

## 一、选择题 (10分/10)

1-5: ACABB 6-10:ABDCC

## 二、简答题(18分/3)

1、简述冯诺依曼体系结构的要点并分析其缺陷，现代计算机如何解决其缺陷。(6分)

答：冯诺依曼体系结构将计算机分为：运算器、控制器、内存、输入系统、输出系统；(2分) 使用存储程序及二进制 (1分); 缺陷为：在内存与CPU间因为数据交换频繁而产生瓶颈 (即冯诺依曼瓶颈) (1分)；现代计算机采用多级高速缓冲存储器Cache来从内存中调度数据，如I-cache及D-cache使得取指令和取数据分别访问不同的cache。(2分)

2、为什么采用并行进位能提高加法器的运算速度？(2分) 请简述如何用4位快速加法器SN74181及组间并行进位74182构建一个16位先行进位加法器。(4分)

答：并行进位使得全加器不需要再等待前一全加器的进位而导致的延迟，通过改变进位间的相互依赖关系，各全加器的进位只与C0有关，从而达到提高整体加法器的运算速度。(2分)

需要四个四位SN74181先行进位加法器，但每个SN74181之间仍需要等待上一SN74181的最后一位输出参与下一轮计算，因为需要一个组间并行进位74182模块构成二重进位加法器，即C4, C8,C12,C16的进位可以同时产生，构成组内并行，组间并行的二重16位加法器。(4分，根据学生回答酌情给分)

3、简述复杂指令集计算机CISC与精简指令集计算机RISC的不同，列出至少三条不同之处，并分别列举至少一种典型的CISC及RISC的CPU微架构。(6分)

答：(1) CISC指令系统复杂，支持变长指令字，多种寻址模式；而RISC指令系统相对简单，指令长度一致，指令少，寻址方式少；

(2) CISC指令周期长，绝大多数指令需要多个时钟周期才能完成；RISC指令周期短。以流水线方式工作，除Load/Store指令外，其他简单指令都只需一个或一个不到的时钟周期就可完成；

(3) CISC各种指令都能访问存储器，除了专门的存储器读写指令外，运算指令也能访问存储器；RISC以RR方式工作，除Load/Store指令可访问外，其余指令都只访问寄存器；

(4) CISC多采用微程序控制器；RISC采用硬连线路控制器，不用或少用微程序控制。

(以上答出任意三条得5分，答不全的可酌情给分)

Intel X86是典型的CISC微架构；MIPS, ARM是典型的RISC微架构。(1分)

## 三、计算题(20分，所有计算题请给出计算过程，只有结果没过程不得分)

1、采用浮点数格式补码表示法计算 $x+y$ ，其中 $x=0.625, y=-22.25$ ，设浮点数的阶码为4比特(含阶符)，尾数为6比特(含尾符)，判断是否溢出。(请按浮点运算步骤对阶、尾数运算、规格化、舍入和判断溢出进行，结果不用转成十进制，尾数用0舍1法，如果不按浮点运算方法计算不得分)(8分)

$X_{原}=0.10100*2^{-0000}$ ;  $Y_{原}=1.10110$  (0舍1入)  $*2^{0101}$ ; (2分)

对阶:  $X_{阶}=0000$   $Y_{阶}=0101$ ;  $(-Y_{阶})补=1011$   $X_{阶}-Y_{阶}=X_{阶}补+(-Y_{阶})补=0000+1011=1011$

故 $X_{阶}$ 小于 $Y_{阶}$ ，小阶向大阶对齐:  $X_{补}=0.00001$  (0舍1入)  $*2^{0101}$   $Y_{补}=1.01010*2^{0101}$  (2分)

相加:  $[X+Y]补=[X]补+[Y]补=00.00001*2^{0101}+11.01010*2^{0101}=11.01011*2^{0101}$  (2分)

规格化: 尾数1.01011已是补码规格化

舍入: 尾数不需舍入

溢出判断: 本题无溢出 (以上共2分)

有些同学一直带着所有尾数，直到最后加法结果出来才用0舍1入法保留六位，按题目 $X, Y$ 尾数要求6位，但可以扣不扣分。

2、假设某计算机A有24条指令,指令的操作码字段固定为5位,从00000-10111,该机器的后续机型B中需要增加16条指令,并与A保持兼容。

(1) 试采用操作码扩展方法为计算机B设计指令操作码,写出其操作码范围。 (3分)

A与B需要兼容,故B沿用A的操作码,因为A只剩下11000-11111,共8种,不够16条指令,因此B需要在A的基础上多加一位操作码,达到 $8*2=16$  (2分);

B的操作码范围为: 110000-111111 (1分);

(2) 计算该计算机中所有指令操作码的平均长度。 (3分)

$(24*5+16*6)/40=5.4$  (3分)

3、若某程序编译后生成的目标代码由A、B、C、D四类指令组成,它们在程序中所占比例分别为20%、20%、10%、50%。已知A、B、C、D四类指令的CPI分别为1、2、2、2。现需要对程序进行编译优化,优化后的程序中D类指令条数减少了一半,而其它指令数量未发生变化。假设运行该程序的计算机CPU主频为800MHZ。(保留1位小数)

(1) 优化前后程序的CPI各为多少? (2分)

优化前平均CPI:  $0.2*1+0.2*2+0.1*2+0.5*2=1.8$  (1分)

优化后平均CPI:  $1*20/75+2*20/75+2*10/75+2*25/75=1.7$  (1分)

(3) 优化前后程序的MIPS各为多少? (2分)

优化前MIPS:  $1/(1.8*1/800\text{MHz})=444.4$  (1分)

优化后MIPS:  $1/(1.7*1/800\text{MHz})=462.4$  (1分) (由于前面优化后的CPI 1.7不是整数,

这个答案在460-470之间都正确)

(3) 通过上面的计算结果你能得出什么结论? (2分)

优化了CPI较高的D类指令,降低了整个指令集的平均CPI,导致优化后的MIPS高于优化前的MIPS,故优化成功。

## 四、分析题(22分)

1、

(1)  $74_{10}=1001010$ ,根据题干虚页8页需要3位表示,页大小16个字需要4位表示,因此虚地址格式为:虚页号(3位)|页内字号(4位) (2分)

(2) 用虚页100做为索引查TLB,得到实页为1,主存4个实页,应用2位表示,因此物理地址为01(实页号,2位)|1010(页内字号,4位) (2分)

(3) CPU先用虚页号100查TLB, TLB命中,不需要查页表;主存命中,转换成物理地址011010;根据此物理地址,与Cache对应,此物理地址将分为tag(2位)|组号(1位)|块内字号(3位),该地址所包含的块应该在第1组上,cache第1组包含D和H块,分别是主存第0页和第2页,而该地址属于主存第1页,故cache不命中;CPU继而访问主存,将包含该地址的J块调入cache第1组,同时更新块表。 (3分)

(4) 【注:此小问与第(3)小问彼此独立】C, D, I都命中, J是主存第3块,将进入cache第1组,故替换H, G是主存第4块,将进入cache第0组,故替换C块,第一轮下来: 2个字不命中, 38个字命中;

第二轮: C不命中, 替换I, D命中, I不命中, 替换G; J命中, G不命中, 替换C, 第二轮3个字不命中, 37个字命中;

第三轮: C不命中, 替换I, D命中, I不命中, 替换G; J命中, G不命中, 替换C, 第二轮3个字不命中, 37个字命中;

故命中率为  $(38+37+37)/120=93.3\%$  (共3分, 学生只要有分析过程, 部分正确即使结果不对, 也可酌情给分)

(5) 根据(4), 可知最后Cache中的块为set0: G I块; Set1: D I块 (2分)

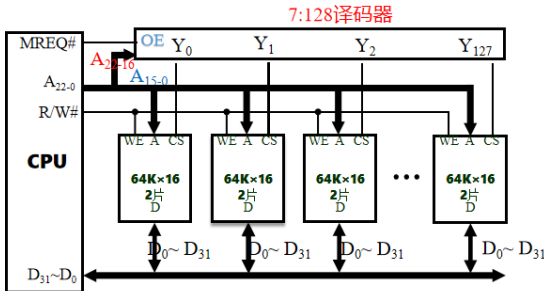
2、

答: (1) 120ns (2分)

- (2) 数据相关 (2分)
- (3) SUB指令推迟2个时钟, AND指令推迟3个时钟进入流水。 (2分)
- 流水时空图如下 (注: 画推迟后的流水时空图, 2分)
- | 指令/时钟 | 1 | 2  | 3  | 4  | 5  | 6  | 7  | 8 |
|-------|---|----|----|----|----|----|----|---|
| ADD   |   | IF | ID | E  | ME | W  |    |   |
|       |   |    | X  | M  | B  |    |    |   |
| SUB   |   |    | IF | ID | E  | ME | W  |   |
|       |   |    |    | X  | M  | B  |    |   |
| AND   |   |    |    | IF | ID | EX | ME | W |
|       |   |    |    |    | M  | B  |    |   |
- (4)  $S = [(100+60+60+120+60)*120]/[(5+120-1)*120] = 3.2$  (2分)

## 五、设计题(30分)

- 1、
- 答 (1)  $(8M*32)/(64K*16) = 2^8 = 256$  (2分)
- (2) 两块64K\*16b的芯片拼在一起构成64K\*32b的芯片 (位扩展), 一共需要128组 (字扩展), 构成8M\*32b的主存; (2分)
- (3)  $8M = 2^{23}$ , 一共需要23位; (1分)
- (4) 16 (小芯片内地址位长) | 7 (芯片组号地址位长) (2分)
- (5) 重要信号线都要有, 如MREQ, 地址线, 数据线, 读写线, 译码器要选对 (5分)



- 2.
- (1) 通用寄存器(R0~R3)和 PC 程序员可见; (1分)
- (2) ALU 共有 7 种操作, 故操作控制信号 ALUop 至少需要 3 位; 移位寄存器有3种操作, 其操作控制信号 SROp 至少需要 2 位; (1分)
- (3) 信号 SRout 所控制的部件是三态门, 用于控制移位器与总线之间数据通路的连接与断开; (2分)
- (4) 端口①、②、③、⑤、⑧都是控制信号, 需要连接到控制部件输出端, 其他都是数据通路; (2分)
- (5) 首先内总线数据需要送 MUX 的 1 号引脚, 因此⑥连接⑨; 另外多路选择器输出应该连接到运算器 B 端, 所以④连接⑦; (1分)
- (6) 指令的二进制为: 0000 1000 1000 1100——>088CH (16进制); (2分)
- (7) 操作码7位, 故该指令系统可定义128条指令; (1分)
- (8) 给出指令ADD R2,R1,(R0)执行阶段的数据流及控制流; (下表一行1分, 共计4分)

时钟	数据流	控制流
1	R0-->MAR,M[MAR]-->MDR	R0out=1,MARin=1,MEMOp=read (写1也可以),MDRinE=1
2	R1-->T	R1out=1,Tin=1
3	MDR-->MUX,SR-->三态门	MDRout=1,MUXOp=1,ALUOp=add,SROp=mov

4	三态门-->R2	SRout=1,R2in=1
---	----------	----------------

(9) (共4分，根据答案酌情给分)

图①～⑨中只有①、②、③、⑤、⑧是控制信号，故操作控制字段只考虑此5个控制信号，其中①为SRout用1位控制；②为SROP三种操作2位控制；③为ALUop七种操作3位控制；⑤为Tin用1位控制；⑧为MUXop二选一多路选择器控制端用1位控制；判断字段2位，下址字段9位（因为控制存储器容量为500字），故微指令格式如下：

操作控制字段					判断字段		下址字段
SRout	SROP	ALUop	Tin	MUXop	P1	P2	微地址
1 bit	2 bits	3 bits	1 bit	1 bit	1 bit	1 bit	9 bits