量远远小于其他cpu体系结构。

pci0: pci@e0008500 {

interrupt-map-mask = <0xf800 0 0 7>;

interrupt-map = <

/\* IRQ5 = 21 = 0x15, IRQ6 = 0x16, IRQ7 = 23 = 0x17 \*/

/\* IDSEL AD14 IRQ6 inta \*/

0x7000 0x0 0x0 0x1 &ipic 0x12 0x8

/\* IDSEL AD15 IRQ5 inta, IRQ6 intb, IRQ7 intd \*/

0x7800 0x0 0x0 0x1 &ipic 0x13 0x8

>;

interrupt-parent = <&ipic>;

interrupts = <66 0x8>;

bus-range = <0x0 0x0>;

ranges = <

0x02000000 0x0 0x90000000 0x90000000 0x0 0x10000000

0x42000000 0x0 0x80000000 0x80000000 0x0 0x10000000

0x01000000 0x0 0x00000000 0xe0300000 0x0 0x00100000

>;

sleep = <&pmc 0x00010000>;

clock-frequency = <66666666>;

#interrupt-cells = <1>;

#size-cells = <2>;

#address-cells = <3>;

reg = <

0xe0008500 0x100 /\* internal registers \*/

0xe0008300 0x8 /\* config space access registers \*/

>;

compatible = “fsl,mpc8349-pci”;

device\_type = “pci”;

};

这是一个mpc8379的dts配置文件,一些属性还是很好理解：

pci@e0008500说明这个pci控制器的寄存器映射基地址为e0008500

reg = <

0xe0008500 0x100 /\* internal registers \*/

0xe0008300 0x8>; /\* config space access registers \*/

再次证明，pci控制器的寄存器映射基地址为0xe0008500，读取配置空间使用的寄存器映射基址为0xe0008300，后面是长度，8个字节。即CFG\_ADDR和CFG\_DATA这两个寄存器。

最最费解的是ranges，用来描述cpu地址空间和pci地址空间的映射关系。

对于e300内核来说，一组配置由6个32位16进制组成。

ranges = <

0x02000000 0x0 0x90000000 0x90000000 0x0 0x10000000

0x42000000 0x0 0x80000000 0x80000000 0x0 0x10000000

0x01000000 0x0 0x00000000 0xe0300000 0x0 0x00100000 >;

这里对rangs属性使用颜色标识出来，总共分红色区、黑色区、粉色区、蓝色区。

红色区的三个是 #address-cells 由 phys.hi、 phys.mid、 phys.low 组成。这3组的第一个32位数表明映射的地址的属性。

Phys.hi：0x02000000是内存映射

Phys.mid：0x42000000也是内存映射，支持预取

Phys.low：0x01000000是IO映射

黑色区的2个32位数表示pci总线的地址空间，用2个双字因为PCI总线可能是支持64位寻址的。

粉色区的1个32位数表示cpu\_address ,即cpu存储器域地址空间，是cpu寻址的空间。e300是32位cpu，如果是e500内核，则需要2个双字了。

蓝色区的2个32位数表示映射长度。

内核启动消息里显示

mpc837x\_rdb\_setup\_arch()

Found FSL PCI host bridge at 0x00000000e0008500. Firmware bus number: 0->0

PCI host bridge /pci@e0008500 (primary) ranges:

MEM 0x0000000090000000 ..0x000000009fffffff -> 0x0000000090000000

MEM 0x0000000080000000 ..0x000000008fffffff -> 0x0000000080000000 Prefetch

IO 0x00000000e0300000 ..0x00000000e03fffff -> 0x0000000000000000

正好与之匹配。

分配配置空间要注意cpu存储器域地址空间，不要跟其他设备的cpu存储器域地址空间重复。