

uboot 启动过程以及配置文件分析

1、Uboot 源码分析

- a) arch 目录: 主控芯片结构 /arch/arm/cpu
- b) board 目录: 具体硬件平台 /board/samsung
- c) board.cfg: 具体平台的配置文件
- d) makefile 文件

2、Uboot 的编译

- a) make tiny4412_config

--在 uboot 顶层目录下 makefile 执行完生成一个 tiny4412_config 配置文件

unconfig:

```
@rm -f $(obj)include/config.h $(obj)include/config.mk \  
$(obj)board/*/config.tmp $(obj)board/*/*/config.tmp \  
$(obj)include/autoconf.mk $(obj)include/autoconf.mk.dep
```

%_config:: unconfig

```
@$(MKCONFIG) -A $(@:_config=)
```

理论上 make xxx_config 都是可以的,但是这里的 xxx 是对应的开发板的名称,在 board.cfg 文件下有备注。

1、从 uboot 源码 board.cfg 文件中找出 tiny4412 的配置信息,利用这些信息生成一些文件。

2、根据配置信息在 include 下面创建指向集体架构的 asm 软件链接,arm 芯片是指向 arch/arm/include/asm, asm/config.h 文件在该目录下,这样编译的时候就能找到对应的文件。

3、根据配置信息创建 include/config.mk 文件,这个文件被顶层的 makefile 文件包含进行了,内容如下:

```
ARCH      =arm  
CPU       =armv7  
BOARD     =tiny4412  
VENDOR    =samsung  
SOC       =exynos
```

4、在 include 下创建了 config.h 文件,内容如下:

```
/* Automatically generated - do not edit */  
#define CONFIG_BOARDDIR board/samsung/tiny4412  
#include <config_defaults.h>  
#include <configs/tiny4412.h>  
#include <asm/config.h>
```

5、

启动流程分析：

- 1、SD EMMC IROM : 存储设备 掉电保存
—— 在这些设备上是不能运行程序的
- 2、DDR3 ISRM: 内存，运行程序，掉电丢失
- 3、Uboot 启动组成: BL1+BL2+uboot.bin+R/W+ZI
作用: 启动引导内核，一旦内核正常启动以后，Uboot 就没有用了。
- 4、Uboot 烧到 sd 卡的哪个位置?
SD 卡的第一块开始
- 5、启动过程: 整个代码还在 SD 中
IROM: 初始化必要的硬件 (时钟、中断...) 选择启动方式
BL1: 把 BL1 搬运到内部的 SRSM 中，重新初始化硬件，把 BL2 搬到 SRAM 中
BL2: 初始化外部 DDR3，把 OS 搬运到 DDR3 内存中运行
- 6、BL2 把 OS (uboot.bin+R/W+ZI) 400k 左右 搬运到 DDR3 中的哪个位置
说明: DDR3 0x40000000