

2019 年 - 2020 学年度第 二 学期 华中科技大学本科生课程考试试卷(A 卷)

课程名称: DSP 原理与应用 课程类别 ☐公共课 ☒专业课 考试形式 ☒开卷 ☐闭卷

所在院系: 人工智能与自动化学院 专业及班级: 自动化、测控 17 级

考试日期: 5 月 28 日

一、简答题 (50 分, 每小题 5 分)

- 1, DSP 器件有哪些主要特征?
- 2, 什么是哈佛结构? TI 的 C2000 采用了怎样的改进哈佛结构?
- 3, 简述流水线技术的作用和 TMS320F2802X 的流水线技术。
- 4, 简述 C2000 DSP 中断结构。
- 5, 简述外设中断的响应过程。
- 6, 简述 TMS320F2802X 的外部中断和 eCAP 的联系与区别。
- 7, 看门狗的作用是什么? 如何避免看门狗计数器溢出?
- 8, 简述 F28027 的 ePWM 模块中 TB、CC、AQ 和 ET 子模块的功能。
- 9, F28027 的 ADC 实现过采样的机制。
- 10, CCS 编译器产生的初始化段和非初始化段各有哪些? 并简述 F280C28x 的 C 语言中的 volatile、interrupt 等标识符的含义。

二、(30 分, 每 0.5 分) 填空题, 按照题目要求或者注释要求填空。本题中 CPU 系统时钟为 60M。注意: 如果是填写寄存器或者是寄存器位的值, 一定要填写具体数值。

- 1, 为 F28027 DSP 配置系统时钟, 要求采用内部晶体振荡器 1:

寄存器 CLKCTL 的相关位取值为:

INTOSC2OFF = (1); OSCCLKSRC2SEL = (2); XCLKINOFF = (3) ;

XTALOSCOFF = (4); OSCCLKSRCSEL = (5); INTOSC1OFF = (6);

寄存器 PLLSTS 的 DIVSEL = 2, 寄存器 PLLCR 的 DIV = (7)。

- 2, 将 CPU 定时器 0 配置成 0.01S 中断一次, 配置完成后定时器 0 运行:

CpuTimer0Regs.PRD.all = (8);

CpuTimer0Regs.TPR.all = 0; CpuTimer0Regs.TPRH.all = (9);

CpuTimer0Regs.TCR.bit.TSS = (10); CpuTimer0Regs.TCR.bit.TRB = (11);

CpuTimer0Regs.TCR.bit.TIF = (12); CpuTimer0Regs.TCR.bit.TIE = (13);

SysCtrlRegs.PCLKCR3.bit.(14) = 1;

3, Flash 和 OPT 的初试化, 要求使能流水线、每次访问使用 4 个 SYSCLKOUT 周期

```
FlashRegs.FOPT.bit.ENPIPE = (15);  
FlashRegs.FBANKWAIT.bit.PAGEWAIT = (16);  
FlashRegs.FBANKWAIT.bit.RANDWAIT = (17);  
FlashRegs.FOTPWAIT.bit.OTPWAIT = (18);
```

4, 完成 EPWM1 的初始化, 要求: PWM 周期 0.1ms, 互补高有效, ePWM1A 和 ePWM1B 的受 CMPA 控制, 不中断, 死区时间 2.5 微秒, 每次 PWM 过零点触发 SOCA:

```
void InitePWM1() {  
    EALLOW;  
    SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = (19);  
    GpioCtrlRegs.GPAMUX1.bit.GPIO0 = (20);  
    GpioCtrlRegs.GPAMUX1.bit.GPIO1 = (21);  
    EDIS;  
    EPwm1Regs.TBPRD = (22);  
    EPwm1Regs.TBCTL.bit.CLKDIV = (23); //不分频  
    EPwm1Regs.TBCTL.bit.CTRMODE = 2;  
    EPwm1Regs.CMPCTL.bit.LOADAMODE = (24); // load on CTR=Zero or PRD  
    EPwm1Regs.CMPCTL.bit.LOADBMODE = (25); // load on CTR=Zero or PRD  
    EPwm1Regs.AQCTLA.bit.CAU = 1;  
    EPwm4Regs.AQCTLA.bit.CAD = 2;  
    EPwm4Regs.DBCTL.bit.OUT_MODE = (26); // enable Dead-band module  
    EPwm4Regs.DBCTL.bit.POLSEL = (27); // Active Hi complementary  
    EPwm4Regs.DBFED = (28);  
    EPwm4Regs.DBRED = (29);  
    EPwm1Regs.ETSEL.bit.SOCAEN = (30);  
    EPwm1Regs.ETSEL.bit.SOCASEL = (31);  
    EPwm1Regs.ETPS.bit.SOCAPRD = (32);  
    EALLOW;  
    SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = (33);  
    EDIS;  
}
```

5, 完成 eCAP1 的初始化, 要求: 可以测量频率范围从 0.05Hz 到 2000Hz 的脉冲的频率。采用 GPIO5 作为 eCAP1 的信号源, 采用差分时间模式。

```
void InitCAP()  
{  
    EALLOW;  
    GpioCtrlRegs.GPAPUD.bit.GPIO5 = (34); // 使能上拉  
    GpioCtrlRegs.GPAMUX1.bit.GPIO5 = (36); // GPIO5 用作 ECAP1  
    EDIS;  
    ECap1Regs.ECEINT.all = (37); // 禁止所有 CAP 中断
```

```

ECap1Regs.ECCLR.all = (38); // 清除所有 CAP 中断标志
ECap1Regs.ECCTL1.all = (39);
ECap1Regs.ECCTL2.bit.CAP_APWM = (40);
ECap1Regs.ECCTL2.bit.CONT_ONESHT = (41);
ECap1Regs.ECCTL2.bit.SYNCO_SEL = 2;
ECap1Regs.ECCTL2.bit.SYNCl_EN = 0;
ECap1Regs.ECCTL2.bit.TSCTRSTOP = (42); // 允许 TSCTR
ECap1Regs.ECEINT.bit.CEVT4 = (43); // 清除 CEVT4
}

```

6, 完成 ADC 的初始化, 要求: 对 ADCINA7 和 ADCINB7 进行 4 倍过采样, 同步采样模式, ePWM1 的 SOCA 作为触发源 (ePWM1 主体在前面已经初始化), 采集窗口时间 ACQPS = 8

```

void InitADC()
{
    EALLOW; SysCtrlRegs.PCLKCR0.bit.ADCENCLK = (44);
    (*Device_cal)();
    AdcRegs.ADCCTL1.all = (45);
    // ADC BandGap、reference、ADC 上电, 使能 ADC, 选内部参考源
    EDIS;
    DELAY_US(ADC_usDELAY);
    EALLOW;
    AdcRegs.ADCCTL1.bit.INTPULSEPOS = 1;
    AdcRegs.INTSEL1N2.bit.INT1E = 1;
    AdcRegs.INTSEL1N2.bit.INT1CONT = 0;
    AdcRegs.INTSEL1N2.bit.INT1SEL = (46); // EOC7 触发 ADCINT1
    AdcRegs.ADCSAMPLEMODE.all = 0xff;
    AdcRegs.ADCSOC0CTL.all = (47);
    AdcRegs.ADCSOC2CTL.all = (48);
    AdcRegs.ADCSOC4CTL.all = (49);
    AdcRegs.ADCSOC6CTL.all = (50);
    EDIS;
    EPwm4Regs.ETSEL.bit.SOCAEN = (51); // Enable SOC on A group
    EPwm4Regs.ETSEL.bit.SOCASEL = (52); // Enable event CTR = ZERO
    EPwm4Regs.ETPS.bit.SOCAPRD = (53); // Generate pulse on 1st event
    EPwm4Regs.ETCLR.bit.SOCA = (54); // Clear SOCA flag
}

```

8, F28027.CMD 文件的编写, 要求把.ebss 放到 M0, .stack 放到 M1, 其他包括复位向量、变量初值等合理安排。

```

MEMORY {
    PAGE 0:  FLASHA  : origin = 0x3F6000, length = 0x001F80
              BEGIN   : origin = 0x3F7FF6, length = 0x000002
              RESET   : origin = 0x3FFFC0, length = 0x000002
              VECTORS  : origin = 0x3FFFC2, length = 0x00003E
    PAGE 1:  RAMM0   : origin = 0x000050, length = 0x0003B0
              RAMM1   : origin = 0x000400, length = 0x000400
              DRAML0   : origin = 0x008800, length = 0x000800
}
SECTIONS
{
    .cinit          :> FLASHA          PAGE = 0
    .text           : (55)
    codestart       : (56)
    .stack          : (57)
    .ebss           : (58)
    .reset          : (59)
    vectors         : (60)
}

```

三、（20 分）应用题

基于 TMS320F28027DSP 设计一款电动自行车驱动控制器。设计思路如下：

- （1）转把为力矩的指令值，ADCINA7 连接至转把电位器变阻端简称中间点，转把转到不同的位置时电位器中间点电压从 0V 到 3.3V 不等。
- （2）PWM1A/PWM1B 控制自行车的输出力矩，电位器的中心点电压小于 0.5V 时 PWM1A 的占空比为 0，0.5V~1.0V 时 PWM1A 的占空比 20%，1.0V~3.3V 时 PWM1A 的占空比从 20%~100%线性变化。
- （3）CAP1 接转速测量仪，转速测量仪输出的脉冲信号的频率 f 对应于自行车的转速 n ，对应关系为 $n = 0.01f$ ，也就是 f 为 100Hz 时 n 为 1 公里/小时。
- （4）ADCINB7 测量自行车电池电压，0 至 3.3V 分别对应于电池电压 0~60V。
- （5）超速报警（报警时 GPIO2 输出为低）和电池电压报警（报警时 GPIO3 输出为低）

根据应用总需求和下面各小题的提示完成相应的程序设计，**各模块采用前面第二题的初始化。**

全局变量定义如下：

```

float MotorSpeed; //自行车转速，单位公里/小时
float TorqRef;    // 从转把得到力矩指令值(0%~100%)
float CellVolt;   // 电池电压值

```

1. （5 分）eCAP1 中断服务程序的编写，得到自行车的转速 MotorSpeed。
2. （5 分）ADCINT1 中断服务程序的编写，得到转把的力矩给定值 TorqRef 和电池电压值 CellVolt。
3. （5 分）定时器 0 中断服务程序的编写，实现将转把的力矩指令值转化为 PWM1A/PWM1B 的输出、自行车超速报警（高于 25 公里/小时报警）、电池电压异常报警（高于 55V 和低于 25V 报警）。
4. （5 分）主程序中完成以上三个中断服务程序的中断向量填写、PIE 级和 CPU 级的中断使能代码编写。