

文章编号:1001-4373(2008)01-0138-03

## TMS320F2812 外扩 RAM 的 .cmd 文件配置

傅雪军

(兰州交通大学 光电技术与智能控制教育部重点实验室, 甘肃 兰州 730070)

**摘 要:**针对 TMS320F2812 在外扩 RAM 的情况下,描述了其接线方法并给出了原理图,并从其存储空间来理解 cmd 文件,在利用外扩 RAM 时(在 non-DSP/BIOS 工程中),举例说明了 cmd 文件在 flash 上的分配方法。

**关键词:** TMS320F2812; RAM; 文件配置; 存储空间

中图分类号: TP333

文献标识码:A

TMS320F2812 是目前 Ti 公司中应用最广的一种芯片, 以其性价比高占有市场很大份额. 在现今的一些比较大型的项目中, 往往涉及到比较复杂的数字信号方面的运算 (比如做一些大型数组的运算), 而 F2812 的内部 RAM 往往不能很流畅的运行复杂的运算, 因此需要外扩 RAM 来加快运行速度. 而外扩 RAM 必然涉及到. cmd 文件的配置, 如果编写不当, 将使开发不能进行. 本文以 TMS320F2812 芯片结合具体的实例, 研究了程序被烧写到 flash 中运行 (程序中有大型数组) 的. cmd 文件的配置方法.

### 1 接线方法及原理图

### 1.1 F2812 外部扩展接口的访问

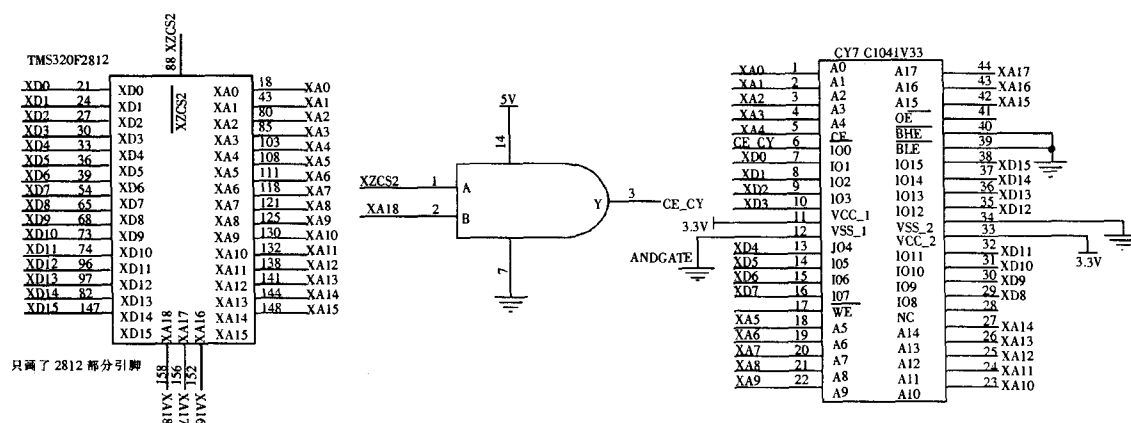


图 1 F2812 外扩 RAM 原理图

**Fig. 1 Schematic diagram of extern RAM of F2812**

收稿日期:2007-10-12

作者简介:傅雪军(1984-),男,江西吉安人,硕士生。

## 2 存储空间分配

TMS320F2812 的 DSP 存储器分为 3 个独立选择的空间—程序空间、数据空间、I/O 空间。表 1 是 F2812 的 flash 存储空间分布<sup>[2]</sup>。

表 1 TMS320F2812 高地址程序空间

Tab. 1 Program spaces of high address in TMS320F2812

寻址空间	程序空间
0x3D7800-0x3D7BFF	OTP 存储器
0x3D8000-0x3F5FFF	FlashB-FlashJ
0x3F6000-0x3F7F7F	FlashA
0x3F7FF6-0x3F7FF7	用于 ROM 入口
0x3F7FF8-0x3F7FFF	用于安全密码

## 3 .cmd 文件在 Flash 上的分配

在 non-DSP/BIOS 应用中,命令文件中使用两个十分有用的伪指令 MEMORY 和 SECTIONS,来指定应用中的存储器结构和进行地址的映射<sup>[3]</sup>。任何用户创建初始化段必须链接到 Flash 中运行(这些段被创建应该用 CODE\_SECTION compiler pragma),而其它被创建的未初始化段如果要链接到 RAM 中运行(应该用 DATA\_SECTION compiler pragma)<sup>[4]</sup>。表 2 描述了当程序烧写到 flash 中运行时,哪些节名应链接到哪。

表 2 Non-DSP/BIOS 工程下的节名链接

Tab. 2 Section linking in the project of Non-DSP/BIOS

节名	链接地方
.cinit	Flash
.cio	RAM
.const	Flash
.econst	Flash
.pinit	Flash
.switch	Flash
.text	Flash
.bss	RAM
.ebss	RAM
.stack	Lower 64Kw RAM
.sysmem	RAM
.esysmem	RAM
.reset	ROM

## 4 实例应用

以下是仿真调试模数转换时的 .cmd 文件的分配(程序烧写到 flash 中运行),在模数转换中已经取得很好的运行。

在分配 .cmd 文件之前,首先得保证程序从

flash 中运行,所以首先得初始化 flash 控制寄存器,在主函数前面需要添加如下程序:

```
#pragma CODE_SECTION (initflash, "RamFuncs");
void initflash(void) //Ti 网站上可以找到
{ ... }
```

在此模数转换中,有一些大数组,如果在内部运行那几乎是不可能完成的事,所以必须到外部 RAM 中去运行,大数组在主函数前必须这样申明,例如:

```
#pragma DATA_SECTION(table, "ramconsts");
float table[4096] = {0}; //申明成全局变量
#pragma CODE_SECTION (initflash, "ramconsts"); //这里 initflash 在前面已经定义
```

另外在主程序里还要对前面这两个做如下说明:

```
main()
```

{//当然这里还有其它的程序这里只写和 .cmd 分配有关的程序

```
initflash();
memcpy(&RamfuncsRunStart,
&RamfuncsLoadStart,
&RamfuncsLoadEnd-
&RamfuncsLoadStart);
memcpy(&ramconstsRunStart,
&ramconstsLoadStart,
&ramconstsLoadEnd-&ramconstsLoadStart);
} //这里象这些 RamfuncsRunStart 等都得在主程序前申明成全局变量
```

### 4.1 包涵外扩 RAM 的 .cmd 文件分配

```
MEMORY
```

```
{
```

```
PAGE 0: /* 程序存储器 */
```

```
ZONE2:origin=0x080000,length=0x080000
```

```
/* 只用到了 ZONE2 */
```

```
OTP:origin=0x3D7800,length=0x000800
```

```
FLASH_ABCDEFGHIJ:origin=0x3D8000,length=0x1FF80
```

```
BEGIN:origin=0x3F7FF6,length=0x000002
```

```
PASSWDS:origin=0x3F7FF8,length=0x000008
```

```
RESET:origin=0x3FFFC0,length=0x000002
```

```
VECTORS:origin=0x3FFFC2,length=0x00003E
```

```
PAGE 1: /* 数据存储器 */
```

```
RAMM0M1:origin=0x000000,length=0x000800
```

```
RAML0L1:origin=0x008000,length=0x02000
```

```
RAMH0:origin=0x3F8000,length=0x002000
```

```
/* 外设 Frame 0 和外设 Frame 1: */
```

```
(这部分省略,因为这部分基本保持不变)
```

```
}
```

```
SECTIONS
```

```

{
/* 分配程序空间: */
codestart;>BEGIN PAGE=0 /* codestart 是代码
开始程序 */
Ramfuncs;
LOAD=FLASH_ABCDEFGHIJ, PAGE=0
RUN=RAML0L1, PAGE=1 LOAD_START(_Ram-
funcsLoadStart),
LOAD_END(_RamfuncsLoadEnd),
RUN_START(_RamfuncsRunStart)
ramconsts;
LOAD=FLASH_ABCDEFGHIJ, RUN = ZONE2 ,
LOAD_START(_ramconstsLoadStart),
LOAD_END(_ramconstsLoadEnd),
RUN_START(_ramconstsRunStart), PAGE = 0
.reset;>RESET PAGE = 0
vectors;>VECTORS PAGE = 0
.cinit;>FLASH_ABCDEFGHIJ PAGE = 0
(.text .pinit .const .econst .switch 和 .cinit 可以一样)
/* Allocate data areas: */
.stack;>RAMM0M1 PAGE = 1
(.bss .ebss .system .esystem .cio 和 .stack 可以一样)
(下面这部分省略)
/* 分配外设 Frame 0 的寄存器结构: */
/* 分配外设 Frame 1 的寄存器结构: */
/* CSM 密码放置: */
}

```

#### 4.2 查看段的分配

程序调试编译后,通过 map 文件可以查看各段的分配情况,包括段的起始地址,使用的字节数等<sup>[5]</sup>. 可以通过这个适当调整上面的. cmd 文件的分配.

## 5 结论

如果对芯片存储器分配图理解的不清楚的情况下,将会导致. cmd 文件配置不好,从而可能会使后面的调试工作无法进行. 在实际过程中. cmd 文件的配置很灵活,要根据实际程序划分好存储空间,使之空间配置达到最优.

## 参考文献:

- [1] 苏奎峰. TMSD320F2812 原理与开发[M]. 北京:电子工业出版社,2004.
- [2] TMS320F2812 Data Manual ,2006[M/OL]. [2007-07-01]. <http://focus.ti.com/lit/ds/sgus051a/sgus051a.pdf>.
- [3] Running an Application from Internal Flash Memory on the TMS320F28xx DSP, 2006[M/OL]. [2007-07-08]. <http://focus.ti.com/general/docs/techdocsabstract.tsp?abstractName=spra958f>.
- [4] 支长义. DSP 原理及开发应用[M]. 北京:北京航空航天大学出版社,2005.
- [5] 万山明. TMS320F281x DSP 原理及应用实例[M]. 北京:北京航空航天大学出版社,2007.

## The Allocating Technique of the .cmd File of Extern RAM of TMS320F2812

FU Xue-jun

(Key Laboratory of Opto-electronic Technology and Intelligent Control, Ministry of Education,  
Lanzhou Jiaotong University, Lanzhou 730070, China)

**Abstract:** In view of the condition of extern RAM, the wiring method of TMS320F2812 is elaborated and its schematic diagram is presented. Then the .cmd file is studied from storage space. Utilizing extern RAM (in the project of non-DSP/BISO), the example is given to illustrate the allocating technique of the .cmd file in the flash.

**Key words:** TMS320F2812; RAM; file configuration; storage space