

TMS320c6678 DSP程序加载方法与实现

宋 昆

陕西长岭电子科技有限责任公司 陕西 宝鸡 721006

摘要: 针对多核DSP系统程序加载复杂问题, 基于TMS320C6678对多核程序加载进行了研究和设计。为了快速生成多核可烧写文件, 采取了多个工具添加启动参数表、启动表、从核入口地址以及数据内容格式的转换。实现了多核程序烧写文件的快速一键生成和加载。

关键词: TMS320c6678; DSP

引言

在以DSP为核心的系统中, 通常将可执行代码存放在外围ROM或者FLASH存储器中, 代码需要在加电后自动加载运行。DSP系统的程序加载是指系统上电复位后, DSP自动将存储在外部存储器中的程序数据版已到内部存储器的过程, 使用外部存储器处理DSP的ROM存储资源可以灵活对系统代码进行升级维护。FLASH是一种电可擦除可重写的非易失性存储器, 容量大、体积小、重量轻、功耗低等优点, 在DSP系统开发过程中广泛应用。本文以DSP芯片TMS320C6678和FLASH存储器N25Q128A为例, 介绍了DSP系统TMS320C6678的基于FLASH程序自引导方法。

1 C6678 引导加载原理

系统上电后, 复位信号为低, 芯片复位。在复位信号上升沿处, 锁BOOTMODE[4:0]信号决定了芯片的存储器映射方式(memory map)、地址0处的存储器类型以及加载模式。该芯片提供3种加载方式^[1]: 不加载, ROM加载和主机引导。

1.1 不加载: CPU直接从地址0处开始执行程序, 地址0处的存储器可以是DSP内部存储器或者是外部SDRAM或SBSRAM。如果使用的是外部SDRAM存储器, CPU会先挂起, 直到SDRAM初始化完成;

1.2 ROM加载: DSP上电复位结束时, 自动通过DMA/EDMA控制器, 从外部空间ROM中搬移64K大小的程序代码到程序存储器的0地址处, 内部复位完成后, DSP开始从地址0处执行该代码。但这个程序量不能满足正常工程程序的需求, 因此需要在内部一级引导程序基础上设计二次boot程序, 完成应用程序从片外向片内加载的工作;

1.3 主机加载: CPU处于内部复位状态, 外部主机通过主机口初始CPU的存储器空间, 包括必要的EMIF或其他外设配置。当主机完成初始化配置, 它向主机口控制寄存器的DSPINT位写1, 结束加载过程, CPU退出内部复位状态, 开始从地址0执行指令。

2 系统概述

本系统采用TI公司的DSP芯片TMS320C6678, 该芯片目前广泛应用于信号处理、图像识别、数据统计等方面, 采用集成可视化开发环境CCS作为DSP的开发环境。

2.1 DSP和FLASH简介

TMS320C6678具有8个TMS320C66x内核, 每个内核可同时运行8项浮点乘法运算, 加之高达1.4GHz的时钟频率, 具有很高的浮点处理能力。CPU内核具有2套处理单元和两套寄存器组, 8个功能单元(.M1、.L1、.D1、.S1、.M2、.L2、.D2、.S2)都具备每个时钟周期执行一条指令的能力。功能单元执行所有乘法运算, 执行一组通用的算术、逻辑和分支函数。单元完成寄存器数据加载和保存。C66x内核A&B结构使.M单元的16位乘法器增至每个功能单元16个, 实现内核原始计算能力提升4倍。

系统中采用MICRON公司容量为128M的串行SPI FLASH存储器N25Q128A。该芯片支持1.8V单电压读写操作, 64个64Kb区块, 任意区块都可以被单独擦除; 通过时钟将指令、地址、数据串行的输入到芯片内部, 然后经过内部控制逻辑处理后, 按照指令要求对芯片进行操作。其支持标准SPI、DualSPI、QuadSPI三种模式, 快读速度最大支持133MHz。FLASH与TMS320C6678的信号接口如图1所示。

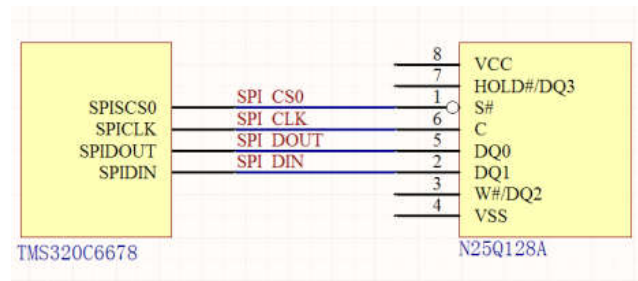


图1 DSP与FLASH存储器接口信号

3 FLASH 程序加载设计

由于TMS320C6678外部接口的丰富多样性, 其程序加

载也存在多种实现方式,如EMIF16, SRIO, Ethernet, PCIe, I²C, SPI和HyperLink七种方式。各种方式的区别仅在于接口类型以及基本配置的不同,对于程序加载来说,这七种方式的工作机制是相同的。本文以SPI接口为例,完成程序加载基于TMS320C6678的程序加载设计。

3.1 单核程序加载设计

C6678的一级引导程序能识别的SPI Flash中的数据存储器格式包括:SPI启动参数表和启动表两部分,其中启动参数表含CPU时钟、待加载表类型及偏移地址等信息;启动表是待加载表的一种,含程序段的大小、目的地址等信息;启动配置表常用与初始化DDR3。

在CCS开发环境中,针对调试的应用程序,编写相应的命令文件(.cmd),通过编译连接形成COFF格式的可执行文件(.out),通过开发环境自带的hex6x可执行文件,对实现Flash内存定位的命令文件进行处理,将.out可执行文件转换成ROM存储器识别的十六进制(.hex)或二进制文件(.bin),再将该文件烧写到ROM(Flash)中存储后,便可实现上电自启动。

3.2 多核程序加载设计

TMS320C6678中共有八个处理器,分别命名为core0~core7。其中,core0为主核,只有主核具有执行一级引导程序的权限,即处理器上电后,首先跳转至core0的入口地址执行。其他7个核在处理器上电后进入IDLE状态,等待core0向其发送IPC中断,即其他7个核是在core0的控制下来启动执行的。

TMS320C6678在每个核中预留了一块固定的内存,称为BOOT_MAGIC_ADDRESS,用来存放每个核开始执行后需要跳转到的入口地址。一级程序执行完毕后,便自动跳转至core0的入口地址0x1087FFFC执行,而其他7个核在接收到core0的IPC中断后,便向各个核对应的BOOT_MAGIC_ADDRESS读取相应的入口地址,然后跳转执行。

多核程序中,每个核中执行的程序可能不同,经过CCS分别编译后生成的.out文件也不同。C66以前的DSP文件格式转换,只需要使用hex6x十六进制转换工具;C6678需要增加后续多个工具进行数据处理。hex6x文件和.rmd文件配合使用,.rmd文件描述了输出控制、引导选项等内容,经过处理生成片上引导加载器加载程序所需的引导表。最终烧写至FLASH的程序数据包括的一个.dat文件里,需要把经过hex6x处理得到的8个.btbl合并成一个.btbl文件^[2]。

调用hex6x文件需要创建一个批处理文件spiboot_multi_8cores,存储hex6x调用的命令行选项和文件名的命令文件。spiboot_multi_8cores创建方式如下:

- a)首先新建批处理文件spiboot_multi_8cores.bat;
- b)批处理文件中输入调用工具hex6x及命令文件;

例如:“hex6x simple0.rmd hex6x simpleX.rmd……”

c)新建.rmd文件,编辑输入需要转换的.out文件,转换选项,定制转换的ROMS及SECTIONS伪指令,以0核为例命令代码见下图;

```

1 simple0.out
2 -a
3 -boot
4 -e_c_int00
5 -order L
6 -map SPIBOOT0.map
7 ROMS
8
9 {
10     ROM1: org = 0x0880, length = 0x2000000, memwidth = 32, romwidth = 32
11         files = { simple0.btbl }
12 }

```

d)批处理文件增加mergebtbl工具部分,将8个.btbl合并成一个.btbl文件,增加以下命令行:

“mergebtbl simple0.btbl simpleX.btbl……”。

多核.btbl文件格式首先是4B的0核程序入口地址,随后是0核段的段数据,将1核至7核的段数据依次加入0核段数据的末尾,合并过程中删除辅核程序的入口地址。

e)批处理文件增加以下命令行:

“b2i2c_ex simple.btbl simple.btbl.i2c
b2ccs_ex simple.btbl.i2c simple.i2c.ccs”

b2i2c_ex将.btbl文件转换成.i2c文件增加组长度和校验和,b2ccs_ex将.i2c文件转换成CCS可识别文件格式.ccs,并添加ccs header(1651---固定标示符,1---十六进制,10000---基地址,1---程序,97dc7---数据长度);

f)批处理文件增加“qfparse”,将boot参数表添加到.ccs文件前部;

g)批处理文件增加“modify i2crom.ccs spirom_le.dat”,器件是小端模式,需要将数据按字节翻转,生成.dat文件;

以上过程完成8核DSP程序的转换,生成统一的.dat文件可供外部烧写。

4 设计验证

本文采用一个简单多核亮灯工程进行测试。该工程包括8个核的.out文件,分别命名simple0.out~simple7.out,simple0.out大小为1845KB;simple1.out~simple7.out大小为367KB,执行相同程序。0核开始执行程序后向core1~core7发送IPC中断,core1~core7成功收到中断执行内部各核亮灯程序,多次试验程序成功执行。

结论:本文基于TMS320C6678设计并实现了FLASH的多核加载,试验验证表明,多核bootloader加载设计工作稳定,可靠性高,具有一定的应用价值。

参考文献

- [1]TM S320C620x/ C670x Boot Modes and Configuration Reference Guide [M]. Texas Instruments,2003, 12-14.
- [2]Texas Instruments. Multicore programming guide[Z]. SPRAB27A, August 2009.