DSP引导流程分析及镜像制作

1.DSP启动流程

DSP上电或复位后，首先会从片上的只读存储器中（on-chip）执行ROM Boot Loader(RBL)，RBL的起始地址为0x20B00000。RBL为固化在DSP芯片内部的一段引导程序，其通过读取DEVSTAT寄存器的值，进行初始化引导等工作，该寄存器的值是在上电或复位期间，对BOOT配置引脚进行采样而得到的，具体参见器件手册。整个RBL引导过程可以分为两步：初始化和Boot处理

2.RBL初始化流程

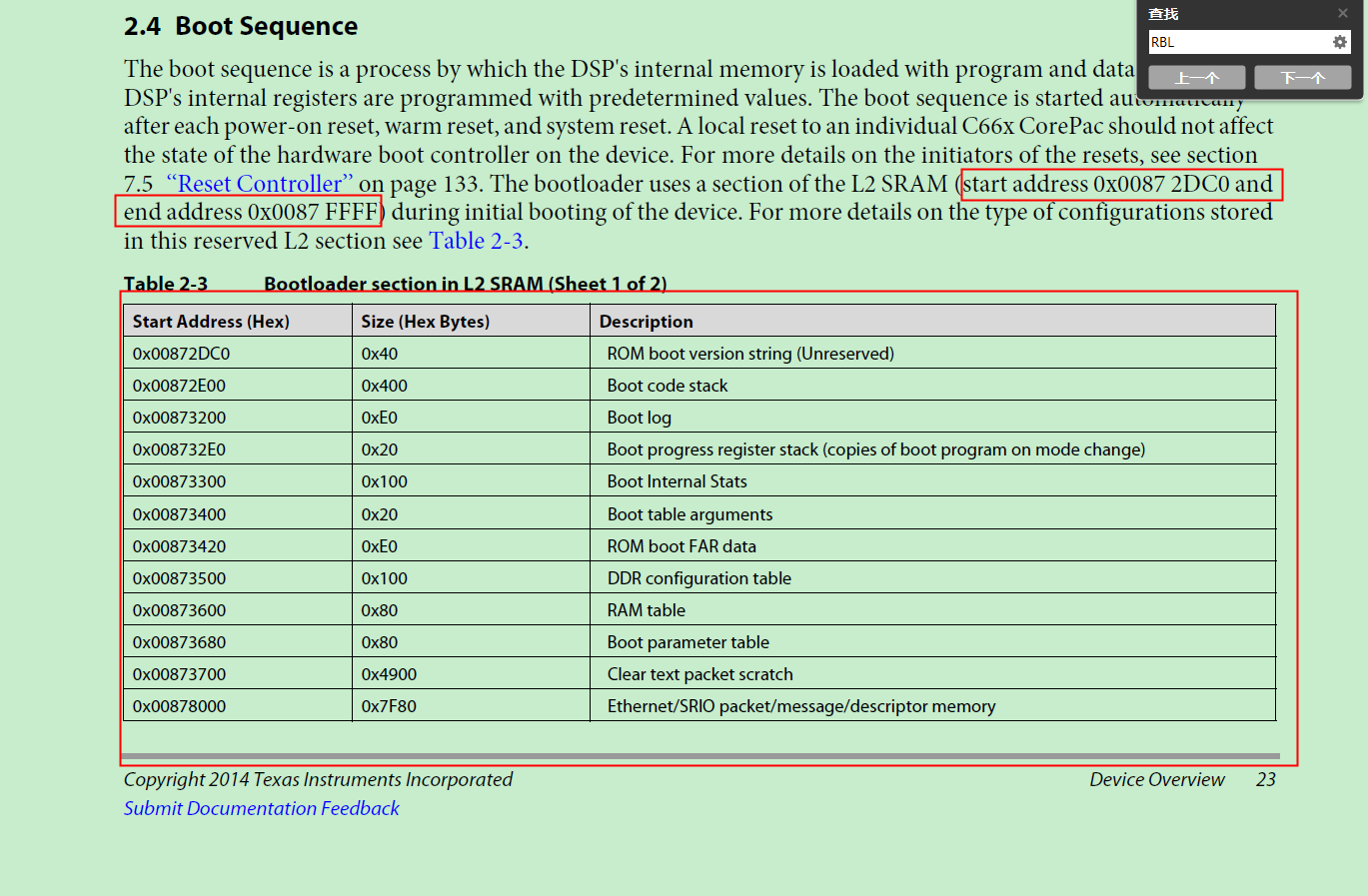
2.1 RBL在支持的外设中启用复位隔离

2.2为引导所需的外设启动电源和时钟域

2.3 配置系统PLL，设置设备的速度

2.3 在no-boot、SPI、I2C模式下，主PLL采用旁路模式

2.4 除了EMIF16启动模式外，RBL会在器件的L2中保留以短空用来进行引导处理，具体的起始地址和大小查看数据手册



2.5 禁用大部分中断

2.6 引导过程中，RBL会让其他的从核执行IDLE指令，并等待中断。当从核的应用程序加载完成后并且填充了BOOT\_MAGIC\_ADDRESS后，核0的应用程序可以触发IPC中断唤醒从核从BOOT\_MAGIC\_ADDRESS指定的地址开始执行程序

2.7 所有的L1P和L1D都配置为高速缓存，L2配置为存储器

2.8 RBL提供了配置DDR EMIF的能力，配置结构存储在L2中，当需要把程序加载到外部存储器时

2.9 RBL根据DEVSTAT寄存器值来厨师配置结构，即引导参数表。该表存储在L2的保留部分，不同的引导模式有不同的结构

2.10 RBL引导完成后，通过BOOTCOMPLETE引脚指示

3.RBL boot处理流程

针对不同的引导模式，boot处理可以分为三类：

RBL从辅助存储器加载镜像（内存引导）；

镜像从知道器件内存映射的主机加载

镜像从不知道器件内存映射的主机加载

针对每种引导模式，boot流程可以进一步分为以下三个部分：

初始化过程，根据所选的引导模式执行特定的初始化过程

镜像加载过程

切换过程，RBL切换到镜像执行

3.1内存引导模式：

该模式下，RBL控制引导设备的镜像下载过程，引导镜像需要转换为RBL可以理解的格式。在该模式下，镜像被转换为引导表（boot table），RBL从辅助存储器读取引导表，并将其放置与设备内部适当的存储器位置。RBL完成镜像下载过程后，会将程序计数器移动到引导表中捕获的C\_int00地址，并开始执行镜像。

3.2镜像从知道器件内存映射的主机加载

在此模式下，RBL等待主机加载镜像。主机加载完镜像后，需要提供给RBL c\_int00地址，所以，主机需要在L2中提供的保留区域的适当地址写入c\_int00地址，具体地址参见数据手册。

3.3镜像从不知道器件内存映射的主机加载

在主机启动模式的情况下，主机不知道启动设备的存储器映射，将镜像加载到设备中取决于RBL解码主机发送的数据的能力。在这种情况下，应将镜像转换为引导表。将此引导表镜像传输到设备取决于特定的引导模式。但是一旦此引导镜像到达设备，RBL就会对引导表进行解码并将镜像加载到特定的内存位置。一旦RBL完成将镜像下载到设备中，它就会将程序计数器移动到引导表中捕获的c\_int00地址并开始执行引导映像。

4.boot配置格式

RBL使用一组表来执行引导过程。RBL使用三种不同类型的表：

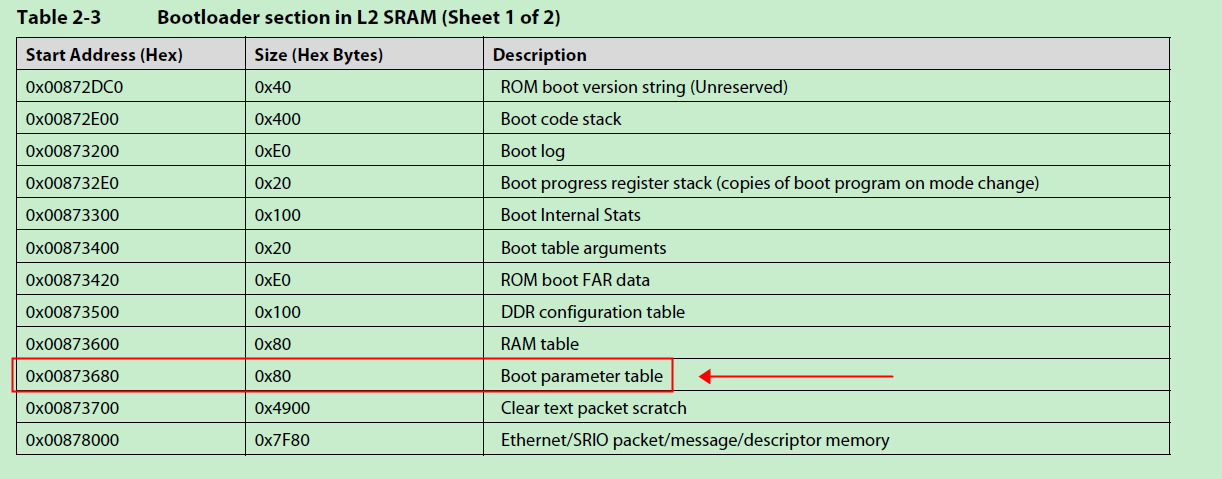
引导参数表（Boot parameter table）

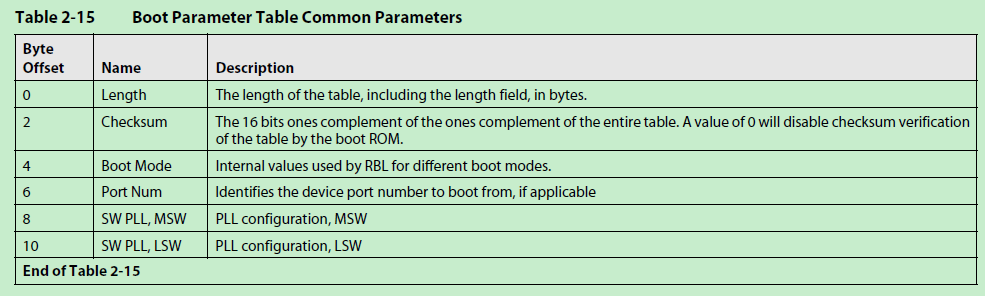
引导表（Boot table）

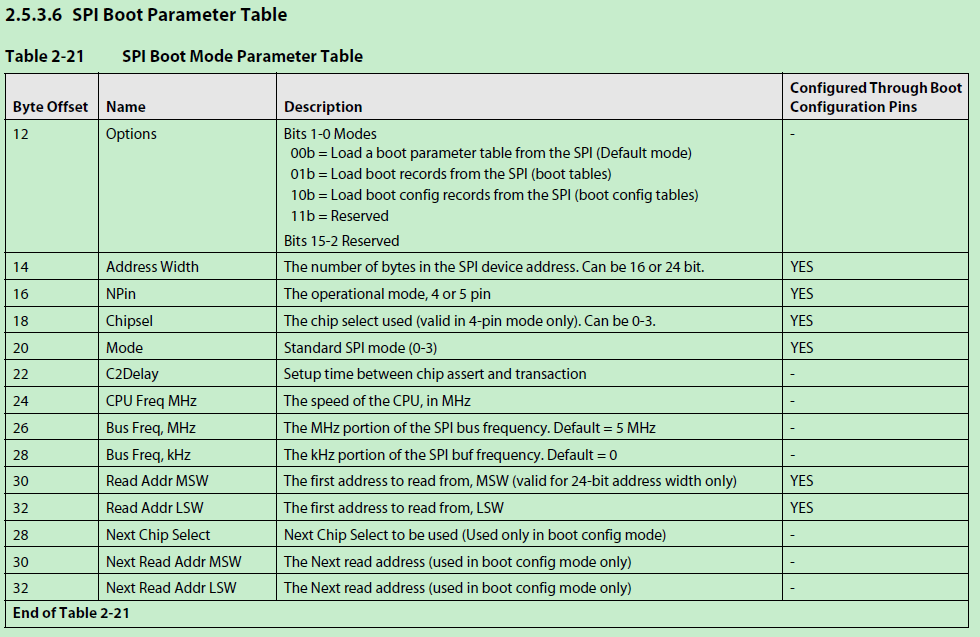
引导配置表（Boot configuration table）

4.1引导参数表（Boot parameter table）

引导参数表是RBL用于确定引导流的最常用格式。在所有引导模式下，引导参数表在表中具有前几个参数，而表格格式的其余部分取决于所选的引导模式。有关不同引导模式的引导参数表格式，请参见特定于设备的数据手册。 RBL将每个引导模式的默认引导参数表复制到CorePac0的保留L2部分，并根据通过引导引脚选择的引导配置修改默认值。此表格形成RBL执行引导过程的映射。







4.2引导表（Boot table）

镜像需要转换为RBL可识别的格式，该格式就是引导表。代码和数据有hex转换工具自动插入到引导表中。hex转换工具使用应用程序文件中链接器嵌入的信息来确定每个部分的目标地址和长度。将这些部分添加到引导表不需要用户进行特殊干预。hex转换工具将应用程序中的所有初始化部分添加到引导表。

每个部分都以相同的格式添加到引导表中。第一个条目是32位计数，表示以字节为单位的节长度。下一个条目是32位目标地址，其中复制了该部分的第一个字节。

RBL继续读取和复制这些部分，直到遇到字节数为0的部分。这表示引导表的结束。然后，引导加载程序分支到入口点地址（在引导表的开头指定）并开始执行应用程序。

引导表的格式：

32位标头记录，指示引导加载程序在完成数据复制后应分支的位置（c\_int00地址）

每个初始化section：

32位section长度

32位section地址（目标地址的副本）

需要拷贝的数据

一个32位的结束符（0x00000000）

4.3引导配置表（Boot configuration table）

如果在加载应用程序之前必须使用与其复位值不同的值对某些外设进行编程，则使用引导配置表。例如，如果需要将应用程序加载到DDR内存中，则可以使用引导配置表对DDR寄存器进行编程，并在将应用程序代码加载到DDR之前启用DDR外设。

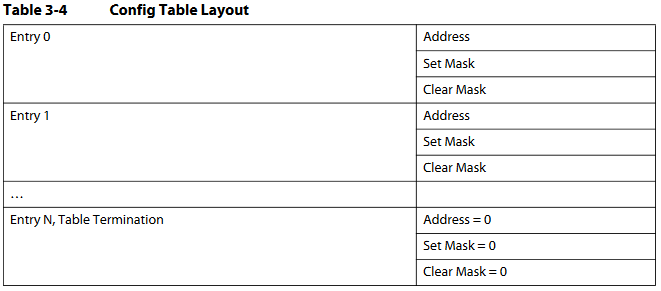
引导配置表中的每个表条目都有三个元素：

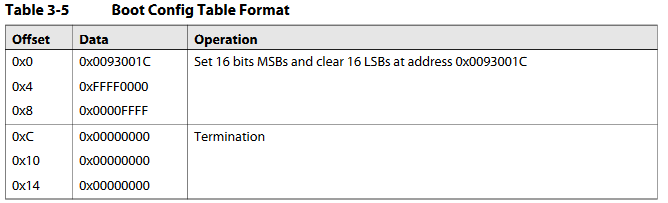
需要修改的地址

设置的掩码

清除的掩码

RBL读取指定的地址，然后设置在set mask元素中设置的任何位，并清除在clear mask元素中设置的任何位。如果set和clear掩码元素都为0，则地址字段中的值通过标准调用分支，返回地址存储在寄存器B3中。当所有三个元素都为0时，引导配置表终止





目前没有提供添加的工具

5用于生成不同表的实用程序

hex6x.exe 用于生成boot表

6.I2C引导

I2C引导模式分为两种，主模式和从模式，主模式下为DSP驱动存储镜像的I2C从器件。从模式为DSP被I2C主器件驱动。我们主要使用I2C EEPROM存储镜像，所以我们主要选择主模式。

加载Boot镜像从EEPROM

在该模式下，第一块将包含引导参数表，RBL读取引导参数表，并根据参数表配置相关硬件，然后RBL解析引导表以及加载镜像到指定的内存中。如果加入了引导配置表，则RBL会根据配置表设置相关寄存器，配置完成后RBL开始加载镜像。当镜像加载完成后，RBL修改程序计数器到c\_int00地址开始执行镜像

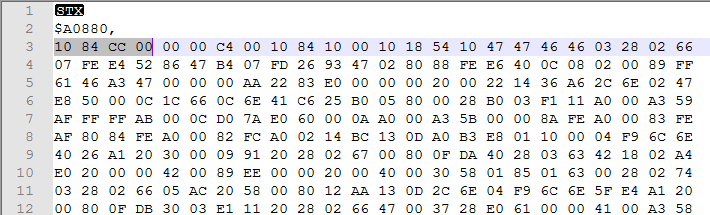
7.SPI引导

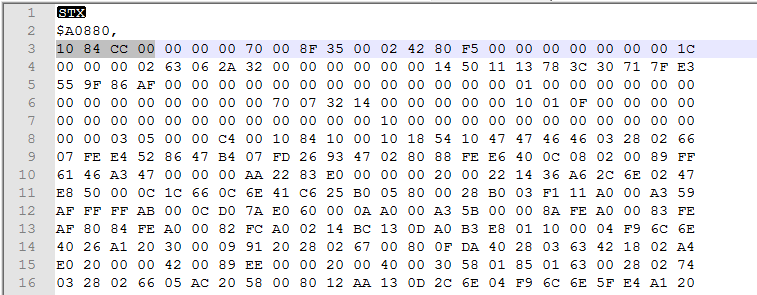
SPI引导流程和I2C基本类似，此处不再赘述。

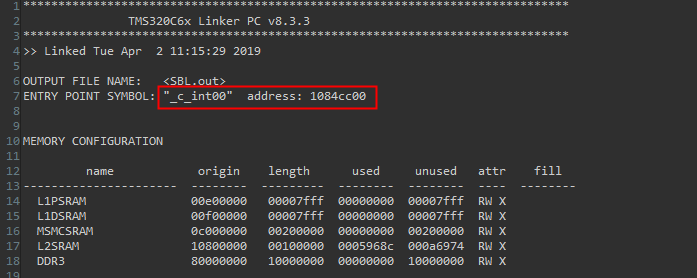
8.boot镜像实例展示：

完整的烧录镜像组成顺序依次为：boot参数表-boot配置表（可选）-boot表

经过hex6x转换后生成的引导表







经过一些列工具转换生成的可烧录镜像

当RBL开始复制section数据时，当RBL检测到DDR configuration table地址时（6678为：0x00873500），RBL会执行引导参数表配置

