1. **知识表达题（每小题4分，共24分）**

**（**评分标准：核心要点全部给出，评分给4分；给出部分要点，评分3分；其余，答对几个要点，给几分，不超过2分；空白，评分为0。**）**

1. **浮点数的阶（指数）为什么要用移码表示？它的精度和范围取决于什么？**

在浮点数的加减运算中，要进行对阶操作，需要比较两个阶的大小。用移码表示阶码后，使得所有数的阶码都相当于一个正整数，比较大小时，只要按高位到低位顺序比较就行了，从而简化阶码的比较过程。浮点数的精度取决于尾数的位数，范围取决于阶码的位数。

1. **在补码加法器中，如何实现补码减法运算？**

答：补码减法的规则是：两个数差的补码可用第一个数的补码加上另一个数的负数的补码得到。由此可见，减法运算可在加法器中运行，只要在加法器的第二个输入端输入减数的负数的补码。求一个数的负数的补码电路称为“负数求补电路”，可以通过“各位取反、末位加1”来实现“负数求补”，其中，“末位加1”通过将加法器的低位进位设为1来实现。

1. **CPU中标志寄存器的功能是什么？有哪几种基本标志？**

CPU中的标志寄存器又称为状态寄存器、程序状态字寄存器，用来存放运算类指令产生的状态标志信息，包括零标志ZF、溢出标志OF、符号标志SF和进位/借位标志CF等。这些标志信息作为条件转移指令是否进行转移的依据。

**4、CPU是如何发现内部异常和外部中断的？两者最本质的不同是什么？**

当CPU在执行当前程序或任务的第i条指令时，若检测到一个异常事件，或在执行第i条指令后发现有一个中断请求信号，则CPU会中断当前程序的执行，跳转到操作系统中相应的异常或中断处理程序去执行。

内部异常与外部中断最本质的不同在于它们的起因和与当前执行指令的关联性。

起因：内部异常是由CPU内部事件或程序执行中的事件引起的，与外部设备或随机事件无关；而外部中断则是由计算机外设或其他随机事件产生的，与当前执行的指令无关。

与当前执行指令的关联性：内部异常与当前执行的指令有直接关联，是执行指令时检测到的异常；而外部中断则与当前执行的指令无关，是由外部设备或随机事件触发的。

**5、流水线处理器中的时钟周期如何确定？理想情况下单条流水处理机的CPI为多少？**

流水线处理器将每条指令的执行规整化为若干个流水段，流水段的个数是以最复杂指令的执行过程所需的流水段个数为标准设计的。每个流水段的宽度也是以最复杂流水段所需宽度来设计，即流水线处理器的时钟周期设置为最慢的流水段所需时间。

理想情况下，当指令源源不断进入流水线，经过若干个周期后，能在每个时钟周期内执行完一条指令，因此CPI等于1。但是指令流水线中存在各种冒险都会阻塞流水线的执行。

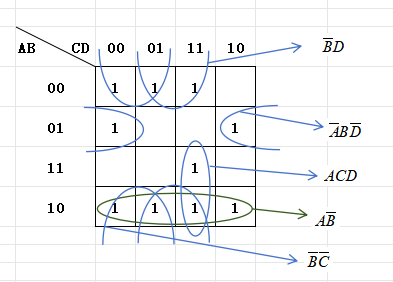
**6、简述DMA控制方式下如何实现I/O操作？**

答：DMA控制方式下实现I/O操作包括如下步骤：1）当外设准备好数据，或准备好接收数据时，发“选通”信号，使数据送数据缓冲寄存器，同时DMA请求触发器置“1”。2）DMA请求触发器向控制/状态端口发“Ready”信号，同时向DMA控制器发“DMA请求”信号。3）DMA控制器向CPU发“总线请求”信号。4）CPU完成现行机器周期后，响应DMA请求，发出“总线响应”信号。DMA控制器接受到该信号后，发出“DMA响应”信号，使DMA请求触发器复位。此时，CPU浮动它的总线，让出总线控制权，由DMA控制器控制总线。5）DMA控制器给出内存地址，并在其读/写线上发出“读”或“写”命令，随后在数据总线上给出数据。6）根据读写命令，将数据总线上的数据写入存储器中，或写入数据端口，并进行主存地址增量，计数值减1，若采用“CPU停止法”，则循环第6步，直到计数值为“0”。若采用“周期挪用法”，则释放总线（下次数据传送时再按过程(1)到(6)进行）。

**二、综合分析题（每小题7分，共56分）**

**（**评分标准：分析步骤和结果都正确，评分给7分；若有两个小题，则均分，各3.5分；部分步骤或结果正确，评分为4-5分或2-3分；其余，答对几个要点，给几分，不超过5分或3分；空白，评分为0。**）**

**1、**求解：

图1 卡诺图化简

**2、答**：根据真值表利用无关项进行化简的卡诺图如图2所示，对应逻辑表达式，其对应的电路图如图3所示。分析逻辑表达式，其乘积项中A、B、D这3个变量都存在逻辑相反的情况。进一步分析，在3个变量B、C、D从000取值到111的8种组合下，输出不可能会出现的情况；同样在3个变量A、C、D从000取值到111的8种组合下，输出不可能会出现的情况；但是，在A=1,B=0,C=1时，会出现，此时当D从1变为0时，会在输出端形成短暂的毛刺，因此图3对应电路存在竞争冒险，可通过增加冗余项的方式消除，此时**。**

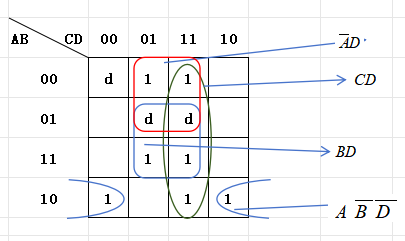


图2 卡诺图化简

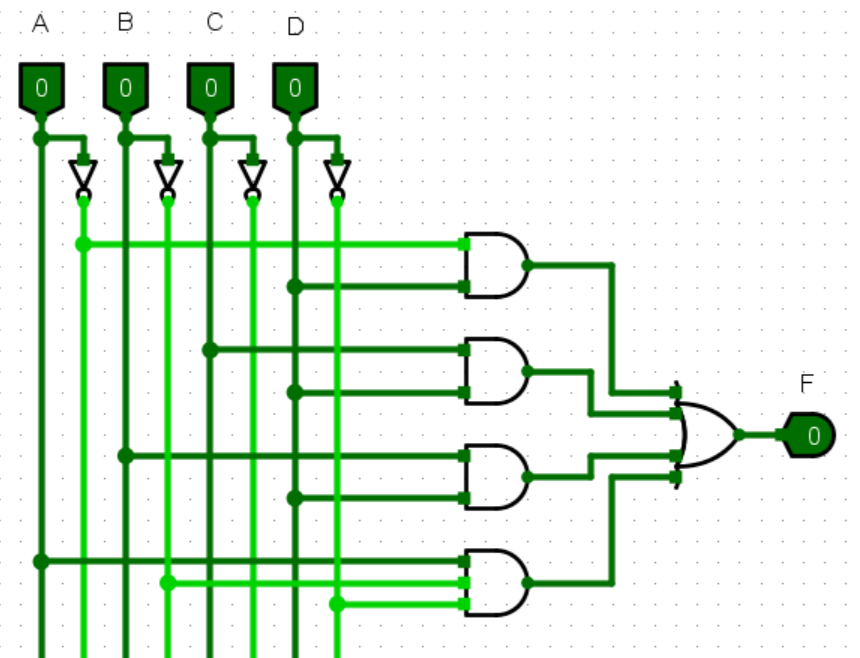


图3 对应的电路图

**3、** 答：设S0表示初始状态以及当X和Y输入0的累计次数为5的倍数时的状态，S1表示0的累计输入次数为模5余1时的状态，S2表示0的累计输入次数为模5余2 时的状态，S3表示0的累计输入次数为模5余3 时的状态，S4表示0的累计输入次数为模5余4 时的状态，则得到电路的状态图（转移条件表达式格式为XY/Z)如图4所示。

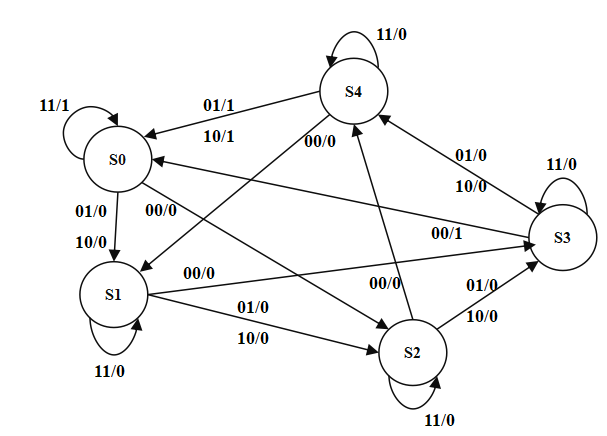


图4 时序逻辑电路状态图

**4、**答：module SaturatingCounter(

input clk,rst,

input predict\_right,predict\_wrong,

output predict\_taken

);

//补充代码如下

reg [2:0] cnt;

always @(posedge clk or posedge rst)begin

if(rst)cnt <= 3´b0;

else if (predict\_right && cnt !=3´b111)

cnt = cnt +1´b1;

else if (predict\_wrong && cnt !=3´b000)

cnt = cnt - 1´b1;

end

assign predict\_taken = cnt[2];

endmodule

**5、采用IEEE754单精度浮点数格式计算表达式的(0.75)-（-65.25）值，请明确给出详细计算步骤。**

解：x=0.75=0.110...0B=(1.100...0)2×2-1

y=-65.25=-1000001.01000...0B=-(1.000001010...0)2×26

用IEEE754标准单精度格式表示为：

Ex = -1+127 = 126 = 01111110B Ey = 6+127 = 133 = 10000101B

[x]浮=0 01111110 10...0 [y]浮= 1 10000101 000001010...0

Ex=01111110, Mx=0(1).10....0, Ey=10000101,My=1(1).000001010...0

尾数Mx和My中小数点前面有两位，第一位为数符，第二位加了括号，是隐藏位“1”。

以下是计算机中进行浮点数加减运算的过程（假定保留2位附加位：保护位和舍入位）

（1）0.75-(-65.25)

①对阶：[∆E]补=[Ex]移+[-[Ey]移]补 （mod 2n)

=0111 1110 + 0111 1011=1111 1001

∆E=-7，根据对阶规则可知需要对x进行对阶，结果为：Ex=Ey=10000101,Mx=00.000000110...0**00**,x的尾数Mx右移7位，符号不变，数值高位补0，隐藏位右移到小数点后面，最后移出的2位保留。

②尾数相减：Mb=Mx-My=00.000000110...0 - 11.000001010...0（注意小数点在隐藏位后）

根据原码加/减运算规则，得:Mb=Mx - My=00.000000110...000 - 11.000001010...000

=01.00001000...000,上式尾数中最左边第一位是符号位，其余都是数值部分，尾数后面两位是附加位（加粗）。

③规格化：根据所得尾数的形式，数值部分最高位为1，所以不需要进行规格化；

④舍入：把结果的尾数Mb中最后两位附加位舍入掉，从本例来看，不管采用什么舍入法，结果都一样，都是把最后两个0去掉，得01.00001000...000

⑤溢出判断：在上述阶码计算和调整过程中，没有发生“阶码上溢”和“阶码下溢”的问题。因此，阶码Eb=10000101

最后结果为Eb=10000101,Mb=0(1).00001000...000,即：+66。

1. 答：Store指令(M[R[Rs]+SEXT[imm16]]←R[Rt]，将寄存器Rt的内容写入内存，内存单元地址为寄存器内容加立即数）执行过程包括4个阶段，各自完成的功能、所需的有效控制信号及其取值如下：

（1）取指令并计算下条指令地址（公共操作），记为 IFecth，根据PC读指令并保存到IR，PC+4

有效控制信号及其取值如下：

① R[IR] ← M[PC]：PCout=1、MARout=0、MemWr=0、IRWr=1

② PC ← PC+4：Add1MUX=0、Add2MUX=1、PCWr=1

③ 其他写使能信号（如MARWr、CCWr、MDRWr、ALUoutWr、RegWr）全部为0

（2）译码并取数（公共操作），记为 Rfecth/ID，IR的OP及CC送控制器译码，并根据Rs和Rt读取寄存器中数据，加法器空闲，投机计算主存地址,有效控制信号及其取值为：ExtOP=1(符扩)、Add1MUX=1、Add2MUX=0、MARWr=1、其他写使能信号全部为0。

Store指令：地址已投机计算，还需两个时钟周期，记为 swExec、swFinish状态，将Rt寄存器中的内容写入MDR，再将MDR的内容写入投机计算好的主存单元中（地址在MAR中）。

（3）swExec阶段的功能是：将Rt寄存器中的内容写入MDR并直送总线，有效控制信号及其取值：MARout=1、PCout=0、MemWr=0、MDRMUX=0、MDRWr=1、其他写使能信号全为0。

（4）swFinish阶段的功能是：将MDR送入总线的数据写入主存。有效控制信号及其取值：RegMUX=0、RegWr=1、其他写使能信号全为0。

7、答：1）数据在主存和Cache间的传送单位为64B，按字节编址，64=26，所以，块内地址为6位。 Cache数据区容量32KB=215B=29×26 B=512行×64B/行=128组×4行/组×64B/行。主存大小为512MB=229B=223x26 B=223块 × 64B/ 块。29位的主存地址划分为三个字段：标记t=16位,cache组号q=7位，块内地址b=6位。

2）主存地址1540240CH展开成29位二进制数为 1 0101 0100 0000 0010 0100 0000 1100，所以主存地址划分为：1 0101 0100 0000 001，0 0100 00，001100，访问1540240CH单元的过程为：首先根据地址中间7位0 0100 00，找到cache第16组，将高16位标记1 0101 0100 0000 001与cache第16组中的4个行的标记进行比较，若有一个相等并且对应有效位为1，则命中，此时，根据低6位块内地址001100从该行中取出信息送CPU；若都不相等或有一个相等但有效位为0，则不命中，此时，需要将1540240CH单元所在的主存第1 0101 0100 0000 0010 0100 00块（即第5570704块）复制到cache第10000组（第16组）的任意一个空行中，并置有效位为1，置标志位为1 0101 0100 0000 001。

8、答：1）中断处理次序改为L1→L3→L2→L4。

表1 中断屏蔽位设置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中断处理程序 | 中断处理屏蔽位 | | | |
| L1级 | L2级 | L3级 | L4级 |
| L1中断处理程序 | 1 | 1 | 1 | 1 |
| L2中断处理程序 | 0 | 1 | 0 | 1 |
| L3中断处理程序 | 0 | 1 | 1 | 1 |
| L4中断处理程序 | 0 | 0 | 0 | 1 |

**2）** L1 L3 L2 L4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **L4** |  |  |  |  |  |
| **L3** |  |  |  |  |  |
| **L2** |  |  |  |  |  |
| **L1** |  |  |  |  |  |
| **用户程序** |  |  |  |  |  |

图5 中断程序执行流程

**三、综合设计题(每小题10分，共20分)**

**（**评分标准：设计步骤和结果都正确，评分给10分；部分步骤或结果正确，评分为6-9分；其余，答对几个要点，给几分，不超过4分；空白，评分为0。**）**

1、答：（评分说明：第1、2、4小题各评分2分；第3小题评分4分）

（1）DRAM芯片容量为1M×8位=1MB，每个内存条需要16MB/1MB=16个DRAM芯片。

（2）构成容量为32MB的主存时，需要32MB/16MB=2个内存条。

（3） 按字节编址，主存最大寻址空间64MB=226B，所以**主存地址共有26位**，即A25A24…A2A1A0

，容量为64MB的主存需要64MB/16MB=**4个**内存条，因为每次在总线上传输32位数据，所以内存条内由4个芯片同时进行读写，每个芯片有8个位平面，因而4个芯片可同时读出32位数据，芯片按交叉方式编址，每个内存条有16MB=224B，故**内存条内存储单元的地址占24位**，即A23A22…A2A1A0，其中内存地址的**最高两位A25A24经过2：4译码器用来选择内存条**。而内存地址最低两位A1A0表示芯片号，用于选择芯片，**A23A22…A2表示芯片内的存储单元地址**，A23A22…A13为DRAM芯片内的**行地址**，A12A12…A2为DRAM芯片内的**列地址**。

1. DRAM芯片为2048行，若芯片的刷新周期为2ms，采用异步式刷新，则产生刷新信号的间隔为2ms/2048=2000us/2048=2000 000ns/2048=997ns。

**2、答：（评分说明：前面分析5分，后面每种指令格式各1分，共10分）**

至多有64种操作，故操作码字段只需要6位；有8个通用寄存器，故寄存器编号至少占3位；寻址方式有4种，故寻址方式位至少占2位；直接地址和立即数都是16位；任何通用寄存器都可做变址寄存器，故指令中要明显指定变址寄存器，其编号占3位；指令字长为16的倍数；指令中需要明确给出每个操作数的寻址模式，二地址指令中给出两个操作数地址，因此，每个操作数各需要2位寻址模式。此外，指令格式应尽量规整，指令长度应尽量短。按照上述要求设计RR型、RI型、RS型、XI型和SS型5种二地址指令格式如下：

用专门的“寻址方式”字段分别说明两个操作数的寻址方式。其定义如下：00--立即寻址，01--寄存器直接寻址，10--寄存器间接寻址，11--变址寻址。

RR型：

OP(6位) 01 Rt(3位) 01 Rs(3位)

OP(6位) 01 Rt(3位) 00 Imm16(16位) 000

000

RI型：

OP(6位) 01 Rt(3位) 10 Rs(3位)

RS型：

OP(6位) 01 Rx(3位) Offset16(16位) 00 Imm16(16位) 000

11 Rx(3位)

XI型：

SS型： RR型：

OP(6位) 10 Rt(3位) 10 Rs(3位)

其中，RI、XI指令格式中添加了3个0，是为了补足位数，以使指令长度为16的倍数。

存储器存取宽度为16位，每次从存储器取出16位。因此，读取16、32和48位指令分别需要1、2和3次存储器访问。各类指令的功能和访存次数分别说明如下（M[x]表示存储器地址x的内容，R[x]表示寄存器x中的内容）。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 指令 | 功能 | 指令长度 | 取指访存 | 执行访存 | 总访存次数 |
| RR | R[Rt]←R[Rt] op R[Rs] | 16位 | 1 | 0 | 1 |
| RI | R[Rt]←R[Rt] op Imm16 | 32位 | 2 | 0 | 2 |
| RS | R[Rt]←R[Rt] op M[R[Rs]] | 16位 | 1 | 1 | 2 |
| XI | M[R[Rx]+Offset]  ← M[R[Rx]+Offset] op Imm16 | 48位 | 3 | 2 | 5 |
| SS | M[R[Rt]]←M[R[Rt]] op M[R[Rs]] | 16位 | 1 | 3 | 4 |