

基于 TI Davinci 架构的双核  
嵌入式应用处理器 OMAPL138 开发入门  
V1.0



LastRitter

superyongzhe@163.com

2011 年 4 月 4 日

## 一、简介

TI 的达芬奇架构嵌入式应用处理器使用 DSP 与 ARM 结合的非对称多核结构，当然现在也有管脚全兼容的单核。本文要介绍的就是其中的一款低功耗处理器 OMAP L138。

OMAP L138 包括一个主频 300M 的 ARM9 处理器内核和一个 300M 的 C6748DSP 内核(均是 32 位处理器)。此外还有大量外设在此不做说明，与通用的 ARM 与 DSP 内核相比有如下一些不同点：

1.内存映射，该处理器的外设与内存统一编址，DSP 与 ARM 共享 4G 内存空间。但其中有一部分 DSP 仅可见（如 DSP 的数据指令缓存），一部分 ARM 仅可见（如 ARM 的内部 RAM），其余的两者都可以访问，具体的请参考其芯片资料。

2.中断，与通用的 DSP 与 ARM 相比，达芬奇架构并没有在这两个核中增加用于双核通信及相互控制的指令（也许是本人没发现吧！如果你发现相关的说明，请发个 email 告诉我，谢谢!），然而却增加两个核之间的中断，共计 7 个。当双核需要进行通信时，首先把数据放在双方可以访问的内存上，然后给对方一个中断，对方在中断中接收传递过来的数据，以此实现双方的通信，个人认为这就是 DSP Link 实现的基本原理吧：)

3.能源与休眠配置模块（PSC），这个模块主要负责整个系统的能源管理，可以使能或者休眠 ARM 核，DSP 核以及大部分外设 ARM 核与 DSP 核的相互控制主要就依赖这个模块。对于 OMAPL138，上电时默认的是 ARM 核被禁止，DSP 核被使能。说以开机时是 DSP 核先启动，然后是 DSP 的 ROM Bootloader 做一些初始化后使能 ARM 核后，ARM 核才开始运行。不同的达芬奇处理器是不同的，比如 OMAPL137 就是反过来的，网上资料鱼龙混杂，弄不清楚的话就去看 PSC 模块中的 ARM 和 DSP 核的默认值就可以了。

## 二、 启动过程

1.开机上电,用户在 **BOOT** 管脚上配置的启动方式被锁定采样到 **SYSCFG** 模块的 **BOOTCFG** 寄存器,从而确定可处理器的启动方式。

2.PSC 模块中的启动默认值是: **ARM** 核休眠, **DSP** 核使能。所以这时候 **DSP** 启动, **DSP** 从片内 **DSP L2 ROM** 处读取指令执行。这里有 **TI** 已经固化的仅 **DSP** 核可以访问的 **ROM Bloder**, 简称 **DSP RBL**。

3.**DSP** 的 **RBL** 做一些简单的初始化后, 就通过 **PSC** 模块使能 **ARM** 核, 休眠自己。然后 **ARM** 就开从片内的 **RAM Local ROM** 处读取指令并执行, 这里面有 **TI** 已经固化好的 **ARM ROM Bootloader**, 简称 **ARM RBL**。

4.**ARM** 的 **RBL** 做一些初始化后, 根据 **BOOTCFG** 寄存器的设定, 从指定的地方读取用于启动 **U-Boot** 的 **Bootloader**, 即 **U-Boot BootLoader**, 简称 **UBL**。**UBL** 可以放在 **Nandflash**, 或者通过串口下载等方式取得。值得说明的是, 这里的 **UBL** 是(**AIS Application Image Script**)格式, 而不是通常的 **BIN** 格式。

5.当 **ARM RBL** 读取到 **UBL** 后就根据 **AIS** 格式中的命令初始化、加载并运行 **UBL**。然后就是 **UBL** 加载运行 **U-Boot**, **U-Boot** 根据启动参数启动 **Linux**, **Linux** 根据启动参数加载根文件系统。

6.这时 **ARM** 核上的 **Linux** 就运行起来了, 但 **DSP** 还处于休眠状态。在 **Linux** 经过一系列初始化后会执行跟文件系统中 **/etc/profile** 的指令。我们在这里添加插入 **DSP Link** 内核模块的命令, 就会在 **/dev** 目录下产生一个叫 **dsplink** 的设备文件。这个 **DSP Link** 内核模块就是通过前面所说的中断和 **PSC** 设定就可以完成与 **DSP** 的通信与控制。

7. 然后再执行我们自己的 **ARM** 端程序, 在 **ARM** 端程序中使用 **DSP Link** 的库函数来访问 **dsplink** 设备, 从而在用户空间实现对 **DSP** 核的控制与通信。在我们写的 **ARM** 端程序中使用 **DSP Link** 库函数加载 **DSP** 端的应用程序到内存, 使用 **DSP Link** 的 **PROC** 函数启动 **DSP** 核。然后 **ARM** 核和 **DSP** 核就都各自独立运行起来了, 通过 **DSP Link** 来进行通信。

### 三、 OMAPL138 相关软件包及开发工具

开发环境说明：除了 CCS3.3 是在 WindowsXP 上外，其他的都是在 Fedora14 上。

1. OMAP\_L138\_1\_00\_00\_11.bin - TI 的 OMAPL138 SDK，对于其他的达芬奇架构处理器都有对应的 DVSDK 可供下载，内容大同小异。

这里面包含的有如下内容：

pspdriers - DSP BIOS 平台驱动，包括 csl 等，编写驱动程序需要参考其中内容。

biosutils - DSP 端 OS 工具集，如 csl 片级支持库等。

codec\_engine - 基于 DSP Link 与 CMEM 之上的更高级的双核间通信方式。

dsplink -用于 ARM 与 DSP 间通信。

REL\_EDMA3\_LDD\_BIOS5 - DSP 端的 DMA 驱动库，方便 DMA 的管理及使用。

fc(Framework Components) - DSP 端软件框架，包括资源管理等。

linuxutils - Linux 端工具集，cmem 就在其中。

xdais - TI DSP 算法接口标准库。

ase(Audio SOC Example) -使用 DSP Link 进行音频编解码的例子。

Davinci-PSP-SDK - TI 提供的 Davinci 平台软件包，包括 nandwriter, ubl, uboot 和 rootfs 等。

2. bios\_setuplinux\_5\_41\_02\_14.bin - DSP BIOS 5 DSP 端的实时操作系统。

3. ti\_cgt\_c6000\_6.1.9\_setup\_linux\_x86.bin - TI DSP C6000 交叉编译器。

4. xdctools\_setuplinux\_3\_16\_01\_27 - RTSC(实时软件组件)工具集，这是 Eclipse 基金会有一个开源项目。用于工程管理，类似于 Make，但比 Make 更强大。

5. cs1omapl138\_1\_00\_00-v2\_setup\_linux.bin -基于 OMAP L138 的 Codec Engine 服务器例子，其中有 jpeg 等编解码算法的例子程序。

6. sourcery g++ - ARM9 交叉编译器，可以自己去 sourcery 的官方去下载。

7. CCS - 这里使用的是 CCS3.3，主要用于编译 nandwriter, ubl, 烧写 ubl 和 uboot。

#### 四、 BootLoader

相关说明:

1.OMAPL138 中的 nandwriter 是运行在 ARM 上的, 用于把 AIS 格式 UBL 和 U-Boot 烧到 Nandflash 上的一个程序, 使用 CCS 编译, 然后用仿真器加载在 ARM 核上运行, 通过交互式对话框就可以把 UBL 和 U-Boot 烧到指定设备上。

2.UBL 也是使用 CCS 创建的 ARM 端裸机程序, 首先编译成 bin 格式的目标程序, 然后使用 AIS 生成工具转换成 AIS 格式, 最后使用 nandwriter 烧到 nandflash 中。

编译烧写步骤:

1. 用 CCS 打开 Davinci-PSP-SDK 中提供的 ubl, 编译生成 bin 目标文件。
- 2.使用 AIS 生成工具, 再根据开发板进行配置, 把 bin 文件转换成 AIS 文件。
- 3.用 CCS 打开 Davinci-PSP-SDK 中提供的 nandwriter, 编译并使用仿真器加载运行。
- 4.在出现的交互式对话框中依次填入要烧写 ubl 核 uboot.bin 路径即可。

## 五、 DSP Link

DSP Link 是 TI 为达芬奇架构处理器提供的一种双核通信架构。在 DSP 端，DSP Link 作为 DSP BIOS 的一个驱动而存在。在 ARM 端，DSP Link 最为一个 Linux 的外设而存在，并通过应用层的函数库访问这个设备的形式进行操作。

进行 DSP Link 的软件开发需要以下的软件包：

sourcery - ARM 交叉编译器。

linux - Linux 内核，编译 DSP Link 内核模块需要。

cgt600 - DSP 交叉编译器。

DSP BIOS - DSP 实时操作系统，DSP 端使用 DSP Link 就必须这个。

xdctools - RTSC 工具集，DSP BIOS 以及很多 TI 的软件包都使用这个进行工程管理。

dsplink - DSP Link 软件包。

编译步骤：

1. 解压或安装以上软件包到同一个目录，如/opt/OMAPL138\_SDK。
2. 然后修改 dsplink 顶层目录中 Makefile 中的软件包路径到正确的为只可。
3. 在命令行中输入 make 编译即可。

额外说明：

使用我修改过的软件包可以大大简化编译过程，具体说明残开软件包顶层 Makefile 中的说明即可。

## 六、 CMem

由于 DSP Link 中的共享内存分配功能比较弱，无法分配太大的连续内存，所以 TI 又推出了 CMem，用于分配大片连续共享内存。

## 七、 Codec Engine

由于 DSP Link 比较底层,使用起来也不方便,所以 TI 又在此基础上设计了 Codec Engine。Codec Engine 的底层实现依然是依赖于 DSP Link 和 CMem 的,在上层对 TI 的 xDAIS 算法接口标准进行扩展 xDAIS-DM,简称 xDM。

xDM 使用面向对象的编程方法,把算法抽象成 4 类对象:视频,图像,语音和音频,即 VISA。

Codec Engine 工程管理使用 XDC 工具,所以所有模块间的访问,模块定义或模块实现等等都要遵循 RTSC 标准。RTSC 软件包的配置使用 xdcscript 脚本语言,模块接口定义使用 xdcspec 脚本语言,这两种脚本语言都是 JavaScript 脚本语言的扩展。包括用来进行 DSP BIOS 配置的 TextConf 语言也是 JavaScript 的扩展。JavaScript 实际上是 ECMA-262 标准中的 ECMAScript 脚本语言,参考其中内容将有利于理解这几种脚本语言。

## 八、 双核技术展望

目前 TI 的 DSP BIOS 6 已经改名为 SYS BIOS 6, 不仅仅可以在 DSP 上运行, 也可以在 ARM 等其他处理器上运行, 支持多个不同的处理期间的通信。在 DSP 端把原本的任务间通信和现在的处理器间通信合并在一起并进行了增强, 叫 IPC。在非 DSP BIOS 操作系统上叫 SYS Link, 取代原来的 DSP Link, 同样也对其进行了增强, 不再需要 CMem 了。而且这些模块的配置都可以动态进行, API 的功能也越来越强大, 使用很灵活。但可惜的是 Codec Engine 只能支持旧版的 DSP Link。

今年, 双核的处理器已经进入手机领域, 不久的将来双核将无处不在。在达芬奇的多核架构中, DSP 多用于处理计算任务, 而 ARM 处理, 存储核显示网络等任务。这仅仅是功能的划分, 但试想一下: 如果两个核都是 ARM, 一个运行 RTOS, 一个运行 Linux 呢? 在 RTOS 上处理实时任务, 在 Linux 处理非实时任务, 也许可以解决当前 Linux 不实时, RTOS 软件资源匮乏的现状呢? 这只是一种臆想, 仅仅是抛砖引而已。

## 九、 参考文档

### DataSheet:

OMAP-L138 Low-Power Applications Processor  
TMS320C6745C6747 DSP System Reference Guide  
TMS320C674x DSP Megamodule Reference Guide  
OMAP-L138 Applications Processor System Reference Guide  
TMS320C674x OMAP-L1x Processor Peripherals Reference Guide

### BootLoader:

Using the OMAP-L1x8 Bootloader

### SDK:

Building The OMAP-L1 SDK

### RTSC&XDCTools:

Real Time Software Components for Embedded Platforms

### DSP BIOS:

DSP BIOS TextConf User's Guide  
TMS320C6000 DSPBIOS 5.x Application Programming Interface (API) Reference Guide

### xDAIS:

TMS320 DSP Algorithm Standard Rules and Guidelines User's Guide  
TMS320 DSP Algorithm Standard API Reference User's Guide

### xDM:

xDAIS-DM (Digital Media) User Guide

### DSP Link & CMem :

Mastering the Art of Memory Map Configuration for Systems  
Multicore Programming Guide  
DSP BIOS Link-Based GPP-DSP System

### Codec Engine :

A Technical Overview of eXpressDSP-Compliant Algorithms for DSP Software Producers  
Codec Engine Application Developer User's Guide  
Codec Engine Server Integrator User's Guide  
Codec Engine Algorithm Creator User's Guide

## 十、 文档说明

人在网络，不能总是予取予求。

特此把自己在 TI 双核开发上的点点肤浅经验贡献出来。

愿对同行有所帮助，共同勉励前行！

个人理解难免存在错误，欢迎批评指正！

欢迎任意转载，但请保留出处！

本文最初发布在 [www.ourdev.com](http://www.ourdev.com) 的 ARM 分坛上，相关软件包，工具和版本更新，如无特说明，均在此处。

## 十一、 更新日志

2011-04-04 V1.0 创建