

南京航空航天大学

第1页 (共7页)

二〇一八~二〇一九 学年 第1学期 《DSP 实用技术》 考试试题

考试日期: 2018 年 12 月 24 日 试卷类型: A

试卷代号: 030052

题号	一	二	三	四	五	六	七	八	九	十	总分
得分	43	53									96

本题分数	46
得分	43

一、分析题意, 回答问题 (共46分)。

1. 说明TI公司目前应用较广的3个DSP系列是什么, 并说明在电机控制领域一般采用哪个系列。(4分)

【答】 TMS320C2000, TMS320C5000, TMS320C6000

电机控制领域 - 一般采用 TMS320C2000,

2. F28335 DSP控制器的片内存储器总线总共有6组 (PAB、PRAB、PWAB、PRDB、DRDB和DWDB)。PAB和DWAB这两条总线分别为多少位? 在流水线F2阶段, CPU从哪条总线上将指令取出, 放入指令队列? 在流水线R1阶段, CPU将操作数地址送到哪条总线上? (4分)

【答】 PAB: 12位 DWAB: 32位

F2阶段, CPU从程序计数器总线 PRDB 将指令取出

R1阶段, CPU将操作数地址送到程序计数器总线 PRDB, PRAB

3. 说明C28x的执行单元中, 输入定标单元和乘法单元的作用, 并说明对乘积的移位和执行整数逻辑运算这两种操作是否要占用ALU的时钟开销。(4分)

【答】 输入定标单元: 将来自总线的16位数据转换为32位并送至32位算术逻辑单元ALU。
乘法单元: 完成32位×32位或16位×16位的乘法运算。
算术逻辑运算不会占用ALU的时钟开销, 但要移位在ALU中。

程序空间最大寻址范围 4MW.

4. F28335 DSP控制器的数据空间最大寻址范围为多大? 其片内34K字的SARAM分了多少个1K字的块和多少个^{4K}字的块? 其外部存储器接口XINTF的作用是什么? (4分)

【答】 4GW

两个1K字的块和8个4K字的块。
扩展片外程序或数据存储

5. 说明F28335的时钟源模块中, 振荡器模块和PLL模块的作用。时钟源模块产生的六种时钟(OSCCLK、WDCLK、CLKIN、SYSCLKOUT、HSPCLK和LSPCLK)中, 哪个是片内外设的全局时钟, 机器周期指的是哪个时钟的周期。 (4分)

【答】 振荡器模块, 用于产生外部时钟 OSCCLK.

PLL 模块, 把 OSCCLK 直接分频或倍频后得到 CLK, 作为 CPU 的输入时钟。 (稳定, 可编程)

片内外设的全局时钟是 SYSCLKOUT, 机器周期是 CLKIN 的周期。

6. 说明汇编器、连接器和 C 编译器各自的作用, 以及连接器命令文件的作用。 (4分)

【答】 汇编器, 将程序代码或数据汇编至目标代码中。

连接器, 建立可执行的 COFF 输出模块, 并分配模块连接各数据地址 (定位)。

C 编译器, 将 C 源代码转换为汇编代码并输出。

连接器命令文件 (.CMD): 指定程序数据和运行过程中方便连接。

7. F28335 DSP 控制器片内分别有多少 SCI 接口资源和 SPI 接口资源? 哪种是异步串行通信方式? 哪一种的通信速率更快? (4分)

【答】 3个 SCI 接口资源, 1个 SPI 接口资源

SCI 是异步串行通信方式

SPI 的通信速率更快。

8. 说明直接寻址、间接寻址和堆栈寻址三种寻址方式的最大寻址空间分别为多大, 分别借助于哪个 (或哪些) CPU寄存器实现。(6分)

【答】

直接寻址: 4M. 数据寄存器 DP
 间接寻址: 4G. 辅助寄存器 XAR0 ~ XAR7
 堆栈寻址: 64k. 堆栈指针 SP

9. 说明在CCS中, 如何生成可执行代码 (写出各步骤名称, 及各步骤分别需要操作哪个菜单中哪个选项)。(6分)

【答】

1. 新建工程: 单击 "Project" → "New", 在弹出的 "Project Creation" 中完成工程的创建。

2. 向工程中添加文件: 将 Volume1 中的 Volume.C, Volume.h 等文件复制到工程所在文件夹, 单击 "Project" → "Add Files to Project", 在弹出的对话框中, 将复制过来的文件除 sine.dat 之外的其他文件添加到工程中。

3. 生成可执行代码: 单击 "Project" → "Rebuild All" 对整个工程进行编译、汇编和链接。

10. 若需采用F28335的CPU定时器2的中断进行定时, 从中断允许的角度, 说明应该设置哪些寄存器的哪些功能位, 及如何设置 (设置为0还是1)。(6分)

【答】

中断允许寄存器 IER, 置0 (禁止CPU中断)

中断标志寄存器 IFR, 置0 (清除所有CPU中断标志)

CPU定时器2寄存器中的 TIMER2 TCR

其中, CPU定时器中断允许位 TIE, 置1

(定时器中断允许)

CPU定时器停止位 TSS, 置0

IER, INT14中断允许位, 置1 (CPU级中断允许)

PIE控制寄存器 PIECTRL, PIE向量表允许位 CNP1E, 置1 (PIE级中断允许)

7/11/14 - 2

本题分数	54
得分	53

二、编程和分析题：(共 54 分)

1. 试根据代码后边的注释提示和要求，在下列代码的空白处填写合适的代码或数据，以便完善 C 语言程序段。(每个空 1 分，共 14 分)

ALLOW ;

//宏指令，允许访问受保护的寄存器。

GpioCtrlRegs. GPAMUX1.bit.GPIO0 = 1 ; //配置 GPIO0 为 PWM1A 引脚。

GpioCtrlRegs. GPAMUX1.bit.GPIO1 = 0 ; //配置 GPIO1 为 I/O 引脚。

GpioCtrlRegs. GPADIR.bit.GPIO1 = 1 ; //配置 GPIO1 为输出。

GpioDataRegs. GPADAT.bit.GPIO1 = 1 ; //GPIO1 输出低电平。

GpioCtrlRegs. GPAMUX1.bit.GPIO2 = 0 ; //配置 GPIO2 为 I/O 引脚。

GpioCtrlRegs. GPAMUX1.bit.GPIO2 = 0 ; //配置 GPIO2 为输入。

EDIS ;

//宏指令，禁止访问受保护的寄存器。

2. 如图 2.1 所示，系统频率 SYSCLKOUT=100MHz，F28335 的 ePWM1A 通道产生 PWM 波，其中，在该 ePWM 模块中，分频系数设置为 HSPCLKDIV=0x5，CLKDIV=0x4。(共 24 分)

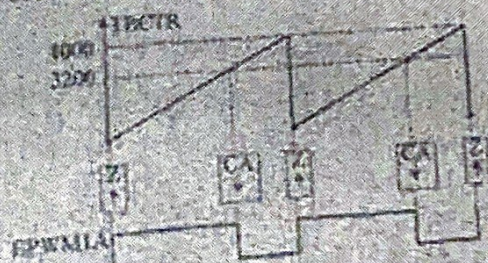


图 2.1 ePWM1A

- (1). 请计算出 TB 模块时钟 TBCLK 的周期。(3 分)

$$f_{TBCLK} = \frac{f_{SYSCLKOUT}}{2^{CLKDIV} \times (2 \times HSPCLKDIV)} = \frac{100M}{2^4 \times 2 \times 5}$$

$$T_{TBCLK} = \frac{1}{f_{TBCLK}} = 1.6\mu s$$


```
EPwm1Regs.TBCTL.bit.HSPCLKDIV = 0x5;
```

```
EPwm1Regs.TBCTL.bit.CLKDIV = 0x4;
```

```
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
```

```
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // 零事件装载
```

```
EPwm1Regs.AQCTLA.bit.CAU = AQ_CLEAR; // 比较匹配清除 (0)
```

```
EPwm1Regs.AQCTLA.bit.ZRO = AQ_SET; // 比较匹配置位 (1)
```

3. 如图 2.2 所示, 利用 F28335 的 ADCINA1 采集一路航空直流电源电压信号 (直流 28.5V)。要求 F28335 的 ADC 模块工作于级联排序器模式, 软件触发, 在中断服务程序中采集数据共 128 次, 存储在 VolA 数组中。请根据要求, 完成下列几个问题。(共 16 分)

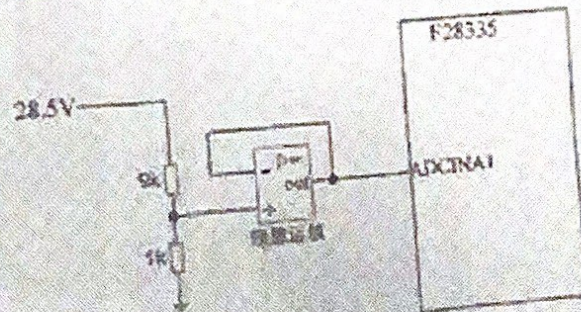


图 2.2 AD 采集航空电源电压信号

- (1) 根据图 2.2, 计算 28.5V 直流电压经信号调理之后, 进行 AD 转换之后的数字量大小 (十六进制表示) (2 分)。

ADCINA1 的数字量: 0xF34

- (2) 根据题目要求和注释, 将以下 ADC 模块的初始化和中断程序补充完整 (每个空 1 分, 共 14 分)。

```
void Adc_setup()
```

```
{AdcRegs.ADCCTRL1.bit.SEG = 0x3C; // 设置 ADC 的排序器模式
```

```
AdcRegs.ADCCTRL1.bit.MODE_SEL = 0; // 设置为顺序采样模式
```

```
AdcRegs.ADCMAXCONV.all = 0; // 设置最大转换次数
```



```

AdcRegs.ADCCHSELSEQ1.bit.CONV00= 1; //设置转换顺序(通道)
AdcRegs.ADCTRL1.bit.CONT_RUN= 0; //工作于启动/停止转换模式
AdcRegs.ADCTRL2.bit.SOC_SEQ1= 1; //软件启动 SEQ
AdcRegs.ADCTRL2.bit.INT_ENA_SEQ1= 1; //允许 AD 中断
ConversionCount=0; //采集的中断计数器次数为 0
}

__interrupt void adc_isr(void)
{
    Vol[A[ConversionCount]]=AdcRegs.ADCRESULT0 <del>为4</del>; //存储数据
    if(ConversionCount==127)
    {
        ConversionCount = 0; //计数器清 0
    }
    else
    {
        ConversionCount++;
    }
    PieCtrlRegs.PIEACK.all= PIEACK_GROUP1; //清除 PIE 应答位, 使 AD 再次响应中断
}

```


南京航空航天大学

第1页 (共8页)

二〇一七~二〇一八 学年 第1学期 《DSP 实用技术》 考试试题

考试日期: 2017年12月4日 试卷类型: B 试卷代号: 030061

班号 0314201 学号 031420102 姓名 尚晓凤

题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

本题分数	40
得分	38

一、分析题意, 回答问题 (共40分)。

1、采用流水线技术有什么好处? F28335 DSP控制器采用几级流水线? 达到流水线哪个阶段之后的指令不能从流水线中清除掉? (4分)

【答】(1) 由于指令执行的每个阶段需要一个机器周期, 故一条指令的执行需要8个周期。但每个周期同时启动8条流水线, 每条流水线处理一条指令。由于这8条指令同一时刻分别使用片内的6条总线, 故不会发生冲突。流水线启动后, 每个周期均有8条指令同时激活, 而且在任一周期均有指令执行, 仿佛一个周期完成一条指令。

(2) 采用8级流水线。

(3) 达到D2阶段的指令不会被从流水线中清除。

2、F28335 DSP控制器具有专门的硬件乘法器, 可以在单周期内完成乘法运算。相关硬件包括哪些? 乘积移位方式由状态寄存器ST0中哪个功能位段确定(写出其名称)? (5分)

【答】包括: 32位的临时寄存器XT、32位的乘积寄存器P、32位的乘积移位器。

ST0中的PM功能位段确定。
乘积移位模式

3、F28335 DSP控制芯片内大部分存储器均同时映射至数据空间和程序空间的同一地址，但是有一种存储资源比较特别，仅仅映射到了数据空间。请问该存储资源叫什么名字？其作用是什么？为什么要对其中的一些存储单元进行写保护？对受保护单元进行写操作前需要使用什么指令解除写保护？写操作完毕需要使用什么指令恢复写保护？（5分）

【答】④ 外设寄存器帧 (PE)

(2) 除了CPU个别寄存器外, 所有片内外设及其中断向量表和Flash存储器的寄存器(包括控制、数据和状态寄存器)均映射在该区域。

(3) 因为实际 CPU 存储器总线上表现为先读后写的顺序, 因而在期望先写后读的特设应用中, 将导致执行顺序错误。

4. 用 "EALLOW" 解除写保护

(5) 用 "EDIS" 恢复写保护。

4、如果由片外30MHz的晶振为F28335 DSP控制器提供时基信号，应如何设置时钟源模块的寄存器（说明需要设置哪些寄存器的哪些功能位段），使F28335的CPU时钟频率初始化为120MHz？（共4分）

【答】应设置 PLLSTS 寄存器的 DIVSEL 位域的分频系数和 PLLCR 寄存器的 DIV 位域的分频系数, 且将 PLLSTS[PLLOF] 设为 0, 使锁相环开启。

可设: $PLLCR[DIV] = 1$, $PLLSTS[DIVSEL] = 1$ } $f_{SYCLKOUT} = f_{CLKIN} = 4f_{JOSCLK}$
或 $PLLCR[DIV] = 1000$, $PLLSTS[DIVSEL] = 2$ } $\underline{\hspace{1cm}} = 120MHz$

5、什么是DSP最小硬件系统？为F28335 DSP控制器设计最小硬件系统时，若系统CPU内核工作频率为150MHz，其内核电源和I/O电源的电压分别应该为多少伏特？（4分）

【答】(1) DSP最小系统是指用尽量少的外围电路构成的可以使DSP正常工作实现基本功能的最简单的系统。一般包括: DSP芯片、电源电路、复位电路、时钟电路和JTAG接口。

(2) 内核电源电压为 $1.9V$

I/O电源电压为3.3V.

本题分数	60
得分	

二、编程和分析题: (共 60 分)

1、说明以下语句分别使用了哪种寻址方式。(4分)

SUB AH, @VAR

直接寻址

SUB AH, T

寄存器寻址

MOVL ACC, *XAR6[AR0]

间接寻址

SUB *SP++, AL

堆栈寻址

2、试在下列空白处填写合适的的数据或者代码, 以便完善C语言程序段。要求: 配置GPIO引脚中的GPIO0和GPIO50配置为普通的I/O引脚, 其中GPIO50作为输入, GPIO0作为输出; 并实时读取GPIO50的输入状态, 从GPIO0输出。(每空1分, 共12分)

unsigned int temp;

EALLOW;

GpioCtrlRegs. GPAMUX1. bit.GPIO0= 0 ;//GPIO0 为普通 I/O 引脚

GpioCtrlRegs. GPBMUX2. bit.GPIO50= 0 ;//GPIO50 为普通 I/O 引脚

GpioCtrlRegs. GPADIR. bit.GPIO0= 1 ;//GPIO0 作输出

GpioCtrlRegs. GPBDIR. bit.GPIO50= 0 ;//GPIO50 作输入

While(1)

{temp= GpioDataRegs. GPBDAT. bit.GPIO50; //读取 GPIO50 引脚输入;

GpioDataRegs. GPADAT. bit.GPIO0=temp; //从 GPIO0 输出

EDIS;

如图1所示, 系统频率SYSCLKOUT=150MHz, F28335的ePWM2B通道产生PWM波, 其中, 在该ePWM模块中, 分频系数HSPCLKDIV=0x6, CLKDIV=0x5。(共20分)

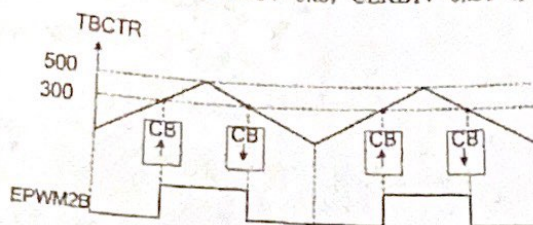


图1 EPWM2B 的波形

- (1) 请计算出 TB 模块时钟 TBCLK 的周期为多少微秒。(3 分)

【答】
$$f_{TBCLK} = \frac{f_{SYSCLKOUT}}{2^{CLKDIV} (2 \times HSPCLKDIV)} = \frac{150MHz}{2^5 (2 \times 6)} = 290625 Hz$$

$$T_{TBCLK} = \frac{1}{f_{TBCLK}} = 2.56 \mu s$$

- (2) 其时基定时器 TBCTR 应工作于哪种计数模式? 系统载波周期为多少个 TBCLK 周期? 并算出具体载波周期为多少毫秒。(4 分)

【答】(1) 工作于连续增/减计数模式

(2) 载波周期为 1000 个 TBCLK 周期

(3) $T_{pwm} = 2 \times 500 \times T_{TBCLK} = 2.56 ms.$

- (3) 产生的 PWM 波的占空比为多少(假设高电平有效)? 若希望增大有效脉冲宽度应调节哪个寄存器的值, 如何调节(增大还是减小)(4 分)

【答】(1) $D = \frac{2 \times (500 - 300)}{2 \times 500} = 0.4$

(2) CMPB, 减小.

(4) 请将以下对 ePWM 模块进行初始化的程序补充完整 (9 分)

```

void initePWM2()
{
    EPwm2Regs.TBPRD = 500 / ; // 设置周期寄存器的值
    EPwm2Regs.CMPB = 200 / ; // 设置 CMPB 寄存器的值
    EPwm2Regs.TBPHS = 0; // Set Phase register to zero
    EPwm2Regs.TBCTR = 0; // clear TB counter
    EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN / ; // 设置计数模式
    EPwm2Regs.TBCTL.bit.PHSEN = TB_DISABLE; // 禁止相位装载
    EPwm2Regs.TBCTL.bit.PRDL = TB_SHADOW;
    EPwm2Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;
    EPwm2Regs.TBCTL.bit.HSPCLKDIV = 0x6 / ;
    EPwm2Regs.TBCTL.bit.CLKDIV = 0x5 / ;
    EPwm2Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
    EPwm2Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // 零事件装载
    EPwm2Regs.AQCTLB.bit. CBU / = AQ_SET / ; // 比较匹配置位
    EPwm2Regs.AQCTLB.bit. CB0 / = AQ_CLEAR / ; // 比较匹配复位
}

```

1. 阅读以下CPU定时器2中断程序，并为带下划线语句加上注释（写在“//”后的横线上）。(8分)

```

#include "DSP2833x_Device.h"
#include "DSP2833x_Examples.h"

interrupt void cpu_timer2_isr(void);

```

(3822

238


```

void main(void)
{
    InitSysCtrl(); //step1 初始化系统控制
    InitGpio(); //step2 初始化 GPIO

    DINT;
    // 禁止 CPU 中断
    InitPieCtrl();
    IER = 0x0000;
    IFR = 0x0000;
    InitPieVectTable();

    EALLOW; // 宏指令, 允许访问受保护寄存器
    PieVectTable.TINT2 = &cpu_timer2_isr;
    // 重新映射使用的中断向量, 使其指向中断服务程序.

    EDIS; // 宏指令, 恢复寄存器的保护状态
    InitCpuTimers();
    ConfigCpuTimer(&CpuTimer2, 150, 1000000);
    CpuTimer2Regs.TCR.bit.TIE = 1;
    // 允许定时器 2 中断
    CpuTimer2.InterruptCount = 0;
    CpuTimer2Regs.TCR.bit.TSS = 0;
    // 设置 TSS 为 0 启动定时器工作
    IER |= M_INT14; // 允许 CPU 的 INT 14 中断, 该中断连接至 TINT2
    EINT; // 清除全局屏蔽位 INTM 以允许屏蔽中断
    ERTM;
    for(;;);
}

interrupt void cpu_timer2_isr(void)
{
    CpuTimer2.InterruptCount++;
}

```


5、要求F28335的ADC模块工作于双排序器、顺序采样模式，使用SEQ2对ADCINA5、ADCINB5始化程序补充完整。(每个空1分，共16分)

```
void Adc_setup()
{
    AdcRegs.ADCTRL1.bit. SEQ_CASC / = 0 / ; //设置 ADC 工作于双排序器模式
    AdcRegs.ADCTRL3.bit. SMODE_SEL / = 0 / ; //设置顺序采样模式
    AdcRegs.ADCMAXCONV.bit. MAX_CONV2 / = 0x1 / ; //设置最大转换状态数
    AdcRegs. ADCHSELSEQ3 / .bit. CONV08 / = 0x5 / ; //设置转换顺序 (通道)
    AdcRegs. ADCHSELSEQ3 / .bit. CONV09 / = 0xD / ; //设置转换顺序 (通道)
    AdcRegs.ADCTRL1.bit. CONT_RUN / = 0x1 / ; //工作于连续转换模式
    AdcRegs.ADCTRL2.bit. SDC_SEQ2 / = 0x1 / ; // 软件启动 SEQ2
}
```