

-boot-sunxi走读

○.背景

本文所属目录层次为:

- > 2.系统移植
 - > 1.基于linux-sunxi SDK的系统移植
 - > 2. u-boot-sunxi走读

经过前面一节的开发环境搭建，大家应该能够初次编译uboot了，但是如果没有思考过2星问题的朋友还是对uboot没有什么印象，本节就带大家走读uboot，了解其工程结构和工作原理。

(交流QQ群: 573832310, 上车口令: 爱荔枝)

一. u-boot-sunxi 目录走读

大致查看下u-boot-sunxi每个目录的文件（仅列出了在调试DDR时比较重要的文件，其它目录请自行打开浏览）

```

├── api                                //封装一些平台无关的操作，如字符串打印，显示，网络，内存
├── arch                               //
│   ├── arm
│   │   ├── cpu
│   │   │   ├── armv7
│   │   │   │   ├── sunxi            //cpu相关的一些操作，如定时器读取
│   │   │   │   │   ├── u-boot-spl.lds //spl的放置方法
│   │   │   │   │   └──
│   │   └── dts
│   │       ├── sun5i-a13.dtsi        //sun5i系列芯片的一些配置
│   │       ├── sun5i-a13-lichee.dts
│   │       ├── sun5i-r8-chip.dts
│   │       ├── sun5i-r8.dtsi
│   │       └── sun5i.dtsi
│   ├── lib                           //一些库文件
│   └── mach-sunxi
│       ├── board.c                   //board_init_f
│       ├── dram_sun4i.c              //ddr的操作，复位，时钟，延时，odt, etc.
│       └── dram_helpers.c            //ddr的设置及读写测试
├── board
│   ├── sunxi
│   │   ├── board.c                   //sunxi_board_init 入口
│   │   └── dram_sun5i_auto.c         //DRAM的一些默认参数

```

```

|   |   └─dram_timings_sun4i.h    //根据时钟生成的DRAM参数
├─ cmd                        //Uboot命令行的一些命令
├─ common                    //含spl
├─ configs                    //menuconfig里的默认配置,比如dcdc的各路电压
|   └─A13-Lichee_defconfig
├─ disk                      //硬盘分区的驱动
├─ doc
├─ drivers                    //外设驱动
├─ dts
├─ examples
├─ fs                        //多种文件系统
├─ include
|   └─ configs
|       └─ sunxi_common.h //预配置的参数,如串口号等
|       └─ sun5i.h
├─ lib                        //加密压缩等算法
├─ net                        //nfs,tftp等网络协议
├─ post
└─ scripts

```

如果需要移植新的版型，如上文所示，主要修改的就是**dts**文件和**configs**目录下的默认配置文件。具体**dts**文件的语法请搜索相关资料。

二.u-boot-sunxi启动流程走读

要让芯片跑起系统，对其**bootloader**必须非常了解，不然移植初期出现的问题将无从下手。笔者调试荔枝板的时候犯了个错误，在没有确认**DDR**正常运行的时候就贸然进行调试，结果出现的问题是如同堆栈溢出，或者指针指飞后的诡异情形，每次出错的位置总是不同，导致多次跟踪徒劳无返。

不过也正是因为犯了前面这个错误，导致笔者翻遍了**u-boot-sunxi**的代码，特别是**SPL**部分代码，使得调试稳定**DDR**后，移植中出现的问题马上就能找到原因，很快完成了**bootloader**的适配。

由于启动部分内容较多，现将**A13**的**bootloader**启动流程分成三个流程图将以展示，可以在github的资源文件目录下找到。

A13整体启动流程：简明介绍A13启动的几个过程

u-boot-sunxi spl部分流程：详细介绍spl部分启动流程

u-boot-sunxi uboot部分流程：详细介绍uboot部分启动流程

流程图只是起到一个路线图的作用，当你不太熟悉启动流程，而启动失败的时候，可以根据流程查看目前的启动进度，以及是在哪里卡住，可以快速定位出错的大致位置。

³三.关于**A13**的裸机用法

群里有朋友问**A13**的裸机用法，其实在**SPL**或**Uboot**下就是**A13**的裸机使用。

SPL下未初始化**DDR**，只能使用片内**48KB SRAM**，就完全是普通单片机用法，只是程序是需要从**tf**卡载入到内存里运行而已。这样使用可以不加**DDR**，只是内部**SRAM**较小。

Uboot下已经初始化了**DDR**，而且对多数外设已经有了驱动，使用起来会更方便，内存也可以充分利用。（像**RTT**之类的小型**RTOS**应该可以直接编译进**Uboot**了吧。。）

感兴趣的朋友可以尝试下在**SPL**或者**Uboot**下编程，可以对启动过程更有了解。

³四.小结

对于多数**linux**开发者来说，此部分内容只作了解即可。

此部分内容主要是**BSP**工程师的工作。下节打算讲下移植中遇到的最大的坑，也就是**DDR**的调试问题，这是关系到系统能否跑起来的关键问题，也是软硬件结合最紧密的问题，所以单独写一节来总结。