

通过TMS320VC5410烧写FLASH实现并行引导装载的研究

叶海炳

重庆大学通信工程学院(400044)

2008-09-05

摘要: 随着DSP的广泛应用,如何对DSP系统上的FLASH重新编程的问题日益突出。在介绍FLASH存储器 Am29LV200B操作方法的基础上,通过TMS320VC5410对FLASH进行程序烧写,从而实现TMS320VC5410上电后用户程序并行自举引导。

关键词: DSP FLASH TMS320VC5410 并行 引导装载 编程

FLASH是一种可在线进行电擦写而掉电后信息又不丢失的存储器。它具有功耗低、容量大、擦写速度快等特点,并且使用内部嵌入算法对芯片进行操作,因而在很多采用DSP+FLASH自举引导装载方案的数字信号处理系统中得到了广泛的应用。本文通过一个完整的实例,介绍Am29LV200B FLASH存储器的烧写方法,实现TMS320VC5410(以下简称C5410)上电后用户程序的并行引导装载。

1 Am29LV200B FLASH存储器

1.1 FLASH存储器简介

Am29LV200B是AMD公司生产的FLASH存储器,其主要特点^[6]有:3V单电源供电,可使内部产生高压进行编程和擦除操作;支持JEDEC单电源FLASH存储器标准;只需向其命令寄存器写入标准的微处理器指令,具体编程和擦除操作可由内部嵌入的算法实现,并且可以通过查询特定的引脚或数据线监控操作是否完成;可以对任一扇区进行读、写或擦除操作,而不影响其它部分的数据。本文中128K×16位Am29LV200B FLASH映射为C5410的片外数据存储区,地址为:0x8000~0xFFFF,数据总线16位,用于16位方式的并行引导装载。128K的FLASH ROM用32K地址分四页进行访问,上电加载程序时使用FLASH ROM的第3页。

1.2 Flash存储器的操作命令

只要向FLASH存储器的特定寄存器写入地址和数据命令,就可对FLASH存储器编程。但要按一定的顺序操作,否则就会导致FLASH存储器复位。由于编程指令不能使“0”变为“1”,只能使“1”变为“0”,而擦除命令可使“0”变为“1”,所以正确顺序是先擦除、后编程。下面介绍几个常用的操作命令:编程命令、擦除命令、读数据命令、复位命令。

(1)编程命令。该命令向FLASH的指定地址写入数据,需要四个总线周期,前两个周期是解锁周期,第三个周期建立编程命令,最后一个周期则向编程地址写入编程数据,如表1所示。

表1 编程命令

周期	1(解锁)	2(解锁)	3(建立)	4(编程)
地址	0x555	0x2AA	0x555	pa(编程地址)
数据	0xAA	0x55	0xA0	pa(编程数据)

由于向每个编程地址写入数据都需要四个周期,所以在循环写FLASH时使用宏比较简单。FLASH ROM的首地址为0x8000,故偏移地址0x555对应的物理地址就为0x8555。_WRI TECOMMAND是实现一个周期编程的写命令宏,而_WRI TEFLASH则是对指定地址进行编程的四个完整周期。

具体编程程序如程序1、程序2所示。

```

程序1
_WRI TECOMMAND .macro pa,pd
; 单一周期编程的写命令宏,
; pa是编程地址,pd是编程数据
PSHM AR1
STM pa,AR1 ; AR1指向编程地址
LD pd,A
STL A,*AR1 ; 把编程数据放入AR1的编程地址
RPT #12
NOP
POPM AR1
.endm
程序2
_WRI TEFLASH .macro par,pdr

```

美国德州仪器公司

- 公司介绍
- 联系公司
- 公司新闻
- 加入收藏夹

热点专题

- 2008嵌入式技术创新及应用高峰论坛
- 2008飞思卡尔技术论坛
- Altera公司SOPCWorld 2008专题报道
- 第十届高交会电子展
- 科技闪耀北京奥运
- ADLINK DAY—2008年量测与自动化技术国际高峰论坛
- 中国电子学会Xilinx杯开放源码硬件创新大赛
- 赛灵思公司Virtex-5系列FPGA
- 3G知识
- IPTV
- 触摸屏技术
- RoHS

杂志精华

- 基于CC2430的无线传感器...
- 无线传感器网络应用系统综述
- 无线传感器网络在野外测量中的...
- 基于竞争的无线传感器网络
- 用于矿井环境监测的无线传感器...
- 具有自适应通信能力的无线传感...
- 基于传感器网络技术的深孔测径...
- 基于无线传感器网络的家庭安防...
- 基于ATmega128L与C...
- 无线传感器网络中移动节点设备...

```

; 编程宏, par是编程地址寄存器
; pdr是存放编程数据的寄存器
_WRI TECCOMMAND #8555H, #0AAH ;周期1(解锁)
_WRI TECCOMMAND #82AAH, #0A0H ;周期2(解锁)
_WRI TECCOMMAND #8555H, #0A0H ;周期3(建立)
LD pdr, A ;周期4(编程)
STL A, par ;把pdr寄存器中数据放入par寄存器的地址
RPT #12
NOP
_JUDGE par, pdr
; 检测编程是否正确, 见FLASH的操作检测
.endm

```

(2)擦除命令。该命令有片擦除和扇区擦除两种, 都需要6个总线周期, 前两个是解锁周期, 第三个是建立周期, 第四、第五两个是解锁周期, 最后是片擦除或扇区擦除周期, 如表2所示。

表 2 擦除命令

周期	1(解锁)	2(解锁)	3(建立)	4(解锁)	5(解锁)	6(片擦除)	6(扇区擦除)
地址	0x555	0x2AA	0x555	0x555	0x2AA	0x555	SA(扇区地址)
数据	0xAA	0x55	0x80	0xAA	0x55	0x10	0x30

一旦执行编程或擦除命令后, 就启动FLASH的内部编程或擦除算法, 自动完成编程或擦除操作。擦除程序如程序3所示。

```

程序3
_ERASEFlash .macro ;擦除宏
_WRI TECCOMMAND #8555H, #0AAH ;周期1(解锁)
_WRI TECCOMMAND #82AAH, #055H ;周期2(解锁)
_WRI TECCOMMAND #8555H, #080H ;周期3(建立)
_WRI TECCOMMAND #8555H, #0AAH ;周期4(解锁)
_WRI TECCOMMAND #82AAH, #055H ;周期5(解锁)
_WRI TECCOMMAND #8555H, #010H ;周期6(片擦除)
STM #8555H, AR3
LD #010H, A
STL A, *AR5
_JUDGE *AR3, *AR5
;检测擦除是否结束, 见FLASH的操作检测
.endm

```

(3)读数据命令。上电或内部编程、擦除操作结束后就进入读数据状态, 写入要读取的地址即可读出该地址的数据。

(4)复位命令。该命令使存储器复位, 进入读数据状态。向任何一个地址写入数据0xF0就能使FLASH存储器复位。在进行编程、擦除之前, 都应先复位, 在编程或擦除等正常操作中出现问题时也要复位。复位程序如程序4所示。

```

程序4
_RESETFlash .macro ;复位宏
_WRI TECCOMMAND #8001H, 0F0H
;向8001H写入0F0H使FLASH复位
RPT #12
NOP
.endm

```

1.3 Flash的操作检测

FLASH内部的编程或擦除算法可自动完成编程或擦除操作, 但必须了解其内部的操作检测机制, 以便知道操作是否完成或正确。常用的检测状态位有: 跳变位(DQ6)、超时标志位(DQ5)、数据查询位(DQ7)和Ready/Busy引脚(RY/)。检测的方法有三种。第一种是判断引脚RY/的状态。在编程、擦除或擦除挂起操作过程中, RY/引脚一直为“0”, 操作完成后变为“1”。第二种是检测跳变位DQ6, 在编程或擦除时对任何地址进行连续的读均引起DQ6连续跳变, 直至操作结束才停止跳变。最后一种是使用数据线的DQ7、DQ5。DQ7位在编程或擦除过程中输出的数是写入该位数据的反码, 当操作完成时输出才变为写入该位的数据; DQ5的状态为“1”时表示操作超时, 此时应再读一次DQ7的状态, 若DQ7输出仍不是写入的数据, 则操作失败, 复位FLASH。其流程如图1所示, 检测程序如程序5所示。

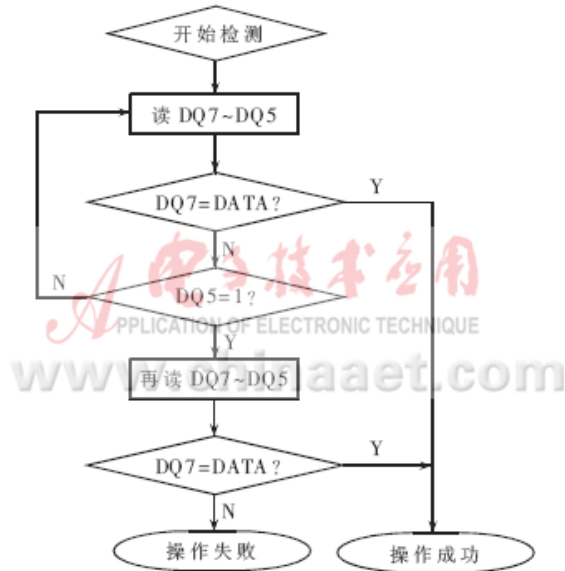


图 1 使用数据线的 DQ7、DQ5 检测时的流程图

程序5

```

JUDGE .macro par,pdr
;检测宏程序。par是编程地址寄存器,
;pdr是存放编程数据的寄存器
_JUDGEBEGIN?: PSHM AR1
LD pdr,B; 获取写入的数据
AND #00FFh,B ;取被写入数据的D07~D00
LD par,A; 读被烧写地址的数据
ADN #00FFh, A ;取DQ7~D00
STL A, TEMP; 保存
LD A, -7,A ;读DQ7状态
XOR B, -7,A ;是否是反码?
BC_JUDGESUCCESS?,AEQ
;DQ7不是反码而是写入数据表示操作成功
BITF TEMP, #20h
BC_JUDGEBEGIN?芽,ntc
;DQ5=1表示操作超时
LD par,A ;再读被烧写地址的数据
AND #00FFh,A
LD A, -7,A
XOR B, -7,A
BC_JUDGESUCCESS?. AEQ
;DQ7不是写入数据,表示操作失败
_JUDGEERRO?
_RESETFI ash ;复位FLASH
_JUDGESUCCESS?
POPM AR1
.endm
  
```

2 C5410 的自举引导

脱离仿真器独立运行程序一般有两种方式:一种是上电后用户程序直接在FLASH存储器中运行,这种方式运行速度比较慢;另一种是上电或复位后将用户程序从FLASH存储器引导到高速数据存储器中运行,此方法最常用,可以较低的成本实现高速的运行。为了实现这个过程就必须运用DSP的自举引导功能。

2.1 自举引导^[1~3]

C5410上电复位后,首先检查MP/MC状态:若为高电平,说明DSP处于微处理器工作方式,即从外部程序存储器0FF80H地址处开始执行用户程序;若为低电平,说明DSP被设置为微计算机工作方式,从片内ROM的0FF80H地址处开始执行程序。0FF80H地址存放的是中断向量表,它实为一条分支转移指令(BD 0F800H),使程序跳转至0F800H执行自举引导程序(Boot loader)。Boot loader是固化在DSP芯片内ROM中的一段代码程序,其功能是将用户程序从外部加载至片内RAM或扩展的RAM中,使其高速运行。在搬运程序之前,Boot loader首先完成初始化工作:使中断无效、内部RAM映射到程序/数据区(OVLY=1)、对程序和数据区均设置7个等待状态等。C5410有以下几种自举引导方式^[4~5]:主机接口HPI、并行口(8/16位)、标准串行口(MCBSP0是16位引导模式,MCBSP2是8位引导模式)以及I/O口(8/16位)自举引导方式。

2.2 并行自举引导

这种方式是比较常用的一种方式,外部存储器的字宽为8位或16位。在自举引导时,通过外部并行接口总线将这些代码从数据存储空间传送到程序存储空间,而且可以重新设置SWWSR及BSCR寄存器的内容。并行自举引导方式首先从地址为0FFFFH的I/O口读取自举表首地址的内容,如果此内容不符合8位或16位的引导方式,就从地址为0FFFFH的数据存储器读取,进行8位或16位并行自举引导。

所以在烧写FLASH数据的同时,也要在0FFFFH烧入自举表的首地址。引导流程如图2所示。

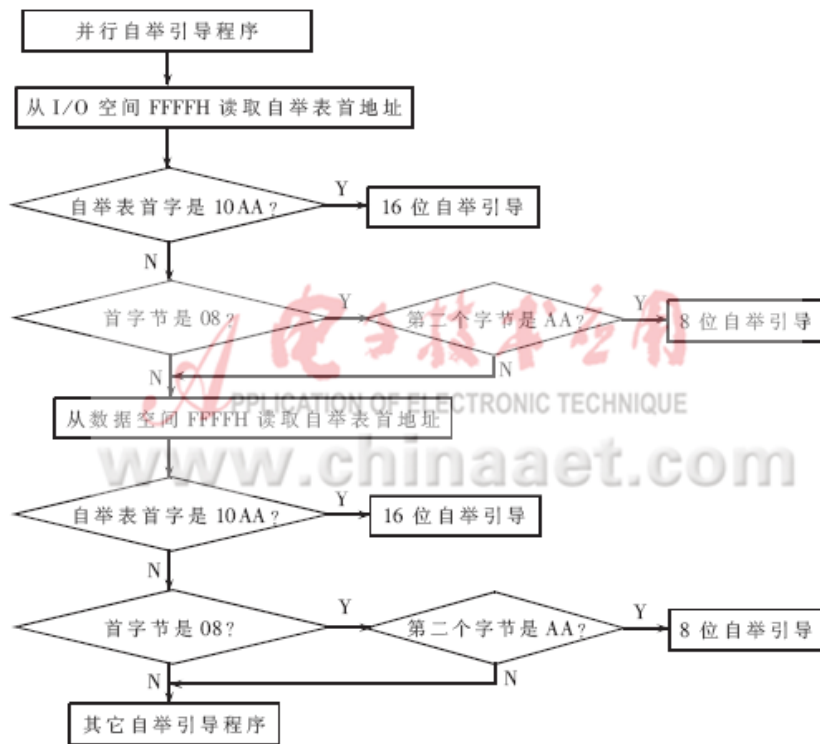


图 2 并行自举引导流程图

2.3 建立自举表

自举表内容不仅包括欲加载的各段代码,而且包括各段代码长度、各代码段存放的目标地址、程序入口地址等信息。若要完成自举引导功能,必须建立正确的自举表。自举表可以由hex500格式转换器自动生成;也可以手动建立自举表,就是把被烧写的程序直接放在烧写程序中,根据被烧写程序的相关信息手动建立自举表。

3 通过C5410烧写FLASH及并行自举引导

下面通过一个FLASH烧写实例,介绍怎样将用户程序烧写进FLASH,以及怎样手动建立自举表,并且脱离仿真器以并行自举引导方式使用用户程序独立运行。被烧写和烧写程序如程序6所示。

程序6

```

.title " FLASH"
.mmregs
SWCR .set 002BH
.set 0060H
.data
.sect " .BOOT"
.label BOOTTABLE ;自举表开始
.word 10AAH ;16位自举标记
.word 7FFFH ;7位等待周期(SWWSR)
.word 0F000H ;块转换寄存器(BSCR)
.word 0000h ;程序入口XPC
.word 0200h ;程序入口地址(MAIN_START)
.word LOADEND - LOADSTART
;程序块长度(0116H)
.word 0000h ;存放目标XPC
.word 0100h ;存放目标地址
LOADSTART: ;中断向量表开始地址
.copy " vector.asm" ;复位处跳转MAIN_START
MAIN_START: ;被烧写的主程序
STM #0F7h, SP
STM #012Ch, PMST
; I PTR=01(中断向量指针为100,指向目标址)
; MP/MC=0, OVLY=1, AVIS=0, DROM=1, CLKOFF=1
LOOPF: RSBX XF ;XF置低
CALL DELAY ;延时
SSBX XF ;XF置高
CALL DELAY
B LOOPF
DELAY: PSHM AR6
  
```

```

STM #0090H, AR6
DELAY_LOOP;
RPT #0FF0h
NOP
BANZ DELAY_LOOP, *AR6-
  POPM AR6
  RET
LOADEND:      ;被烧写的程序结束
.space 20h
.mmregs
.label FINDTABLE
.word 8000h
.text
ERASE_WRITE_FLASH: ;烧写程序开始
STM #OFFA0H, PMST
STM #07FFFH, SWWSR
STM #OFFFFH, SWCR
_RESETFLASH      ;FLASH复位
_ERASEFLASH      ;擦除FLASH
WRITE_FLASHSTART: ;开始编程FLASH
SSBX SXM
RSBX OVM
_RESETFLASH      ;FLASH复位
STM#8000H, ARO   ;FLASH起始地址8000H
STM BOOTTABLE, AR5 ;被烧写的源地址(自举表首地址)
STM #(LOADEND-BOOTTABLE), AR4
;写入011E个字WRITE_RPT
_WRITEFLASH *ARO, *AR5 ;调入编程宏
LD *ARO+, A
LD *AR5+, A      ;完成ARO和AR5地址自动加1
BANZ WRITE_RPT, *AR4-
STM #OFFFFH, ARO ;ARO指向数据空间的FFFF地址
STM FINDTABLE, AR5
_WRITEFLASH *ARO, *AR5
;向数据空间的FFFF地址写入自举表的首地址8000H
ENDD: NOP
B ENDD
.end

```

被烧写的主程序是从MAIN_START开始,一直到LOADEND。程序的主要功能是不断改变XF引脚的状态。LOADSTART是中断向量文件(vector.asm)的开始,在中断复位(RESET)处放入一条跳转到MAIN_START的指令(BD MAIN_START)。ERASE_WRITE_FLASH是烧写程序的开始,只要程序指针(PC)指向ERASE_WRITE_FLASH,开始运行就可以完成对FLASH的烧写操作。从自举表首地址BOOTTABLE(0F8H)到LOADEND(0216H)存放的数据就是要写入FLASH的内容,从LOADSTART到LOADEND的数据是DSP自举程序从FLASH搬运到片内RAM的程序。根据以上信息就可以完成自举表的内容,如表3所列。

表 3 自举表

数据区地址	内容	含义
8000H	10AA	16位自举标记
8001H	7FFF	SWWSR
8002H	F000	BSCR
8003H	0000	程序入口 XPC
8004H	0200	程序入口地址
8005H	0123	程序段长度
8006H	0000	存放目标 XPC
8007H	0100	存放目标地址
8008H	F273	程序代码 1
8009H	0200	程序代码 2
...
811DH	FC00	程序代码
811EH	0000	结束
...
FFFFH	8000	自举表存放首地址

整个并行自举引导过程为: C5410上电复位后,判断MP/MC=0 处于微计算机工作方式,从片内ROM的0FF80H处执行中断向量表的分支转移指令(BD 0F800H),使程序跳转至0F800H处执行自举引导程序。自举引导程序完成初始化后,读取数据空间的0FFFFH地址的内容,找到自举表首地址8000H,从8000H处开始读取内容。首先是16位自举标记(10AA)。然后分别是寄存器SWWSR及BSCR的内容、程序入口地址、代码段长度、存放代码段的目标地址等信息。最后,根据这些信息把FLASH的8008H~811EH的程序搬运到片内RAM的100H开始的地址中,跳转至片内RAM 100H,即PC为100H、XPC为0,开始执行用户程序,完成用户程序的并行自举过程。

把程序烧写入FLASH后,复位C5410,使其处于微计算机工作方式;使用示波器测试XF引脚,观察程序运行正确与否。通过上述方法可完成C5410对Am29LV200B FLASH的烧写,很好地实现C5410上电后的用户程序自举引导功能。

参考文献

- 1 韦江维,廖义奎,农建波等. TMS320VC5402 DSP的并行I/O引导装载方法研究. 电子技术应用, 2000; 15(8): 42~44
- 2 韩峰,侯亚峰,张浩等. 两类DSP芯片的引导过程分析. 电子技术应用, 2002; 17(10): 65~67
- 3 杜志秀. TMS320C542系统的最小硬件设计. 微计算机信息, 1999; 15(6): 62~64
- 4 TMS320VC5410 Fixed-Point Digital Signal Processor Data Manual. Texas Instrument, December. 2000
- 5 TMS320VC5410 Bootloader Application Report. Texas Instrument, April 2000
- 6 Am29LV200B Data Sheet. AMD, April 2002

在线联系

[添加到收藏夹](#)

关于“通过TMS320VC5410烧写FLASH实现并行引导装载的研究”,我有如下需求或意向:

用户名: 密码: 验证码: 5829 欢迎注册

相关应用

- 一种基于FFT的直扩通信系统中窄带干扰信号参数的估计方法
- 用FPGA实现DSP与液晶显示器的快速接口
- 高速ADC的性能测试
- 基于DSP的上网方案的软硬件设计与实现
- 多媒体网关系统中模拟电话语音卡的设计
- 智能天线实验平台研究

[版权声明](#) | [投稿须知](#) | [《电子技术应用》投稿](#) | [网站地图](#) | [帮助中心](#) | [广告中心](#) | [关于我们](#) | [管理员信箱](#)

[回到顶端](#)

《电子技术应用》编辑部版权所有

地址: 北京海淀区清华东路25号电子六所大厦

联系电话: 82306084 / 82306085 传真: 62311179 京ICP备05053646号

推荐分辨率1024*768 IE6.0版本

