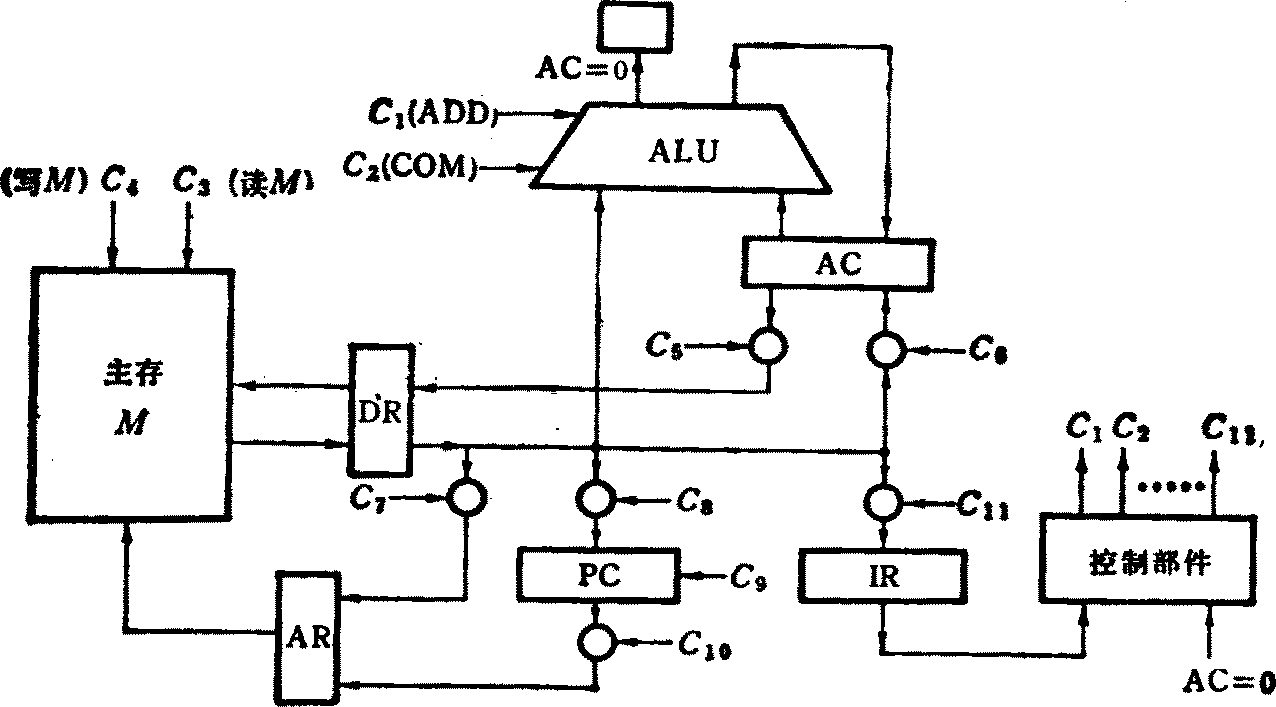
1. CPU的结构如图A19.1示，其中AC为累加器，AR为主存地址寄存器，DR为主存数据寄存器，DR(OP)为DR的操作码字段，DR(ADR)为DR的地址码字段，IR为指令寄存器，PC为程序计数器，M为主存储器，表(1)列出CPU控制信号，表(2)列出指令组助记符及其功能，并给出每条指令的操作码。

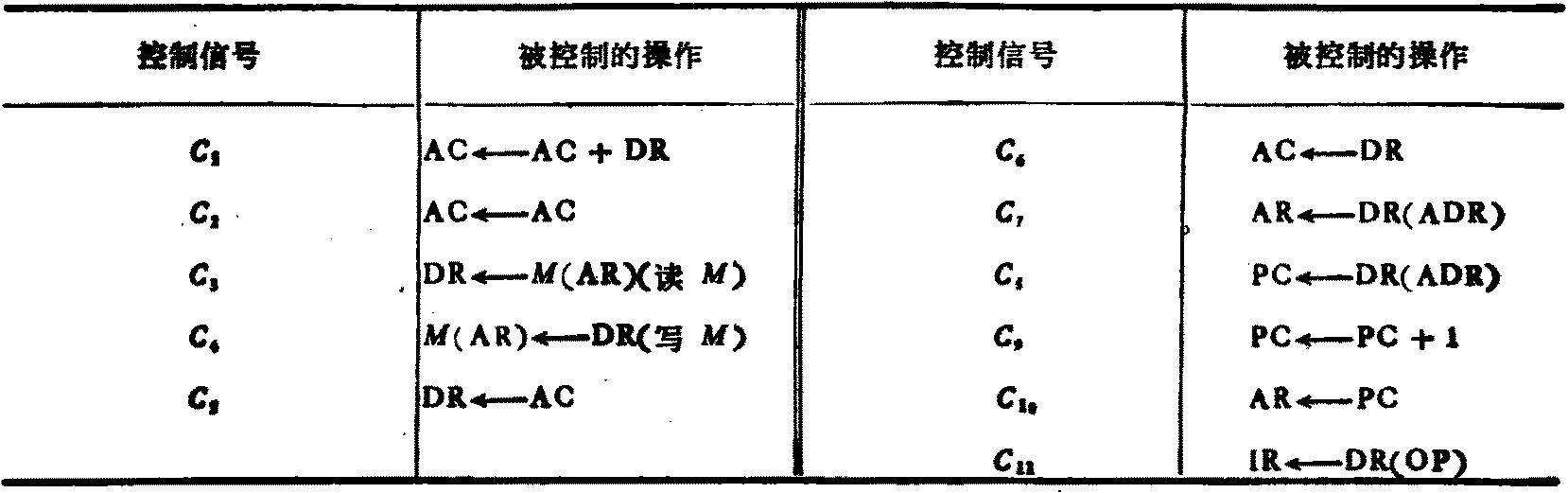
请设计：（1）满足所给条件的微指令格式（直接控制法）。

（2）设计表（2）中6条指令的微程序流程图，标明每条微指令在控存中的地址。

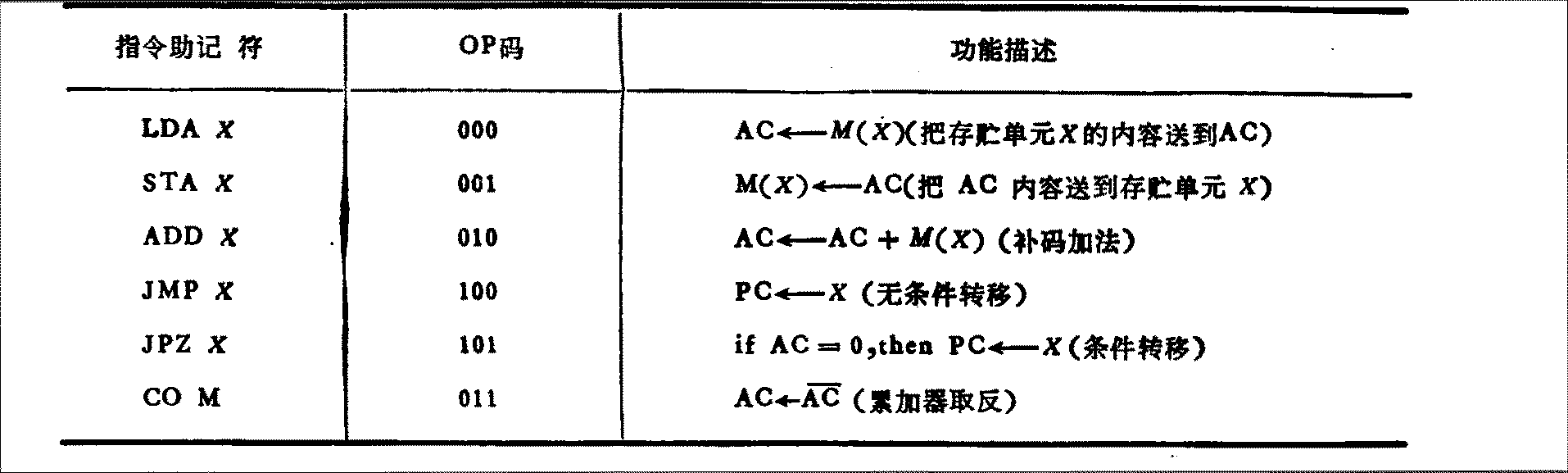


图A19.1

表 (1)



表（2）



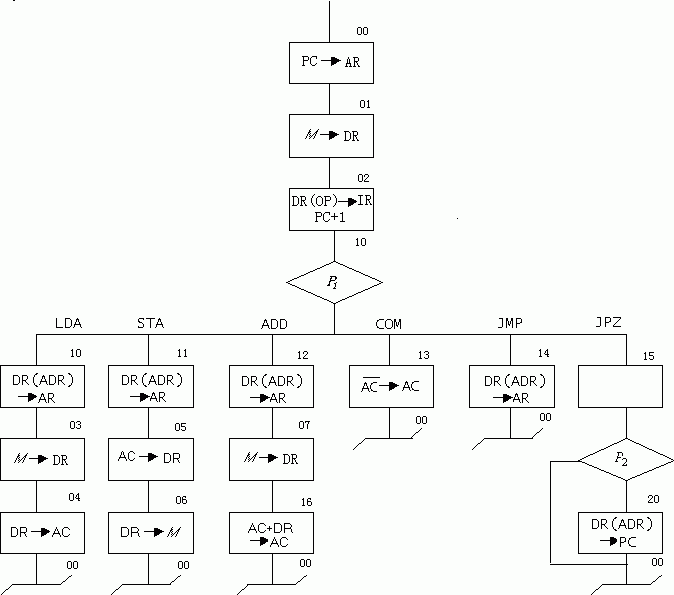
解：(1)根据表(1)，共有１１个控制信号，因此微指令控制字段共由１１位组成。顺序控制采用断定方式，判别测试位２位，下址字段５位。５位地址可提供CM的容量为３２个单元，就是说CM中可存放３２条微指令，这足以实现６条指令的全部微程序。

　　　　　　C1 C2…….C10 C11 P1 P2 μA1 μA2 ……..μA5

|  |  |  |
| --- | --- | --- |
| 操作控制字段 | 判别 | 下址字段 |

　　　　　　　 11　　位　　　　２位　　　　５　位

　　　　(2)六条机器指令的微程序示于图A19.4中，图A19.4中每一框表示一条微指令，右上角注明了该微指令在CM中的地址（八进制表示），其中P1测试时，微程序出现６路分支，测试条件是指令的操作码。根据操作码不同，各个指令的微程序入口地址分别是１0 ,11 ,12, 13, 14, 15。在P2处，根据标志“AC =0”进行测试，如条件满足，微地址修改为２０，否则保持００不变。



图A19.4

2. 某16位机器所使用的指令格式和寻址方式如下所示，该机有两个20位基址寄存器，四个16位变址寄存器，十六个16位通用寄存器，指令汇编格式中的S(源)， D(目标)都是通用寄存器，m是主存的一个单元，三种指令的操作码分别是MOV(OP) =（A） H,STA(OP)=(1B)H, LDA(OP)=(3C)H, MOV是传送指令，STA为写数指令，LDA为读数指令。

15 10 9 8 7 4 3 0

|  |  |  |  |
| --- | --- | --- | --- |
| OP | — | 目标 | 源 |

MOV S , D

15 10 9 8 7 4 3 0

|  |  |  |  |
| --- | --- | --- | --- |
| OP | 基址 | 源 | 变址 |
| 位 移 量 | | | |

STA S, M

15 10 9 8 7 4 3 0

LDA S, M

OP — D

20 位 地 址

要求（1）分析三种指令的指令格式和寻址方式特点

（2）处理机完成哪一种操作所花时间最短？那一种最长？第二种指令的执行时间有时会等于第三种指令的执行时间吗？

（3）下列情况下每个十六进制指令字分别代表什么操作？其中有编码不正确时，如何改正才能成为合法指令？

①(F0F1)H (3CD2)H ② (2856)H ③ (6FD6)H ④ (1C2)H

解**：**(1)第一种指令是单字长二地址指令，RR型；第二种指令是双字长二地址指令RS型，其中S采用基址寻址或变址寻址，R由源寄存器决定；第三种也是双字长二地址指令，RS型，其中R由目标寄存器决定，S由20位地址（直接寻址）决定。

（2）处理器完成第一种指令所花的时间最短，因为是RR型指令，不需要访问存储器。第二种指令所花的时间最长，因为是RS型指令，需要访问存储器，同时要进行寻址方式的变换运算（基址或变址），这也要时间。第二种指令的执行时间不会等于第三种指令，因为第三种指令虽也访问存储器，但节省了求有效地址运算的时间开销。

（3）根据已知条件：MOV(OP) = 0010101 STA(OP) = 011011 LDA(OP) = 111100， 将指令的十六进制格式转换成二进制代码且比较后可知：

①（F0F1）H(3CD2)H 指令代表LDA指令，编码正确，其含义是把主存

（13CD2）H地址单元的内容取至15号寄存器。

②（2856）H代表MOV指令，编码正确，含义是把6号源寄存器的内容传送至6号目标寄存器。

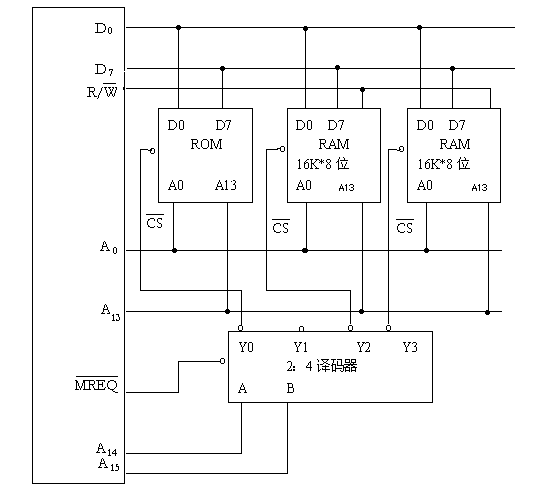
③ (6FD6)H是单字长指令，一定是MOV指令，但编码错误，可改正为(28D6)H

④（1C2）H是编码错误，可改正为（28C2）H,代表MOV指令。

3 某机器中，配有一个ROM芯片，地址空间0000H—3FFFH。现在再用几个16K**×**8的芯片构成一个32K**×**8的RAM区域，便其地址空间为8000H—FFFFH。假设此RAM芯片有CS和WE信号控制端。CPU地址总线为A15—A0,数据总线为D7—D0,控制信号为RD(读)，WR(写)，MREQ(存储器请求)，当且仅当MREQ和RD（或WR）同时有效时，CPU才能对有存储器进行读(或写)，试画出此CPU与上述ROM芯片和RAM芯片的连接图。

解：CPU与芯片连接如图A19.3：

图A19.3



4. 现有四条机器指令，其操作码分别为MOV（OP）=00，ADD（OP）=01，COM（OP）=10，ADT（OP）=11。根据数据通路部分设计的微程序流程图如A18.1所示：（1）第一条是“取指”微指令，功能是将一条机器指令从主存M取出后放到指令寄存器IR；(2)P1测试指令类型，测试条件是指令操作码；(3)P2测试(a+b+c)是否大于9，测试条件是进位标志Cj；（4）公操作中硬件自动执行，程序计数器PC🡪AR (主存地址寄存器)和PC+1动作。

现设定控制存储器EPROM容量最大为16个单元，其字长符合微指令格式要求。

1. 请给微程序流程图中每条微指令分配当前微地址与下一微地址（用二进制）编码给出。
2. 写出微地址转移逻辑表达式。
3. 画出微地址转移逻辑

1

M🡪IR

P1

MOV ADD COM 　　　ADT

rs🡪rd

R2+R1🡪R2

r s+r d🡪r d

r s🡪r d

2 3 4 5

6

R2+R3🡪R2

Cj=1

P2

Cj=0

R2-R3🡪R2

7

图A18.1

解：(1)因EPROM容量为16单元，微地址寄存器4位即可，设为μA3~μA0

七条微指令地址分配如下表所示，一条微指令只占一个微地址，（可直接填写在流程图右上角和右下角）

微指令序号 当前微地址 下一微地址

1 0000 1000

2 1000 0000

3 1001 0000

4 1010 0000

5 1011 1111

6 1111 0000

7 0100 0000

（2）从流程图看出，P1处微程序出现四个分支，对应4个微地址，用OP码作为测试条件。P2处微程序出现2个分支，对应2个微地址

微地址转移逻辑表达式如下：

μA2=P2×CJ×T4

μA1=P1×IR1×T4

μA0=P1×IR0×T4

其中IR1，IR0是指令类寄存器中存放操作码的触发器，T4表示某个节拍脉冲时修改微地址寄存器。

（3）画出逻辑图如图A18.4

Q Q Q Q Q Q Q Q

μA1

D

μA0

D

μA2

D

μA3

D

T1

CM3 CM2 CM1 CM0

T4

P2 P1 P1

Cj IR1 IR0 图A18.4

CM为EPROM读出信号，T1时打入微地址寄存器，而T4时进行修改。

5. 用16K×8位的DRAM芯片构成64K×32位存储器，要求：

（1）画出该存储器的组成逻辑框图

（2）设存储器读/写周期为0.5us，CPU在1us内至少访问一次，试问采用哪种刷新方式比较合理？两次刷新的最大时间间隔是多少？对全部存储单元刷新一遍所需的实际刷新时间是多少？

解：（1）连接图如图A18.3

片内地址线：A13—A0 ；片选信号由A15，A14两位通过2**:**4线译码器给出；

8位数据线D7—D0并接。

A13 A0

CS3 CS2 CS1 CS0

16K\*8

D7—D0

CS0 CS1 CS2 CS3

2 : 4译码器

A14 A15 图 A18.3

(2) 根据已知条件，CPU在1μs内至少需要访存一次，所以整个存储器的平均读/写周期与单个存储器片的读/写周期相差不多，应采用异步式刷新方式比较合理。

DRAM存储器来讲，两次刷新的最大时间间隔是2ms.

DRAM芯片读/写周期为0.5μs。假定16K×1位的RAM芯片由128×128矩阵存储元构成，刷新时只对128行进行异步式刷新，则刷新间隔为2ms/128 =15.6μs，可取刷新信号周期为15μs .

（11分）假设某计算机的运算器框图如图A14.1 所示，其中ALU为16位的加法器（高电平工作），SA,SB为16位锁存器。4个通用寄存器由D触发器组成，Q端输出，其读写控制如下表示：

读控制 写控制

R RA0 RA1 选择 W WA0 WA1 选择

1 0 0 R0 1 0 0 R0

1 0 1 R1 1 0 1 R1

1 1 0 R2 1 1 0 R2

1 1 1 R3 1 1 1 R3

0 × × 不读出 0 × × 不写入

要求：(1)设计微指令格式

(2)画出ADD,SUB两条指令微程序流程图.

/

16位数据总线

ALU

16位/ SB→ALU 16位 SB→ALU

SA

LDSA SB

LDSB CLR

/ 16位

RA0

读选择

R RA1

4个通用寄存器 WA0

写选择

W WA1

图A14.1

解：（1）微指令格式如下所示，各个字段意义如下：

1位 2位 1位 2位 1位 1位 1位 1位 1位 1位

R RA0RA1 W WA0WA1 LDSA LDSB SB→ALU SB→ALU CLR ~ P字段 下地址

R----通用寄存器读命令

W----通用寄存器写命令

RA0RA1----读R0-R3的选择控制

WA0WA1----写R0-R3的选择控制

LDSA----打入SA的控制信号

LDSB----打入SB的控制信号

SB→ALU ----打开非反相三态门的控制信号

SB→ALU ----打开反相三态门的控制信号，并使加法器最低位加1

CLR---清锁存器SB为零

~ ---- 一段微程序结束，转入取机器指令的控制信号

(2 ) ADD指令和SUB指令的微程序流程图如下所示：

取指

测试

SUBADD

R3🡪SA

R0🡪SA

R1🡪 SB

R2🡪SB

SA-SB🡪R3

SA+SB🡪R1

图A14.3

7. 设有两个浮点数x=2Ex×Sx,y=2Ey×Sy,，EX=(-10)2,，Sx=(+0.1001)2,Ey=(+10)2, Sy=(+0.1011)2, 若尾数4位，阶码2位，阶符1位，求x+y=? 并写出运算步骤及结果。

解：因为x+y = 2Ex×(Sx+Sy)(Ex=Ey),所以求x+y要经过对阶、尾数求和及规格化等步骤。

1. 对阶：

△J= Ex-Ey=(-10)2 -(+10)2 =(-100)2 所以Ex＜Ey，则Sx右移4位，Ex+(100)2 =Ey。

Sx右移4位后Sx =0.00001001,经过舍入处理后，Sx =0001，经过对阶、舍入后，x=2(10)2×(0.0001)2。

1. 尾数求和：Sx+Sy

0．0 0 0 1（Sx）

＋ 0．1 0 1 1（Sy）

0．1 1 0 0（Sx+Sy）

结果为规格化数，所以

x+y = 2(10)2×（Sx+Sy）= 2(10)2×(0.1100)2=(11.00)2

8. CPU执行一段程序时，cache完成存取的次数为1900次，主存完成存取的次数为100次，已知cache存取周期为50ns,主存存取周期为250ns.

求：（1）cache/主存系统的效率 （2）平均访问时间

解：（１）命中率H = Nc / (Nc + Nm) = 1900 / (1900 + 100) = 0.95

主存慢于cache的倍率　　r = tm / tc = 250ns / 50ns = 5

访问效率　　e = 1 / [r+(1-r)H] = 1 / [5+(1-5)]**×**0.95 = 83.3%

（２）平均访问时间　ta = tc / e = 50ns / 0.833 = 60 ns

某16位机运算器框图如图A2.1所示，其中ALU为加法器，SA,SB为锁存器，4个通用寄存器的读/写控制信号如下表所示：

读控制 写控制

R RA0 RA1 选择 R RA0 RA1 选择

1 0 0 R0 1 0 0 R0

1 0 1 R1 1 0 1 R1

1 1 0 R2 1 1 0 R2

1 1 1 R3 1 1 1 R3

0 × × 不读出 0 × × 不写入

16位数据总线

ALU

SB→ALU SB→ALU

16位

SA SB CLR

LDSA LDSB

16位

RA0 读选择

R RA1 读选择

4个通用寄存器 WA0 写选择

W WA1 写选择

图A2.1

1. 请设计微指令格式（只考虑控制字段）。

“ADD R0,R1”指令完成（R0）+ (R1) 🡪 R1的操作，画出微程序流程图。

解：(1)微指令格式如下：

1 2位 1 2位 1 1 1 1 1 1

R RA0RA1  W WA0A1  LDSA LDSB SB→ALU SB→ALU CLR ～ P字段 下址字段

其中LDSA,LDSB为锁存器打入信号，

CLR为SB清零信号

SB→ALU为SB送原码控制信号

SB→ALU为SB送反码控制信号

～ 为公操作标志信号

(2)ADD指令的微程序流程图如图A2.4所示

取指

P(1)

ADD

R0 SA

ADD

R1 SB

SA+SB R0

图A2.4

（10分）图A8.1给出了微程序控制的某计算机的部分微指令序列，图中每一框代表一条微指令。分支点a由指令寄存器IR5,IR6两位决定，分支点b由条件码标志Co决定，现采用断定方式实现微程序的顺序控制，已知微地址寄存器长度为8位，要求：

1. 设计实现该微指令序列的微指令字顺序控制字段格式。
2. 画出微地址转移逻辑图。

A

B

IR5·IR6=00 IR5·IR6=01 IR5·IR6=10 IR5·IR6=11

a

E

C0=1 C0=0

b

G

K

O

D

F

C

H

J

I

L

M

N

图A8.1

（１）已知微地址寄存器长度为８位。故推知控存容量为256单元。所给条件

　　　中微程序有两处分支转移。如下不考虑其他分支转移，则需要２位判别测试位P1P2

(直接控制)，故顺序控制字段共１０位，其格式如下，μAi表示微地址寄存器的某

　　　一位。

　　　　　　　　P1 P2 μA1 …. μA8

|  |  |
| --- | --- |
| 判别字段 | 下址字段 |

　　　　(2)微程序在a处有4路转移,用P1测试.在b处有2路转移,用P2测试故转移逻辑

表达式如下：

　　　　　　μA8 = P1×IR6×T4 μA7 = P1×IR5×T4 μA6 = P2×C0×T4

其中T4为节拍脉冲信号。在P1条件下，当IR6 = 1时，T4脉冲到来时微

　　　　　　地址寄存器的第８位μA8将置“１”，从而将该位“０”修改为“１”。

　　　　　　如果IR6 = 0 ,则μA8的“０”状态保持不变。μA7,μA8的修改也类似。

微地址转移逻辑图如图A8.3

Q Q Q Q Q Q

SD ° μA8 SD ° μA7　　　 SD　 ° μA6

D D D

T2

ROM ROM ROM

T4

P1 IR6 P1 IR5 P2 C0

图A8.3

运算器结构如图A10.1示。IR为指令寄存器，R1—R3为三个通用寄存器，其中任何一个可作为源寄存器或目标寄存器,A和B是三选一多路开关，通路的选择分别由AS0,AS1和BS0,BS1控制。（如BS0BS1 = 01选择R1,10选择R2,11选择R3）。S1S2是ALU的操作性质控制端，功能如下：

S1S2 = 00 ALU输出B S1S2 = 01 ALU输出A + B

S1S2 = 10 ALU输出A – B S1S2 = 11 ALU输出B

现有四条机器指令，其操作码OP功能如下：

|  |  |  |
| --- | --- | --- |
| 指令名称 | OP | 指令功能 |
| MOV | 00 | 从源寄存器传送一个数到目标寄存器 |
| ADD | 01 | 源寄存器和目标寄存器内容相加后送目标寄存器 |
| COM | 10 | 源寄存器内容取反后送目标寄存器 |
| ADT | 11 | 十进制加法指令，修正量6在R3.a,b数在R1和R2 |

BUS

CJ

ALU→BUS LDCJ

S1

ALU

S2 +1 R/W

A B 主存M

AS0 BS0

AS1 11 10 01 11 10 01 BS1

R3 R2 R1 R3 R2 R1

R1

R2

R3

IR

LDIR LDR3 LDR2 LDR1

图A10.1

1. 请设计微指令格式。

假定“取指”微指令完成从主存M取指令到IR,画出四条机器指令的微程序流程图。请标出具体微地址和测试标志。

解:(1)假设控存容量为16单元.从总框图看到:控制信号共有12个,CM容量为16单元, 需占用4位下地址字段,判别测试字段需2位.

分析机器指令级的指令格式与A,B两个多路开关的控制方式后发现, AS0, AS1和 BS0,BS1四个控制信号可以直接由机器指令级上的源字段和目标字段控制,但ADT指令例外. 为此微指令中设A,B二个微命令,用以产生AS0, AS1 ,BS0,BS1信号.另外,LDR1—LDR3三个控制信号可以由微指令级提供一个控制信号LDRi,然后与机器指令级上的目标字段进行组合译码后产生

微指令格式如下:

A B S1 S2 +1 ALU-BUS LDRi LDIR P1 P2 μA3-μA0

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |

1 1 2 1 1 1 1 2位 4位

（2）微指令流程如图A10.4所示：

0000

1000

P(1)

MOV ADD COM ADT

1000 1001 1010 1011

R2+R1→R1

rs→rd

rs+rd→rd

rs→rd

r

0000 0000 0000

1111

R2+R3→R2

0000

Cj=1

P(2)

Cj=0

0100

R2-R3→R2

图A10.4 0000

四位运算器框图如图A12.1所示。ALU为算术逻辑单元，A和B为三选一多路开关，预先已通过多路开关A的SW门向寄存器R1、R2送入数据如下，R1=0101，R2=1010，寄存器BR输出端接四个发光二极管进行显示，其运算过程依次如下：显示灯

1. R1（A）+ R2（B）🡪 BR(1010)； （2）R1（A）+ R2（B）🡪 BR(1111)；

(3) R1（A）+ R2（B）🡪 BR(1010)； （4）R1（A）+ R2（B）🡪 BR(1111)；

(5)R1（A）+ BR（B）🡪 BR(1111)； （6）R1（A）+ BR（B）🡪 BR(1010)；

试分析运算器的故障位置与故障性质（“1”故障还是“0”故障），说明理由。

显示灯

BR

LDBR BUS

S3

ALU

74181

S2 M

S1 CS

S0

B

A

AS0  ADC BDC BS0

01 10 11

01 10 11

AS1 BS1

R1 R2 BR R1 R2 BR

LDR1

R1

解：运算器的故障位置在多路开关B，其输出始终为R1的值。

分析如下：

（1） R1（A） + R2（B） = 1010， 输出结果错

（2） R2（A） + R1（B） = 1111， 结果正确，说明R2（A），R1（B）无错

（3） R1（A） + R1（B） = 1010， 结果正确，说明R1（A），R1（B）无错

由此可断定ALU和BR无错

（4）R2（A） + R2（B） =1001， 结果错，由于R2（A）正确且R2（A）=1010

推知R2（B）=0101，显然多路开关B有问题

* + 1. R2（A） + BR（B） =1111，结果错，由于R2（A）=1010，BR（B）=1111，

但现推知BR（B）=0101，证明开关B输出有错

* + 1. R1（A）+BR（B） =1010，结果错，由于R1（A）=0101，本应BR（B）

=1111，但现推知BR（B）=0101，证明开关B输出有错

综上所述，多路开关B输出有错。故障性质：多路开关B输出始终为0101，这有两种可能：一是控制信号BS0，BS1始终为01，故始终选中寄存器R1；二是多路开关B电平输出始终嵌在0101上。