



4、冯.诺依曼型计算机的主要设计思想是什么?它包括哪些主要组成部分?

主要设计思想:

由运算器、控制器、存储器、输入设备、输出设备五大部分构成计算机硬件系统概念结构;

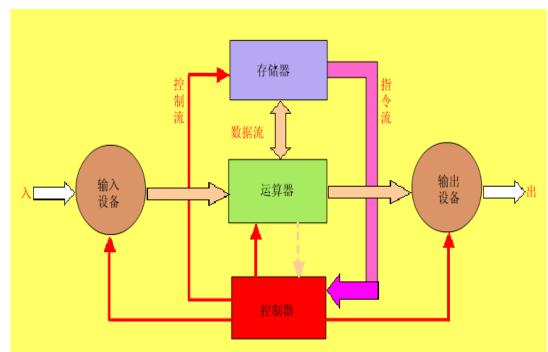
采用二进制代码表示数据和指令;

采用存储程序控制方式(指令驱动)。



4、冯.诺依曼型计算机的主要设计思想是什么?它包括哪些主要组成部分?

组成部分及概念结构:





7、指令和数据均存放在内存中,计算机如何区分他们是指令还是数据?

指令流: 在取指周期中从内存中读出的信息流称为指令流

,它通过总线、CPU内部数据通路流向控制器。

数据流:在执行周期中从内存、I/O设备中读出的信息流称为数据流,它通过总线、CPU内部数据通路流向运算器。



7、指令和数据均存放在内存中,计算机如何区分他们是指令还是数据?

时间上:取指令事件发生在取指周期(取指令阶段),取数据事件发生在执行周期(执行指令阶段);

空间(处理部件)上:指令一定送给控制器,数据一定送给运算器。



1、写出下列各整数的8位原码、补码、反码表示。

(1) -35 (2) 128 (3) -127 (4) -1

| 4 | 二进制真值。 | | 原码。 | 补码。 | 反码。 |
|-------|-------------------|-------------------------|------------|-------------|------------|
| -35₽ | −23H ₂ | -0010 0011 | 1010 0011 | 1101 1101 | 1101 1100₽ |
| 128₽ | +80H | +1000 00004 | 不能表示。 | 不能表示。 | 不能表示。 |
| 127₽ | +7FH₽ | +0111 11114 | 0111 11114 | 0111 11114 | 0111 11114 |
| -14 | −01H _€ | -0000 0001 | 1000 0001 | 1111 11114 | 1111 1110₽ |
| -127₽ | -7FH₽ | -0111 1111 ₀ | 1111 11114 | 1000 0001 | 1000 0000₽ |
| -128₽ | -80H ⁶ | -1000 0000¢³ | 不能表示。 | 1000 000043 | 不能表示。 |



5、已知X和Y,用变形补码计算X+Y,同时指出结果是否溢出。



5、已知X和Y,用变形补码计算X+Y,同时指出结果是否溢出。

X+Y=-10111



9、设阶码3位,尾数6位,按浮点运算方法,完成下列取值的

```
[X+Y], [X-Y]运算。
(1) X=2<sup>-011</sup> * 0.100101, Y=2<sup>-010</sup> * (-0.011110)
```

(2) $X=2^{-101}*$ (-0.010110), $Y=2^{-100}*$ (0.010110)



(1) $X=2^{-0.11} * 0.100101$, $Y=2^{-0.10} * (-0.011110)$

[X+Y]:设阶码、尾数均用补码表示

(1) 对阶: -3 -2

X: 1110 0010010 (1) (右移一位)

(2) 尾数加: 00 010010(1)

+ 11 100010

11 110100 (1) 未溢出、非规格化

(3) 规格化 (2位): 1100 1010010

(4) 无舍入: X+Y = 2-100* (-0.101110)



(1) $X=2^{-0.11} * 0.100101$, $Y=2^{-0.10} * (-0.011110)$

[X-Y]:设阶码、尾数均用补码表示

X: 1 101 0 100101 Y: 1 110 1 100010

(1) 对阶 -3 V: 1 110 0 010010 (1)

X: 1 110 0 010010 (1)

(2) 尾数减:

00 0 1 0 0 1 0 (1) [-Y]补=0 011110

+ 00 0 1 1 1 1 0

00 110000 (1) 未溢出、且是规格化的

(3) 0舍1入: 1110 00 110001 = 2-010* (+0.110001)



(2) $X=2^{-101} * (-0.010110) , Y=2^{-100} * (0.010110)$

[X+Y]: 设阶码、尾数均用补码表示

X: 11 011 11 101010 Y: 11 100 00 010110

(1) 对阶: -5 -4

X: 11 100 11 110101 (0) (右移一位)

(2) 尾数加:

11 110101 (0)

+ 00 010110

100001011(0) 未溢出、非规格化

(3) 规格化 (2位): 1010 00 101100

(4) 无舍入: X+Y = 2⁻¹¹⁰* (0.101100)



(2) $X=2^{-101}* (-0.010110)$, $Y=2^{-100}* (0.010110)$ [X-Y]: 设阶码、尾数均用补码表示 (1) 对阶: -5 X: 11 100 11 110101 (0) (右移一位) (2) 尾数加: 11 1 1 0 1 0 1 (0) + 11 101010 111 011111(0) 未溢出、规格化

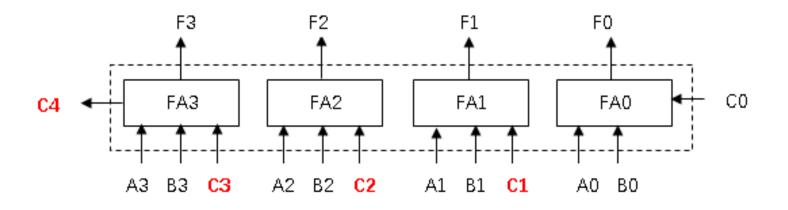
(3) 0舍1入: X+Y = 11 100 11 011111= 2⁻¹⁰⁰* (-0.100001)

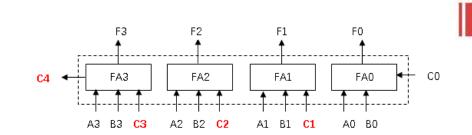


11、某加法器进位链小组信号为C4C3C2C1,低位来的进位信号为C0,请分别按照下述两种方式写出C4C3C2C1的逻辑表达式。

(1) 串行进位方式

(2) 并行进位方式





- (1) 直接从全加器的进位公式推导;
- (2) 设被加数A=A3A2A1A0,加数B=B3B2B1B0,且有:

$$G0 = A0B0$$
 $P0 = A0 + B0$

$$G1 = A1B1$$
 $P1 = A1 + B1$

$$G2=A2B2$$
 $P2=A2+B2$

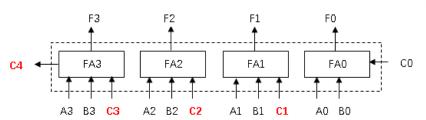
$$G3 = A3B3$$
 $P3 = A3 + B3$

Gi: 进位产生函数, 表示两个数位都为1

Pi: 进位传递函数,表示某位上的两个数位有一个为1, 如果来自低位

的进位为1,则肯定会产生进位。





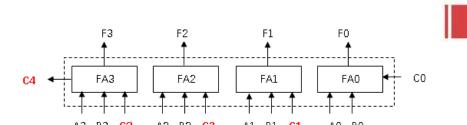
(3) 串行加法器的串行进位方式:某位的运算必须等到下一位的进位传递来以后,才能开始。也就是进位从最低位向最高位逐级传递,速度慢。

$$C1=G0+P0C0$$

$$C2 = G1 + P1C1$$

$$C3 = G2 + P2C2$$

$$C4 = G3 + P3C3$$



(4) 并行加法器的并行进位方式: 所有进位可以同时产生, 实际上只依赖于数位本身、来自最低位的进位C0。

从逻辑表达式可以看出,所有的Ci只与Gi、Pi、C0有关,而Gi、Pi、C0可以从已知条件A、B、C0同时产生,因此所有的Ci可以同时产生、并行进位。



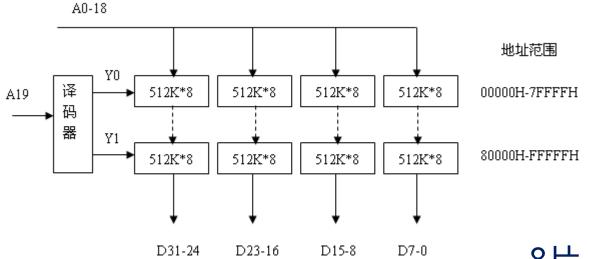
1、设有一个具有20位地址和32位字长的存储器,问:

(1) 该存储器能存储多少个字节的信息?

 $2^{20} \times 32 \text{bits} = 1 \text{M} \times 4 \text{B} = 4 \text{MB}$



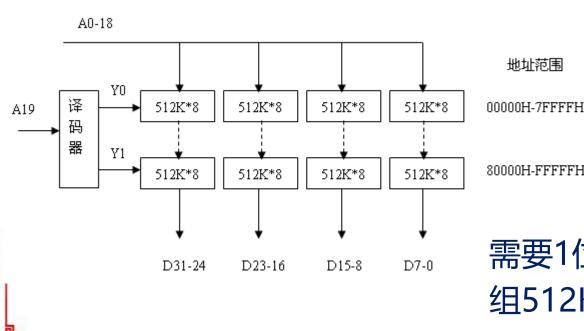
- 1、设有一个具有20位地址和32位字长的存储器,问:
 - (2) 如果存储器由512K*8位SRAM芯片组成,需要多少片?



8片 512K× 8位SRAM



- 1、设有一个具有20位地址和32位字长的存储器,问:
 - (3) 需要多少位地址做芯片选择?



需要1位做芯片选择: A₀₋₁₈接片 组512K的地址线, A₁₉片选。

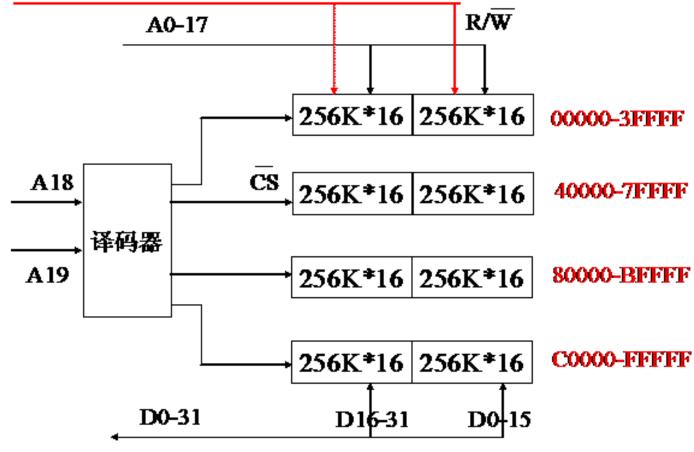


5、要求用256K*16位SRAM芯片设计1024K*32位的存储器。

SRAM芯片有两个控制端: 当CS有效时,该片选中。当R/W=1

时执行读操作,当R/W=0时执行写操作。





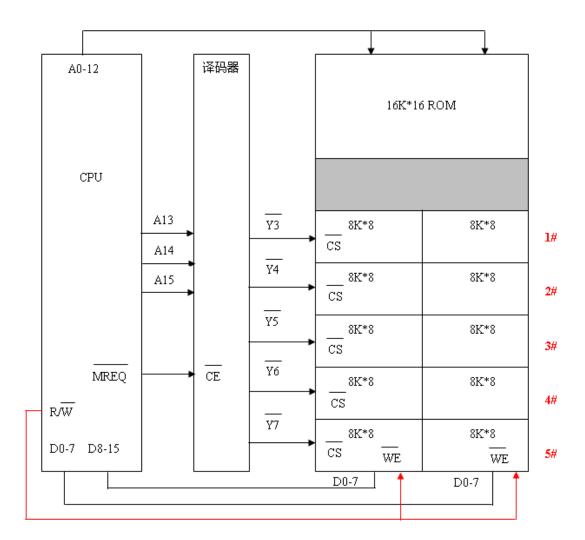


- 7、某机器中,已知配有一个地址空间为0000H--3FFFH的ROM 区域。现在再用一个RAM芯片(8K*8)形成40K*16位的RAM区 域,起始地址为6000H。假设RAM芯片有CS和WE信号控制端。 CPU的地址总线为A15-A0,数据总线为D15-D0,控制信号为 R/W (读/写), MREQ (访存), 要求:
 - (1) 画出地址译码方案。
 - (2) 将ROM与RAM同CPU连接。

| 16K*16 ROM #0 8K | | 0000H | 0000 0000 0000 0000 |
|-------------------------|--------|--------------|-----------------------------------|
| | | 3FFFH | <mark>00</mark> 11 1111 1111 1111 |
| | | 4000H | 0100 0000 0000 0000 |
| | | 5FFFH | 0101 1111 1111 1111 |
| 8K*8 | 8K*8 | 6000H | 0110 0000 0000 0000 |
| RAM | RAM #1 | 7FFFH | <mark>011</mark> 1 1111 1111 1111 |
| 8K*8 | 8K*8 | 8000H | 1000 0000 0000 0000 |
| RAM | RAM #2 | 9FFFH | 1001 1111 1111 1111 |
| 8K*8 | 8K*8 | A000H | 1010 0000 0000 0000 |
| RAM | RAM #3 | BFFFH | 1011 1111 1111 1111 |
| 8K*8 | 8K*8 | C000H | 1100 0000 0000 0000 |
| RAM | RAM #4 | DFFFH | 1101 1111 1111 1111 |
| 8K*8 | 8K*8 | Е000Н | 1110 0000 0000 0000 |
| RAM | RAM #5 | FFFFH | 1111 1111 1111 1111 |



| $\overline{\text{CS1}} = \overline{\text{A15}} \text{ A14 A13} = 011 = \overline{\text{Y3}}$ |
|--|
| $\overline{\text{CS2}} = \text{A15 A14 A13} = 100 = \overline{\text{Y4}}$ |
| $\overline{\text{CS3}} = \text{A15 A14 A13} = 101 = \overline{\text{Y5}}$ |
| CS4 = A15 A14 A13 = 110 = Y6 |
| $\overline{\text{CS5}} = \text{A15 A14 A13} = \text{111} = \overline{\text{Y7}}$ |







8、设存储容量为64M,字长为64位,模块数m=8,分别用顺序和交叉方式进行组织。存储周期T=100ns,数据总线宽度为64位,总线传送周期t=50ns。求:顺序存储器和交叉存储器的带宽各是多少?



[解]

以读出8个字进行考虑,信息总量: q=64*8=512bits 顺序存储器读出8个字的时间: t1=mT=8*100ns=8*10⁻⁷s 交叉存储器流水读8个字时间:

t2=T+(m-1)t=100ns+(8-1)*50ns=450ns=4.5* 10⁻⁷s 顺序存储器的带宽: W1=q/t1=512/(8* 10⁻⁷s)=64* 10⁷bps

交叉存储器的带宽: W2=q/t2=512/(4.5*10⁻⁷s)=114 *10⁷bps



9、CPU执行一段程序时,cache完成存取的总次数为2420次,主存完成存取的总次数为80次,cache存储周期为40ns,主存存储周期为240ns,求cache/主存系统的效率和平均访问时间。

[解]

```
h=Nc/(Nc+Nm)=2420/(2420+80)=2420/2500=0.968

r=tm/tc=240/40=6

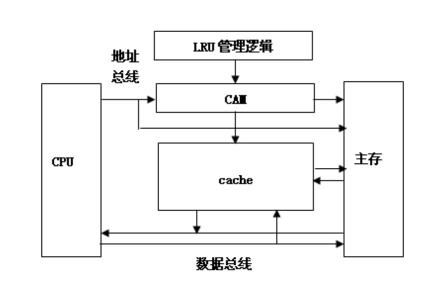
e=1/(r+(1-r)h)=1/(6+(1-6)0.968)=84.7\%

ta=tc/e=40/0.847=47ns
```



[补1] 高速缓冲存储器cache原理图如下图所示,请说明高速缓冲存储器cache的工作原理。

- (1) 主存和cache均按照约定长度 划分为若干块,存储单元物理地址=块 地址(块编号)+ 块内地址;
- (2) 主存中一个存储块调入(交换) 到cache中,则将存储块地址(块编号) 存放到相联存储表CAM中,将数据块 内容存放在cache中;

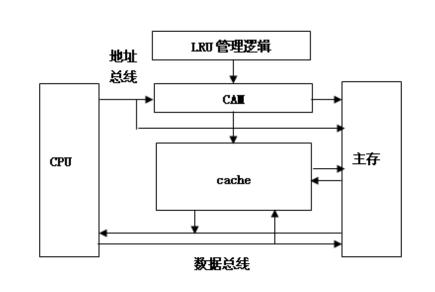




[补1] 高速缓冲存储器cache原理图如下图所示,请说明高速缓冲存储器cache的工作原理。

(3) 当CPU访问主存时输出物理地址,然后根据物理地址的高位部分(块地址、块编号),判断是否在相联存储器CAM中?

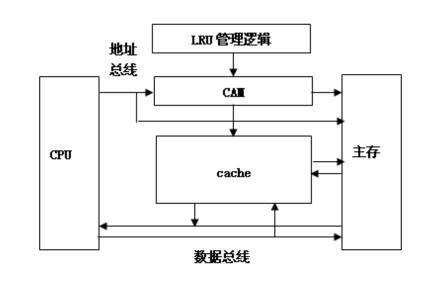
若在,则命中,直接访问cache;若不在,则未命中,直接访问主存,并将该单元所在存储块交换到cache中,后续对该存储块的访问则在cache命中。





[补1] 高速缓冲存储器cache原理图如下图所示,请说明高速缓冲存储器cache的工作原理。

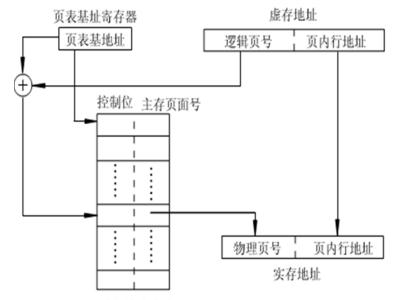
- (4) 通过cache和主存之间的存储 块动态交换,尽量争取CPU访存操作在 cache命中,从而总体提高访存速度;
- (5) 基于程序和数据的局部性访问原理, cache实际上是主存的当前最活跃部分,即主存的一个子集。





[补2] 请阐述页式虚拟存储器的原理及地址变换过程。

- (1) 将物理地址空间、虚拟地址空间均按照约定长度划分为若干页,主存和外存之间的调入调出以页为基本单位。
- (2) 物理地址空间的页称为物理页(实
- 页) , 物理地址 = 实页号 + 页内地址。
 - (3) 虚拟地址空间的页称为逻辑页 (虚
- 页),虚拟地址 = 虚页号 + 页内地址。



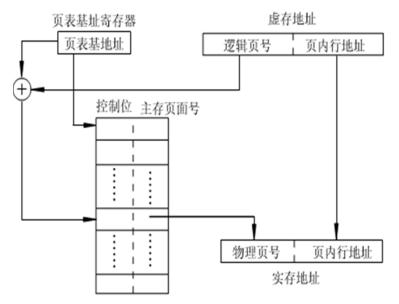
页表(在主存中) 页表构成=[虚页号]+ 控制位 + 实页号



[补2] 请阐述页式虚拟存储器的原理及地址变换过程。

(4) 页表:由"虚页号+装载标志+实页号+修改标志位+……"构成,每个虚页在页表中占一行。

当某个虚页交换到内存时,在页表中对应的虚页行上记载实页号、装载标志等信息。



页表(在主存中) 页表构成=[虚页号]+ 控制位 + 实页号

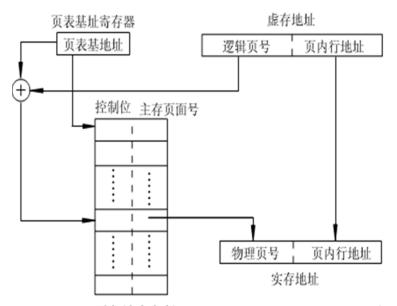


[补2] 请阐述页式虚拟存储器的原理及地址变换过程。

(5) 查表法: 当CPU访问主存时,根据虚拟地址中的虚页号查页表。

若已装入,则取实页号,物理地址 =实页号+页内地址;

若未装入,则产生缺页中断,从外 存调入页到内存。



页表(在主存中) 页表构成=[虚页号]+ 控制位 + 实页号



[补3] 设CPU的地址总线20根(A0-15, A0为低位), 双向数据 总线16根(D0-15),控制总线中与主存有关的信号有MREQ(允许 访存, 低电平有效), R/W(高电平为读命令, 低电平为写命令)。 现有如下SRAM存储器芯片: 8K×8位, 16K×8位, 32K×8位 ,64K×8位,128K×8位(控制端有CS、RD、WR)。 请从上述芯片中选择适当芯片设计该计算机主存储器,画出 主存储器逻辑框图,注意画出选片逻辑(可选用门电路及3:8译码 器74LS138)与CPU 的连接,说明选哪些芯片,选多少片。

第三章

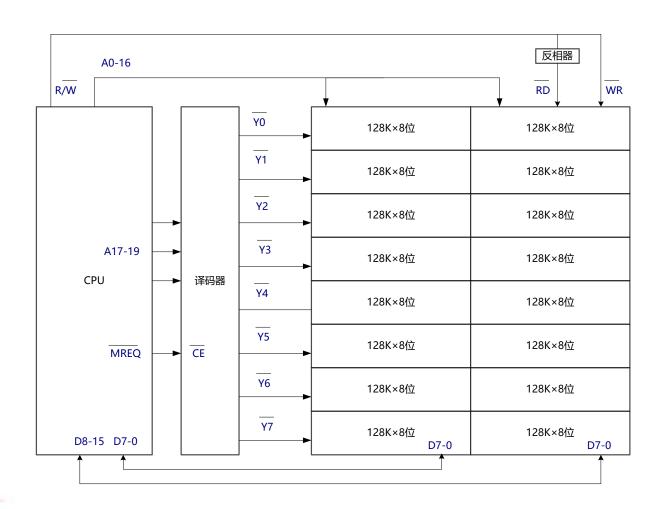


| 128K×8位 | 128K×8位 |
|-----------------|---------|
| 128K×8位 | 128K×8位 |
| 128K×8 <u>位</u> | 128K×8位 |
| 128K×8 <u>位</u> | 128K×8位 |

| 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 00000H |
|--------------|--------------|--------------|--------------|--------------|--------------|------------------|
| 0001 0010 | 1111 0000 | 1111 0000 | 1111 0000 | 1111 0000 | 1111 0000 | 1FFFFH 20000H |
| 0011 0100 | 1111 0000 | | | 1111 0000 | 1111 0000 | 3FFFFH 40000H |
| 0101 0110 | 1111 0000 | | 1111 0000 | | 1111 0000 | 5FFFFH 60000H |
| 0111 1000 | 1111 0000 | | | 1111 0000 | 1111 0000 | 7FFFH 80000H |
| 1001 1010 | 1111 0000 | 1111 0000 | 1111 0000 | 1111 0000 | 1111 0000 | 9FFFFH A0000H |
| 1011 1100 | 1111 0000 | 1111 0000 | 1111 0000 | 1111 0000 | 1111 0000 | BFFFFH C0000H |
| 1101 1110 | 1111 0000 | 1111 0000 | 1111 0000 | 1111 0000 | 1111 0000 | DFFFFH E0000H |
| 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | FFFFFH |

第三章







3、指令格式结构如下所示,试分析指令格式及寻址方式特点。

| 15 | 10 | | 7 4 | 3 | 0 |
|----|----|------------|-------|---|------|
| OP | | — - | 目标寄存器 | | 源寄存器 |

- (1) 操作码 操作码字段6位,可64种操作;定长操作码字段。
- (2) 操作数 (地址码)

双操作数指令;源、目的操作数都在通用寄存器中(可分别指定2⁴=16个寄存器),寄存器寻址方式;RR型指令。

(3) 指令字长度 单字长;等长指令字结构。



4、指令格式结构如下所示, 试分析指令格式及寻址方式特点。



- (1) 操作码
 - 操作码字段6位,可64种操作;定长操作码字段。
- (2) 操作数 (地址码)

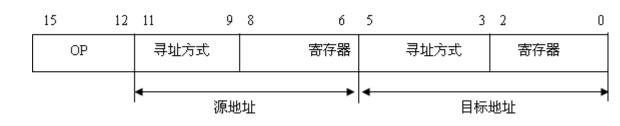
双操作数指令;源操作数在通用寄存器中(可分别指定24=16个寄存器)

- ,寄存器寻址方式;目的操作数在内存单元中,变址寻址方式(变址寄存器
- +16位偏移量); RS型指令。
 - (3) 指令字长度

双字长;等长指令字结构。



5、指令格式结构如下所示, 试分析指令格式及寻址方式特点。



(1) 操作码

操作码字段4位,可16种操作;定长操作码字段。

(2) 操作数(地址码)

双操作数指令;源、目的操作数都是由寻址方式字段、寄存器字段构成,可以分别构成2³=8种寻址方式;双操作数的类型不确定,可能RR、RS、SS型指令。

(3) 指令字长度 指令字长度不确定;指令字结构不确定。



6、一种单地址指令如下所示,其中I为间接特征,X为寻址模式,D为形式地址。I、X、D组成该指令的操作数有效地址E。设R为变址寄存器,R1为基址寄存器,PC为程序计数器,请在下表中第一列位置填入适当的寻址方式名称。

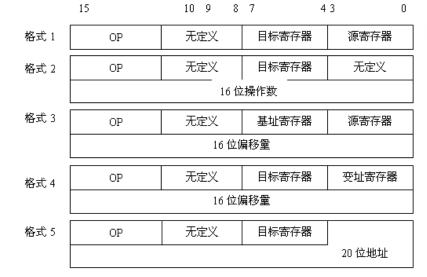
| ОР | I | X | D | | |
|-----------|---|----|-----------------------------------|--|--|
| 寻址方式名称 | I | Х | 有效地址E | | |
| 直接寻址 | 0 | 0 | E=D | | |
| 相对寻址 | 0 | 1 | E= (PC) +D | | |
| 变址寻址 | 0 | 10 | E=(R)+D | | |
| 基址寻址 | 0 | 11 | E= (R1) +D | | |
| 间接寻址 | 1 | 0 | E= (D) | | |
| 基址寻址+间接寻址 | 1 | 1 | E= ((R1) +D) D=0 即E= ((R1)) | | |



[补]某16位机器内存地址空间1M,所使用的指令格式如下图所示,要求:

- (1) 分析五种指令的指令格式与寻址方式特点;
- (2) 哪种指令执行时间最长? 为什么?

| | 15 | 10 9 8 | 7 4 | 3 0 | | | | | |
|------|---------|--------|-------|--------|--|--|--|--|--|
| 格式 1 | OP | 无定义 | 目标寄存器 | 源寄存器 | | | | | |
| 格式 2 | OP | 无定义 | 目标寄存器 | 无定义 | | | | | |
| | | 16 位 | 操作数 | | | | | | |
| 格式 3 | OP | 无定义 | 基址寄存器 | 源寄存器 | | | | | |
| | 16 位偏移量 | | | | | | | | |
| 格式 4 | OP | 无定义 | 目标寄存器 | 变址寄存器 | | | | | |
| | 16 位偏移量 | | | | | | | | |
| 格式 5 | OP | 无定义 | 目标寄存器 | | | | | | |
| | | | | 20 位地址 | | | | | |



- 一、第一种指令
 - (1) 操作码

操作码字段6位,可64种操作;定长操作码字段。

(2) 操作数 (地址码)

双操作数指令;源操作数、目的操作数均在通用寄存器中(可分别指定

- 24=16个寄存器),寄存器寻址; RR型指令。
 - (3) 指令字长度

单字长;变长指令字结构。



- 一、第二种指令
 - (1) 操作码

操作码字段6位,可64种操作;定长操作码字段。

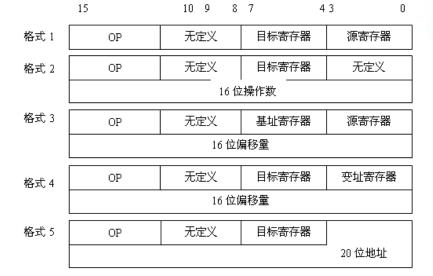
(2) 操作数 (地址码)

双操作数指令;源操作数是立即数,在内存单元中,立即寻址;目的操作数

|在通用寄存器中(可指定24=16个寄存器),寄存器寻址;RS型指令。

(3) 指令字长度

双字长;变长指令字结构。



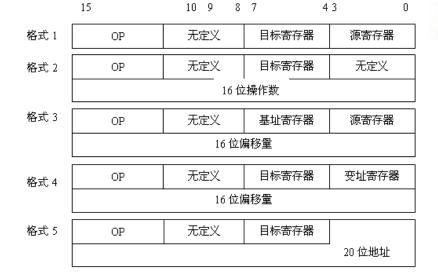
- 一、第三种指令
 - (1) 操作码

操作码字段6位,可64种操作;定长操作码字段。

(2) 操作数 (地址码)

双操作数指令;源操作数在通用寄存器中(可指定2⁴=16个寄存器),寄存器寻址;目的操作数在内存单元中,基址寻址(基址寄存器+16位偏移量); RS型指令。

(3) 指令字长度: 双字长; 变长指令字结构。



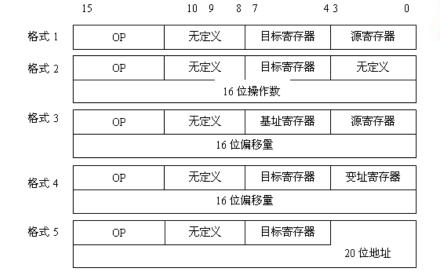
- 一、第四种指令
 - (1) 操作码

操作码字段6位,可64种操作;定长操作码字段。

(2) 操作数 (地址码)

双操作数指令;源操作数在内存单元中,变址寻址(变址寄存器+16位偏移

- 量);目的操作数在通用寄存器中(可指定2⁴=16个寄存器),寄存器寻址; RS型指令。
 - (3) 指令字长度: 双字长; 变长指令字结构。



- 一、第五种指令
 - (1) 操作码

操作码字段6位,可64种操作;定长操作码字段。

(2) 操作数 (地址码)

双操作数指令;源操作数在内存单元中,直接寻址(20位直接地址随指令一起保存);目的操作数在通用寄存器中(可指定24=16个寄存器),寄存器寻址;RS型指令。

(3) 指令字长度:双字长;变长指令字结构。





二、执行时间比较

第三种指令执行时间最长,因为(1)计算有效地址地址:基址寄存器内容

+16位偏移量; (2) 目的操作数是基址寻址,保存结果还需要访问内存。

第一种指令执行时间最短。格式1<格式2 <格式5 <格式4 <格式3。



2、参见图5.15的数据通路。画出存数指令"STA R1, (R2)"的指令周期流程图,其含义是寄存器R1的内容传送到(R2)为地址的主存单元中。标出各微操作信号序列。

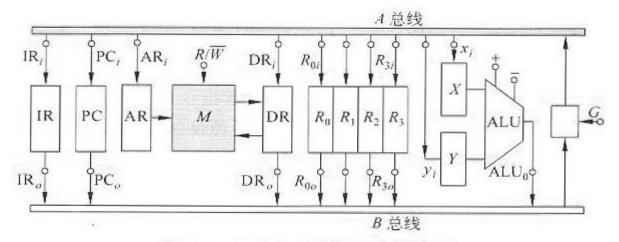


图 5.15 双总线结构机器的数据通路

STA R1, (R2)

 $PC \rightarrow AR$ PCo, G, ARi

 $M \rightarrow DR R/\overline{W}=R$

DR → IR DRo, G, IRi

 $R2 \rightarrow AR$ R2o, G, ARi

 $R1 \rightarrow DR$ R1o, G, DRi

 $DR \rightarrow M R/W=W$

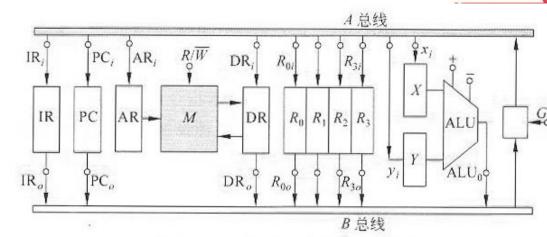


图 5.15 双总线结构机器的数据通路



3、参见图5.15的数据通路。画出取数指令"LDA (R3), R0"的指令周期流程图, 其含义是将(R3)为地址主存单元的内容取至寄存器R0中。标出各微操作信号序列。

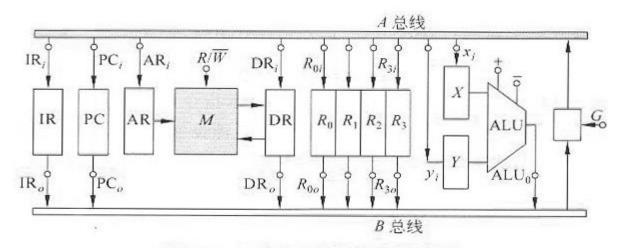


图 5.15 双总线结构机器的数据通路

LDA (R3), R0

$$PC \rightarrow AR$$
 PCo, G, ARi

$$M \rightarrow DR R/\overline{W}=R$$

$$R3 \rightarrow AR$$
 R3o, G, ARi

$$M \rightarrow DR R/W=R$$

$$DR \rightarrow R0$$
 DRo, G, R0i

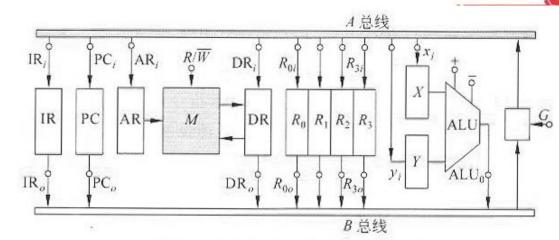
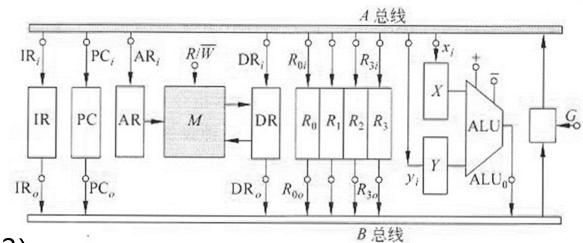


图 5.15 双总线结构机器的数据通路



 $PC \rightarrow AR PCo, G, ARi$ R2o, G, ARi $R2 \rightarrow AR$ R/W=R $M \rightarrow DR$ R/W=R $M \rightarrow DR$ DR → IR DRo, G, IRi DRo, G, R0i $DR \rightarrow Y$ -、ALUo、G、DRi R3o, G, ARi $R3 \rightarrow AR$ $X+Y \rightarrow DR$ R/W=RR2o, G, ARi $M \rightarrow DR$ $R2 \rightarrow AR$ R/W=W DRo, G, R0i $DR \rightarrow X$ $DR \rightarrow M$

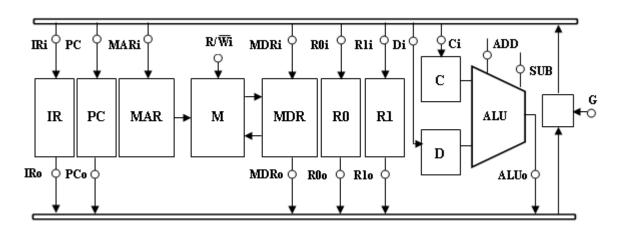


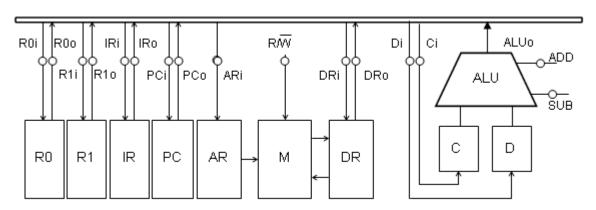
10、某计算机有如下部件:ALU、移位器、主存M、主存地址寄存器MAR、主存数据寄存器MDR、指令寄存器IR、通用寄存器R0、R1、R2、R3、暂存器C和D。

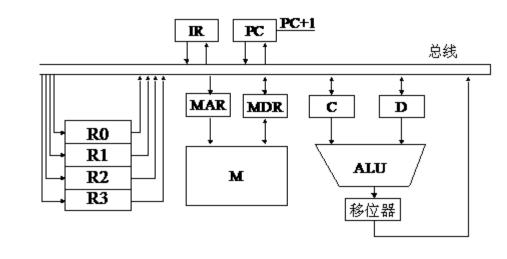
- (1) 请将各逻辑部件组成一个数据通路,并标明数据流动方向。
- (2) 画出 "ADD (R1), (R2) +" 指令的指令周期流程图,指令的含义与PDP-11相同。

```
((R1)) + ((R2)) → ((R2)) ; 寄存器间接寻址(R2) +1 → (R2) ; R2寄存器自增
```

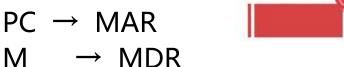








$$ADD (R1) , (R2) +$$



 $M \rightarrow MDR$

$$MDR \rightarrow IR$$

R1 $\rightarrow MAR$

$$M \rightarrow MDR$$

$$R2 \rightarrow MAR$$

 $MDR \rightarrow C$

$$M \rightarrow MDR$$

 $MDR \rightarrow D$

$$C+D \rightarrow MDR$$

$$R2 \rightarrow MAR$$

$$MDR \rightarrow M$$

$$R2 \rightarrow D$$

$$D+1 \rightarrow R2$$



- 13、指令流水线有取指IF、译码ID、执行EX、访存MEM、写回寄存器WB五个过程段,共有20条指令连续输入此流水线。
 - (1) 画出流水处理的时空图,假设时钟周期为100ns。

| 空间(| 1 | | | | 10 | I1 | 12 | 13 | I4 | |
|-----|----|----|----|-----|-----|-----|-----|-----|----|----|
| | | | | | 1 | 1 | 1 | 1 | 1 | |
| | | | | | ₩B | ₩B | ₩B | ₩B | ₩B | |
| | | | | IEI | TET | TET | IEI | TET | | |
| | | | EX | EX | EX | EX | EX | | | |
| | | ID | ID | ID | ID | ID | | | | |
| | IF | IF | IF | IF | IF | | | | | |
| | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 | t9 | 时间 |



13、指令流水线有取指IF、译码ID、执行EX、访存MEM、写回寄存器WB五个过程段,共有20条指令连续输入此流水线。

(2) 求流水线的实际吞吐率(单位时间里执行完毕的指令数)。

k+(n-1)=5+(20-1)=24 20条指令连续输入此流水线,并经过五个过程段全部输出,需要24个时钟周期。

每个时钟周期为100ns。

故实际吞吐率为(每秒多少条指令): 20条 / (24*100ns)



13、指令流水线有取指IF、译码ID、执行EX、访存MEM、写回寄存器WB五个过程段,共有20条指令连续输入此流水线。

(3) 求流水线的加速比。

20条指令采用非流水线处理需要的时钟周期数为:

nk=20*5=100

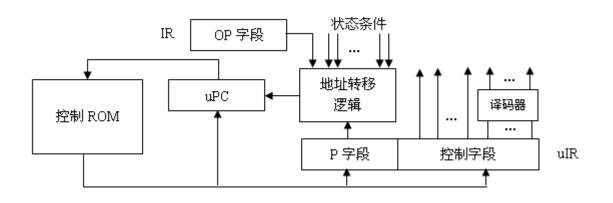
20条指令采用流水线处理需要的时钟周期数为:

k + (n-1) = 5 + (20-1) = 24

加速比为: nk / (k + (n-1))=100/24=4.17



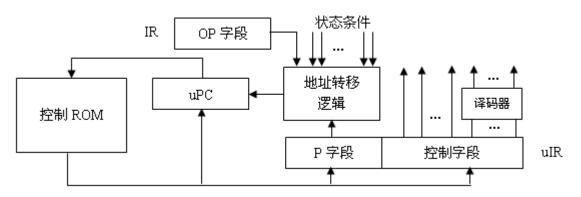
[补] 微程序控制器构成如下图所示,试说明它的基本原理?



设计层面: 首先,根据CPU的数据通路结构、指令操作定义等,画出每条指令的指令周期流程图(具体到每个时钟周期、微操作、微命令)。然后,根据微指令格式、指令周期流程图编写每条指令的微程序。最后,把整个指令系统的微程序(其中取指令的微程序段是公用的)固化到控制存储器中。



[补] 微程序控制器构成如下图所示, 试说明它的基本原理?



运行层面: 首先,逐条执行取指令公用微程序段,控制取指令操作。然后,根据指令的操作码字段,经过变换,找到该指令所对应的特定微程序段,从控制存储器中逐条取出微指令,根据微操作控制字段,直接或经过译码产生微命令(控制信号),控制相关部件完成指定的微操作。一条微指令执行以后,根据微地址字段取下一条微指令。