

高等学校计算机专业核心课

名师精品·系列教材

教育部高等学校软件工程专业教学指导委员会

软件工程专业系列教材

计算机 组成原理

微课版

华中科技大学计算机科学与技术学院 组编

谭志虎 主编

秦磊华 吴非 肖亮 副主编

习题参考答案

COMPUTER ORGANIZATION
AND ARCHITECTURE

立足经典知识体系，注重系统性与完整性，培养计算机系统能力
理论实践一体化，游戏闯关式在线虚拟仿真实验，轻松破解实验难题
紧扣考研大纲，精品慕课、微课同步导学，优质教辅资源助力教学



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

目录

第 1 章	计算机系统概论	1
第 2 章	数据信息的表示	2
第 3 章	运算方法与运算器	5
第 4 章	存储系统	6
第 5 章	指令系统	10
第 6 章	中央处理器	12
第 7 章	指令流水线	15
第 8 章	总线系统	18
第 9 章	输入输出系统	19

第1章 计算机系统概论

1.2 选择题

- (1) D (2) C (3) C (4) A (5) D
(6) D (7) C (8) D (9) D (10) C

1.5 $CPI_1=3.25$ $MIPS_1=185$ $CPI_2=2.65$ $MIPS_2=302$

1.6 (1) 优化前 $CPI=1.6$ 优化后 $CPI=1.75$

(2) 优化前 $MIPS=312.5$ 优化后 $MIPS=285.7$

(3) 优化后 CPI 增加, $MIPS$ 减少, 优化失败, 应尝试降低 CPI 最大指令的比例。

第2章 数据信息的表示

2.2 选择题

- (1) B (2) A (3) B (4) D (5) A
(6) A (7) D (8) A (9) A (10) B (11) C

2.4

#	真值	原码	反码	补码
1	0	0.00...0	0.00...0	0.00...0
2	-0	1.00...0	1.11...1	0.00...0
3	0.10101	0.10101	0.10101	0.10101
4	-0.10101	1.10101	1.01010	1.01011
5	0.11111	0.11111	0.11111	0.11111
6	-0.11111	1.11111	1.00000	1.00001
7	-0.10000	1.10000	1.01111	1.10000
8	0.10000	0.10000	0.10000	0.10000

2.5

补码	真值	补码	真值
$[x]_{\text{补}} = 0.10010$	$x = 0.10010$	$[x]_{\text{补}} = 1.10010$	$x = -0.01110$
$[x]_{\text{补}} = 1.11111$	$x = -0.00001$	$[x]_{\text{补}} = 1.00000$	$x = -1.00000$
$[x]_{\text{补}} = 0.10001$	$x = 0.10001$	$[x]_{\text{补}} = 1.00001$	$x = -0.11111$

2.6

解:

输出结果如下:

$x = 4294967295 = -1$;

$u = 2147483648 = -2147483648$

- 1) %u 以无符号输出, %d 输出真值
- 2) 在计算机中整数以补码形式表示和存储。
- 3) $x = -1$, 先求 -1 的 32 位补码, 机器码是 $2^{32} - 1 = 4294967295$ 。所以第一行输出是分别是机器码和真值。
- 4) $u = 2^{31}$ 是一个无符号数, 无溢出, 由于首位为 1, %u 输出机器码就是 2147483648, %d 输出是真值, 将该机器码按补码转换成真值, 所以是 -2147483648 。

2.7

解:

- 1) 16 位无符号数: $0 \sim 1111\ 1111\ 1111\ 1111$, 即 $0 \sim 2^{16}-1=65535$
- 2) 16 位原码定点小数: $1.111\ 1111\ 1111\ 1111 \sim 0.111\ 1111\ 1111\ 1111$, 即 $-(1-2^{-15}) \sim 1-2^{-15}$
- 3) 16 位补码定点小数: $1.000\ 0000\ 0000\ 0000 \sim 0.111\ 1111\ 1111\ 1111$, 即 $-1 \sim 1-2^{-15}$
- 4) 16 位补码定点整数: $1000\ 0000\ 0000\ 0000 \sim 0111\ 1111\ 1111\ 1111$, 即 $-2^{15} \sim 2^{15}-1$

2.8

解: 8 位补码的表示范围为 $-128 \sim 127$, 模为最高位 x_0 的进位位的权值, 所以模为 256。

2.9

解:

- (a) $1100\ 0000\ 1101\ 0100\ 0000\ 0000\ 0000\ 0000 = (C0D40000)_{16}$
- (b) $0100\ 0000\ 0100\ 1001\ 0000\ 1111\ 1101\ 1011 = (40490FDB)_{16}$
- (c) $0100\ 0111\ 0111\ 1010\ 0000\ 0000\ 0000\ 0000 = (477A0000)_{16}$

2.10

解: 十进制数=296。

2.11

解: $f_{max} = 2^{127} \times (2-2^{-23})$ $f_{min} = -2^{127} \times (2-2^{-23})$

2.12

解: (1) 有可能, 例如 $N_1 = 2^3 \times 0.1$, $N_2 = 2^4 \times 0.001$, 此时 $m < n$, 却有 $N_1 > N_2$ 。

(2) 不可能。因为规格化浮点数要求尾数的最高位为非 0 数码, 即当尾数的值不为零时, 其绝对值应大于或等于 $(1/2)_{10}$, 那么 M_1 和 M_2 都必须是 $0.1 \times \dots \times$ 的形式。这时, 若 $m < n$, 则一定有 $N_1 < N_2$ 。

2.13

解:

#	阶码	尾码	真值
最大正数	011	0.111111	$2^3 \times (1-2^{-6})$
最小正数	100	0.000001	$2^{-4} \times 2^{-6}$
最大负数	100	1.111111	$-2^{-4} \times 2^{-6}$
最小负数	011	1.000000	-2^3

2.14

解:

- (1) $57/128 = 1111, 0111001000$ (2) $-69/128 = 0000, 1011101100$

2.15

解: 奇校验码: 010110110 偶校验码: 010110111

如果接收方收到的 $x=010110100$ (只有 1 位出错, 最后一个 0 是校验位), 如果采用奇校验, 接收方计算检错位 $G=1$, 表明数据一定发生了错误。如果采用偶校验, 接收方计算检错位 $G=0$, 表明数据高概率正确。

2.16

解: 则 $X_1 X_2 X_3 X_4$ 处的比特分别为 1110; $X_5 X_6 X_7 X_8$ 处的比特分别为 1000; $X_9 X_{10} X_{11} X_{12}$ 处的比特分别为 1011; Y_1 和 Y_2 处的字符分别为 I 和 7。

2.18

解: 被检验位有 8 位, 设检验位有 r 位, 因为 $8+r \leq 2^r - 1$ 所以 $r=4$

具体分组关系如下表:

海明码	H ₁	H ₂	H ₃	H ₄	H ₅	H ₆	H ₇	H ₈	H ₉	H ₁₀	H ₁₁	H ₁₂
检错码/位置	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100
映射关系	P₁	P₂	0/D ₁	P₃	1/D ₂	1/D ₃	0/D ₄	P₄	1/D ₅	1/D ₆	1/D ₇	0/D ₈

海明码为: 110011011110

接收方接收到的海明编码为 110011011111, 只有 D8 位出错

$$G_1 = P_1 \oplus D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7 = 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 0$$

$$G_2 = P_2 \oplus D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7 = 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 0$$

$$G_3 = P_3 \oplus D_2 \oplus D_3 \oplus D_4 \oplus D_8 = 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 = 1$$

$$G_4 = P_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus D_8 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 1$$

指错码 $G_4 G_3 G_2 G_1 = 1100 = 12$, 如果假设只有一位错, 则是海明码 H_{12} 出错, 也就是 D_8 出错, 将对应位取反即可。

2.19

解: 作模二除法: $\frac{M(x) \cdot X^3}{G(x)} = \frac{1001000}{1101} = 1111 + \frac{011}{1101}$

所以循环码为: 1001011。

若接收到的数据信息 $x'=1101$, $\frac{1101011}{G(x)} = 1000 + \frac{011}{1101}$,

将余数 011 继续补零作除法, 经过两次运算余数为 001, 所以是第 2 位出错, 将左侧起第 2 位的取反即可, 也可以通过查表法快速定位出错位置。

第3章 运算方法与运算器

3.2 选择题

(1) D (2) B (3) A (4) C (5) B (6) D (7) D

3.3

解: (1) $[x+y]_{\text{补}}=01.10001$, 正溢出 (2) $[x+y]_{\text{补}}=00.01001$, 未溢出
(3) $[x+y]_{\text{补}}=10.10001$, 负溢出

3.4

解: (1) $[x-y]_{\text{补}}=11.11110$, 未溢出。 (2) $[x-y]_{\text{补}}=11.11001$, 未溢出。
(3) $[x-y]_{\text{补}}=11.11010$, 未溢出。

3.5

解: (1) $x \times y = -0.1110000011$ (2) $x \times y = 0.0100011110$

3.6

解: (1) $[x \times y]_{\text{补}}=1.1110111110$ (2) $[x \times y]_{\text{补}}=0.001011110010$

3.7

解: (1) $x \div y = 0.11000$, 余数 $r = 0.11 \times 2^{-5}$ (2) $x \div y = -0.111$, 余数 $r = 0$

3.8

解: (1) $[x+y]_{\text{补}}=\underline{000}10, \underline{00}101110$ (2) $[x+y]_{\text{补}}=\underline{11}100, \underline{11}011001$

3.9

解: (1) $0.625 + (-12.25) = (1 \ 10000010 \ 011101000000000000000000)_2$
 $= (C13A0000)_{16}$

(2) $0.625 - (-12.25) = (0 \ 10000010 \ 100111000000000000000000)_2$
 $= (414E0000)_{16}$

3.10

解:

(1) $(R1)=86H, (R5)=90H, (R6)=7CH$

(2) $m=-122, k1=-112$ 。

(3) 能。

(4) 如果符号位进位 C_f 和最高数据位进位 C_d 不同, 则结果溢出。最后一条语句执行时会发生溢出。

4.8

解：产生刷新信号的时间间隔为 $7.8125\mu\text{s}$ 。若采用集中刷新，存储器刷新一遍至少需要 256 个读写周期，CPU 的死时间是 $256 \times 0.5\mu\text{s} = 128\mu\text{s}$ 。

4.9

解：11 个。

4.10

答：采用异步刷新方式或分散刷新比较比较合适。刷新信号的产生周期为 $15.625\mu\text{s}$ 。

4.11

解：

(1) 全相联映射

cache 行	标志	数据
0	000000	87568536
1	000002	87792301
2	004001	9ABEFC68
3	007FFF	4FFFC68
4	3FFFFE	01BF2460

(2) 直接相联映射

cache 行	标志	数据
0000	00	87568536
0002	00	87792301
0001	01	9ABEFC68
3FFF	01	4FFFC68
3FFE	FF	01BF2460

(3) 组相联映射

cache 组	标志	数据
000	000	87568536
002	000	87792301
001	004	9ABEFC68
0FFF	007	4FFFC68
0FFE	3FF	01BF2460

4.12

解：

(1) 主存地址位数为 19 位，cache 地址位数为 13 位

(2)

标记(tag)	组索引(index)	块内偏移(offset)
8	4	7

4.13

解：(1) cache 命中率 = 98.7%

(2) 使用了 cache 后速度提高了 8.95 倍

4.14

解：(1)程序 A 中具有很好的空间局部性，不存在时间局部性；

而程序 B 空间局部性不佳，同样也不存在时间局部性；

(2)变量 sum 在循环中被多次使用，故具有良好的时间局部性。

(3)for 循环中的指令会被反复循环执行，因此具有较好的时间局部性，另外循环体中的机器指令序列通常会顺序执行，因此也具有一定的空间局部性。

4.15

解：VPN 的位数为 19 位，VPO 和 PPO 的位数均为 12 位，PPN 的位数为 11 位。

4.16

解：(1)失效页分别是 2、3、5、7

(2)

主存地址 ₁₀	主存地址 ₂	VPN	PPN	实存地址 ₂	实存地址 ₁₆
0	<u>000</u> 00 0000 0000	0	3	<u>11</u> 00 0000 0000	3072
3028	<u>010</u> 11 1101 0100	2	1	<u>01</u> 11 1101 0100	2004
1023	<u>000</u> 11 1111 1111	0	3	<u>11</u> 11 1111 1111	4095
2048	<u>010</u> 00 0000 0000	2	缺页		
4096	<u>110</u> 00 0000 0000	6	0	<u>00</u> 00 0000 0000	0
8000	<u>111</u> 11 0100 0000	7	缺页		

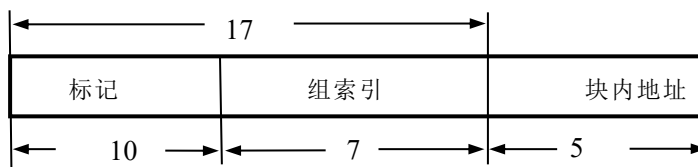
4.17

解：(1)虚拟地址的高 14 位表示的是虚页号，低 17 位表示的是页内偏移地址

故虚页号的高 12 位为 TLB 标记，低 2 位为 TLB 组索引。

(2) 高 5 位为物理页号，低 17 位表示页内偏移地址。

(3)则主存地址划分如下：



4.18

解：(1) A~G 的位数分别为 19、19、11、13、9、9、6

TLB 中标记字段 B 的内容是虚页号，表示该 TLB 项对应哪个虚页的页表项。

(2)映射的 cache 组号为 3，对应的 H 字段内容为 0 0000 1000B。

(3)因为缺页处理需要访问磁盘，而 cache 缺失只要访问主存，cache 缺失带来的开销小，而处理缺页的开销大。

(4)因为采用写穿策略时需要同时写快速存储器和慢速存储器，而写磁盘比写主存慢得多，所以 cache 可以采用写穿策略，而虚存则应采用写回策略。

4.19

解：

(1) 主存地址位数为 28。

(2) TLB 采用全相联映射。TLB 应采用 SRAM，读写速度快，但成本高。

(3)极简实现只需要在 cache 行中增加 1 位 LRU 位用于替换计数；另外每行还需要增加 1 位脏位，如果脏位为 1 则需要写回内存。

cache 总容量为 558 字节。有效位用来指出所在 cache 行中的信息是否有效。

4) 对应物理地址是 0040040H，cache 不命中。虚拟地址 0007 C260H 映射到 cache 第 3 组。

第5章 指令系统

5.2 选择题

- (1) A (2) A **(3) A** (4) C (5) C (6) A (7) D
(8) D (9) D (10) B (11) C (12) A (13) A

5.4

答：(1) 立即数寻址 (2) 直接寻址 (3) 寄存器寻址 (4) 寄存器间接寻址

5.5

解：

- (1) 单操作数指令最多有 512 条指令。
- (2) 双操作数指令最多有 $2^2=4$ 条。
- (3) 直接寻址地址范围为 0~7。
- (4) 变址寻址地址范围为 $2^{16}=64K$ 。

5.6

解：单操作数指令条数 = $(16-m)*64-n/64$ 条。

- (2) 双操作数指令=15 条，单操作数=63 条，无操作数指令=64 条

5.7

解：

- (1) 转移指令第 2 字节为 1FH，第 3 字节为 00H。
- (2) 转移指令第 2 字节为 EBH，第 3 字节为 FFH。

5.8

解：

- (1) EA=D=20H (2) EA=1166H
(3) EA=1258H (4) EA=58H

5.9

解：(1) 可以采用扩展操作码方式，保留 4 个状态用于扩展指令，将操作码扩展到地址字段，只需要占用地址字段 3 位即可表示 32 条新指令。

- (2) 平均长度为=7.04 位。

5.10

解：汇编指令为：add \$s0,\$a1,\$t7

5.11

解:

汇编指令	十六进制机器指令
add \$t3,\$t4,\$t5	018D5820H
mult \$t3,\$t6	016E0018H
mflo \$t3	00005812H
div \$t3,\$t7	016F001AH
mflo \$t3	00005812H

5.12

解:

- (1) 该指令系统最多可有 16 条指令；该机最多有 8 个通用寄存器；
- (2) MDR 和 MAR 至少各需 16 位。
- (3) 转移指令的目标地址范围是 0000H~ FFFFH ($0 \sim 2^{16}-1$)。

(4) 汇编语句 “add (R4), (R5) +”, 对应的机器码为 0010 0011 0001 0101B=2315H。该指令执行后, 寄存器 R5 和存储单元 5678H 的内容会改变。执行后 R5 的内容从 5678H 变成 5679H。存储单元 5678H 中的内容变成该加法指令计算的结果 $5678H + 1234H = 68ACH$ 。

5.13

解:

(1) 按字节编址, offset 采用 8 位补码表示, 所以表示范围位-128~127, 可以向前跳跃 128, 向后跳跃 127 条指令。

(2) 指令中 $C=0$, $Z=1$, $N=1$, 故应根据 ZF 和 NF 的值来判断是否转移。

当 $CF=0$, $ZF=0$, $NF=1$ 时, 需转移。已知指令中偏移量为 1110 0011B=E3H, 符号扩展后为 FFE3 H, 左移一位 (乘 2) 后为 FFC6 H, 故 PC 的值 (即转移目标地址) 为 $200CH + 2 + FFC6H = 1FD4H$ 。

当 $CF=1$, $ZF=0$, $NF=0$ 时不转移, PC 的值为: $200CH + 2 = 200EH$ 。

第6章 中央处理器

6.2 选择题

- (1) B (2) B (3) B (4) D (5) A
(6) B (7) D (8) C (9) C (10) C

6.4

答：(1) a: DR; b: IR c: AR; d: PC

- (2) 取指阶段数据通路: $PC \rightarrow AR \rightarrow \text{主存 } M \rightarrow DR \rightarrow IR$; $PC \rightarrow PC+1$ 。
执行阶段数据通路: $IR(A) \rightarrow AR \rightarrow \text{主存 } M \rightarrow DR \rightarrow AC$

6.6

答：(1) RegWrite 恒 0 故障, lw, sw, addi, add, slt 指令故障。RegWrite 恒 1 故障, beq, j 指令故障。

(2) RegDst 恒 0 故障, add、slt 指令故障, RegDst 恒 1 故障, lw, addi 指令故障。

(3) MemWrite 恒 0 故障, sw 指令故障, MemWrite 恒 1 故障, lw, beq, addi, add, slt, j 指令故障。

6.8

答：

(1) PCSrc 恒 0 故障, beq, j 指令故障。PcSrc 恒 1 故障, 所有顺序执行的指令都无法顺序执行。

(2) MemToReg 恒 0 故障, lw 指令故障。恒 1 故障时, addi, add, slt 故障。

(3) IorD 恒 0 故障, lw 指令会出现故障, 恒 1 故障时, 所有指令都会在取指令阶段发生故障, 无法正常取指令到 IR 寄存器。

6.10

解：

单总线 MIPS 处理器 $CPI=7.6$, $T_{cpu}=1368\text{ s}$

单周期 MIPS 处理器 $CPI=1$, $T_{cpu}=530\text{ s}$

多周期 MIPS 处理器 $CPI=4$, $T_{cpu}=800\text{ s}$

6.11

解：R 型运算指令中计算的结果需要先缓存在 C 寄存器中, 再送寄存器堆写回, 可以直接将 ALU 运算结果送寄存器写回数据端, 这样可以减少一个时钟。 $CPI=3.5$, $T_{cpu}=700\text{ s}$ 。

6.12

解:存储器是瓶颈,减少存储器延迟可以提升性能,但当存储器延迟为 110ps 时,性能优化到达极限,此时成本最低。

6.13

解:寄存器文件关键参数增加一倍,性能会降低 15%,但功耗降低一半,在追求功耗的场所这种方案还是非常有价值的。

6.18

解:

$S4 = addi + slt \quad S3 = beq + sw \quad S2 = addi + beq + lw$

$S1 = addi + slt + beq \quad S0 = slt + sw$

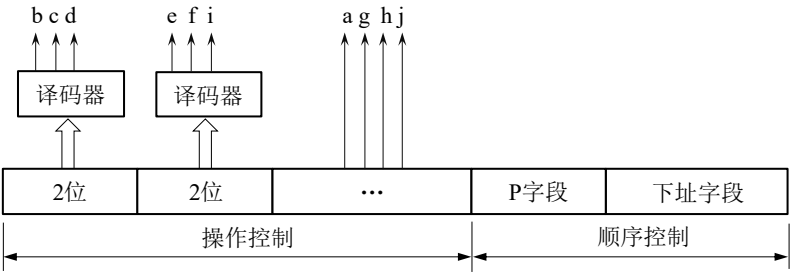
$M_{OP1} = P2 \overline{P1} equal \quad M_{op0} = \overline{P2} P1$

6.20

解: (1) 下址字段为 7 位,判别测试条件 3 位,操作控制字段 22 位。
(2) 参考教材 Page 231 图 6.49。

6.21

答: 可以发现两个互斥组 (b, c, d), (e, f, i), 可以将这两个互斥组采用字段译码法进行编码, 其余的 a,g,h,j 等四个微命令采用直接表示法, 具体如下图所示



6.23

答: 参考答案, 实际还可以有不同的方法。

时钟	功能	有效控制信号
C5	$MAR \leftarrow (R1)$	$R1_{out}, MAR_{in}$
C6	$MDR \leftarrow M(MAR)$ $A \leftarrow (R0)$	$MemR, MDR_{inE},$ $R0_{out}, A_{in}$
C7	$AC \leftarrow (MDR) + (A)$	MDR_{out}, Add, AC_{in}
C8	$MDR \leftarrow (AC)$	AC_{out}, MDR_{in}
C9	$M(MAR) \leftarrow (MDR)$	$MDR_{outE}, MemW$

6.24

解：(1)通用寄存器(R0~R3)和 PC 程序员可见。因为采用了单总线结构，若无暂存器 T，则 ALU 的 A、B 端口会同时获得两个相同的数据，数据通路不能正常工作。增加暂存器后可以使 A，B 端口输入不同的数据。

(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。

(3)信号 SR_{out} 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。

(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。

(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。

(6)因为每条指令的字节长度是 2，顺序寻址时应该实现 (PC)+2 的逻辑，所以这里 MUX 的一个输入端为 2，便于执行(PC)+2 操作。

6.25

解：

(1)寻址方式 1 位，寄存器编号 2 位，操作码位数=16-3*(1+2)=7 位，最多可定义 $2^7=128$ 条指令。

(2)各条指令的机器代码分别如下：

① “inc R1” 的机器码为：0000001 0 01 0 00 0 00，即 0240H。

② “shl R2, R1” 的机器码为：0000010 0 10 0 01 0 00，即 0488H。

③ “sub R3, (R1), R2” 的机器码为：0000011 0 11 1 01 0 10，即 06EAH。

(3)各标号处的控制信号或控制信号取值如下：

①0；②mov；③mova；④left；⑤read；⑥sub；⑦mov；⑧SR_{out}。

(4)指令 “sub R1, R3, (R2)” 的执行阶段至少包含 4 个时钟周期；

指令 “inc R1” 的执行阶段至少包含 2 个时钟周期。

第7章 指令流水线

7.2 选择题

(1) C (2) A (3) D (4) B (5) C (6) A (7) D (8) A (9) C (10) B

7.8

解:

CLKs	取指 IF	译码 ID	执行 EX	访存 MEM	写回 WB
1	addi \$s0, \$s0, 4				
2	lw \$s1, (\$s0)	addi \$s0, \$s0, 4			
3	add \$s2, \$s2, \$s1	lw \$s1, (<u>\$s0</u>)	addi <u>\$s0</u> , \$s0, 4		
4	add \$s2, \$s2, \$s1	lw \$s1, (<u>\$s0</u>)	Bubble	addi <u>\$s0</u> , \$s0, 4	
5	add \$s2, \$s2, \$s1	lw \$s1, (<u>\$s0</u>)	Bubble	Bubble	addi \$s0, \$s0, 4
6	and \$s3, \$s1, \$s2	add \$s2, \$s2, <u>\$s1</u>	lw <u>\$s1</u> , (\$s0)	Bubble	Bubble
7	and \$s3, \$s1, \$s2	add \$s2, \$s2, <u>\$s1</u>	Bubble	lw <u>\$s1</u> , (\$s0)	Bubble
8	sub \$s4, \$s2, \$s2	add \$s2, \$s2, <u>\$s1</u>	Bubble	Bubble	lw <u>\$s1</u> , (\$s0)
9	Next Instr	sub \$s4, \$s2, \$s2	add \$s2, \$s2, \$s1	Bubble	Bubble

7.9

解:

CLKs	取指 IF	译码 ID	执行 EX	访存 MEM	写回 WB
1	addi \$s0, \$s0, 4				
2	lw \$s1, (\$s0)	addi \$s0, \$s0, 4			
3	add \$s2, \$s2, \$s1	lw \$s1, (<u>\$s0</u>)	addi <u>\$s0</u> , \$s0, 4		
4	and \$s3, \$s1, \$s2	add \$s2, \$s2, \$s1	lw \$s1, (<u>\$s0</u>)	addi <u>\$s0</u> , \$s0, 4	
5	and \$s3, \$s1, \$s2	add \$s2, \$s2, \$s1	Bubble	lw \$s1, (<u>\$s0</u>)	addi \$s0, \$s0, 4
6	sub \$s4, \$s2, \$s2	and \$s3, \$s1, \$s2	add \$s2, \$s2, <u>\$s1</u>	Bubble	lw \$s1, (<u>\$s0</u>)
7	Next Instr	sub \$s4, \$s2, \$s2	and \$s3, \$s1, \$s2	add \$s2, \$s2, <u>\$s1</u>	Bubble

7.10

解: 程序执行周期数=508。

7.11

解: 程序执行周期数=310。

7.12

解: 程序执行周期数=80。

7.13

解：程序执行周期数=62。

7.14

解： $T_{min_clk} = \max(T_{if_max}, T_{id_max}, T_{ex_max}, T_{ex_max}, T_{mem_max}, T_{wb_max})$ ，所以应该优化最慢的功能段，这里 ID 段最慢，应该优化其中的寄存器堆读延迟 $T_{RF_read}=150ps$ ，当这个时间延迟优化到 100 时，ID 段时延和 IF、EX、MEM 相同，再进一步优化没有意义，只会增加成本。

7.15

解：（1） $[x]_{\text{补}}=1111\ 1101\ 1111\ 1111B$ ，即指令执行前(R1)=FDFFH，右移 1 位后为 1111 1110 1111 1111B，即指令执行后(R1)=FEFFH。

（2）至少需要 $5+(n-1)=5+(4-1)=8$ 个时钟周期

（3） I_3 的ID段被阻塞的原因：因为 I_3 与 I_1 和 I_2 都存在数据相关，需等到 I_1 和 I_2 将结果写回寄存器后， I_3 才能读寄存器内容，所以 I_3 的ID段被阻塞，流水线中插入了3个气泡。 I_4 的IF段被阻塞的原因：因为 I_4 的前一条指令 I_3 在ID段被阻塞，所以 I_4 的IF段被阻塞。

需要注意的是此题中寄存器读写不能在一个时钟周期内完成，所以ID段和WB段的数据相关也必须插入气泡，相比教材上的方式多插入一个气泡。

（4）因 $2 \times x$ 操作有左移和加法两种实现方法，故 $x=x*2+a$ 对应的指令序列为

```
I1  LOAD    R1, [x]
I2  LOAD    R2, [a]
I3  SHL     R1      //或者ADD    R1, R1
I4  ADD     R1, R2
I5  STORE   R2, [x]
```

这 5 条指令在流水线中执行过程如下图所示。

	时间单元																
指令	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
I1	IF	ID	EX	M	WB												
I2		IF	ID	EX	M	WB											
I3			IF			ID	EX	M	WB								
I4						IF				ID	EX	M	WB				
I5										IF				ID	EX	M	WB

故执行 $x=x*2+a$ 语句最少需要 17 个时钟周期。

7.16

解：

(1)已知计算机 M 采用 32 位定长指令字，一条指令占 4B，观察表中各指令的地址均

为 32 位，且相邻指令的地址差为 4 个地址单位，正好对应指令长度 4B，所以该计算机是按字节编址的。

(2)左移 2 位相当于以乘 4，根据汇编代码可知数组间的数据间隔为 4 个地址单位，而计算机按字节编址，所以数组 A 中每个元素占 4B。

(3)bne 指令的机器代码为 1446FFFAH，根据题目给出的指令格式，后 2B 的内容为 OFFSET 字段，所以该指令的 OFFSET 字段为 FFFAH，用补码表示，值为-6。

当系统执行到 bne 指令时，PC 自动加 4，PC 的内容就为 08048118H，而跳转的目标是 08048100H，两者相差了 18H，即-24 个字节距离， $-24/-6=4$ 。可知 bne 指令的转移目标地址计算公式为 $(PC)+4+OFFSET*4$ 。

(4)由于数据相关而发生阻塞的指令为第 2、3、4、6 条，第 6 条指令会发生控制冒险。当前循环的第 5 条指令与下次循环的第 1 条指令虽然有数据相关，但由于第 6 条指令后有 3 个时钟周期的阻塞，因而消除了该数据相关。

第8章 总线系统

8.3 选择题

- (1) C (2) C (3) B (4) C (5) C
(6) A (7) C (8) A (9) B (10) B

8.4

解：数据传输率=400MB/s，可采用以下3方案：(1) 将总线数据位宽增加一倍；(2) 将时钟频率增加一倍；(3) 每个时钟周期传输2个数据。

8.5

解：(1) 字符传送速率为：480 字符/秒；
(2) 每个数据位的时间为 0.26ms
(3) 3840 位/秒

8.8

解：总线可寻址空间 64K，数据传输率 4MB/S

8.9

解：

读速率=16.7MB/S 写速率=15.4MB/S

8.10

解：设总线频率为 f

- (1) 总线数据传输率=16f bps，存储器数据传输率=10.67f bps。
(2) 总线数据传输率=51.2f bps，存储器数据传输率=28.4f bps。

8.11

解：(1) 每秒总线事务次数=1/225ns=4.44M 次/s
(2) 每秒总线事务次数=1/285ns=3.51M 次/s

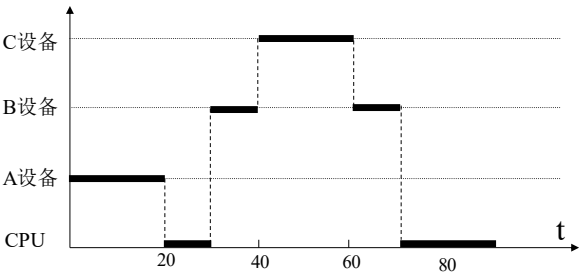
第9章 输入输出系统

9.2 选择题

- (1) D (2) D (3) D (4) A (5) A (6) B
(7) B (8) B (9) C (10) D (11) D (12) B

9.4

解：

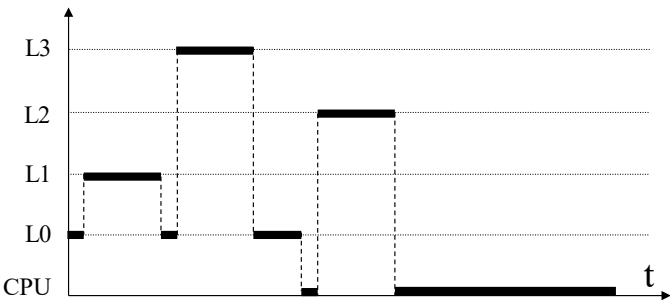


9.5

解：(1) 中断屏蔽字如下表所示：

设备名	中断屏蔽字			
	L0	L1	L2	L3
L0	1	0	1	0
L1	1	1	1	1
L2	0	0	1	0
L3	1	0	1	1

(2) CPU 运行轨迹如图所示：



9.6

解：外设的最大数据传输率为 20KBps，缓冲区为 2Byte，每次中断传输 2Byte。
因此每秒钟产生的中断数为 $20KB/2B=10000$ 次。
每次的执行为 500 个周期。则中断占 CPU 时间的比率为：

$500 \times 10000 / (500 \times 10^6) = 1\%$, 对 CPU 的影响不大, 可以采用中断方式。
若最大数据传输率为 2MBps, 则每秒钟产生的中断数为 $2\text{MB} / 2\text{B} = 10^6$
CPU 占用率为 $(500 \times 10000) / (500 \times 10^6) = 100\%$, 故不能采用中断方式。

9.7

解:

(1) 每传送一个 ASCII 字符, 需要传输的位数有 1 位起始位、7 位 ASCII 数据位、1 位奇校验位和 1 位停止位, 故总位数为 $1+7+1+1=10$ 。

I/O 端口每秒钟最多可接收 $1/0.5 \times 10^{-3} = 2000$ 个字符。

(2) 一个字符传送时间包括: 设备 D 将字符送 I/O 端口的时间、中断响应时间和中断服务程序前 15 条指令的执行时间。时钟周期 $T = 1/(50 \text{ MHz}) = 20 \text{ ns}$, 设备 D 将字符送 I/O 端口的时间为 $0.5 \text{ ms} / 20 \text{ ns} = 2.5 \times 10^4$ 个时钟周期。一个字符的传送时间大约为 $2.5 \times 10^4 + 10 + 15 \times 4 = 25070T$ 。完成 1000 个字符传送所需时间约为 $1000 \times 25070 = 2.507 \times 10^7 T$ 。

CPU 用于该任务的时间大约为 $1000 \times (10 + 20 \times 4) = 9 \times 10^4 T$ 。

在中断响应阶段, CPU 主要进行关中断、保护断点、中断识别操作。

9.8

解:

(1) 设备 A 每隔 $4\text{B} / 2\text{MB} = 2\mu\text{s}$ 就会产生新数据, 为保证数据不丢失, 每隔 $2\mu\text{s}$ 必须查询一次才能保证数据, 每秒的查询次数至少是 $1\text{s} / 2\mu\text{s} = 5 \times 10^5$, 每秒 CPU 用于设备 A 输入/输出的时间至少为 $5 \times 10^5 \times 10 \times 4 = 2 \times 10^7$ 个时钟周期, 占整个 CPU 时间的百分比至少是 $2 \times 10^7 / 500\text{M} = 4\%$ 。

(2) 中断响应和中断处理的时间为 $400 \times (1/500\text{M}) = 0.8\mu\text{s}$, 而 B 设备每隔 $4\text{B} / 40\text{MB} = 0.1\mu\text{s} < 0.8\mu\text{s}$, 设备 B 不适合采用中断 I/O 方式。

(3) 在 DMA 方式中, 只有预处理和后处理需要 CPU 处理, 设备 B 每秒 DMA 传输次数为 $40\text{MB} / 1000\text{B} = 40000$, CPU 用于设备 B 输入输出时间为 $40000 \times 500 = 2 \times 10^7$ 个时钟周期, 占 CPU 总时间的百分比最多为 $2 \times 10^7 / 500\text{M} = 4\%$ 。