

基于 DM642 的大容量 FLASH 引导加载方法研究与实现

李兴友, 游志胜

(四川大学 图象图形研究所, 四川 成都 610065)

(lixing-you@163.com)

摘 要: TMS320DM642 数字媒体处理器基于 C6000DSP 家族中的 C64xCPU。DM642 整合了许多外部设备, 用于视频和图像应用的开发。本文详细介绍了大容量 FLASH 存储与多级引导程序的开发过程和实现方法, 为以高性能媒体处理器为基础平台开发高端多媒体嵌入式系统提供了新思路 and 解决途径, 具有很大的实用性。

关键词: 数字媒体处理器; 大容量 FLASH; 引导加载

中图分类号: TP37 **文献标识码:** A

Study and implementation of the bulk FLASH memory bootloader based on DM642

LI Xing-you, YOU Zhi-sheng

(Institute of Image and Graphics, Sichuan University, Chengdu Sichuan 610065, China)

Abstract: The TMS320DM642 Digital Media Processor is based on the C64x CPU, a member of the C6000 DSP family. The DM642 integrates a number of peripherals to address the development of video and imaging applications. The development process and implementation methods of the multi-level bootloader for bulk Flash memory bootloading on DM642 Platform was introduced in detail, and a new way to develop top multimedia embedded system based on TMS320DM642 Digital Media Processor was given. It is proved that the way introduced here is practical and has a good prospect.

Key words: digital media processor; bulk FLASH memory; bootstrap

0 引言

TMS320DM642 是基于 C6000DSP 家族中的 C64xCPU, 主要是针对多媒体处理领域应用的一款高性能媒体处理器, 它是在 C64x 的基础上, 整合了许多的外围设备和接口, 包括三个独立可配置的视频端口: 视频输入, 视频输出和传输流输入。这些视频端口能支持 BT. 656 视频输入/输出、支持 10Bit 的数字高清晰电视 (HDTV) 的 Y/C 和 RGB 输入和输出以及 MPEG-2 传输流输入等。DM642 上还包括一个 10/100Mbps 的以太网 MAC (EMAC), 一个能支持标准音频接口模式的多通道音频串行接口 (McASP), 包括串行音频接口 (IIS) 和支持 S/PDIF、IEC60958-1、AES-3 和 CP-430 的集成数字音频接口发送器 (DIT) 两种音频模式, 一个支持 66MHz 时钟频率 32-Bit 的 PCI2.2 规范总线接口, 以及众多其他的外设。为了满足视频和图像处理的需要, DM642 将 VelociTI 体系结构和超长指令 (VLIW) 结构二者完美结合, 使得在一个指令周期能够并行执行 8 条指令, 工作主频从 600MHz 到 1.1GHz, 其处理能力从 4800MIPS 到 8800MIPS, 堪称业界最快的 DSP。因此, 它是视频图像处理与多媒体应用最为理想的硬件平台, 具有广阔的应用前景。

笔者在相关开发项目中, 均采用以 DM642 数字媒体处理器为中央处理器和 Xilinx 公司的高性能、低成本、通用性 Spantan-3 系列 FPGA 为视频图像协处理器为硬件平台, 与各种软硬件开发工具相结合, 较为圆满地完成项目开发与实施

工作。事实上, 这些项目的应用程序通常都非常庞大, 具有嵌入式系统的一个共同的特点——开发嵌入式系统的自引导程序。就其上述项目而言, 大型应用程序的自动引导加载与可靠稳定运行与否是关系到系统成败的关键, 因此, 开发基于高端 DSP 的大型应用程序的大容量存储与自引导加载并可靠稳定运行, 就成为了项目系统开发中的重点和难点之一。在深入研究和分析 DM642 硬件接口的基础, 结合项目自身特点, 给出了一套较为简单可行的软硬件实现方案。

1 FLASH 的外部接口和内部结构

FLASH 的外部接口简化连接如图 1 所示, 采用一片 TI 公司 TMS320DM642GDK600 和一片 Xilinx 公司 Spantan-3 系列的 XC3S400-4PQ208C 以及外挂一片 AMD 公司 AM29LV320D-TSOP48 或者是 TOSHIBA 公司 TC58FVB641FT-TSOP48 构成 4M 或 8M 字节的大容量 FLASH 存储池, 位于 DM642 的 CE1 地址空间, 宽度为 8 bits, 由于 CE1 的地址空间小于 FLASH, 且为 FLASH 只映射 (0x90000000 ~ 0x9007FFFF) 512K 字节的空间, 因此, 必须扩展地址线才能完全访问 FLASH。笔者采用 Xilinx 公司的 FPGA, 一方面除了用作视频信号预处理工作等外, 另一方面还可以用来产生 3 或 4 根 FLASH 扩展地址线, 并跟 DM642 一起协同工作, 把 FLASH 分为 8 或 16 页, 每页 512K 字节, 在系统上电或复位时, FLASH 位于第 0 页。在系统的初始化完成后, 可以通过 FPGA 控制选页, 从而实现 FLASH 各页的读写。另外, FLASH 除了用于存储一、二级引

导程序和应用程序外,同时还要存储 FPGA 的配置数据。从系统硬件方面解决了大型应用程序的存储与可靠稳定加载的问题。

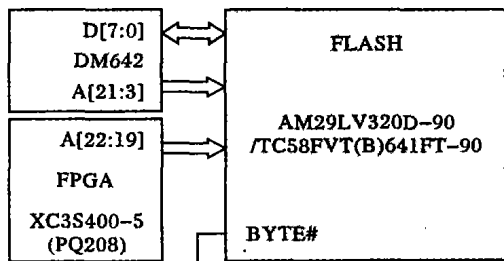


图1 FLASH 的外部接口简化连接

FLASH 的内部数据结构如图2所示,FLASH 第0页的前半页(256K 字节)为 boot_loader 段,用于存放用户的一、二级引导程序(Bootloader)(只能用汇编语言编写),以及小于 256K 字节的用户应用程序或者大型应用程序的二级加载用户程序(C 语言编写)。后半页(256K 字节)为 FPGA 配置数据段,用于存放 FPGA 的程序(XC3S400 配置大小为 1 699 136 字节),而第1页(512K 字节)为系统预留数据空间,以便特殊使用。第2页至第7/15 页用于存放大型应用程序,最大可存放 3M 或 7M 字节。

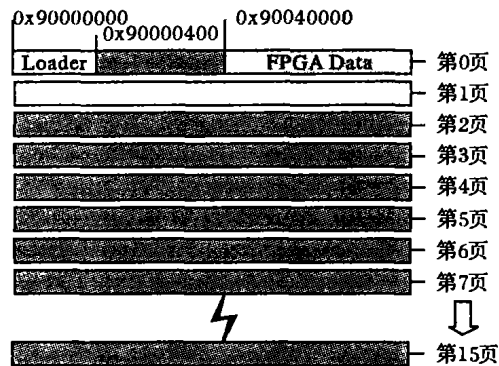


图2 FLASH 的内部数据结构

2 大容量 FLASH 引导模式实现

在嵌入式系统中,DM642 是以 ROM 方式引导系统的,当 DSP 上电或复位时,内核处于复位状态,并自动以 ROM 的读写时序从 FLASH 的第0 页起始地址开始复制 1K 字节的代码到 DSP 的片内内存起始地址为 0 开始的地址空间上。当代码传输完成后,DSP 复位状态结束,紧接着从 0 地址处开始执行指令,完成应用程序的加载,指向程序的入口点 c_int00() 地址处,并开始执行应用程序。

2.1 设计 FPGA 的 FLASH 扩展地址线的 VHDL 程序

利用 VHDL 语言设计产生有关 FLASH 扩展地址线的 VHDL 语言代码,并跟其他视频预处理 VHDL 语言代码一起,根据系统时序要求,建立 FPGA 的综合时序约束文件,笔者采用的是 Synplicity 公司 Synplify Pro 7.7 综合工具软件,经过编译、综合和反复处理优化处理与分析后,并产生最理想的综合和优化结果的 EDIF 文件;再利用 Mentor 公司针对 Xilinx 器件 ModelSim XE 5.6 专用仿真工具,进一步进行有关行为模式、转换、映射和布局布线的优化仿真设计;接着采用 Xilinx 公司最新高性能设计开发工具 ISE 6.3i 版,经过转换、映射、布局实现以后,生成器件相应的 bit 编程文件,再经过 ISE 内嵌的 iMPACT 配置工具将 bit 编程文件转换为 PROM

格式的 hex 配置文件,最后通过笔者所开发的转换软件,将 FPGA 的 hex 配置文件转换为 TI 公司的 ASCII hex 文件(ahex),以便供集成开发环境 Code Composer Studio(CCS)的 FlashBurn 实用程序对 FLASH 进行在线烧写。

2.2 开发一级引导程序(boot_load)

一级引导程序的大小不能超过 1K 字节,且必须用汇编语言编写。它最主要的功能:①是配置 DSP 的 EMIFA 控制寄存器,包括配置全局控制寄存器(GBLCTL)和各个空间控制寄存器(CExCTL)以及 SDRAM 的控制寄存器(SDCTL)、时序控制寄存器(SDTIM)和扩展控制寄存器(SDEXT)等,值得特别注意的是:各个空间的存储的类型和读写时序配置是否合理直接关系到系统是否稳定可靠地工作,笔者在此处感受最深;②是从 FLASH 的第0 页的前半页起始地址为 0x90000400 处开始把程序的各个段和数据拷贝到指定的存储器物理地址中;③跳到 C 程序的入口点 c_int00 处,建立 C 语言的运行环境,包括系统堆栈定义、初始化堆栈、初始化全局和静态变量等,为 DSP 运行 mian() 函数做好准备工作;④是 DSP 从 main() 处开始执行应用程序。

下面是一级引导程序代码片断:

```
.title "Flash bootup utility for DM642 Embedded Board"
.option D, T
.length 102
.width 140

; Address of the generated boot-table
COPY_TABLE .equ 0x90000400
; EMIF Register Base Address for c64x family
EMIF_BASE .equ 0x01800000
.sect ".boot_load"
.global _boot
_boot:
; * Debug Loop - Comment out B for Normal Operation
zero B1
_myloop:
; [!B1] B_myloop
nop 5
_myloopend: nop
; * Configure EMIFA registers
为配置 EMIFA 控制寄存器代码部分
; * Copy code sections
movkl COPY_TABLE, A3 ; load copy table pointer
movkh COPY_TABLE, A3
ldw *A3 ++, B1 ; load entry point
copy_section_top:
ldw *A3 ++, B0 ; byte count
ldw *A3 ++, A4 ; ram start address
nop 3
[!B0] b copy_done ; have we copied all sections?
nop 5
copy_loop:
ldb *A3 ++, B5
sub B0, 1, B0 ; decrement counter
[ B0] b copy_loop ; setup branch if not done goto copy_loop
[!B0] b copy_section_top ;
zero A1 ; clear zero
[!B0] and 3, A3, A1 ;
stb B5, *A4 ++ ;
[!B0] and -4, A3, A5 ; round address up to next multiple of 4
[ A1] add 4, A5, A3 ; round address up to next multiple of 4
; * Jump to entry point
copy_done:
```

```
b .S2 B1
nop 5
```

2.3 开发二级引导程序(Secondary_boot_load)

通常情况下,应用程序都是大于256K字节,同时跟小于256K字节的应用程序存储在FLASH中的位置也不同。因此,必须也要用汇编语言编写二级引导程序代码,把在FLASH中第2页到第7/15页的应用程序的各个段和数据拷贝到系统指定的存储器物理地址中,去完成大型应用程序的加载引导工作。当然,二级引导程序也可以加载引导小于256K字节的应用程序,只不过应用程序存储在FLASH中的位置不同而已。篇幅限制代码省略。

2.4 开发可引导(Bootable)的应用程序

对于在CCS集成开发环境中,开发的应用程序都是通过仿真环境自身的加载程序去完成的,要前应用程序具有自启动功能,下面将详细加以叙述:

(1)对于小于256K字节的应用程序的比较简单。其方法是:在CCS集成开发环境中,BIOS配置工具中的存储管理器中,对于小于256K字节应用程序必须建立一个起始地址为0长度为0x400的“BOOT”的内存段,同时,将“ISRAM”段的起始地址从0x00移到0x400处,且将原长度相应减去0x400;添加“.boot_load"代码段,即一级汇编语言引导程序;创建连接命令文件,并将“.boot_load"段代码映射到“BOOT”内存区。

(2)对于大于256K字节的应用程序的相对较为复杂。首先,必须开发跟小于256K字节的应用程序相类似的加载用户程序,主要的功能是:初始化的板级设备支撑库(Board Support Library)、从FLASH第0页后半页(0x90040000)中加载FPGA的配置数据并完成FPGA的从串行模式(Slave Serial Mode)上电初始化配置任务、设置加载应用程序FLASH页面指针、屏蔽全局中断、然后执行二级汇编语言引导代码,去完成应用程序的数据传输加载工作。在加载用户应用程序中,必须追加“.snd_boot_load"代码段,连接命令文件,并将“.snd_boot_load"段代码也映射到“BOOT”内存区。需要注意的是,必须确保FPGA能正确到FLASH进行选页操作,否则无法正确地加载应用程序。

应用程序的建立方法是:在存储管理器中建立两个存储段,一个是“ISRAM”段,跟上面小程序一样,另一个是“SDRAM”段,基地址是0x80000000,长度为SDRAM的大小即可。

(3)根据不同的FLASH修改FlashBurn插件程序。笔者是根据FLASH的数据手册修改FBTC6416系统的FBurnCmd.c程序有关代码而成。

(4)利用TI公司的Hex6x.exe二进制文件转换工具。用户应用程序和加载用户应用程序经过编译,生成的COFF格式的.out文件需转换成ASCII二进制.hex文件后,才能烧入FLASH。需正确地用户应用程序和加载用户应用程序设置Hex6x程序的命令行参数,下面分别是加载用户程序和用户应用程序的转换工具命令行参数文件内容。

加载用户应用程序命令行参数文件内容:

```
Release\load_userbigpro.out
/* 加载用户程序的 COFF 格式文件 */
-a /* Create ASCII image */
-memwidth 8 /* Width of ROM/Flash memory */
```

```
-map Release\load_userbigprohex.map
/* Create a hex map file */
-boot /* Create a boot table for all initialized sects */
-bootorg 0x90000400 /* Address of the boot/copy-table */
-bootsection .boot_load 0x90000000
/* Section containing system first boot routine */
ROMS
{
    FLASH: org = 0x90000000, len = 0x40000, romwidth = 8, files
    = {load_userbigpro_ahex.hex}
}
用户应用程序命令行参数文件内容:
Release\mpeg4enc.out /* 用户应用程序的 COFF 格式文件 */
-a /* Create ASCII image */
-memwidth 8 /* Width of ROM/Flash memory */
-map Release\mpeg2enchex.map /* Create a hex map file */
-boot /* Create a boot table for all initialized sects */
-bootorg 0x90100000 /* Address of the copy-table */
ROMS
{
    FLASH: org = 0x90000000, len = 0x200000, romwidth = 8,
    files = {mpeg4enc_ahex.hex}
}
```

其中,0x90100000为FLASH的第2页的起始地址,0x200000为2M字节文件长度,视应用程序的规模大小,可以改变其文件大小,但不能超过项目系统的最大3M或7M字节。

(5)利用FlashBurn实用程序烧录程序。首先,必须完全擦除FLASH中的全部内容,按照先烧录FPGA程序,接着再烧录小于256K字节的应用程序或者加载用户应用程序。对于烧录大于256K字节的应用程序时,必须在前面的两个程序烧录结束后,重新启动系统,待FPGA正常工作时,再烧录应用程序。

3 结语

大容量FLASH存储与多级引导程序的开发过程和实现方法,解决了在实际应用中的大型应用程序、大数据量的特殊要求,实践证明,本文提出的方法是可行的,为以高性能媒体处理器为基础平台,开发高端多媒体嵌入式系统提供了新思路 and 解决途径,具有很大的实用性。

参考文献:

- [1] 李方慧,王飞,何佩琨. TMS320C6000系列DSPs原理与应用[M]. 第2版. 北京:电子工业出版社,2003.
- [2] 孙航. Xilinx可编程逻辑器件的高级应用与设计技巧[M]. 北京:电子工业出版社,2003.
- [3] Texas Instruments Inc. TMS320DM642 Video/Imaging Fixed-Point Digital Signal Processor(spr5200b)[Z]. May 2003.
- [4] Texas Instruments Inc. Creating a Second-Level Bootloader for FLASH Bootloading on TMS320C6000 Platform With Code Composer Studio(spr999a)[Z]. August 2004.
- [5] Texas Instruments Inc. TMS320C6000 Peripherals Reference Guide(spru190d)[Z]. February 2001.
- [6] Xilinx Inc. Xilinx DS099 Spartan-3 Complete Data Sheet[Z]. January 2003.
- [7] AMD Inc. Am29LV320D Data Sheet[Z]. July 2003.
- [8] TOSHIBA Inc. TC58FV7641/B641FT-10 Data Sheet[Z]. August 2000.