第二章

4. 假定机器数为 8 位（1 位符号，7 位数值），写出下列各二进制数的原码和补码表示。
+0.1001，–0.1001，+1.0，–1.0，+0.010100，–0.010100，+0，–0

参考答案：

	原码	补码
+0.1001：	0.1001000	0.1001000
–0.1001：	1.1001000	1.0111000
+1.0：	溢出	溢出
–1.0：	溢出	1.0000000
+0.010100：	0.0101000	0.0101000
–0.010100：	1.0101000	1.1011000
+0：	0.0000000	0.0000000
–0：	1.0000000	0.0000000

5. 假定机器数为 8 位（1 位符号，7 位数值），写出下列各二进制数的补码和移码表示。
+1001，–1001，+1，–1，+10100，–10100，+0，–0

参考答案：

	移码	补码
+1001：	10001001	00001001
–1001：	01110111	11110111
+1：	10000001	00000001
–1：	011111111	11111111
+10100：	10010100	00010100
–10100：	01101100	11101100
+0：	10000000	00000000
–0：	10000000	00000000

6. 已知 $[x]_{补}$，求 x

（1）$[x]_{补}$=1.1100111　　（2）$[x]_{补}$=10000000

（3）$[x]_{补}$=0.1010010　　（4）$[x]_{补}$=11010011

参考答案：

（1）$[x]_{补}$=1.1100111　　x = –0.0011001B

（2）$[x]_{补}$=10000000　　x = –10000000B = –128

（3）$[x]_{补}$=0.1010010　　x = +0.101001B

（4）$[x]_{补}$=11010011　　x = – 101101B = – 45

7. 假定一台 32 位字长的机器中带符号整数用补码表示，浮点数用 IEEE 754 标准表示，寄存器 R1 和 R2 的内容分别为 R1：0000108BH，R2：8080108BH。不同指令对寄存器进行不同的操作，因而，不同指令执行时寄存器内容对应的真值不同。假定执行下列运算指令时，操作数为寄存器 R1 和 R2 的内容，则 R1 和 R2 中操作数的真值分别为多少？

（1） 无符号数加法指令

参考答案：

R1 = 0000108BH = 0000 0000 0000 0000 0001 0000 1000 1011b

R2 = 8080108BH = 1000 0000 1000 0000 0001 0000 1000 1011b

（1）对于无符号数加法指令，R1 和 R2 中是操作数的无符号数表示，因此，其真值分别为 R1：108BH, R2：8080108BH。

11. 下列几种情况所能表示的数的范围是什么？

（1） 16 位无符号整数

（2） 16 位原码定点小数

（3） 16 位补码定点小数

（4） 16 位补码定点整数

（5） 下述格式的浮点数（基数为 2，移码的偏置常数为 128）

参考答案：

（1）无符号整数：$0 \sim 2^{16}-1$。

（2）原码定点小数：$-(1-2^{-15}) \sim +(1-2^{-15})$。

（3）补码定点小数：$-1 \sim +(1-2^{-15})$。

（4）补码定点整数：$-32768 \sim +32767$。

（5）浮点数：负数：$-(1-2^{-7})\times 2^{+127} \sim -2^{-7}\times 2^{-128}$。

正数：$+2^{-135} \sim (1-2^{-7})\times 2^{+127}$。

12. 以 IEEE 754 单精度浮点数格式表示下列十进制数。

+1.75，–1/8

参考答案：

+1.75 = +1.11B = $1.11B \times 2^0$，故阶码为 0+127=01111111B，数符为 0，尾数为 1.110…0，小数点前为隐藏位，所以+1.7 表示为 0 01111111 110 0000 0000 0000 0000 0000，用十六进制表示为 3FE00000H。

–1/8 = – 0.125 = – 0.001B = – 1.0×2^{-3}，阶码为–3+127 = 01111100B，数符为 1，尾数为 1.0…0，所以–1/8 表示为 1 01111100 000 0000 0000 0000 0000 0000，用十六进制表示为 BE000000H。

17. 假定在一个程序中定义了变量 x、y 和 i，其中，x 和 y 是 float 型变量（用 IEEE754 单精度浮点数表示），i 是 16 位 short 型变量（用补码表示）。程序执行到某一时刻，x = –0.125、y=7.5、i=100，它们都被写到了主存（按字节编址），其地址分别是 100，108 和 112。请分别画出在大端机器和小端机器上变量 x、y 和 i 在内存的存放位置。

参考答案：

–0.125 = –0.001B = -1.0×2^{-3}

x 在机器内部的机器数为：1 01111100 00…0 (BE00 0000H)

7.5= +111.1B= $+1.111 \times 2^2$

y 在机器内部的机器数为：0 10000001 11100…0 (40F0 0000H)

100=64+32+4=1100100B

i 在机器内部表示的机器数为：0000 0000 0110 0100（0064H）

	大端机		小端机
地址	内容		内容
100	BEH		00H
101	00H		00H

102	00H	00H
103	00H	BEH
108	40H	00H
109	F0H	00H
110	00H	F0H
111	00H	40H
112	00H	64H
113	64H	00H

第三章

1. 假设某字长为 8 位的计算机中，有两个整数 x 和 y，x=-60，y=-84，采用补码形式（含一位符号位）表示，x 和 y 分别存放在寄存器 A 和 B 中。另外，还有两个寄存器 C 和 D。A、B、C、D 都是 8 位的寄存器。请回答下列问题：（要求最终用十六进制表示二进制序列）

（1）寄存器 A 和 B 中的内容分别是什么？

C4H ACH

（2）x 和 y 相加后的结果存放在 C 寄存器中，寄存器 C 中的内容是什么？结果是否正确？加法器最高的进位 Cout 是什么？溢出标志位 OF 是什么？符号标志位 SF 是什么？零标志 ZF 是什么？

70H　不正确，1，　1，　0, 0

（3）x 和 y 相减后的结果存放在 D 寄存器中，寄存器 D 中的内容是什么？结果是否正确？加法器最高的进位 Cout 是什么？溢出标志位 OF 是什么？符号标志位 SF 是什么？零标志 ZF 是什么？

18H　正确,1, 0, 0, 0

2. 假设某字长为 8 位的计算机中，x 和 y 为无符号整数，x=60，y=84，x 和 y 分别存放在寄存器 A 和 B 中。另外，还有两个寄存器 C 和 D。请回答下列问题：（要求最终用十六进制表示二进制序列）

（1）寄存器 A 和 B 中的内容分别是什么？

3CH 54H

（2）x 和 y 相加后的结果存放在 C 寄存器中，寄存器 C 中的内容是什么？结果是否正确？加法器最高的进位 Cout 是什么？符号标志位 SF 是什么？零标志 ZF 是什么？进位借位标志 CF 是什么？

90H　正确　0, 1, 0, 0

（3）x 和 y 相减后的结果存放在 D 寄存器中，寄存器 D 中的内容是什么？结果是否正确？加法器最高的进位 Cout 是什么？符号标志位 SF 是什么？零标志 ZF 是什么？进位借位标志 CF 是什么？

E8H，正确，　0，　1，　0，　1

3．考虑以下 C 语言程序代码：

```
int     func1(unsigned word)
{
        return    (int) (( word <<24) >> 24);
```

```
}
int     func2(unsigned word)
{
        return    ( (int) word <<24 ) >> 24;
}
```

假设在一个 32 位机器上执行这些函数，该机器使用二进制补码表示带符号整数。无符号数采用逻辑移位，带符号整数采用算术移位。请填写下表，并说明函数 func1 和 func2 的功能。

W		func1(w)		func2(w)	
机器数	值	机器数	值	机器数	值
0000 007FH	127	0000 007FH	+127	0000 007FH	+127
0000 0080H	128	0000 0080H	+128	FFFF FF80H	–128
0000 00FFH	255	0000 00FFH	+255	FFFF FFFFH	–1
0000 0100H	256	0000 0000H	0	0000 0000H	0

函数 func1 的功能是把无符号数高 24 位清零（左移 24 位再逻辑右移 24 位），结果一定是正的有符号数；而函数 func2 的功能是把无符号数的高 24 位都变成和第 25 位一样，因为左移 24 位后进行算术右移，高 24 位补符号位（即第 25 位）。

5. 以下是两段 C 语言代码，函数 arith( )是直接用 C 语言写的，而 optarith( )是对 arith( )函数以某个确定的 M 和 N 编译生成的机器代码反编译生成的。根据 optarith( )，可以推断函数 arith( ) 中 M 和 N 的值各是多少？

```
#define    M
#define    N
int    arith(int x, int y)
{
       int    result = 0 ;
       result = x*M + y/N;
       return result;
}

int optarith ( int x,    int y)
{
       int    t = x;
       x << = 4;
       x - = t;
       if ( y < 0 )    y += 3;
       y>>2;
       return x+y;
```

参考答案：

可以看出 x*M 和“int t = x; x << = 4; x-=t;”三句对应，这些语句实现了 x 乘 15 的功能（左移 4 位相当于乘以 16，然后再减 1），因此，M 等于 15；

y/N 与“if (y < 0) y += 3; y>>2;”两句对应，功能主要由第二句“y 右移 2 位”实现，它实现了 y 除以 4 的功能，因此 N 是 4。而第一句“if (y < 0) y += 3;”主要用于对 y=–1 时进行调整，若不调整，则–1>>2=–1 而–1/4=0，两者不等；调整后 –1+3=2，2>>2=0，两者相等。

思考：能否把　　if (y < 0)　y += 3; 改成 if (y < 0)　y += 2; ？

不能！因为 y = - 4 时不正确。

7．已知 x = 10，y = – 6，采用 6 位机器数表示。请按如下要求计算，并把结果还原成真值。

（1）求$[x+y]_{补}$，$[x-y]_{补}$。

（2）参考答案：

$[10]_{补}$ = 001010　　$[-6]_{补}$ = 111010　　$[6]_{补}$ = 000110　　$[10]_{原}$ = 001010　　$[-6]_{原}$ = 100110

（1）　$[10+(-6)]_{补}$= $[10]_{补}$+$[-6]_{补}$= 001010+111010 = 000100 (+4)

$[10-(-6)]_{补}$= $[10]_{补}$+$[-(-6)]_{补}$ = 001010+000110 = 010000 (+16)

9．在 IEEE 754 浮点数运算中，当结果的尾数出现什么形式时需要进行左规，什么形式时需要进行右规？如何进行左规，如何进行右规？

参考答案：

(1) 对于结果为±1x . xx……x 的情况，需要进行右规。右规时，尾数右移一位，阶码加 1。右规操作可以表示为：$M_b \leftarrow M_b \times 2^{-1}$，$E_b \leftarrow E_b+1$。右规时注意以下两点：

a) 尾数右移时，最高位“1”被移到小数点前一位作为隐藏位，最后一位移出时，要考虑舍入。

b) 阶码加 1 时，直接在末位加 1。

(2) 对于结果为±0.00……01x……x 的情况，需要进行左规。左规时，数值位逐次左移，阶码逐次减 1，直到将第一位“1”移到小数点左边。假定 k 为结果中“±”和左边第一个 1 之间连续 0 的个数，则左规操作可以表示为：$M_b \leftarrow M_b \times 2^k$，$E_b \leftarrow E_b-k$。左规时注意以下两点：

a) 尾数左移时数值部分最左 k 个 0 被移出，因此，相对来说，小数点右移了 k 位。因为进行尾数相加时，默认小数点位置在第一个数值位（即：隐藏位）之后，所以小数点右移 k 位后被移到了第一位 1 后面，这个 1 就是隐藏位。

b) 执行 $E_b \leftarrow E_b-k$ 时，每次都在末位减 1，一共减 k 次。

第四章

PPT 列出来的？ 在主存中建立了一递减空堆栈，堆栈指针 SP 的内容为 300H（H 表十六进制），栈顶内容是 2000H，一条双字长子程序调用指令位于主存地址 1000H 和 1001H 中，1001H 的内容是地址字段，内容为 3000H。试完成：

（1）子程序调用指令读取之前 SP、PC 及栈顶内容。

（2）子程序调用指令执行之后 SP、PC 及栈顶内容，并画出此时堆栈示意图。

（3）从子程序返回主程序之后 SP、PC 及栈顶内容。

解：（1）SP=300H，PC=1000H，栈顶内容：2000H

（2）SP=2FFH，PC=3000H，栈顶内容：1002H

（3）SP=300H，PC=1002H，栈顶内容：2000H

3. 假定某计算机中有一条转移指令，采用相对寻址方式，共占两个字节，第一字节是操作码，第二字节是相对位移量（用补码表示），CPU 每次从内存只能取一个字节。假设执行到某转移指令时 PC 的内容为 200，执行该转移指令后要求转移到 100 开始的一段程序执行，则该转移指令第二字节的内容应该是多少？

参考答案：

因为执行到该转移指令时 PC 为 200，所以说明该转移指令存放在 200 单元开始的两个字节中。因为 CPU 每次从内存只能取一个字节，所以每次取一个字节后 PC 应该加 1。

该转移指令的执行过程为：取 200 单元中的指令操作码并译码→PC+1→取 201 单元的相对位移量→PC+1→计算转移目标地址。假设该转移指令第二字节为 Offset，则 100=200+2+Offset，即 Offset = 100–202 = –102 = 10011010B =9AH

（注：没有说定长指令字，所以不一定是每条指令占 2 个字节。）

4. 假设地址为 1200H 的内存单元中的内容为 12FCH，地址为 12FCH 的内存单元的内容为 38B8H，而 38B8H 单元的内容为 88F9H。说明以下各情况下操作数的有效地址和操作数各是多少？

（1）操作数采用变址寻址，变址寄存器的内容为 12，指令中给出的形式地址为 1200H。

（2）操作数采用一次间接寻址，指令中给出的地址码为 1200H。

（3） 操作数采用寄存器间接寻址，指令中给出的寄存器编号为 8，8 号寄存器的内容为 1200H。

参考答案：

（1） 有效地址 EA=000CH+1200H=120CH，操作数未知。

（2） 有效地址 EA=(1200H)=12FCH，操作数为 38B8H。

（3） 有效地址 EA=1200H，操作数为 12FCH。

6. 某计算机指令系统采用定长指令字格式，指令字长 16 位，每个操作数的地址码长 6 位。指令分二地址、单地址和零地址三类。若二地址指令有 k2 条，无地址指令有 k0 条，则单地址指令最多有多少条？

参考答案：

设单地址指令有 k1 条，则 $((16-k2)\times 2^6-k1)\times 2^6=k0$，所以 $k1=(16-k2)\times 2^6-k0/2^6$

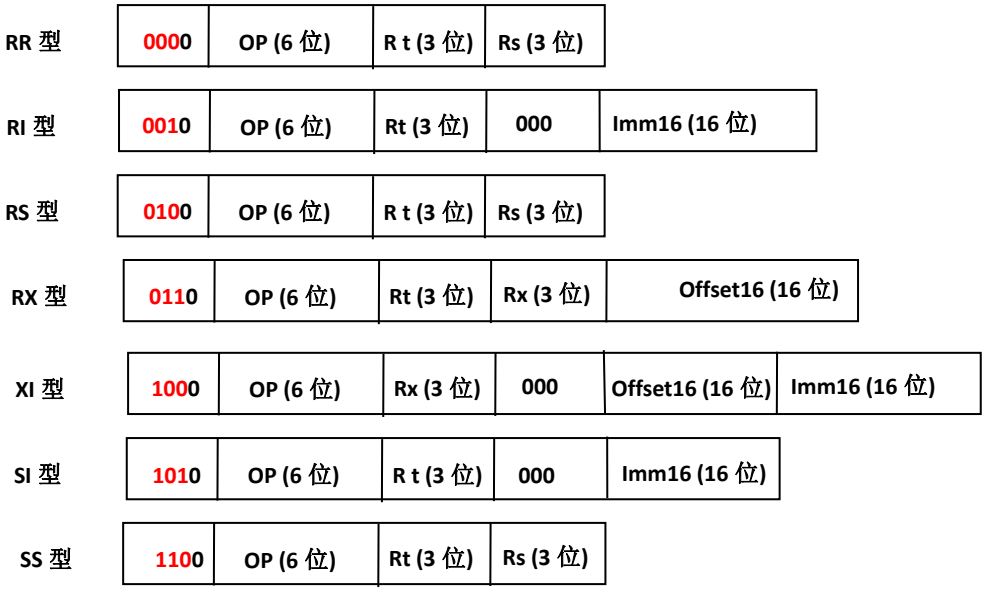
7. 某计算机字长 16 位，每次存储器访问宽度 16 位，CPU 中有 8 个 16 位通用寄存器。现为该机设计指令系统，要求指令长度为字长的整数倍，至多支持 64 种不同操作，每个操作数都支持 4 种寻址方式：立即（I）、寄存器直接（R）、寄存器间接（S）和变址（X），存储器地址位数和立即数均为 16 位，任何一个通用寄存器都可作变址寄存器，支持以下 7 种二地址指令格式（R、I、S、X 代表上述四种寻址方式）：RR 型、RI 型、RS 型、RX 型、XI 型、SI 型、SS 型。请设计该指令系统的 7 种指令格式，给出每种格式的指令长度、各字段所占位数和含义，并说明每种格式指令需要几次存储器访问？

参考答案：

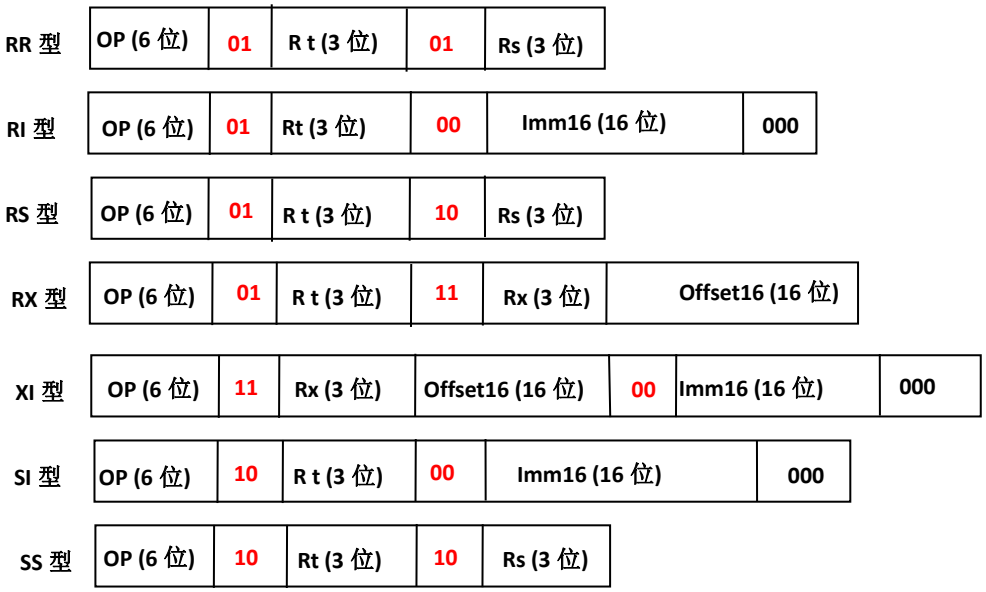
指令格式可以有很多种，只要满足以下的要求即可。

操作码字段：6 位；寄存器编号：3 位；直接地址和立即数：16 位；变址寄存器编号：3 位；总位数是 8 的倍数。

指令格式例 1：



指令格式例 2：



寻址方式字段(2 位)----00：立即；01：寄直；10：寄间；11-变址

第五章

2. 某计算机字长 16 位，采用 16 位定长指令字结构，部分数据通路结构如图 5.8 所示。假设 MAR 的输出一直处于使能状态。加法指令“ADD (R1)，R0”的功能为 M[R[R1]]←M[R[R1]]+R[R0]。

表 5.2 给出了上述指令取指和译码阶段每个节拍(时钟周期)的功能和有效控制信号，

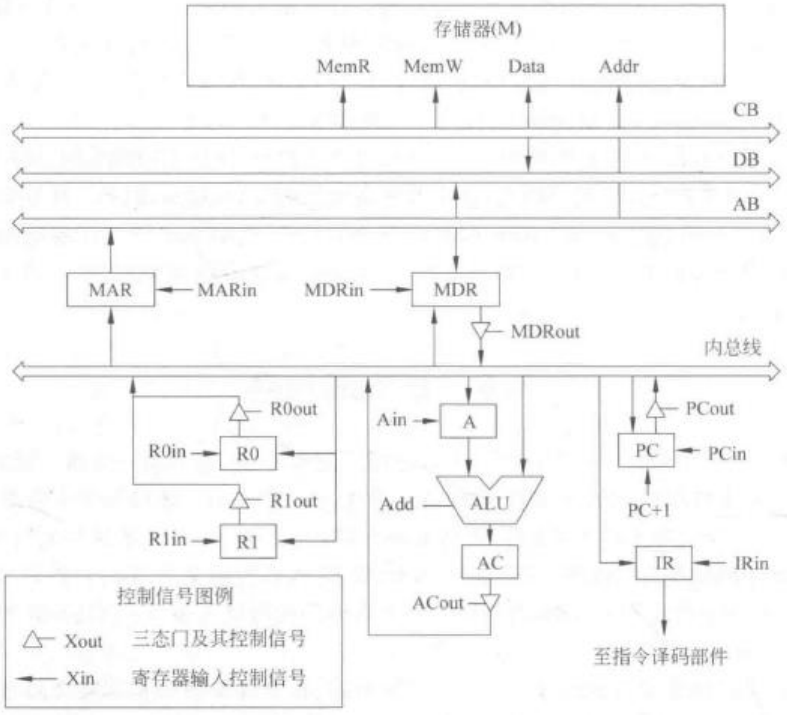


图 5.8 题 2 中的数据通路

请按表中描述方式列出指令执行阶段每个节拍的功能和有效控制信号，并说明需要多少节拍。

表 5.2　题 2 中取指令阶段的控制信号

时钟	功　能	有效控制信号
C1	MAR←(PC)	PCout,MARin
C2	MDR←M(MAR) PC←(PC)+1	MemR PC+1
C3	IR←(MDR)	MDRout,IRin
C4	指令译码	无

【分析解答】

加法指令“ADD(R1),R0”的执行阶段每个节拍的功能和控制信号如表 5.3 所示。

表 5.3　题 2 中指令执行阶段的控制信号

时钟	功　能	有效控制信号
C5	MAR←(R1)	R1out,MARin
C6	MDR←M(MAR) A←(R0)	MemR R0out,Ain

续表

时钟	功　能	有效控制信号
C7	AC←A+(MDR)	MDRout,Add
C8	MDR←(AC)	ACout,MDRin
C9	M(MAR)←MDR	MemW

从表 5.3 可以看出，在 C6 节拍中同时进行了存储器读和寄存器之间传送，这样，该指令的执行阶段共有 5 个节拍。当然，也可将存储器读和寄存器之间传送操作安排在不同的节拍内进行，这样得到表 5.4。此时，执行阶段共有 6 个节拍。

表 5.4　题 2 中指令执行阶段的控制信号

时钟	功　能	有效控制信号
C5	MAR←(R1)	R1out,MARin
C6	MDR←M(MAR)	MemR
C7	A←(MDR)	MDRout,Ain
C8	AC←A + (R0)	R0out,Add
C9	MDR←(AC)	ACout,MDRin
C10	M(MAR)←MDR	MemW

5. 假定某计算机字长16位，CPU内部结构如书中图6.9所示，CPU和存储器之间采用同步方式通信，按字编址。采用定长指令字格式，指令由两个字组成，第一个字指明操作码和寻址方式，第二个字包含立即数Imm16。若一次存储访问所花时间为2个CPU时钟周期，每次存储访问存取一个字，取指令阶段第二次访存将Imm16取到MDR中，请写出下列指

令在指令执行阶段的控制信号序列，并说明需要几个时钟周期。

（1）将立即数Imm16加到寄存器R1中，此时，Imm16为立即操作数。

即：R[R1]←R[R1]+ Imm16

（2）将地址为Imm16的存储单元的内容加到寄存器R1中，此时，Imm16为直接地址。

即：R[R1]←R[R1]+ M[Imm16]

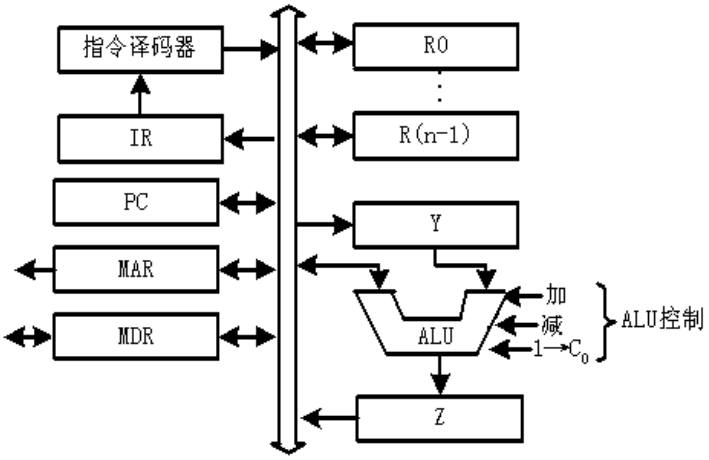
（3）将存储单元Imm16的内容作为地址所指的存储单元的内容加到寄存器R1中。此时，Imm16为间接地址。即：R[R1]←R[R1]+ M[M[Imm16]]

参考答案：

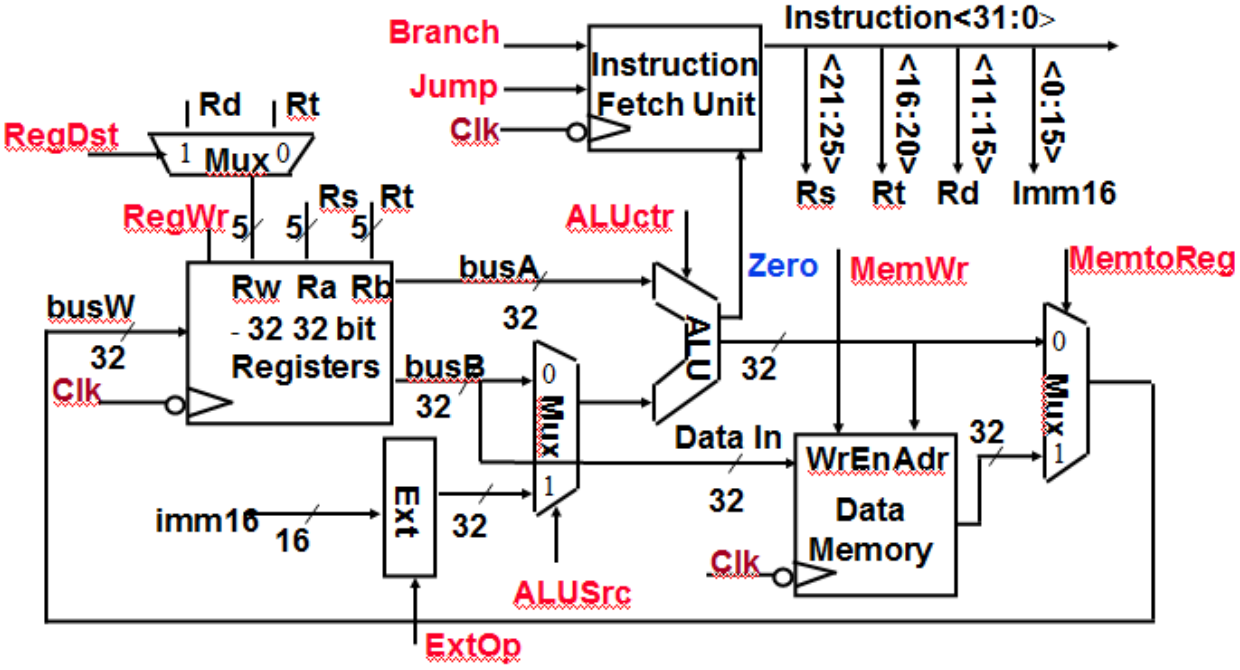
（1） MDRout, Yin
R1out, add, Zin
Zout, R1in
需3个时钟周期

（2） MDRout, MARin
Read1,（R1out, Yin也可以放在该控制信号所在的时钟周期中）
Read2, R1out, Yin
MDRout, add, Zin
Zout, R1in
需5个时钟周期

（3） MDRout, MARin
Read1
Read2
MDRout, MARin
Read1,（R1out, Yin）
Read2, R1out, Yin
MDRout, add, Zin
Zout, R1in
需8个时钟周期



6. 假定图6.24单周期数据通路对应的控制逻辑发生错误，使得在任何情况下控制信号RegWr、RegDst、Branch、MemWr、ExtOp、R-type总是为0，则哪些指令不能正确执行？为什么？



参考答案：

	总是0	总是1
RegWr	则所有需写结果到寄存器的指令（如：R-Type指令、load指令等）都不能正确执行，因为寄存器不发生写操作	不需写结果到寄存器的指令可能会出错（如store,分支,转移指令等）
RegDst	则所有R-Type指令都不能正确执行，因为目的寄存器指定错误	所有非R-Type指令都不能正确执行
Branch	Branch指令可能出错，因为永远不会发生转移	非Branch指令都出错,因为下条指令的地址计算错误
MemWr	Store指令不能正确执行，因为存储器不能写入所需数据	非Store指令都会出错,因为存储器内会写入错误数据
ExtOp	需要符号扩展的指令（如Beq、lw/sw,addiu等）发生错误	必须0扩展的指令会出错(比如ori)

7. 假定图6.24单周期数据通路对应的控制逻辑发生错误，使得在任何情况下控制信号RegWr、RegDst、Branch、MemWr、ExtOp、R-type总是为1，则哪些指令不能正确执行？为什么？

参考答案：见第6题的表格.

15. 微程序控制器容量为1024×48位，微程序可在整个控存内实现转移，反映所有指令执行状态转换的有限状态机中有4个分支点，微指令采用水平格式，微地址由专门的下地址字段确定。请设计微指令的格式，说明各字段有多少位？为什么？

参考答案：

微程序控制器容量为1024×48位，说明微地址占10位，微指令字共48位，其中10位下地址字段用来给出下条微地址；转移控制字段需要对5种情况进行控制，需3位。例如，

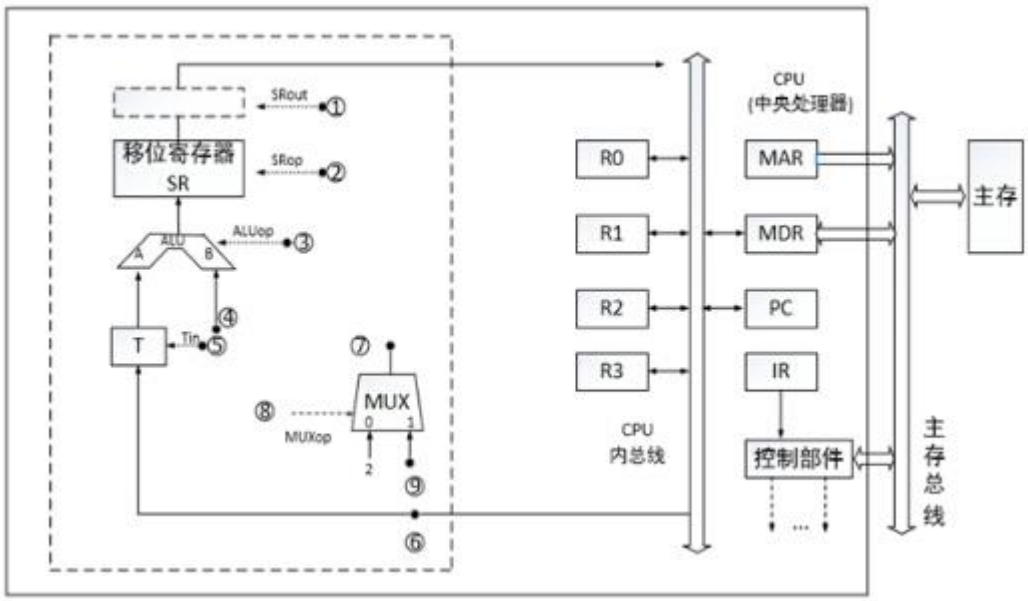
000：下地址字段指出的地址作为下条微地址

100：根据分支1处的条件选择下条微地址
101：根据分支2处的条件选择下条微地址
110：根据分支3处的条件选择下条微地址
111：根据分支4处的条件选择下条微地址
剩下的48-10-3=35位用来表示微操作码字段。
（如果采用计数器法，则转移控制字段需要对6种情况进行控制，比上述5种情况多一种：即顺序执行下条微指令，此时，也要3位。）

也可以用5位作为转移控制字段, 33位作为微操作码字段
00001，00010,00100,01000,10000

15年考研题：

43. (13 分)某 16 位计算机主存按字节编码。存取单位为 16 位;采用 16 位定长指令格式;CPU 采用单总线结构，主要部分如下图所示。图中 R0~R3 为通用寄存器;T 为暂存器;SR 为移位寄存器，可实现直送(mov)、左移一位(left)、右移一位(right)3 种操作，控制信号为 Srop,SR 的输出信号 Srout 控制;ALU 可实现直送 A(mova)、A 加 B(add)、A 减 B(sub)、A 与 B(and)、A 或 B(or)、非 A(not)、A 加 1(inc)7 种操作，控制信号为 ALUop。



请回答下列问题。

(1) 图中哪些寄存器是程序员可见的?为何要设置暂存器 T?

(2) 控制信号 ALUop 和 SRop 的位数至少各是多少?

(3) 控制信号 Srout 所控制邮件的名称或作用是什么?

(4) 端点①~⑨中，哪些端点须连接到控制部件的输出端?

(5) 为完善单总线数据通路，需要在端点①~⑨中相应的端点之间添加必要的连线。写出连线的起点和终点，以正确表示数据的流动方向。

(6) 为什么二路选择器 MUX 的一个输入端是 2?

答案：

(1) 图中程序员可见的寄存器有通用寄存器 R0~R3 和程序计数器 PC;设置暂存器 T 用于暂存数据总线发送的数据。

(2) ALUop 和 SRop 的位数分别为 3,2。

(3) Srout 所控制的部件作用是控制计算机运算结果的输出。

(4) 须连接到控制部件的输出端端点有①②③⑤⑧。

(5) ⑥→⑨，⑦→④。

(6) 使 PC 自增 2 以获取下一条指令地址。

【考查知识点】寄存器相关概念及寄存器的操作，单总线结构

解析： 1） 程序员可见寄存器为通用寄存器（R0～R3）和 PC。因为采用了单总线结构， 因此，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，使数据通路不能正常工作。 【评分说明】回答通用寄存器（R0～R3），给分；回答 PC，给分；部分正确，酌情给分。设置暂存器 T 的原因若回答用于暂时存放端口 A 的数据，则给分，其他答案，酌情给分。

2） ALU 共有 7 种操作，故其操作控制信号 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 SRout 所控制的部件是一个三态门，用于控制移位器与总线之间数据通

路的连接与断开。

【评分说明】只要回答出三态门或者控制连接/断开，即给分。

4） 端口①、②、③、⑤、⑧须连接到控制部件输出端。

【评分说明】答案包含④、⑥、⑦、⑨中任意一个，不给分；答案不全酌情给分。

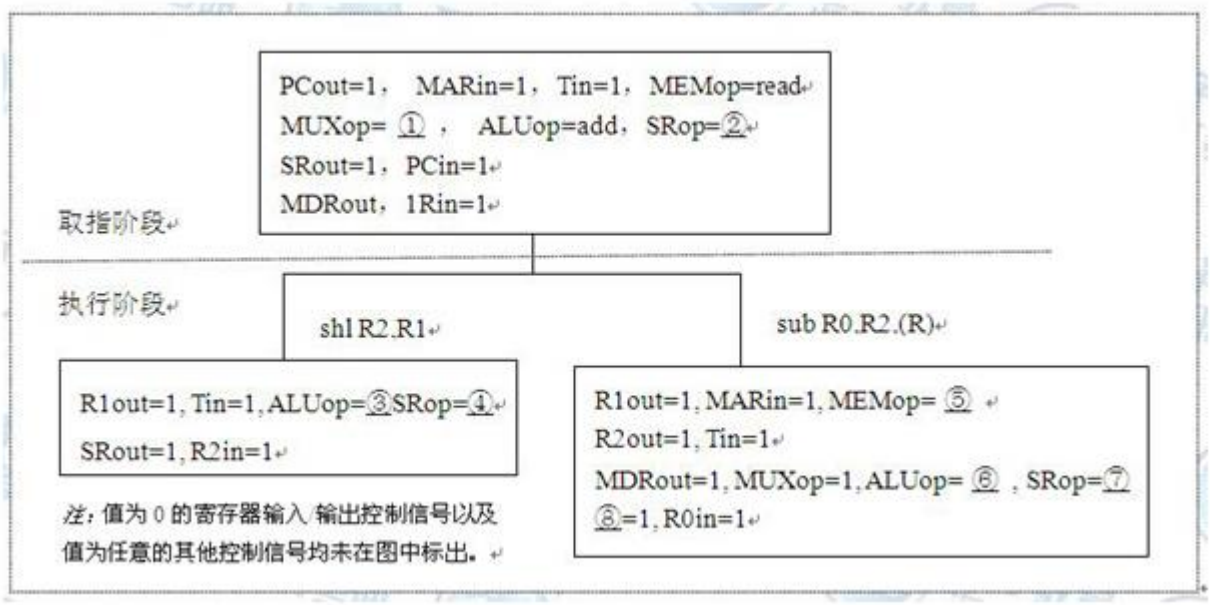
5） 连线 1，⑥→⑨；连线 2，⑦→④。

【评分说明】回答除上述连线以外的其他连线，酌情给分。

6） 因为每条指令的长度为 16 位，按字节编址，所以每条指令占用 2 个内存单

元，顺序执行时，下条指令地址为(PC)+2。MUX 的一个输入端为 2，可便于执行(PC)+2 操作。

44. (10 分)题 43 中描述的计算机，其部分指令执行过程的控制信号如如题 44 图 a 所示。



题 44 图 a 部分指令控制信号

该机指令格式如题 44 图 b 所示，支持寄存器直接和寄存器间接两种寻址方式，寻址方式位分别为 0 和 1，通用寄存器 R0~R3 的编号分别为 0、1、2 和 3。

题 44 图 b 指令格式

请回答下列问题。

(1) 该机的指令系统最多可定义多少条指令?

(2) 假定 inc、shl 和 sub 指令的操作码分别为 01H、02H 和 03H，则以下指令对应的机

器代码各是什么?

① inc R1 ; R1 + 1→R1

② shl R2,R1 ; (R1) << 1→R2

③ sub R3, (R1),R2 ; ((R1)) – (R2) → R3

(3) 假定寄存器 X 的输入和输出控制信号分别为 Xin 和 Xout，其值为 1 表示有效，为 0 表示无效(例如，PCout=1 表示 PC 内容送总线);存储器控制信号为 MEMop，用于控制存储器的读(read)和写(write)操作。写出题 44 图 a 中标号①⑧处的控制信号或控制信号的取值。

(4) 指令“sub R1,R3,(R2)”和“inc R1”的执行阶段至少各需要多少个时钟周期?

解析：
1） 指令操作码有 7 位，因此最多可定义 27
=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；⑧Srout。 【评分说明】答对两个给分。
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。

第七章

4. 用 64K×1 位的 DRAM 芯片构成 256K×8 位的存储器。要求：

（1） 计算所需芯片数，并画出该存储器的逻辑框图。

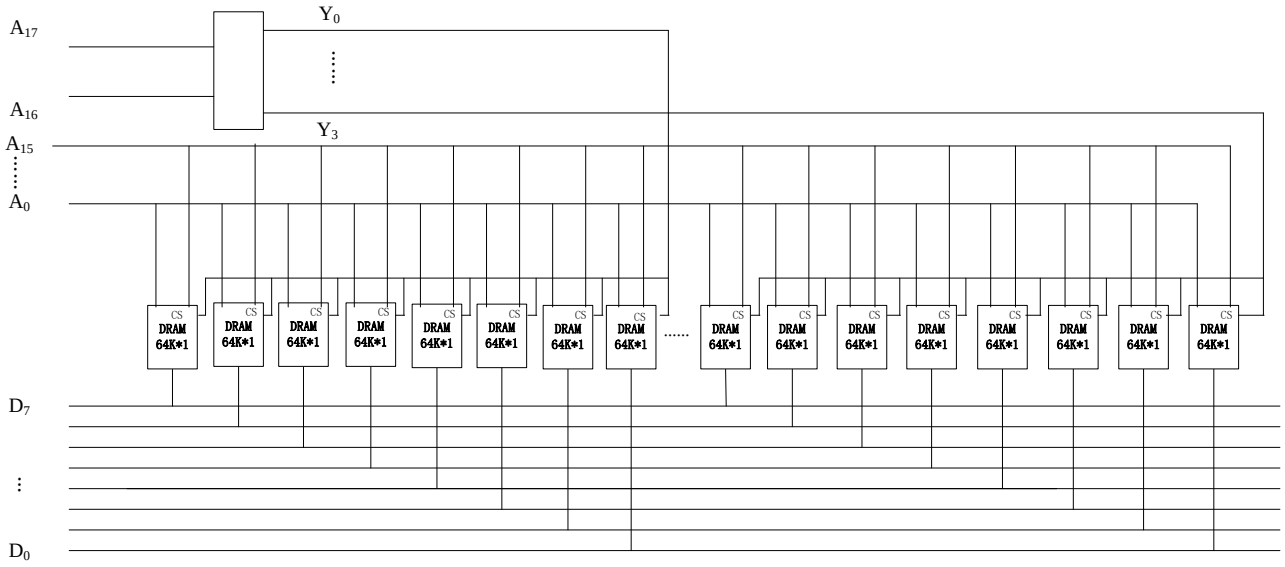
（2） 若采用异步刷新方式，每单元刷新间隔不超过 2ms，则产生刷新信号的间隔是多

少时间？若采用集中刷新方式，则存储器刷新一遍最少用多少读写周期？

参考答案：

（1）256KB / 64K×1 位 = 4×8 = 32 片。存储器逻辑框图见下页（图中片选信号 CS 为高电平有效）。

（2）因为每个单元的刷新间隔为 2ms，所以，采用异步刷新时，在 2ms 内每行必须被刷新一次，且仅被刷新一次。因为 DRAM 芯片存储阵列为 64K=256×256，所以一共有 256 行。因此，存储器控制器必须每隔 2ms/256=7.8μs 产生一次刷新信号。采用集中刷新方式时，整个存储器刷新一遍需要 256 个存储（读写）周期，在这个过程中，存储器不能进行读写操作。



5. 用 8K×8 位的 EPROM 芯片组成 32K×16 位的只读存储器，试问：

（1）数据寄存器最少应有多少位？　　（2）地址寄存器最少应有多少位？

（3） 共需多少个 EPROM 芯片？　　（4） 画出该只读存储器的逻辑框图。

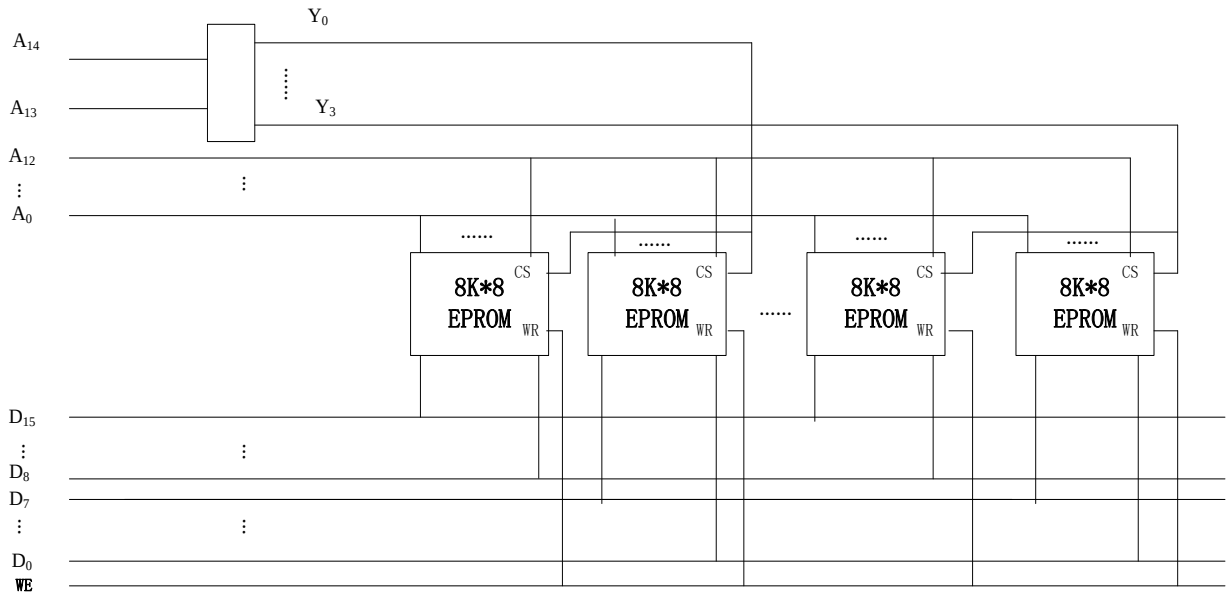
参考答案：

（1）数据寄存器最少有 16 位。

（2）地址寄存器最少有：15 位（若按 16 位的字编址）；16 位（若按字节编址）。

（3）共需要 32K×16 位 / 8K×8 位= 4×2 = 8 片。

（4）该只读存储器的逻辑框图如下（假定按字编址，图中片选信号 CS 为高电平有效）。



10. 假定某机主存空间大小1GB，按字节编址。cache的数据区（即不包括标记、有效位等存储区）有64KB，块大小为128字节，采用直接映射和全写（write-through）方式。请问：

（1）主存地址如何划分？要求说明每个字段的含义、位数和在主存地址中的位置。

（2）cache的总容量为多少位？

参考答案：

（1）主存空间大小为 1GB，按字节编址，说明主存地址为 30 位。cache 共有 64KB/128B=512 行，因此，行索引（行号）为 9 位；块大小 128 字节，说明块内地址为 7 位。因此，30 位主存地址中，高 14 位为标志（Tag）；中间 9 位为行索引；低 7 位为块内地址。

（2）因为采用直接映射，所以cache中无需替换算法所需控制位，全写方式下也无需修改（dirty）位，而标志位和有效位总是必须有的，所以，cache总容量为 512×(128×8+14+1)=519.5K位。

11. 假定某计算机的cache共16行，开始为空，块大小为1个字，采用直接映射方式。CPU执行某程序时，依次访问以下地址序列：2，3，11，16，21，13，64，48，19，11，3，22，4，27，6和11。要求：

（1）说明每次访问是命中还是缺失，试计算访问上述地址序列的命中率。

（2）若 cache 数据区容量不变，而块大小改为 4 个字，则上述地址序列的命中情况又如何？

参考答案

（1） cache 采用直接映射方式，其数据区容量为 16 行×1 字/行=16 字；主存被划分成 1 字/块，所以，主存块号 = 字号。因此，映射公式为：cache 行号 = 主存块号 mod 16 = 字号 mod 16。

开始 cache 为空，所以第一次都是 miss，以下是映射关系（字号-cache 行号）和命中情况。

2-2: miss，3-3: miss，11-11: miss，16-0: miss, 21-5: miss，13-13: miss，64-0: miss、replace，

48-0: miss、replace，19-3: miss、replace，11-11: hit, 3-3: miss、replace，22-6: miss，

4-4: miss，27-11: miss、replace，6-6: miss、replace，11-11: miss、replace。

只有一次命中！

（2）cache 采用直接映射方式，数据区容量不变，为 16 个字，每块大小为 4 个字，所以，cache 共有 4 行；主存被划分为 4 个字/块，所以，主存块号 = [字号/4]。因此，映射公式为：cache 行号 = 主存块号 mod 4 = [字号/4] mod 4。

以下是映射关系（字号-主存块号-cache 行号）和命中情况。

2-0-0: miss，3-0-0: hit，11-2-2: miss，16-4-0: miss、replace，21-5-1、13-3-3: miss，64-16-0、48-12-0、19-4-0: miss, replace，11-2-2: hit，3-0-0: miss、replace，22-5-1: hit，4-1-1: miss、replace，27-6-2: miss、replace，6-1-1: hit，11-2-2: miss、replace。

命中 4 次。

由此可见，块变大后，能有效利用访问的空间局部性，从而使命中率提高！

17. 假设某计算机的主存地址空间大小为64MB，采用字节编址方式。其cache数据区容量为4KB，采用4路组相联映射方式、LRU替换和回写（write back）策略，块大小为64B。请问：

（1）主存地址字段如何划分？要求说明每个字段的含义、位数和在主存地址中的位置。

（2）该cache的总容量有多少位？

（3）若cache初始为空，CPU依次从0号地址单元顺序访问到4344号单元，重复按此序列共访问16次。若cache命中时间为1个时钟周期，缺失损失为10个时钟周期，则CPU访存的平均时间为多少时钟周期？

参考答案：

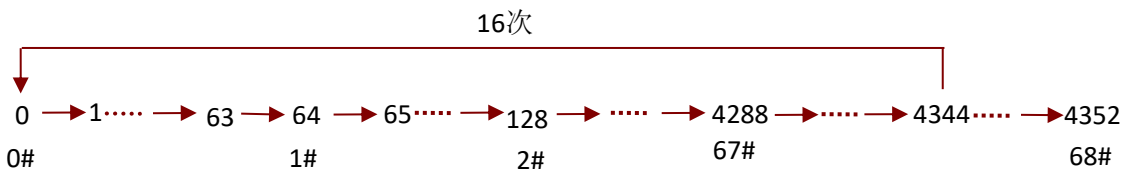
（1）cache 的划分为：4KB = 2^{12}B = 2^4 组×2^2 行/组×2^6 字节/行，所以，cache 组号（组索引）占 4 位。

主存地址划分为三个字段：高 16 位为标志字段、中间 4 位为组号、最低 6 位为块内地址。

即主存空间划分为：64MB = 2^{26}B = 2^{16} 组群×2^4 块/组群×2^6 字节/块

（2）cache 共有 64 行，每行中有 16 位标志、1 位有效位、1 位修改(dirty)位、2 位 LRU 位，以及数据 64B。故总容量为 64×(16+1+1+2+64×8)=34048 位。

（3）因为每块为64B，CPU 访问的单元范围为0～4344，共4345个单元，4345/64=67.89，所以 CPU 访问的是主存前 68 块（第 0～67 块），也即 CPU 的访问过程是对前 68 块连续访问 16 次，总访存次数为 16×4345 = 69520。



cache 共有 16 组，每组 4 行，采用 LRU 算法的替换情况如下图所示：

	第0行	第1行	第2行	第3行
0组	0/64/48	16/0/64	32/16	48/32
1组	1/65/49	17/1/65	33/17	49/33
2组	2/66/50	18/2/66	34/18	50/34
3组	3/67/51	19/3/67	35/19	51/35
4组	4	20	36	52
…	…	…	…	…
…	…	…	…	…
15组	15	31	47	63

根据图中所示可知，第一次循环的每一块只有第一次未命中，其余都命中；以后 15 次循环中，有 20 块的第一字未命中，其余都命中。所以命中率 p 为(69520–68–15×20)/69520 = 99.47%

平均访存时间为：Hit Time + (1–p) × Miss Penalty

=1+10×(1–p) = 1+0.0053×10 = 1.053 个时钟周期