

## 2021-2022 秋冬学期电气工程学院《微机原理与应用》(徐习东老师)

### 一、填空题 (每空 1 分)

- 1, ALU 是 算数逻辑单元, CPU 是 中央处理单元。
- 2, 2812 共有 8 个辅助地址寄存器 (XARn), XARn 都是 32 位的寄存器。
- 3, ST0 中 V 是 溢出 标志, C 是 进位/借位 标志。
- 4, 2812 内部总线中, 程序地址总线是 22 位的, 数据地址总线是 32 位的。
- 5, 2812 有 3 种低功耗模式, 分别是 IDLE、STANDBY 及 HALT 模式。

### 二、简答题 (每题 6 分)

- 1, 2812 在执行一条指令时, 通常分为哪 8 个阶段?  
1获取指令地址 2获取指令内容 3解码指令 4解析操作数地址 5锁定操作数地址 6获取操作数 7CPU执行工作 8将结果写入内存
- 2, 通用定时器计数输入时钟源有哪 3 种? 计数模式有哪四种? 通用定时器什么情况下才会产生 Overflow 中断标志?  
时钟源: 1QEP电路产生 2外部时钟 3内部时钟(HSPCLK->TCLK)  
计数模式: 1启动/停止 2连续增 3连续增/减 4定向增/减  
T1CNT的值为0x0000 (下溢) 或0xFFFF (上溢)

### 三、计算问答题 (16 分)

执行下面语句后, C 标志、V 标志、Z 标志、N 标志的值各为多少? 为什么? (12 分) 写出减法的计算结果, 结果放在哪个单元? (4 分)

```
MOV SP, #0x420      SP = *0x420
                     *0x416 = #0x10
MOV *-SP[10], #0x10  AL = #0x12
                     *0x416 - AL = -0x2 = 0xFFFF
MOV AL, #0x12
SUB *-SP[10], AL
```

结果存放在0x416单元  
运算结果为0xFFFF  
C = 0(有借位)  
V 保持不变(不确定V之前是什么)  
Z = 0(运算结果不为0)  
N = 1(最高位1表示负数)

### 四、汇编语言程序设计 (共 26 分)

- 1, 用直接寻址, 间接寻址以及空间立即寻址的汇编语言实现下面 C 语言的赋值语句 (6 分)

`*(int *) 0x70D5 = 0x3F00;`

- 2, 试编写汇编语言程序, 完成以下计算: (20 分)

1 空间立即寻址  
`MOV AL, #0x3F00`  
`MOV *(0:0x70D5), @AL`  
2 直接寻址  
`MOV DP, #0x70D5 >> 6`  
`MOV @21, #0x3F00`  
3 间接寻址  
`MOV XAR0, #0x70D5`  
`MOV *+XAR0[0], #0x3F00`  
4 间接寻址2  
`MOV SP, #0x70D5`  
`MOV *-SP[0], #0x3F00`

```
2.1
AAAA:
MOV SP, #0x220
MOVU ACC, *-SP[16]
MOV T, #10
RPT #15
||SUBCU ACC, @T
MOV AL, *-SP[0]
ADD AH, @AL
MOV *-SP[32], AH
LRET
```

```
2.2
AAAA:
MOVL XAR4, #0x3F9040
MOV AL, *+XAR4[0]
MOV T, @AL
MPYB P, T, #0x51
MOV AL, #0x20
ADD AL, @PL
MOV *(0:0x3F903F), @AL
LRET
```

①  $*(int *) 0x200 = (*(unsigned int *) 0x210) \% 10 + (*(int *) 0x220);$

②  $*(int *) 0x3F903F = (*(int *) 0x3F9040) * 0x51 + 0x20$

五、C 程序设计

1、试为一段测试程序编写连结命令文件。

① 把程序代码、变量安排在 0x3F8000~0x3F9FFF 空间内，程序所占空间为变量所指空间的 3 倍。要求安排 .text 段、.ebss 段、.cinit 段。程序中用到 PIEIER1、PIEIFR1，这两个寄存器的地址分别为：0xCE2、0xCE3，请在连接命令文件中给这 2 个寄存器安排地址。（8 分）  
写不下，看后面

② 简要说明程序空间，数据空间起始地址，长度的设定过程。（4 分）  
总空间0x3F8000-0x3F9FFF共0x2000长度，程序空间为数据空间的3倍，故程序空间长度0x2000\*3/4=0x1800，若起于0x3F8000则终于0x3F97FFF

③ 写出 C 程序中分配变量到数据段的语句及变量申明语句。（3 分）

```
#pragma DATA_SECTION (PIEIER1,"PIEIR1")
#pragma DATA_SECTION (PIEIFR1,"PIEFR1")
volatile int PIEIER1,PIEIFR1;
```

2、某 2812 DSP 通过 GPIOF 的 GPIOF9 引脚，向外输出占空比为 0.31 的周期信号，周期为 0.01 秒。系统外接晶振频率为 30MHz，要求系统时钟频率设置为 60MHz，关闭监视定时器，关闭所有外设时钟，GPIOF 的其他引脚都设置为输入状态，试编写主程序（10 分），中断服务程序（6 分）并写出有关时钟参数计算过程（3 分）。简要回答：如果未写中断向量会出现什么情况？（2 分）

（要求程序采用 CPU-Timer0 的溢出中断方式。已知：TIMER0PRD、TIMER0TCR、TIMER0TPR、TIMER0TPRH、GPnMUX、GPnDIR、GPnDAT、PLLCR、PIE 及其他相关寄存器已经定位在它们的立即地址上。）

```
MEMORY{
PAGE 0:
PRAMHO : origin = 0x3F8000, length = 0x1800
PAGE 1:
RAMMO : origin = 0x3F9800, length = 0x0800
PIEIER : origin = 0x000CE2, length = 0x0001
PIEIFR : origin = 0x000CE3, length = 0x0001
}
SECTIONS{
.text : >PRAMHO, PAGE = 0
.cinit : >PRAMHO, PAGE = 0
.ebss : >RAMMO, PAGE = 1
PIEIR1 : >PIEIER, PAGE = 1
PIEFR1 : >PIEIFR, PAGE = 1
}
```

注：问过XXD了，题目中的PIEIER1指的是.c文件中的变量名，SECTIONS中的段名和MEMORY中的空间名可以随便取，只要能一一对应上就行

```
void main(void){
asm(" EALLOW");
asm(" DINT");
WDCR = 0x0068; // 关闭监视定时器
PCLKCR = 0x0; // 关闭外设时钟
PLLCR = 0x4; // 60MHz
GPFMUX = 0x0000;
GPFDIR = 0x0200;
asm(" and IFR,#00H");
asm(" or IER,#01H");
PIEIFR1 = 0x0000;
PIEIER1 = 0x0040;
PIECTRL = 0x1;
PIEACK = 0xffff;
LPMCRO = 0x0;

interrupt void Timer0_INT(void);
typedef interrupt void (*PINT)(void);
*(PINT*)0x0D4C = &Timer0_INT;

TIMER0PRD = 99; // 10^4 Hz = 0.1ms
TIMER0TPR = 59;
TIMER0TPRH = 0;
TIMER0TCR = 0xf000; // 开时钟
asm(" EINT");
asm(" EDIS");
int k = 0;
for(;;) {asm(" IDLE");}
```

```
Timer0_INT(void){
k++;
if (k == 69) GPFDAT = 0xFFFF;
else if (k == 100){
GPFDAT = 0xF0FF;
k = 0;
}
PIEACK = 0x1;
TIMER0PCR = 0xf000;
}
```