(1条消息) 摘要1：DTS语法说明_teddy99999的专栏-CSDN博客_dts语法

Device tree是一种简单的节点和属性的树形结构。属性是键值对，而节点可能包括属性和子节点。画一颗电路板上CPU、总线、设备组成的树，内核根据这棵树展开出platform_device、i2c_client、spi_device等设备，并根据节点内容为这些设备分配必要的内存、中断等资源。

节点语法

节点名称

无reg属性　　node-name

有reg属性　　node-name[@unit-address]

compatible属性

指定与driver的对应关系

reg属性

reg = <address1 [[length1] [address2[length2]]] … >

#address-cells　说明address的字节长度

#size-cells　　说明length的字节长度

ranges属性

用于非内存映射设备（总线设备？），指定如何从一个域名将局部地址转换到CPU地址域。

```
external-bus {
      #address-cells = <2>
      #size-cells = <1>;
       ranges = <0 0  0x10100000   0x10000     // Chipselect 1, Ethernet
              1 0  0x10160000  0x10000           // Chipselect 2, i2c controller
              2 0  0x30000000  0x1000000>;      //Chipselect 3, NOR Flash

       ethernet@0,0 {
         compatible = "smc,smc91c111";
         reg = <0 0 0x1000>;
       };

       i2c@1,0 {
         compatible = "acme,a1234-i2c-bus";
         #address-cells = <1>;
         #size-cells = <0>;
         reg = <1 0 0x1000>;
         rtc@58 {
            compatible = "maxim,ds1338";
            reg = <58>;
         };
       };

       flash@2,0 {
         compatible = "samsung,k8f1315ebm", "cfi-flash";
         reg = <2 0 0x4000000>;
       };
    };
```

空ranges属性的出现表示位于子地址空间的地址被1:1的映射到母地址空间。你也许要问为什么当地址可以被1:1映射的时候还要使用地址转换。一些总线(如PCI)拥有完全不同的地址空间，而这些地址空间细节需要出现在操作系统。其它则拥有DMA驱动程序，这些程序需要在总线了解真正的地址。有时设备需要被集合，因为他们都分享相同的软件可编程物理地址映射。

中断

Interrupt-controller

一个空属性表明一个节点作为一个接收中断信号的设备

#interrupt-cell

这是中断控制器节点的一个属性。它代表此中断控制器的interrupt specifier有多少cells

Interrupts

包括interrupt specifier列表设备的一个属性，设备上每个中断输出信号都有一个

DTS关联关系

linux设备驱动模型三要素：device，driver，bus。Device和driver分别注册到bus上，然后通过name匹配，最终走到driver的probe函数中。

传统的ARM Linux定义platform_device和注册过程都是通过静态写在arch/mach下面的文件中，为每一个外设都初始化好一个platform_device，然后在kernel初始化的时候把这些device注册到platform_bus_type中，然后在后续的driver初始化中再把相应的platform_driver结构体注册到platform_bus_type中，然后通过name匹配。

现在有了device tree，platform_device不需要静态定义在c文件中了，而是利用device tree的方法实现了动态生成platform_device。这样同一个内核就可以在不需要修改内核代码的情况下根据不同的DTS文件，动态生成针对不同硬件平台的platform_device。

DTS使用

编译命令　　dtc -Idts -O dtb -@ BB-UART1-00A0.dts > BB-UART1-00A0.dtbo

反编译命令　dtc -I dtb-O dts BB-UART1-00A0.dtbo > BB-UART1-00A0.dts

加载之前，一定记住要把编译好的dtbo文件放到**/lib/firmare/**目录中，否则程序是找不到你的dtbo文件的