

## 《DSP 原理及应用》

### 2020-2021 学年第二学期期中考核题答案

一、简述 DSP 时钟源 (OSCCLK) 的两种配置模式, 说明系统时钟 (CLKIN) 的不同选择设定方式。(10 分)

有2种配置模式, 即使用片内振荡器和外部时钟源。(2分)

1) 使用内部振荡器: 在时钟输入引脚X1/XCLKIN和X2间接一个石英晶体, 构成内部振荡器的反馈电路;

2) 外部时钟源: 将时钟信号直接接到X1/XCLKIN引脚, X2悬空, 此时不使用片内振荡器。(4分)

当复位DSP芯片时,  $\overline{\text{XF\_PLLDIS}}$  的引脚状态决定是否使能PLL。

当系统复位时, 如果  $\overline{\text{XF\_PLLDIS}}=0$  时, CPU直接采用外部时钟或片内振荡器输出作为系统时钟;

如果  $\overline{\text{XF\_PLLDIS}}=1$  时, 外部时钟经过PLL分频或倍频后为CPU提供时钟。(4分)

二、F2812 的中断系统可以分为哪几种中断级别? 试简述 F2812 的中断处理流程。(10 分)

DSP 的中断分为 CPU 级、PIE 级和外设级三个级别。(4 分) 其中 F281× 共支持 17 个 CPU 级中断。外设中断扩展模块 (PIE) 中多个中断源复用 1 个中断信号。PIE 支持多达 96 个外设级中断源, 其中 8 个中断源为一组, 复用一个中断信号, 总共有 12 个中断信号 (INT1~INT12)。PIE 的中断优先级由高到低分别是 INT1—INT12, 组内 8 个优先级由高到低的次序依次是 INTx.1—INTx.8。(6 分)

三、读下列程序, 若高速外设时钟频率为 75MHz, 试分析程序, 写出设置功能。(12 分)

```
void init_eva()
{
    EvbRegs.T1PR = 0x63;
    EvbRegs.T1CMPR = 0x20;
    EvbRegs.T1CNT = 0x0000;
    EvbRegs.T1CON.all = 0x1042;
    EvbRegs.T2PR = 0x63;
    EvbRegs.T2CMPR = 0x10;
    EvbRegs.T2CNT = 0x0000;
    EvbRegs.T2CON.all = 0x1042;
    .....}
```

T1CON, T2CON 设置(计数模式选择)	0x1042: 001,0000,0100,0010 (连续增模式)
T1CON, T2CON 设置(输入时钟预定标因子)	TPS2-TPS0:000 X/1
T1PWM、T2 PWM 比较输出频率	$f = 75\text{MHz}/(99+1)=750\text{KHz}$

T1、T2 比较输出极性选择	T1 比较输出低有效, T2 比较输出高有效
----------------	------------------------

三、编程：使用 CPU 定时器 0 产生定时中断，每次中断（1 秒），两个发光二极管（接 GPIOF10、GPIOF11）交替闪烁；（18 分）

```
#include "DSP28_Device.h"
unsigned int Led_Flag;
interrupt void ISRTimer0(void);
void main(void)
{
    InitSysCtrl();           //初始化系统控制
    DINT;                    //关中断
    IER = 0x0000;
    IFR = 0x0000;
    InitPieCtrl();           //初始化 PIE
    InitPieVectTable();      //初始化 PIE 中断矢量表
    InitPeripherals();       //初始化外设，可以根据需要配置
    EALLOW;
    PieVectTable.TINT0 = &ISRTimer0;
    EDIS;
    ConfigCpuTimer(&CpuTimer0, 150, 1000000); //配置 CPU 定时器 0
    CpuTimer0Regs.TCR.bit.TSS = 0;             //起动 CPU 定时器 0
    PieCtrl.PIEIER1.all = 0x40;                //使能 CPU 定时器 0 中断
    IER |= 0x0001;
    EINT;                                       // Enable Global
    interrupt INTM
    for(;;) {.....}
}
```

（写出主要程序步骤即可得分）（10 分）

系统中断服务程序

```
#define LED1_ON  GpioDataRegs.GPFDAT.bit.GPIOF10=0
#define LED1_OFF GpioDataRegs.GPFDAT.bit.GPIOF10=1
#define LED2_ON  GpioDataRegs.GPFDAT.bit.GPIOF11=0
#define LED2_OFF GpioDataRegs.GPFDAT.bit.GPIOF11=1
（写出定义即可得分）（4 分）
```

// PIE Group 1 - MUXed into CPU INT1

```
interrupt void ISRTimer0(void)
{
    PieCtrl.PIEACK.all |= 0x0001;
    if(Led_Flag == 1)
    {
        LED1_ON;
        LED2_OFF;
    }
}
```

```
        Led_Flag = 0;
    }
else
{    LED1_OFF;
LED2_ON;
    Led_Flag = 1;
}
}
```

(写出主要程序步骤即可得分) (4 分)