

第一章 计算机系统概论

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？

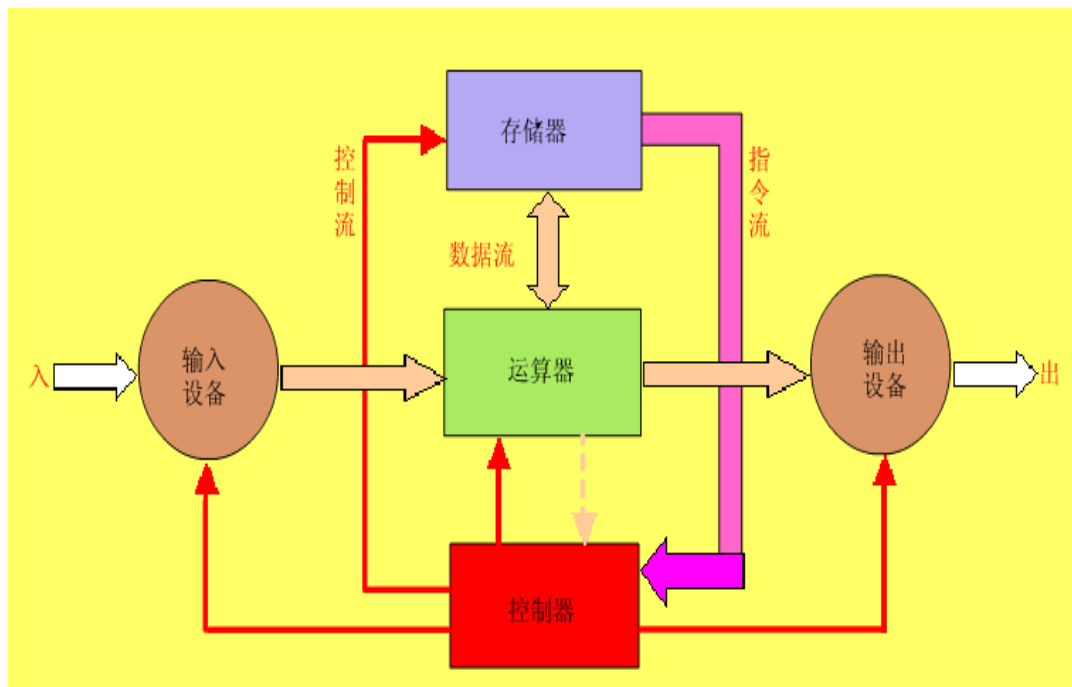
主要设计思想：

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）。

主要组成部分及结构（能简要描述清楚其工作过程）：



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。

从**时间**上来说，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）；

从**空间（处理部件）**上来说，指令一定送给控制器，数据一定送给运算器。

第二章 运算方法和运算器

1、写出下列各整数的原码、补码、反码表示（用 8 位二进制数）。其中 **MSB** 是最高位（符号位），**LSB** 是最低位。

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

[解]:

	二进制真值		原码	补码	反码
-35	-23H	-0010 0011	1010 0011	1101 1101	1101 1100
128	+80H	+1000 0000	不能表示	不能表示	不能表示
127	+7FH	+0111 1111	0111 1111	0111 1111	0111 1111
-1	-01H	-0000 0001	1000 0001	1111 1111	1111 1110
-127	-7FH	-0111 1111	1111 1111	1000 0001	1000 0000
-128	-80H	-1000 0000	不能表示	1000 0000	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。

(1) X=11011, Y=00011

解：

$$\begin{array}{r} [X]_{\text{补}} = 00 \ 1 \ 1 \ 0 \ 1 \ 1 \\ + \quad [Y]_{\text{补}} = 00 \ 0 \ 0 \ 0 \ 1 \ 1 \\ \hline [X+Y]_{\text{补}} = 00 \ 1 \ 1 \ 1 \ 1 \ 0 \quad \text{未溢出} \\ X+Y = 1 \ 1 \ 1 \ 1 \ 0 \end{array}$$

(2) X=-10110, Y= -00001

解：

$$\begin{array}{r} [X]_{\text{补}} = 11 \ 0 \ 1 \ 0 \ 1 \ 0 \\ + \quad [Y]_{\text{补}} = 11 \ 1 \ 1 \ 1 \ 1 \ 1 \\ \hline [X+Y]_{\text{补}} = 1 \ 11. \ 0 \ 1 \ 0 \ 0 \ 1 \quad (\text{丢弃进位 } 1) \quad \text{未溢出} \\ X+Y = 0. \ 0 \ 0 \ 1 \ 1 \ 0 \end{array}$$

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。

$$(1) X=2^{-011} * 0.100101, \quad Y=2^{-010} * (-0.011110)$$

解：

$[X+Y]$:

设阶码、尾数均用补码表示

$$X: \textcolor{red}{1} \textcolor{red}{101} \ 0 \ 100101 \quad Y: \textcolor{red}{1} \textcolor{red}{110} \ 1 \ 100010$$

$$(1) \text{ 对阶:} \quad -3 \quad -2$$

$$X: \textcolor{red}{1} \textcolor{red}{110} \ 0 \ \textcolor{blue}{0}10010 \ (1) \ (\text{右移一位})$$

(2) 尾数加:

$$\begin{array}{r} 00 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ (1) \\ + \quad 11 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \\ \hline \end{array}$$

$$\textcolor{red}{11} \ 1 \ \textcolor{red}{1} \ \textcolor{red}{0} \ 1 \ 0 \ 0 \ (1) \ \text{未溢出、非规格化}$$

$$(3) \text{ 左规 (2 位):} \quad \textcolor{red}{1} \textcolor{red}{100} \ 1 \ 0100\textcolor{blue}{10}$$

$$(4) \text{ 无舍入: } X+Y = 2^{-100} * (-0.101110)$$

[X-Y]:

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

(1) 对阶 -3 -2

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 1 1 0 0 0 0 (1) 未溢出、且是规格化的

(3) 0 舍 1 入: 1 110 00 1 1 0 0 0 1

(4) 结果: $X-Y = 2^{-010} * (+0.110001)$

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

解:

[X+Y]:

设阶码、尾数均用补码表示

$$X: \text{11 011} \quad 11 \quad 101010 \quad Y: \text{11 100} \quad 00 \quad 010110$$

$$(1) \text{对阶:} \quad -5 \quad -4$$

$$X: \text{11 100} \quad 11 \quad 110101 \quad (0) \quad (\text{右移一位})$$

(2) 尾数加:

$$\begin{array}{r} 11 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad (0) \\ + \quad 00 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \\ \hline 1 \quad 00 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad (0) \end{array} \quad \text{未溢出、非规格化}$$

$$(3) \text{左规 (2 位):} \quad \text{1 010} \quad 00 \quad 101100$$

$$(4) \text{无舍入: } X+Y = 2^{-110} * (0.101100)$$

[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 **1**10101 (0) (右移一位)

(2) 尾数加:

$$\begin{array}{r} 11 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad (0) \\ + \quad 11 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \\ \hline 1 \quad 11 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad (0) \end{array}$$

未溢出、规格化

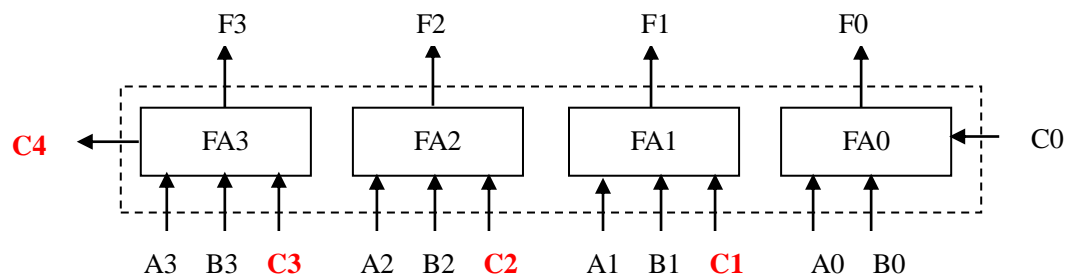
(3) 0 舍 1 入:

$$\begin{aligned} X+Y &= 11 \quad 100 \quad 11 \quad 011111 \\ &= 2^{-100} * (-0.100001) \end{aligned}$$

11、某加法器进位链小组信号为 $C_4C_3C_2C_1$ ，低位来的进位信号为 C_0 ，请分别按照下述两种方式写出 $C_4C_3C_2C_1$ 的逻辑表达式。

(1) 串行进位方式 (2) 并行进位方式

解：



(1) 直接从全加器的进位公式推导；

(2) 设被加数 $A=A_3A_2A_1A_0$ ，加数 $B=B_3B_2B_1B_0$ ，且有：

$$G_0=A_0B_0 \quad P_0= A_0 + B_0$$

$$G_1=A_1B_1 \quad P_1= A_1 + B_1$$

$$G_2=A_2B_2 \quad P_2= A_2 + B_2$$

$$G_3=A_3B_3 \quad P_3= A_3 + B_3$$

G_i ：进位产生函数，表示两个数位都为 1

P_i ：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。

$$C_1=G_0+P_0C_0$$

$$C_2 = G_1 + P_1 C_1$$

$$C_3 = G_2 + P_2 C_2$$

$$C_4 = G_3 + P_3 C_3$$

(3) 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 C_0 。

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 (G_0 + P_0 C_0) = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

$$C_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$$

从逻辑表达式可以看出，所有的 C_i 只与 G_i 、 P_i 、 C_0 有关，而 G_i 、 P_i 、 C_0 可以从已知条件 A、B、 C_0 同时产生，因此所有的 C_i 可以同时产生。

C_1 、 C_2 、 C_3 、 C_4 实现的逻辑电路

第三章 存储系统

1、设有一个具有 20 位地址和 32 位字长的存储器，问：

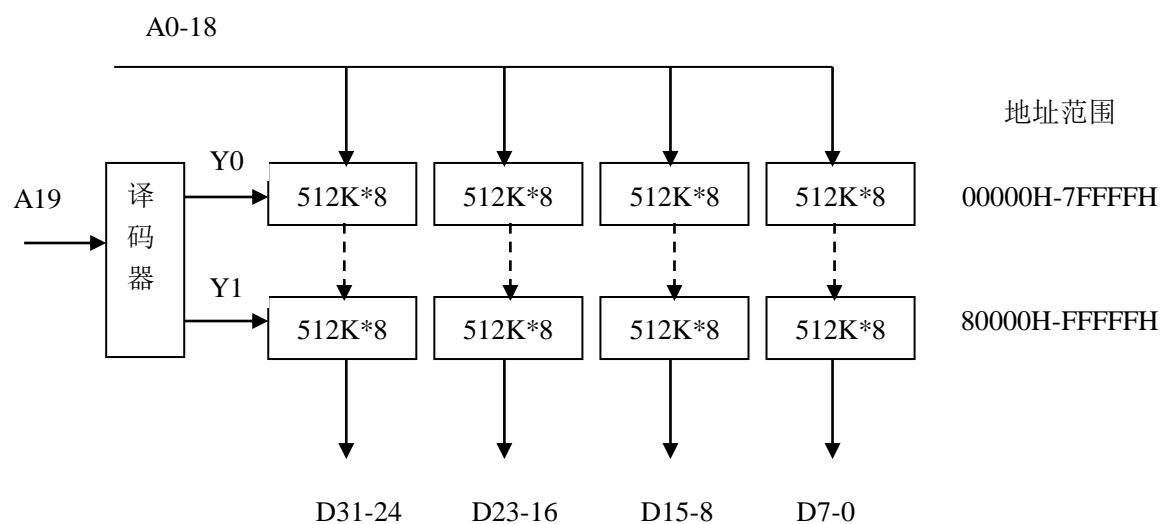
- (1) 该存储器能存储多少个字节的信息？
- (2) 如果存储器由 512K*8 位 SRAM 芯片组成，需要多少片？
- (3) 需要多少位地址做芯片选择？

[解]：

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

(2) $2^{20} * 32\text{bits} = 1024\text{K} * 32 = 2 * [(512\text{K} * 8) * 4]$ ，因此，位扩展 4 片，字扩展 2 组，共需要 8 片 512K*8 位 SRAM 芯片。

(3) 需要 1 位做芯片选择；地址线 A_{0-19} ： A_{0-18} 接片组 512K 的地址线， A_{19} 片选；数据线 D_{0-31} 。

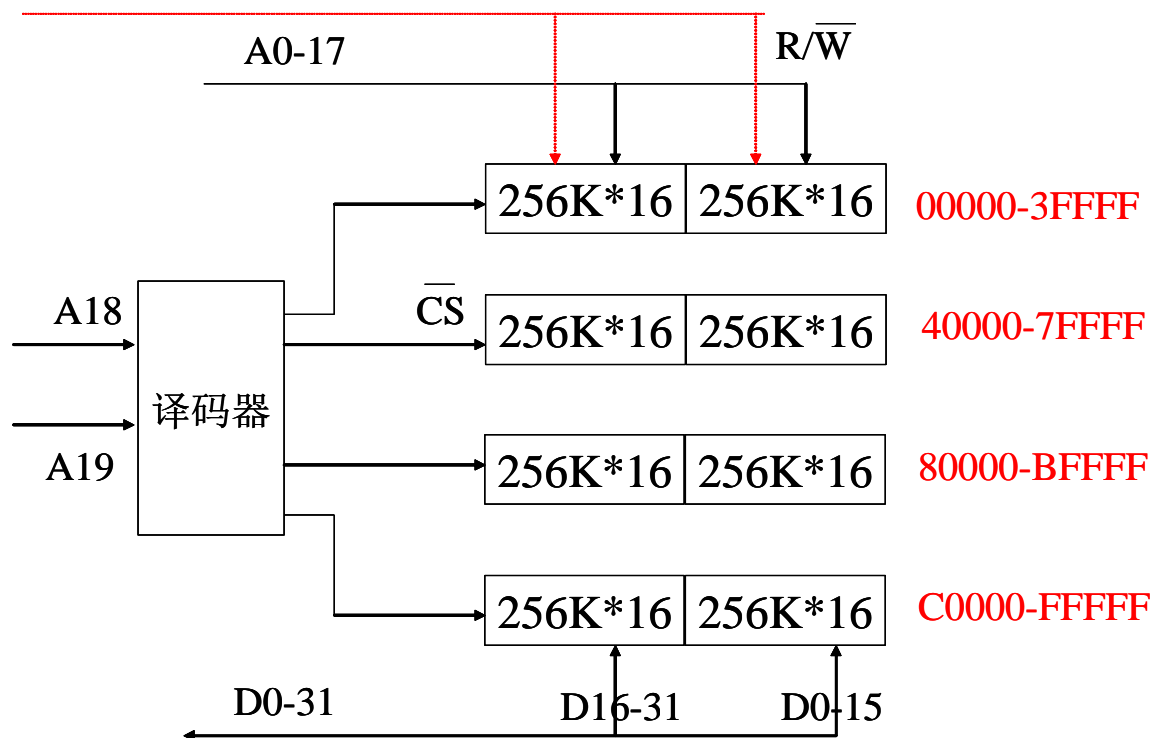


5、要求用 256K*16 位 SRAM 芯片设计 1024K*32 位的存储器。SRAM 芯片有两个控制端：当 \overline{CS} 有效时，该片选中。当 $W/\overline{R}=1$ 时执行读操作，当 $W/\overline{R}=0$ 时执行写操作。

[解]：

$1024K \times 32 \text{ 位} = 4 * (256K \times 16 \text{ 位}) * 2$

地址总线需要 20 位。逻辑图如下：



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 \overline{CS} 和 \overline{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/\overline{W} (读/写), \overline{MREQ} (访存), 要求:

- (1) 画出地址译码方案。
- (2) 将 ROM 与 RAM 同 CPU 连接。

解:

(1) 地址译码方案:

16K*16		0000H	0000	0000	0000	0000
ROM #0		3FFFH	0011	1111	1111	1111
8K		4000H	0100	0000	0000	0000
		5FFFH	0101	1111	1111	1111
8K*8	8K*8	6000H	0110	0000	0000	0000
RAM	RAM #1	7FFFH	0111	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	E000H	1110	0000	0000	0000
RAM	RAM #5	FFFFH	1111	1111	1111	1111

$$CS1 = \overline{A15} \ A14 \ A13 = 011$$

$$CS2 = A15 \ \overline{A14} \ \overline{A13} = 100$$

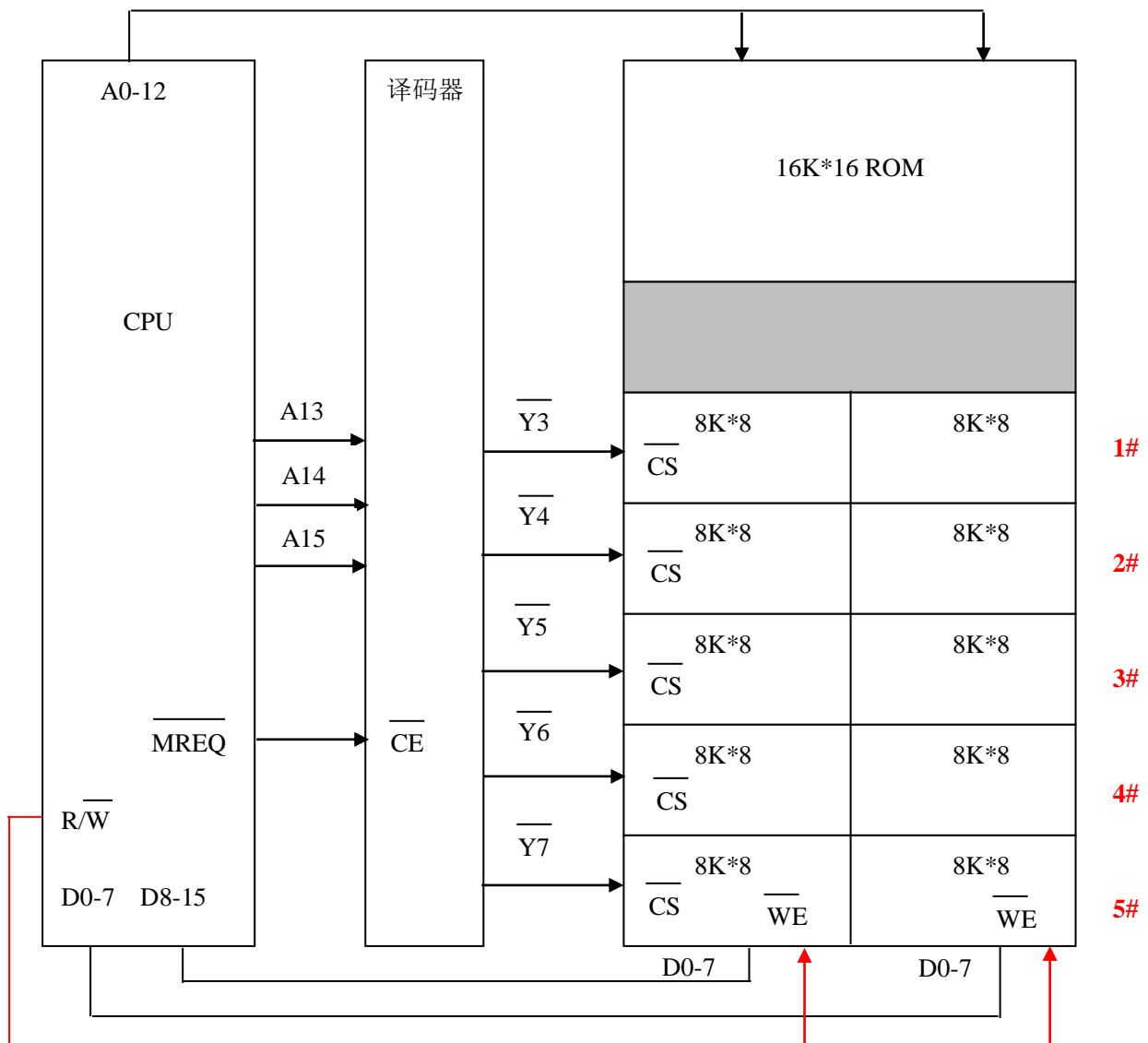
$$CS3 = A15 \ \overline{A14} \ A13 = 101$$

$$CS4 = A15 \ A14 \ \overline{A13} = 110$$

$$CS5 = A15 \ A14 \ A13 = 111$$

显然，可以通过译码器实现。

(2) ROM 与 RAM 同 CPU 连接示意图如下：



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 $T=100\text{ns}$ ，数据总线宽度为 64 位，总线传送周期 $t=50\text{ns}$ 。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： $q=64*8=512\text{bits}$

顺序存储器读出 8 个字的时间： $t_1=mT=8*100\text{ns}=8*10^{-7}\text{s}$

交叉存储器流水读 8 个字时间： $t_2=T+(m-1)t$

$$=100\text{ns}+(8-1)*50\text{ns}$$

$$=450\text{ns}=4.5*10^{-7}\text{s}$$

顺序存储器的带宽： $W_1=q/t_1=512/(8*10^{-7}\text{s})=64*10^7\text{bps}$

交叉存储器的带宽： $W_2=q/t_2=512/(4.5*10^{-7}\text{s})=114*10^7\text{bps}$

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, 已知 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]

$$h = N_c / (N_c + N_m) = 2420 / (2420 + 80) = 2420 / 2500 = 0.968$$

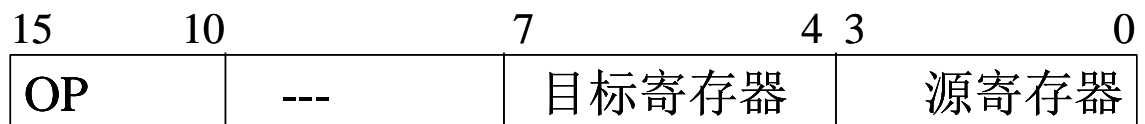
$$r = t_m / t_c = 240 / 40 = 6$$

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

$$t_a = t_c / e = 40 / 0.847 = 47\text{ns}$$

第四章 指令系统

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。

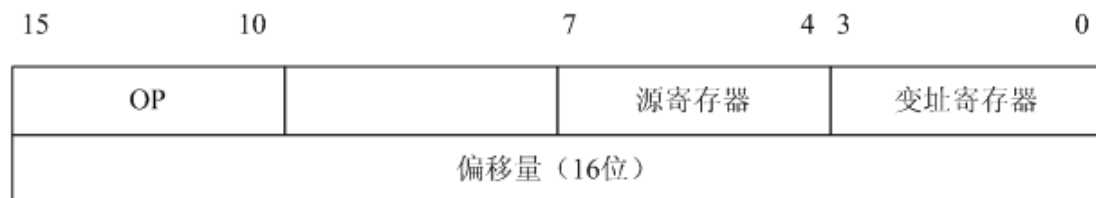


解：

指令格式及寻址方式特点如下：

- (1) 操作码字段 6 位，可 64 种操作；
- (2) 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 $2^4=16$ 个寄存器），属于 RR 型指令；寄存器寻址方式
- (3) 单字长二地址指令。

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。



解：

指令格式及寻址方式特点如下：

- (1) 操作码字段 6 位，可以指定 $2^6=64$ 种操作；
- (2) 双操作数指令。源操作数在通用寄存器中(可分别指定 $2^4=16$ 个寄存器)；目的操作数在主存单元中，采用**变址寻址方式**，即变址寄存器的内容加上随指令一起保存的 16 位偏移量；属于 RS 型指令。
- (3) 双字长二地址指令。

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。

OP	I	X	D
----	---	---	---

解：

寻址方式名称	I	X	有效地址 E
直接寻址	0	00	$E=D$
相对寻址	0	01	$E=(PC)+D$
变址寻址	0	10	$E=(R)+D$
基址寻址	0	11	$E=(R1)+D$
间接寻址	1	00	$E=(D)$
基址寻址+间接寻址	1	01	$E=((R1)+D)$ $D=0$ 即 $E=((R1))$

第五章 中央处理器

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。

解:

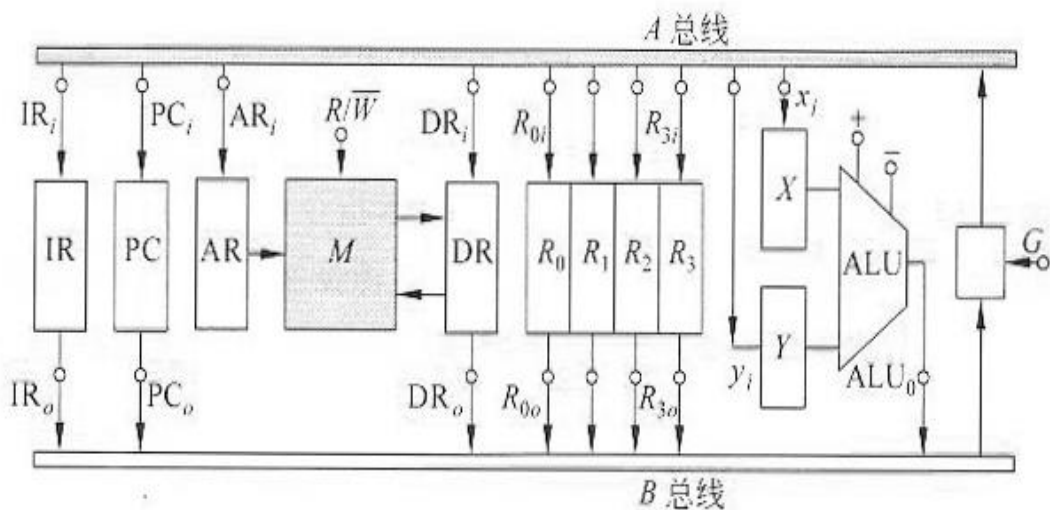
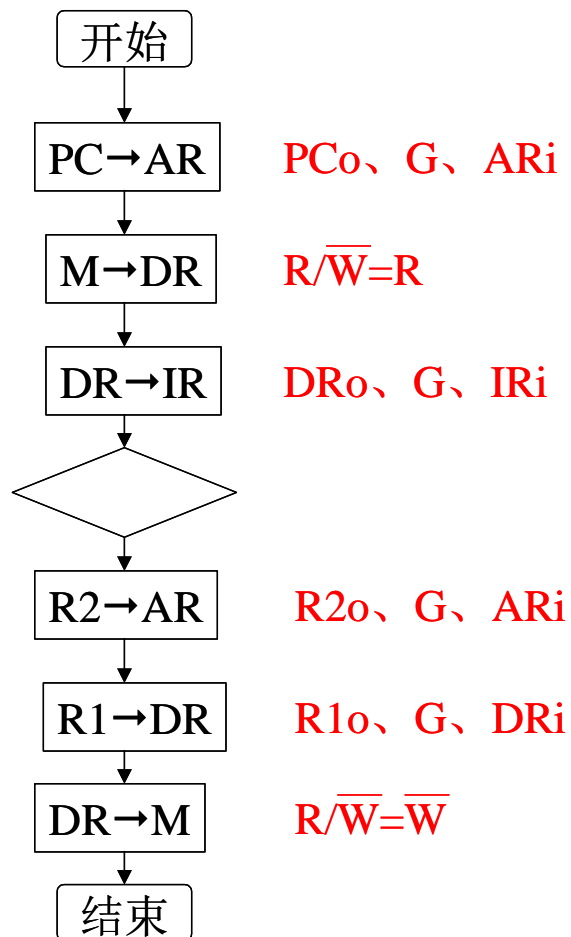


图 5.15 双总线结构机器的数据通路

STA R1, (R2)



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。

解：

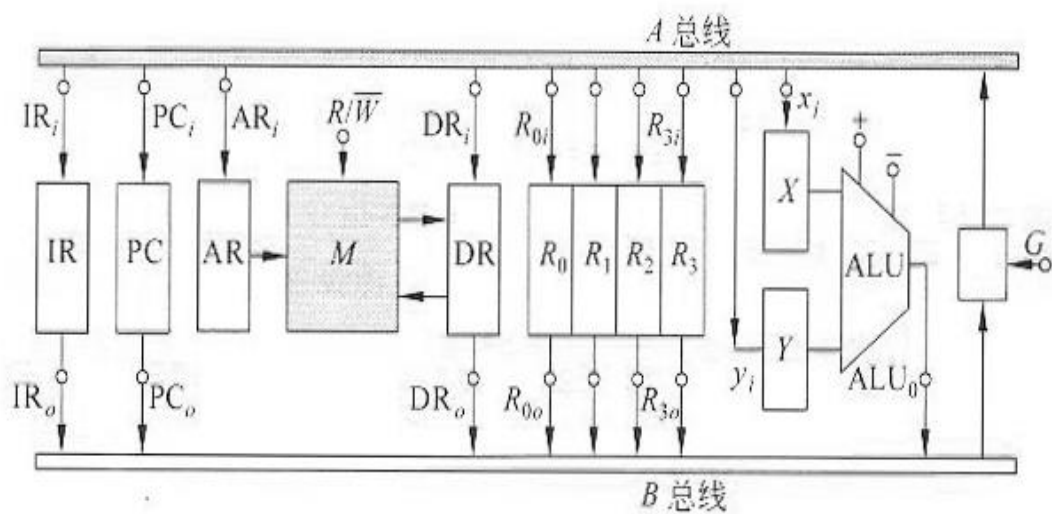
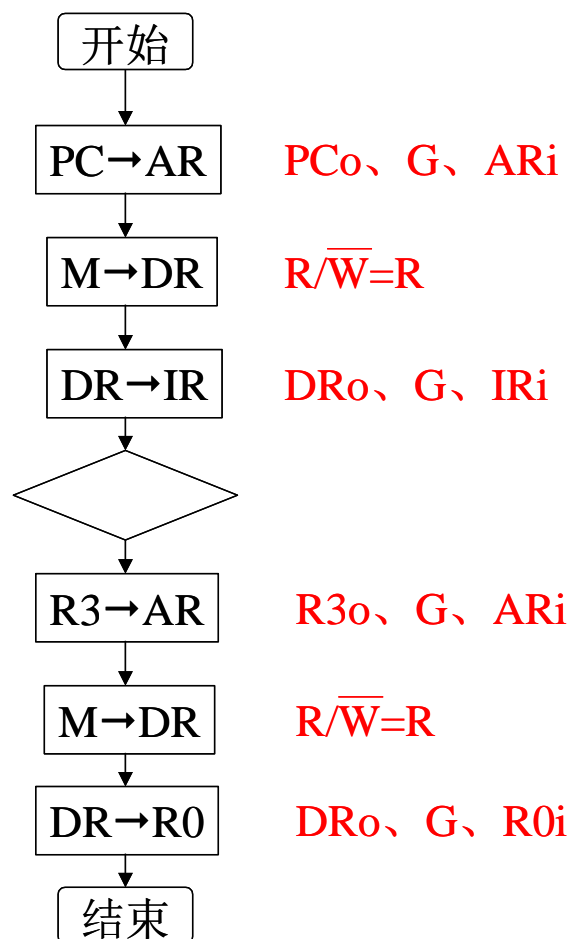


图 5.15 双总线结构机器的数据通路

LDA (R3), R0



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。

解：

ADD (R1), (R2) + 指令

操作： ((R1)) + ((R2)) → ((R2))

(R2) + 1 → (R2)

C、D 双向箭头，是因为在寄存器之间的传送需要通过暂寄存器过渡实现。移位器是为了实现移位运算。

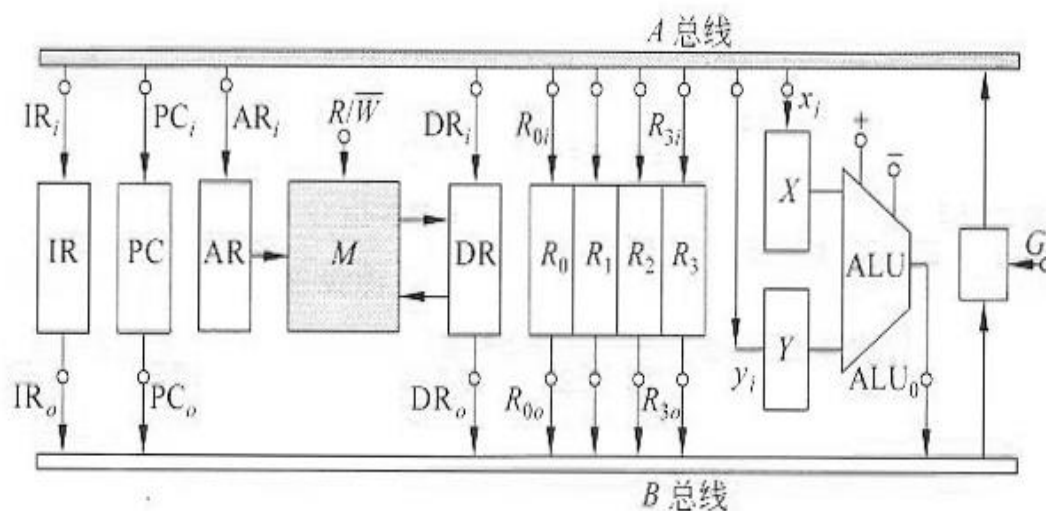
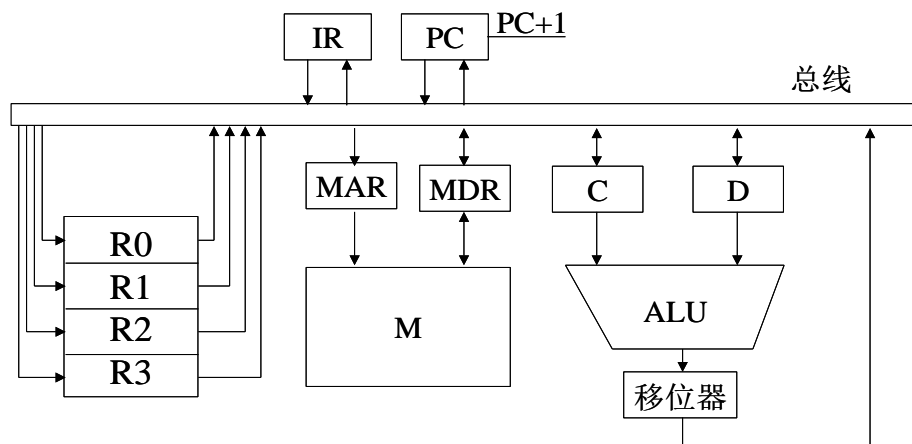


图 5.15 双总线结构机器的数据通路



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。
- (2) 求流水线的实际吞吐率（单位时间里执行完毕的指令数）。
- (3) 求流水线的加速比。

解：

- (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

$$20 \text{ 条} / (24 * 100\text{ns})$$

(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：

$$nk / (k + (n-1)) = 100 / 24 = 4.17$$