

TMS320C30 外部 FLASH 在线烧写及代码搬移的实现

杨东军^{1,2}, 方伟¹, 叶新¹

(1. 中国科学院 长春光学精密机械与物理研究所空间一部, 吉林 长春 130033;

2. 中国科学院研究生院, 北京 100039)

摘要: TMS320C30 内部没有启动自引导机制 (Bootloader), 使得程序上电后从外部 Flash 启动较为复杂; 利用 CPLD 地址译码重映射、Flash 在线编程烧写技术, 解决了贴片式 Flash 程序烧写问题, 实现了中断 LED 程序从外部 Flash 启动; 另外, 鉴于外部 Flash 运行速度慢的缺点, 设计了代码搬移程序, 在 FLASH 启动后, 首先执行搬移程序, 把 FLASH 内的程序代码搬移到外部 RAM 内执行, 实现了代码的高速执行; 实践证明, 该方法简单适用, 运行稳定可靠, 实现了 TMS320C30 上电后的用户程序自动引导, 满足了外部贴片式 Flash 启动和代码高速运行的要求。

关键词: TMS320C30; Flash; 在线烧写; 搬移

Study on On-line Program and Code Remove of TMS320C30's External Flash

Yang Dongjun^{1,2}, Fang Wei¹, Ye Xin¹

(1 Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033,

China; 2 China Graduate School of Chinese Academy of Sciences, Beijing 100039, China)

Abstract: TMS320C30 does not have the bootstrap function, and this makes the program run from Flash very difficultly. Based on the re-coding with CPLD and on-line program of Flash, resolves the problem of the piece package Flash's program, and makes the led interrupt project run in the Flash successfully. On the other hand, for the limit of the Flash speed, designs the code remove program, when the Flash power on runs the code remove program which moves the code from Flash to RAM first, and this improves the code running speed. Experiment shows that this method is simple and stability, and makes the user code bootstrap of TMS320C30 successfully, meeting the requirement of the high speed running from flash when power on.

Key words: TMS320C30; Flash; on-line program; remove

0 引言

作为 TI 的早期浮点处理器 TMS320C30 在工业应用领域已经逐步退出市场, 但 TMS320C30 良好的抗辐射性和宽温度范围 (-55~125℃), 使其成为在航天领域除单片机以外应用最多的处理器, 特别是高达 50MHz 的晶振频率, 适合对工作频率要求较高的航天载荷应用。太阳辐射监测仪作为某卫星的有效载荷, 同时进行太阳跟踪、辐照度测量、转台控制、温度控制, 电路系统复杂, 工作频率要求高, 因此主控制器选用 TMS320C30。但 TMS320C30 的存储空间不足, 需要外扩存储器, 本文选用四片 CY7C1049 和四片 AM29F010 分别作为外部 RAM 和外部 Flash。

用来存放用户程序代码的 Flash 存储器的代码写入最常用的是通过专门 Flash 编程器实现, 但该方法只适用于双列直插

等可插拔封装形式 Flash。随着芯片制造工艺的提高, 使 Flash 存储器正在向小型化、贴片式方向发展。目前贴片式 Flash 烧写多采用将 Flash 芯片安装在用户电路板后进行在线编程^[1]。

程序代码烧写到 Flash 后, 上电直接在 Flash 运行速度慢, 难以发挥 TMS320C30 的速度优势。另外, 由于 TMS320C30 没有系统上电自引导功能, 需要自己设计引导程序, 因此 Flash 启动后的第一件事情就是搬移 Flash 程序内容到 RAM 中运行, 提高系统的运行速度。

本文利用 CPLD 地址译码重映射、程序在线烧写和代码搬移技术, 实现了贴片式 Flash 在线烧写和程序搬移后的高速运行, 实践证明, 该方法简单适用, 运行稳定可靠。

1 实现原理

本系统有定时器 0 中断点灯主程序 led_prj、程序代码搬移程序 move_prj 和 Flash 在线烧写程序 flash_prj 3 个独立程序。系统实现原理框架如图 1, 整个操作步骤如下:

(1) 打开工程文件 move_prj, 编译生成 move_out, 编写引导文件 hex30.cmd, 运行 Hex30.exe 生成 .Hex 格式文件 move.dat, 随后通过格式转换软件 Hex_conv.exe 剔除指示信息后, 生成程序代码文件 move2.dat 并得到代码长度 len1, 同样的步骤生成 led_prj 工程的程序代码文件 led2.dat, 得到相应的代码长度 len2。

收稿日期: 2011-01-30; 修回日期: 2011-03-10。

基金项目: 国家自然科学基金 (61077080); 吉林省青年科研基金 (20090117) 资助项目。

作者简介: 杨东军 (1979-), 男, 山东莱芜人, 在读博士, 主要从事微弱信号监测及太阳辐射计量等方向的研究。

方伟 (1965-), 女, 研究员, 博导, 主要从事空间辐射计量方向的研究。

(2) 打开工程文件 flash.prj, 编译生成 flash.out 文件, 对其进行 load program 加载到外部 RAM 中, 通过编译环境内 File- data- load 加载 move2.dat 到 0x60000 开始的外部 RAM 内, flash.prj 程序读取该部分的内容到数组 RData[i] 内, 运行扇区擦除命令, 擦除 Flash 第一扇区 (0x60000 开始区间), 并把 RData[i] 写入该扇区。同样的方法把 led2.dat 写入第二个扇区 (0x600400 开始区间)。

(3) 通过外部开关切换触发 CPLD 重译码, 使得外部 RAM 和 Flash 映射互换 (Flash 对应存储区由 0x60000 变成 0x0), 系统重新上电, 此时系统从 0x0 运行 (即 Flash), 此时首先执行 move.prj 搬运程序, 把 Flash 第二扇区的内容搬运到外部 RAM (0x60000), 在搬运完成后, move.prj 最后一条程序触发重译码语句, 触发 CPLD 再次重译码, 使 RAM 和 Flash 映射互换 (RAM 存储区间变为 0x0), 复位后, 运行外部 RAM 内 led2.prj 主程序。

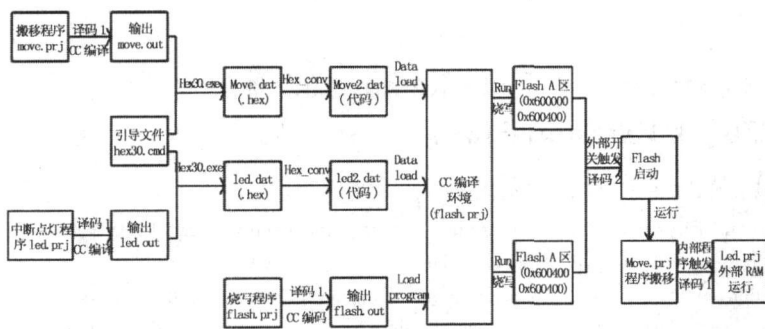


图1 原理框图

2 中断矢量表的映射

TMS320C30 的编译器的程序代码存放位置需要用户自定义, 否则编译结果无法确定程序运行的位置。因此需要通过 .cmd 链接文件进行 DSP 代码的定位。TMS320C30 程序正常启动最重要的复位等中断矢量在 .cmd 文件中的定义如下:

```
MEMORY { vect: org = 0x000000 len = 0x40 ... }
SECTIONS { "vectors" > vect ... }
```

TMS320C30 具有 MP (Microprocessor Mode) 和 MC (Microcomputer) 两种存储器映射方式, 两种方式下中断向量的位置不同。本文 TMS320C30 采用微处理器 (MP) 工作模式, 0x0-0x40 区间主要用于存放中断矢量表和 trap 中断表, 在程序运行过程中, 首先需要初始化中断向量表, 因此该区间内容不能被覆盖。该区间内容的正确映射是程序从外部 Flash 启动的关键, 通过 vectors.asm 汇编文件建立中断矢量表, 具体如下:

```
.sect "vectors"
.word _c_int00 .word _c_int99
.word _c_int01 .word _c_int99
.word _c_int02 .space 20
.word _c_int03 .loop 28
.loop 5 .word _c_int99
.word _c_int99 .endloop
.end loop .space 4
.word _c_int09 .end
```

其中, c_int00 存放复位中断, 对应入口程序, 不需要单

独的中断子程序, 在 rts30.lib 库内 boot.asm 中已经做了复位处理。_c_int99 是空中断, 即中断处理子程序内容为空。_c_int09 对应中断 9 (定时器 0 中断), 其中断子程序内容为定时点灯程序。space 表示保留位, vectors.asm 程序长度刚好为 0x40, 保证了中断矢量表的正确映射。

3 CPLD 重映射

CY7C1049 地址线 A[0...16], 片选 RAMCE, Am29F010 地址线 A[0...18], 片选 FLASHCE。TMS320C30 共 24bit 地址线, 其中 A23、A22 和 A21 连接到 CPLD 进行译码。由于 Flash 在线烧写时程序在外部 RAM 运行, 此时 CPLD 运行译码 1, 当 Flash 程序烧写完成后, 系统从外部 Flash 启动, 此时 CPLD 运行译码 2, CPLD 译码运行流程如图 2 所示。

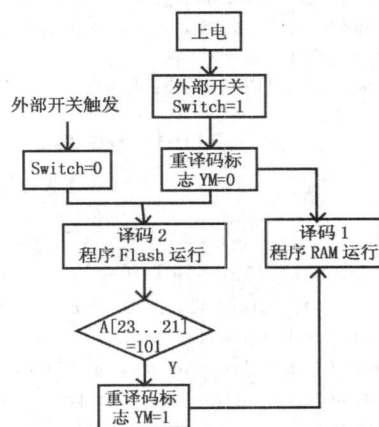


图2 CPLD 译码运行流程

其中, 译码 1 对应: 当 A[23...21] = 000, 片选信号 RAMCE = 0, 存储区间为 0x000000-0x7FFFFF; 当 A[23...21] = 011, 片选信号 FLASHCE = 0, 存储区间为 0x600000-0x61FFFFF。译码 2 对应: 当 A[23...21] = 011, 片选信号 RAMCE = 0, 存储区间为 0x000000-0x7FFFFF; 当 A[23...21] = 000, 片选信号 FLASHCE = 0, 存储区间为 0x600000-0x61FFFFF。两种译码方式通过外部开关触发, 通过判断外部开关的电平来判断采用哪种译码方式。

4 链接文件 (Hex30.cmd) 及格式转换

Code Composer 集成开发环境中的汇编器和链接器创建的目标文件格式为公共目标文件格式, 简称 COFF (Common Object File Format)。Flash 并不接受 COFF 目标文件格式, 因此必须将 COFF 目标文件转换后 (通常为 .HEX 文件) 才能将其写入 EPROM 和 Flash 中。

Code Composer 集成开发环境中提供了一个 HEX 格式转换工具 HEX30.EXE, 用以将 .out 格式的可执行文件转换为具有特定格式的十六进制文件 (.hex), 并在文件中加入了 TMS320C30 引导程序所需的信息字节。HEX30.CMD 文件的内容和含义为:

```
led.out; HEX30.exe 以文件 led.out 为输入;
- map led.mxp; 产生一个印象文件 led.mxp;
- i; 指定输出格式为 INTEL 格式;
- romwidth 32; 指定 ROM 宽度为 32 位;
```

- memwidth 32; 指定系统存储字长为 32 位;
- e 0x 000042; 指定程序入口地址;
- boot; 将各段转换到引导文件中;
- bootorg 0x000000; 指定 Flash 起始映射地址

通过执行上述命令后, 生成了 INTELHEX 格式的 . HEX 文件 led . dat 和 move . dat. IntelHEX 文件是记录文本行的 ASCII 文本文件, 在 IntelHEX 文件中, 每一行是一个 HEX 记录 (由十六进制数组成的机器码或者数据常量), 每条记录占用一行, 包含了 9 个字符前缀 (分别定义了记录的起始符、字节数、加载地址和记录类型) 和 2 个字节后缀 (定义了累加和)^[2]。

通常表贴型 Flash 芯片在线烧写需要进一步将 . HEX 文件转换成二进制。但在此不需要进行此步骤工作, 可以直接烧写 32 位数据宽度内容。但是 . HEX 文件中只有数据字节部分是需要烧写到 Flash 内的程序代码, 因此需要从 . HEX 文件中提取该部分内容, 基于 VC 编写了代码提取软件 HEX _ conv . exe, 主要完成记录类型为“00”的各行内数据段的提取、有效 32 位数据的个数、并以 . dat 文件形式输出数据 (led2 . dat)^[3]。

TMS320C30 内存 0x - 0x11 分别存放 reset、INT0、INT1、INT2、INT3、XINT0、RINT0、XINT1、RINT1、TINT0、TINT1、DINT 共 12 个中断矢量, 在 TMS320C30 上电时 reset 只有在 0x0 位置复位时, DSP 才能正常启动, 另外, 要想外部 Flash 启动时能响应中断, 必须保证 0x1 - 0x11 对应的中断矢量映射正确。

led . out 加载到 RAM 后的反汇编内容, 指示出了各中断的入口地址和程序代码, 通过比较 led2 . dat 打开后的内容, 可以看出中断 0 - 11 的矢量地址与在 RAM 运行时的相同, 因此可以验证中断矢量表的映射是正确的。

5 Flash 在线烧写程序及代码搬移

AM29F010 有复位、读写、片擦除、扇区擦除等操作命令, 由于 Flash 需要进行 led . prj 和 move . prj 两个程序的烧写, Flash 需要分扇区擦除和烧写, 命令时序如表 1, 其中 SA 为扇区起始地址, PA 为要编程地址, PD 为要写入的数据^[4]。

表 1 Flash 操作时序

	扇区擦除	字节编程
周期 1	5555h (AAAAAAAAAh)	5555h (AAAAAAAAAh)
周期 2	2AAAh (55555555h)	2AAAh (55555555h)
周期 3	5555h 80808080h	5555h 0A0A0A0Ah
周期 4	5555h AAAAAAAAh	PA PD
周期 5	2AAAh 55555555h	
周期 6	SA 30303030h	

由于只有擦除指令能使“0”变为“1”, 而编程指令不能使“0”写为“1”, 所以在进行 Flash 存储器烧写时, 应先擦除再进行编程。另外, 在线 Flash 烧写的过程中, 需要了解 Flash 存储器的各状态位, 以监控烧写或擦除是否成功。Flash 提供了数据线的 DQ2 (跳变位)、DQ5 (超时标志位)、DQ6 (跳变位)、DQ7 (数据查询位) 和管脚 RY/BY 等状态位。常用数据轮询 (data - polling) 和检测跳变位两种方法, 本文采

用相对简单的检测跳变位方法, 在对 Flash 存储器进行编程或擦除时, 对任何地址进行连续的读将会引起 DQ6 不断跳变, 而一旦操作结束就停止跳变, 具体代码如下:

```
void WriteOrEraseIsOver() {
    unsigned int LastDQ6, CurrentDQ6;
    LastDQ6 = ( * ( volatile unsigned int * ) FLASH _ BASE _ ADDR) & 0x40404040;
    CurrentDQ6= ( * ( volatile unsigned int * ) FLASH _ BASE _ ADDR) & 0x40404040;
    while( 1) {
        LastDQ6= CurrentDQ6;
        CurrentDQ6= ( * ( volatile unsigned int * ) FLASH _ BASE _ ADDR) & 0x40404040;
        if( LastDQ6= CurrentDQ6)
            break; } }
```

在线烧写的思想: 首先判定需烧写程序是 led . prj (块选择标示 Sel= 1) 还是 move . prj (块选择标示 Sel= 0), 然后将转换后的数据 led2 . dat/ move2 . dat 通过编译环境内 File - data - load 加载到 DSP 的外部 RAM (起始地址 060000), 为确保烧写程序正常运行, 不能将数据加载到烧写程序占据的 RAM 空间, 随后通过 flash . prj 内的语句提取 32 位数据 (长度 len 为 Hex _ conv . exe 显示的字节数), 提取方法为:

```
for(i= 0; i< len; i+ +) {
    RData[i]= * ( unsigned int * ) (0x60000+ i); }
```

最后分别执行块擦除和字节写入命令将 led . prj 和 move . prj 程序烧写到外部 Flash, 具体烧写流程如图 3 所示。

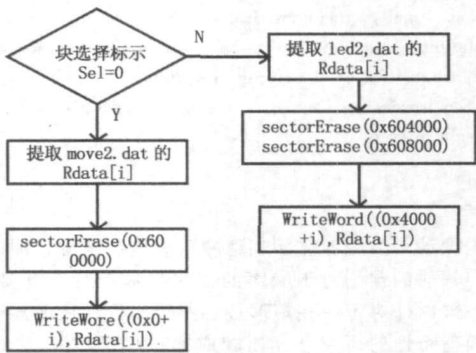


图 3 在线烧写流程

若要查看烧写的情况, 可打开汇编窗口 (菜单 -> windows -> assembly) 对 Flash 目标地址中的内容与 led2 . dat 源文件中的内容进行比对, 看其是否一致。

在 move . prj 程序烧写到 Flash 的 0x0 - 0x600000 区间后, 通过外部开关触发使 CPLD 切换到译码 2, move . prj 程序首先执行搬运操作, 把 0x004000 开始的固定长度内容搬运到 0x600000 外部 RAM 区间内, 具体程序搬移代码如下:

```
unsigned int * srcAddr = ( unsigned int * ) 0x0004000;
unsigned int * desAddr = ( unsigned int * ) 0x600000;
for(i= 0; i< 0x200; i+ +) {
    * desAddr+ + = * srcAddr+ + ;
    IOF= 0x66; IOF= 0x22 } // 搬移过程方波指示
* ( volatile unsigned int * ) 0xa00000= 0xffff; // 程序触发重译码
```

在程序烧写和搬移完成后, 重新上电, 示波器监测 IO 口

(下转 2233 页)

辐射噪声, 轴承设置采用轴输出端游动, 另一端固定方式^[9]。系统的激励频率取系统在 1440 r/min 的稳定转速下, 齿轮传动系统的啮合冲击频率。

在对减速器系统进行谐响应分析后, 由于轴的影响较小, 故在此主要提取了箱体 5 个外表面以及高、低速齿轮表面单元的参数值, 提取出表面各单元的面积 ΔA_m , 单元的总数 N , 各单元中心点坐标 (x, y, z) , 以及单元各节点位移幅值 w_m , 然后利用离散瑞利积分公式计算得到减速器系统各外表面附近的声强值, 结果如图 7 所示。从图中可得到下面结论。

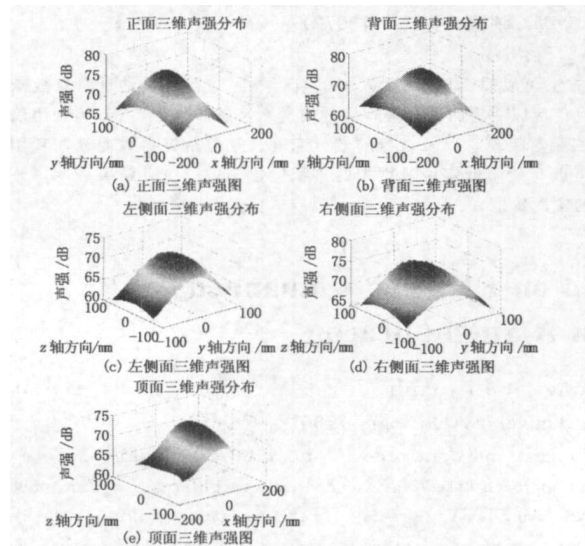


图 7 齿轮箱声强三维分布图

从图 7 中的 a 和图 b 可以看出, 减速器正面 (高速轴输入端) 和背面 (低速轴输出端) 的声强值在 x 和 y 轴方向都是先增大后减小, 且在分别在高速、低速轴承处达到峰值, 这是由于齿轮啮合所产生的轴向力正是通过“压紧”轴承传递给箱体, 并使箱体在“压紧”轴承处的振动也较大。从图 7 中的 c 和图 d 可以看出, 减速器左、右侧面的声强值在 z 轴方向变化平缓, 在 y 轴方向先增大后减小, 这是由齿轮啮合所产生的水平径向力的位置和作用所致。从图 7 中的 e 图可见, 减速器顶面声强在 x 轴方向先增大后减小, 且在 x 轴正方向高速齿轮轴线处达到峰值, 这是由于齿轮啮合所产生的垂直径向力的作用和高速齿轮处加强筋较弱所致。将该结果与文献 [10] 中实验数据相比较, 可见二者吻合较好, 可以说明利用有限元

分析和声场理论相结合的方法对齿轮减速器系统进行研究, 其结果能够与实际减速器系统的辐射的声场情况相符合。

3 结论

(1) 通过有限元软件对减速器系统进行了动力学响应与特性分析, 得到了包括传动系统 (齿轮、传动轴) 和结构系统 (主要是箱体) 以及轴承在内的整个系统的固有频率和模态振型。

(2) 在减速器系统动力学特性分析的基础上, 通过提取各结构表面单元参量, 然后结合声场理论的离散瑞利积分法, 并考虑各壁板隔声量的影响, 通过声场叠加原理计算得到减速器外声场各点处所辐射的声压与声强值, 并将此数值仿真结果与已有的实际测量数据进行比较, 验证了用仿真软件和声学理论相结合的方法来进行噪声预估的正确性, 这也为减速器的噪声控制提供了一种方法。

参考文献:

- [1] 石焱文, 尚志远, 田 华. 基于有限元方法对人工径向裂纹薄圆板振动辐射声场的研究 [J]. 应用力学学报, 2003, 20 (4): 6-9.
- [2] PARK J, MONGEAU L, SIEGMUND T. Influence of support properties on the sound radiated from the vibrations of rectangular plates [J]. J. Sound Vib, 2003, 264 (): 775-794.
- [3] TAO J S, LIU G R, LAM K Y. Sound radiation of a thin infinite plate in light and heavy fluids subject to multi-point excitation [J]. Applied Acoustics, 2001, 62: 573-587.
- [4] 赵志高, 黄其柏, 何 铿. 基于有限元边界元方法的薄板声辐射分析 [J]. 噪声与振动控制, 2008, 2 (1): 39-43.
- [5] Chen Chao, Xia Jianghai, Liu, Jiangping. Nonlinear inversion of potential-field data using a hybrid-encoding genetic algorithm [J]. Computers and Geosciences, 2006, 32 (2): 230-239.
- [6] 姚昊萍, 张建润, 陈 南, 等. 封闭弹性长方体与柱状体连接结构的辐射声场分析 [J]. 振动与冲击, 2007, 26 (8): 125-130.
- [7] 何 琳, 朱海潮, 邱小军, 等. 声学理论与工程应用 [M]. 北京: 科学出版社, 2006.
- [8] 陶泽光, 李润方, 林腾蛟. 齿轮系统有限元模态分析 [J]. 机械设计与研究, 2000, 16 (3): 63-65.
- [9] 姚昊萍, 张建润, 陈 南, 等. 考虑边界条件的弹性长方体封闭结构腔辐射声场分析 [J]. 机械工程学报, 2007, 43 (4): 163-167.
- [10] 李润方, 林腾蛟, 陶泽光. 齿轮箱振动和噪声试验研究 [J]. 机械设计与研究, 2003, 19 (5): 63-65.

(上接 2229 页)

信号 IOF 的波形, 首先显示表示程序正在搬运的周期 $25\mu s$ 方波, 1s 以后经过 MAX692 看门狗复位后, 出现周期 500ms 的定时器方波, 此时程序开始从 RAM 内运行, 结果表明 Flash 启动及程序搬移成功。

6 结论

通过 CPLD 的译码, . HEX 文件格式转换, Flash 在线烧写和程序搬移等方法的有效结合, 解决了贴片式 Flash 的烧写和高速运行问题, 实现了中断 LED 程序的外部 Flash 启动及高速运行。实践证明, 该方法简单适用, 运行稳定可靠, 实现了 TMS320C30 上电后的用户程序自动引导, 克服了专用

Flash 编程器的缺点, 使用灵活, 能满足不同型号 Flash 的烧写。

参考文献:

- [1] 李爱武, 等. FLASH 存储器在 TMS320C3x 系统中的应用 [J]. 新器件新技术, 2003, (6): 43-45.
- [2] 周 华. 基于 C3X/C4X 编译环境的大容量可执行二进制文件转换 [J]. 中国高新技术企业, 2010, 13: 19-20.
- [3] 杨东军, 方 伟, 叶 新, 等. 太阳辐射监测仪遥测信号采集系统设计 [J]. 计算机测量与控制, 2009, 17 (9): 1851-1853.
- [4] 黄天戊, 等. TM S320C32 与 AM29F010 的接口设计与编程技术 [J]. 计算机工程, 2003, (6): 150-152.