

【例】有一个 (7, 4) 码 (即CRC码为7位, 信息码为4位),
已确定生成多项式为: $G(X) = X^3 + X + 1 = 1011$
被传输的信息 $C(x) = 1001$, 求 $C(x)$ 的CRC码。

解: $C(x)$ 左移 $r = n - k = 3$ 位

$$\text{即 } C(x) \cdot 2^r = 1001 \times 2^3 = 1001000$$

将上式模2采用除法 除以给定的 $G(x) = 1011$:

$$1001000 / 1011 = 1010 + 110 / 1011$$

得到余数表达式: $R(x) = 110$

所求CRC码:

$$C(x) \cdot 2^3 + R(x) = 1001000 + 110 = 1001110$$

$$\begin{array}{r}
 1010 \\
 \hline
 1011 \overline{) 1001000} \\
 \underline{1011} \\
 1000 \\
 \underline{1011} \\
 110
 \end{array}$$

(A)

$$\begin{array}{r}
 1010 \\
 \hline
 1011 \overline{) 1001111} \\
 \underline{1011} \\
 1011 \\
 \underline{1011} \\
 1
 \end{array}$$

(B)



CRC码的查错表

CRC	A7	A6	A5	A4	A3	A2	A1	余数	出错
正确	1	0	0	1	1	1	0	000	
某一位出错	1	0	0	1	1	1	1	001	A1
	1	0	0	1	1	0	0	010	A2
	1	0	0	1	0	1	0	100	A3
	1	0	0	0	1	1	0	011	A4
	1	0	1	1	1	1	0	110	A5
	1	1	0	1	1	1	0	111	A6
	0	0	0	1	1	1	0	101	A7

- 收到的CRC码除以约定的生成多项式 $G(x)$ ，如果余数为0则传输无误，否则传输错误，根据所得余数值就可找出错误并取反纠正。
- 上表详细说明了CRC码1001110在传送时某一位出错后的判断与纠正方法 [$C(X) = 1001$ 、 $G(x) = 1011$]。

2.3.2 浮点数的加/减运算

两数首先均为规格化数，进行规格化浮点数的加减运算需经过5步完成：

- (1) **对阶**操作：低阶向高阶补齐，使阶码相等。
- (2) **尾数**运算：阶码对齐后直接对尾数运算。
- (3) **结果规格化**：对运算结果进行规格化处理（使补码尾数的最高位和尾数符号相反）。如溢出则需右规；如不是规格化时应左规。
- (4) **舍入**操作：丢失位进行0舍1入或恒置1处理。
- (5) **判断溢出**：判断阶码是否溢出，下溢则将运算结果置0（机器0），上溢则中断。



例：设 $X=2^{-011} \times 0.110100$ ， $Y=2^{-100} \times -0.101110$ 。按下列运算步骤求 $[X+Y]_{\text{补}}$ ，其中阶码4位（含1位符号位），尾数7位（含1位符号位）。

①求阶差 ②对阶 ③尾数运算 ④结果规格化

解答：

①阶差 ΔE 为1。

②对阶。

Y的阶码小，应使Y的尾数右移1位，阶码加1。此时X的阶码为11101，尾数为11.101001。

③尾数求和。 $00.110100+11.101001=00.011101$ 。

④规格化处理。

结果符号与最高位相同，执行左规。

结果尾数为00.111010，阶码为11100。

(2) IEEE754标准

根据IEEE 754国际标准，常用的浮点数有两种格式：

- 单精度浮点数（32位），阶码8位，尾数24位（内含1位符号位）。
 - 双精度浮点数（64位），阶码11位，尾数53位（内含1位符号位）。
- 由于IEEE754标准约定尾数用原码表示，在小数点左部有一位隐含位，从而实际使得尾数的有效值变为1.M。例如，最小为x1.0...0，最大为x1.1...1。规格化表示，故小数点左边的位恒为1，可省去。
 - 阶码部分采用移码表示，移码值127，1~254 经移码为-126 ~ +127。其中00000000 11111111有特点含义，不用做阶码。
- 格式： $(-1)^S \times 2^E \times (M_0.M_{-1}\cdots M_{-(p-1)})$
 - 最高是数符S，占1位，0表示正、1表示负。
 - 指数项E，基数是2，是一个带有一定偏移量的无符号整数（移码）。
 - 尾数部分M，是一个带有一位整数位的二进制小数真值形式（原码）。其规格化形式应调整阶码使其尾数整数位M₀为1且与小数点一起隐含掉。

主存储器的扩展

- 例：拟扩展存储器容量 $8K \times 8$ 位，可选用的RAM芯片容量为 $2K \times 4$ 位。已知地址总线 $A_{15} \sim A_0$ ，共16位；双向数据总线 $D_7 \sim D_0$ ，共8位，由R/W线控制读写。

问：1. 该存储系统要采用什么形式的扩展方式？

2. 总共需要多少个RAM芯片？

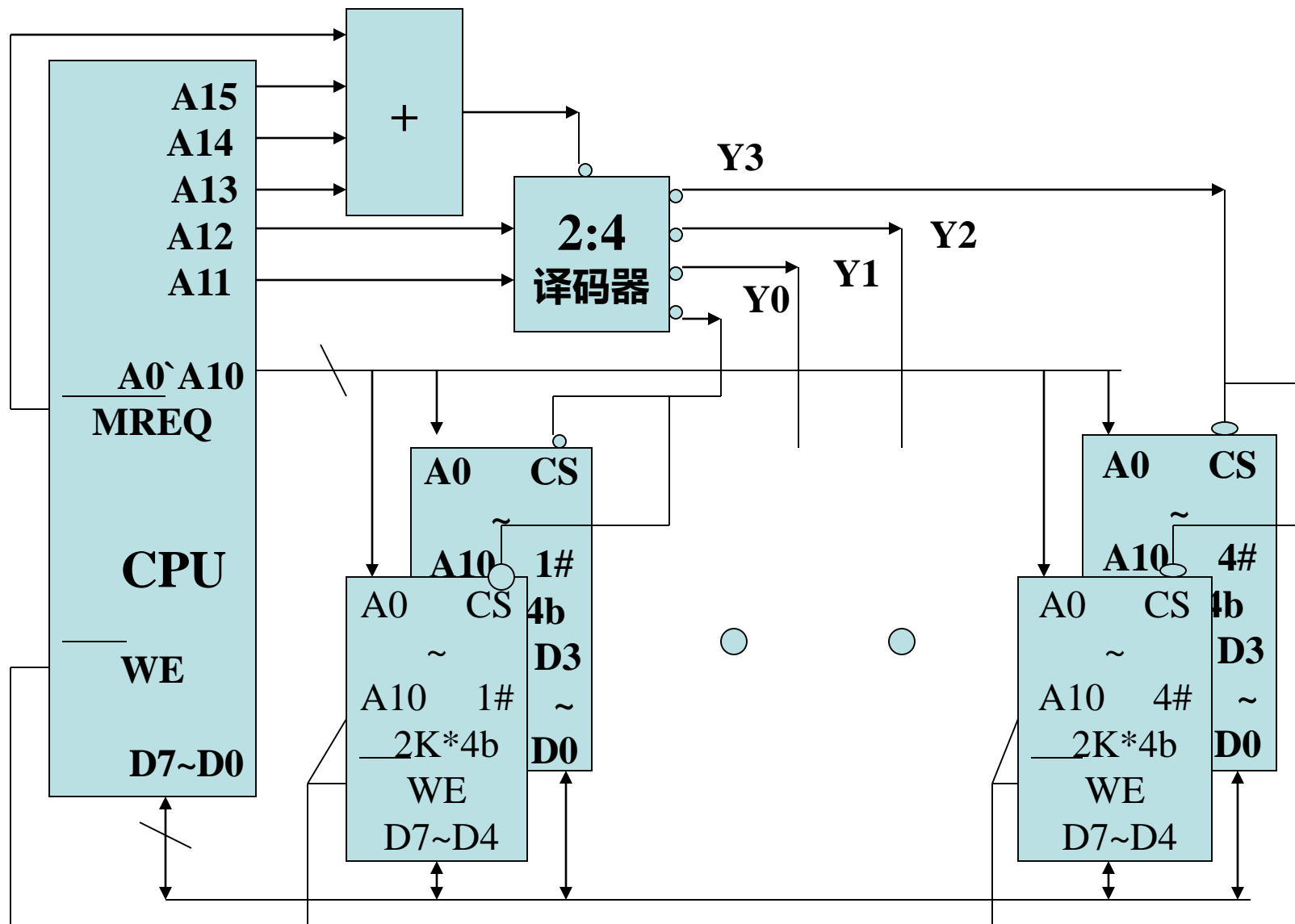
3. 请采用全译码方式，设计并画出该存储器系统的逻辑电路图，并注明存储器地址分配。

- 解：1. 要采用字、位全扩展方式

2. 所需芯片数为 $(8/2) \times (8/4) = 4 \times 2 = 8$ 片

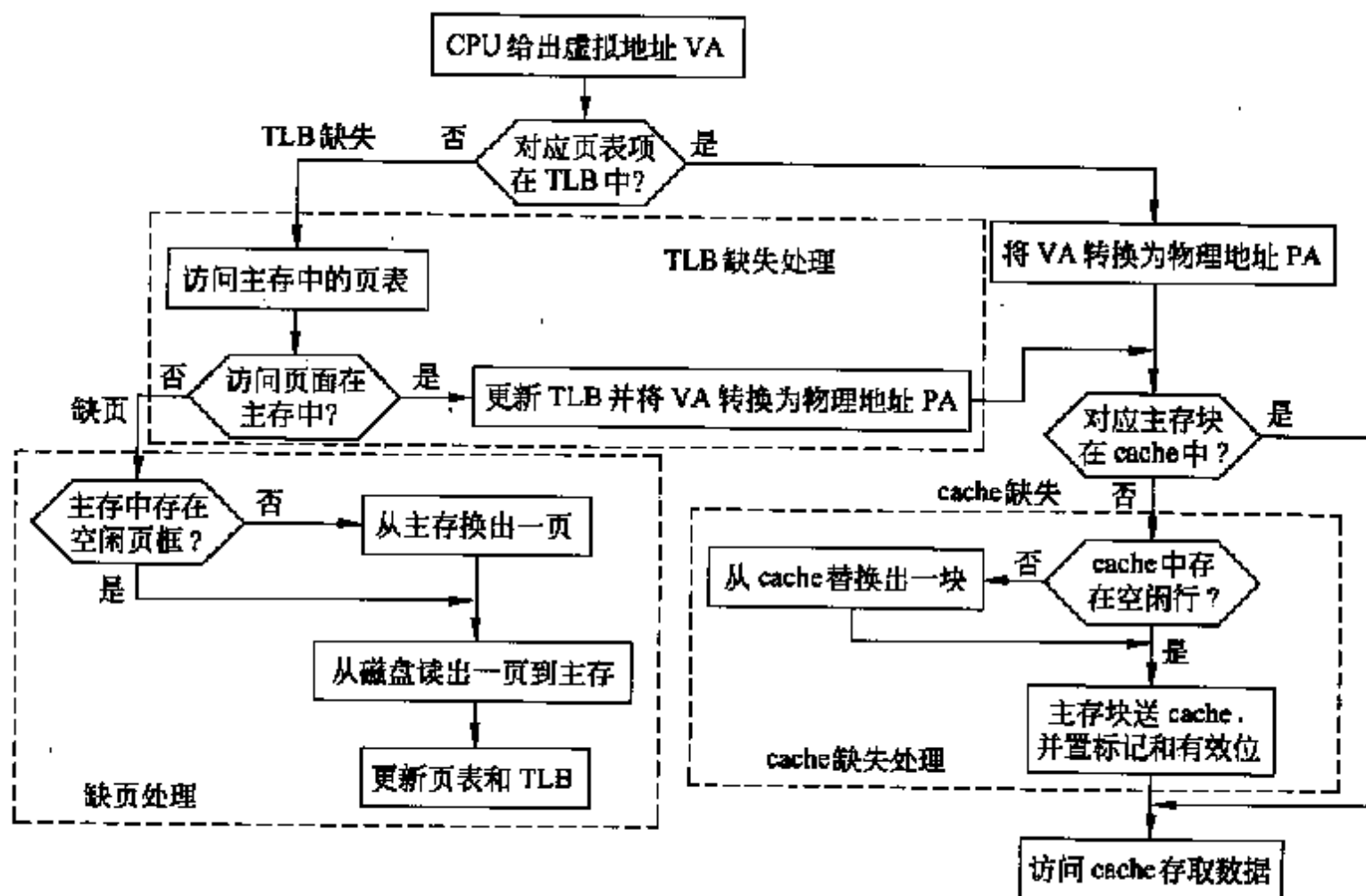
3. 全译码方式要求每一存储单元（字节）都要有唯一的存储地址，不得重叠。已知 $2K \times 4$ 位芯片内字地址为 $A_{10} \sim A_0$ 共11位。而字长8K的存储范围是 $A_{12} \sim A_0$ ，其中 A_{12} 、 A_{11} 是片选地址， $A_{15} \sim A_{13}$ 的连接决定哪种译码方式。全译码方式时其地址不能悬空和乱接。

8K*8b存储系统



- 该系统各存储块的地址范围分别为:
- A15A14A13A12A11A10~A0 16进制数
- 1#是 0 0 0 0 0 0 ~ 0 0000H
- 0 0 0 0 0 1 ~ 1 07FFH
- 2#是 0 0 0 0 1 0 ~ 0 0800H
- 0 0 0 0 1 1 ~ 1 0FFFH
- 3#是 0 0 0 1 0 0 ~ 0 1000H
- 0 0 0 1 0 1 ~ 1 17FFH
- 4#是 0 0 0 1 1 0 ~ 0 1800H
- 0 0 0 1 1 1 ~ 1 1FFFH
- 此8KB全译码方式的地址范围是0000H~1FFFH

CPU访问主存过程



CACHE

例：下列命令组合情况中，一次访存过程中，不可能发生的是（ ）。

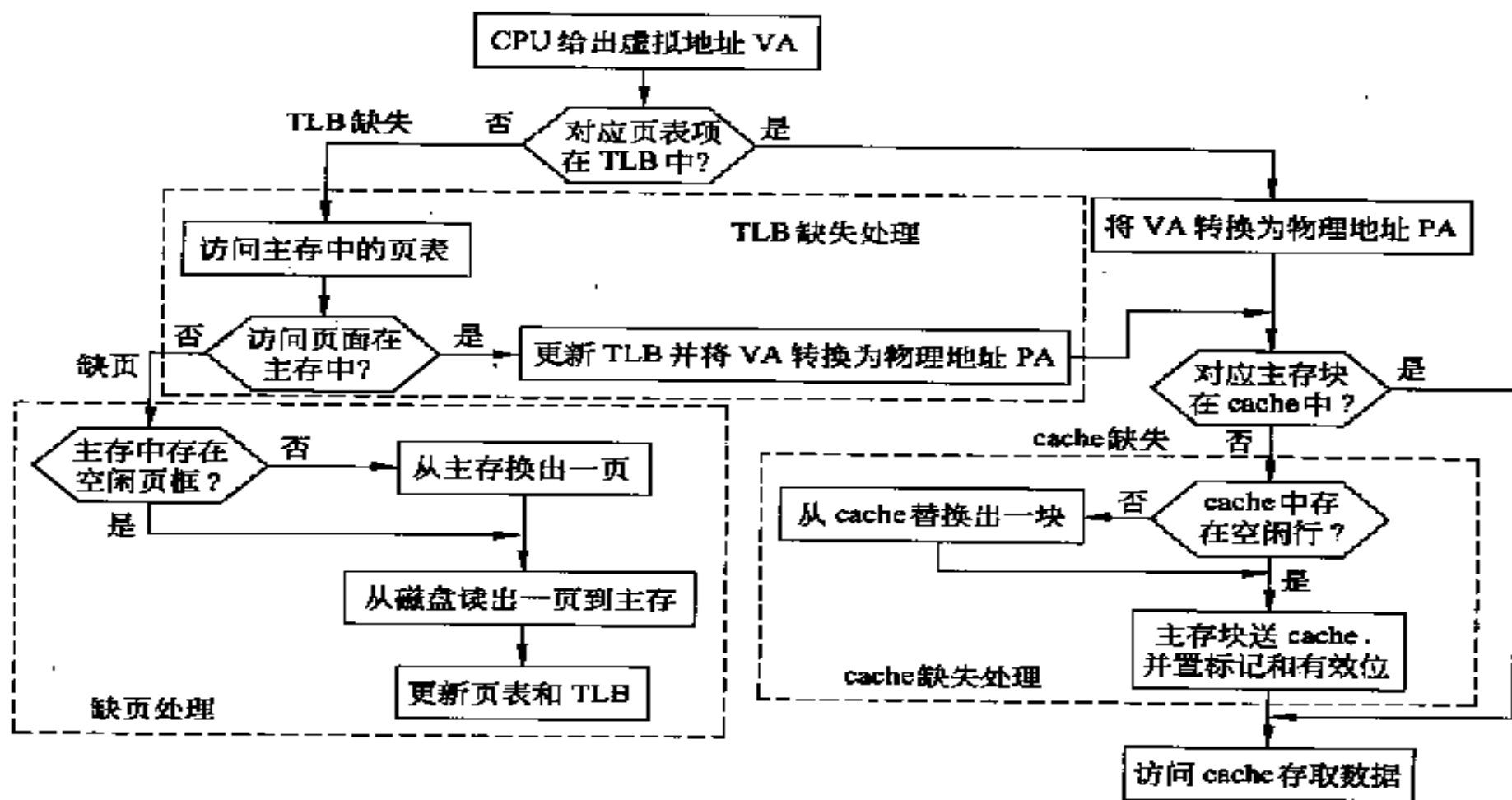
- A. TLB未命中，Cache未命中，Page(页表)未命中
- B. TLB未命中，Cache命中，Page命中
- C. TLB命中，Cache未命中，Page命中
- D. TLB命中，Cache命中，Page未命中



选 D

17. 题详细解答:

CPU访存过程: 在一个具有Cache和虚拟存储器的系统中, CPU的一次访存操作可能涉及到TLB、页表、Cache、主存和磁盘的访问, 其访问过程如下图所示。



从图上中可以看出，**CPU**访存过程中存在**TLB**缺失、**cache**、缺页三种缺失情况。下表给出了三种缺失的几种组合情况。

序号	TLB	page	cache	说 明
1	hit	hit	hit	可能,TLB 命中则页一定命中,信息在主存,就可能在 cache 中
2	hit	hit	miss	可能,TLB 命中则页一定命中,信息在主存,但可能不在 cache 中
3	miss	hit	hit	可能,TLB 缺失但页可能命中,信息在主存,就可能在 cache 中
4	miss	hit	miss	可能,TLB 缺失但页可能命中,信息在主存,但可能不在 cache 中
5	miss	miss	miss	可能,TLB 缺失,则页也可能缺失,信息不在主存,一定也不在 cache
6	hit	miss	miss	不可能,页缺失,说明信息不在主存,TLB 中一定没有该页表项
7	hit	miss	hit	不可能,页缺失,说明信息不在主存,TLB 中一定没有该页表项
8	miss	miss	hit	不可能,页缺失,说明信息不在主存,cache 中一定也没有该信息

快表 (TLB)

- **快表TLB (Translation Look aside Buffer)** 由硬件组成，是专用的高速缓冲存储器，通常称为**转换旁路缓冲器**。用于存放近期经常使用的页表项。
- **TLB**内容只是慢表（指主存中的页表）的小小的副本，比实际的页表小得多，一般在**16行~128行**之间。
- 查表时，由虚页号**同时去查快表和慢表**，当在快表中有此虚页号时，就能很快地找到对应的实页号送入实主存地址寄存器，并使慢表的查找作废。
- 如果在快表中查不到时，那就要费一个访主存时间查慢表，从慢表中查到实页号送入实主存地址寄存器并将此虚页号和对应的实页号同时送入快表。替换快表中某一行内容，这也要用到替换算法。具体的替换算法如先进先出、近期最少使用和随机算法等。

- **平均访问时间：**若 t_c 表示Cache的存取周期， t_m 表示主存的存取周期， $1-h$ 表示未命中率，则Cache—主存系统的平均访问时间 t_a 为：

$$\begin{aligned} t_a &= h t_c + (1-h) (t_m + t_c) \\ &= t_c + (1-h) t_m \\ &\approx h t_c + (1-h) t_m \end{aligned}$$

(Cache访问和主存访问同时启动)

- **访问效率：**设 $r=t_m/t_c$ 表示主存慢于Cache的倍率，则访问效率 e 为：

$$e = t_c / t_a = t_c / [h t_c + (1-h) t_m] = 1 / [h + (1-h) r] = 1 / [r + (1-r) h]$$

为提高访问效率， h 接近 1 好。

- **加速比：** $SP = t_m / t_a$

- 某计算机系统的内存系统中，已知cache存取周期为45ns，主存存取周期为200ns。CPU执行一段程序时，CPU访问内存系统共4500次，其中访问主存的次数为340次，问：1. Cache 命中率H是多少？
- 2. CPU访问内存的平均访问时间Ta是多少？
- 3. Cache/主存系统的访问效率e是多少？

解： 1、Cache 命中率H

$$H = N_c / (N_c + N_m) = (4500 - 340) / 4500 = 0.92$$

2、CPU访存的平均时间

$$\begin{aligned} T_a &= H * T_c + (1 - H) (T_m + T_c) \\ &= 0.92 * 45 + (1 - 0.92) * (200 + 45) = 61 \text{ ns} \end{aligned}$$

3、Cache/主存系统的访问效率

$$e = T_c / T_a * 100\% = 45 / 61 * 100\% = 0.74 * 100\% = 74\%$$

- **7.10** 某程序对页面要求的序列为：
P3P4P2P6P4P3P7P4P3P6P3P4P8P4P6。
- **1)** 设主存容量为**3**个页面，求**FIFO**和**LRU**替换算法是各自的命中率（假设开始时主存为空）。
- 2)** 当主存容量增加到**4**个页面时，两替换算法各自的命中率又是多少？

●答：1) 主存3个页面时的调页情况：

●页面请求

●L ③

●R ②

●U ①

●命中

●F ③

●I ②

●F ①

●O 命中

3	4	2	6	4	3	7	4	3	6	3	4	8	4	6
3	3	3	4	2	6	4	3	7	4	4	6	3	3	8
/	4	4	2	6	4	3	7	4	3	6	3	4	8	4
/	/	2	6	4	3	7	4	3	6	3	4	8	4	6
×	×	×	×	√	×	×	√	√	×	√	√	×	√	×
3	3	3	4	4	2	6	3	3	7	4	4	6	3	8
/	4	4	2	2	6	3	7	7	4	6	6	3	8	4
/	/	2	6	6	3	7	4	4	6	3	3	8	4	6
×	×	×	×	√	×	×	×	√	×	×	√	×	×	×

●采用LRU算法的命中率位 $6/15=40\%$ ，采用FIFO算法的命中率位 $3/15=20\%$ 。

● 2)主存页面为4时的调页情况:

● 页面请求

● L ④

● R ③

● U ②

● ①

● 命中

● F ④

● I ③

● F ②

● O ①

● 命中

3	4	2	6	4	3	7	4	3	6	3	4	8	4	6
3	3	3	3	3	2	6	6	6	7	7	7	6	6	3
/	4	4	4	2	6	4	3	7	4	4	6	3	3	8
/	/	2	2	6	4	3	7	4	3	6	3	4	8	4
/	/	/	6	4	3	7	4	3	6	3	4	8	4	6
×	×	×	×	√	√	×	√	√	√	√	√	×	√	√
3	3	3	3	3	3	4	4	2	2	2	6	7	7	3
/	4	4	4	4	4	2	2	6	6	6	7	3	3	4
/	/	2	2	2	2	6	6	7	7	7	3	4	4	8
/	/	/	6	6	6	7	7	3	3	3	4	8	8	6
×	×	×	×	√	√	×	√	×	√	√	×	×	√	×

● 采用LRU算法的命中率为9/15=60%，FIFO为6/15=40%。

例：设某磁盘有两个记录面，存储区内直径为**2.36英寸**，外直径为**5英寸**，道密度为**1250TPI**(TPI: 磁道数/英寸)，内直径处的位密度为**52400bpi** (位/英寸)，转速为**2400rpm**(rpm: /分)。请解答

①每面有多少磁道，每磁道能存储多少字节？ ②数据传输率是多少？ ③设找道时间在**10ms~40ms**之间，在一个磁道上写上**8000**字节数据，平均需要多少时间？

解答： ① 已知道密度 $D_{\text{cylinder}} = 1250\text{TPI}$ ，内直径 $\text{Dim}_{\text{inner}} = 2.36$ 英寸，外直径 $\text{Dim}_{\text{outside}} = 5$ 英寸，则每面的磁道数 $N_{\text{每面磁道数}}$ 为：

$$N_{\text{每面磁道数}} = D_{\text{cylinder}} \times (\text{Dim}_{\text{outside}} - \text{Dim}_{\text{inner}}) / 2 \\ = 1250 \times 1.32 = 1650$$

每道上存储的字节个数相同，则可存储的字节数可由下公式得到：
 $\text{Num}_{\text{磁道}} = 2\pi R_{\text{inner}} \times D_{\text{bit}} = \pi \times \text{Dim}_{\text{inner}} \times D_{\text{bit}} / 8 = 48562$ 字节。

② 设数据传输率为 f ，则根据数据传输率的定义，由以下公式可以得到解： $f = \pi \times \text{Dim}_{\text{inner}} \times (D_{\text{bit}}/8) \times (2400/60) = 1.94 \text{ MB/s}$

③ 设读取数据所需总的时间为 T ，平均找道时间为 T_{SEEK} ，旋转延迟为 T_{ROTATE} ，数据读取时间为 T_{READ} ，则解过程为：

$$\begin{aligned} T &= T_{\text{SEEK}} + T_{\text{ROTATE}} + T_{\text{READ}} \\ &= (10+40)/2 + [1/(2400/60)/2 + (8 \times 1000)/(1.94 \times 10^6)] \times 10^3 = 41.7 \text{ ms} \end{aligned}$$

例. (12分)某计算机存储器按字节编址，虚拟(逻辑)地址空间大小为**16MB**，主存(物理)地址空间大小为**1MB**，页面大小为**4KB**；**Cache**采用直接映射方式，共**8行**；主存与**Cache**之间交换的块大小为**32B**。系统运行到某一时刻时，页表的部分内容和**Cache**的部分内容分别如题**44-a**图、题**44-b**图所示，图中页框号及标记字段的内容为十六进制形式。请回答下列问题。

虚页号	有效位	页框号	...
0	1	06	...
1	1	04	...
2	1	15	...
3	1	02	...
4	0	-	...
5	1	2B	...
6	0	-	...
7	1	32	...

题 44-a 图 页表的部分内容

行号	有效位	标记	...
0	1	020	...
1	0	-	...
2	1	01D	...
3	1	105	...
4	1	064	...
5	1	14D	...
6	0	-	...
7	1	27A	...

题 44-b 图 Cache 的部分内容

2011年试题和解析

- (1) 虚拟地址共有几位，哪几位表示虚页号？物理地址共有几位，哪几位表示页框号(物理页号)？
- (2) 使用物理地址访问**Cache**时，物理地址应划分成哪几个字段？要求说明每个字段的位数及在物理地址中的位置。
- (3) 虚拟地址**001C60H**所在的页面是否在主存中？若在主存中，则该虚拟地址对应的物理地址是什么？访问该地址时是否**Cache**命中？要求说明理由。



44题解析:

(1) 虚拟地址为**24**位，其中高**12**位为虚页号：

物理地址为**20**位，其中高**8**位为物理页号。

(2) **20**位物理地址中，最低**5**位为存储块块内地址，中间**3**位表示为**Cache**行号，高**12**位为标志区。

(3) 在主存中。

虚拟地址**001C60H=0000 0000 0001 1100 0110 0000B**，故虚页号为**0000 0000 0001B**，查看**0000 0000 0001 B=001H**处的页表项，由于对应的有效位为**1**，故虚拟地址**001C60H**所在的页面在主存中。

页表**001H**处的页框号(物理页号)为**04H=0000 0100B**，与页内偏移**1100 0110 0000B**拼接成物理地址：

0000 0100 1100 0110 0000B=04C60H。

44题解析:

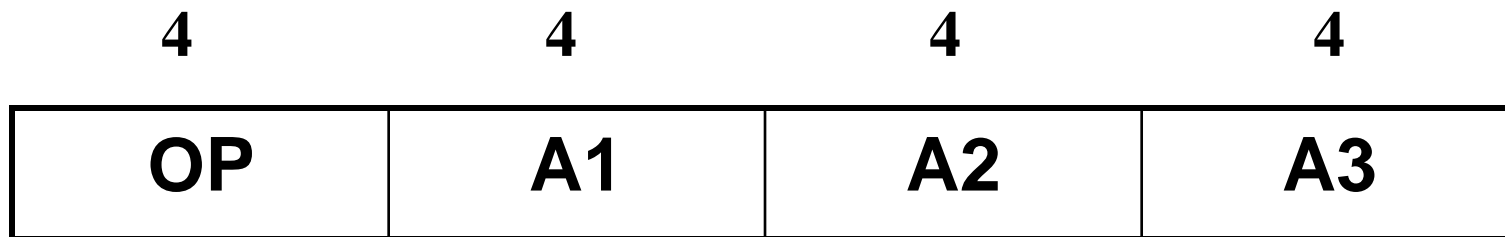
对于物理地址0000 0100 1100 0110 0000B，所在主存块只可能映射到Cache第3行(即第011B行，最低5位是Cache块内地址)；由于该行的有效位：1、标记(值为105H) \neq 04CH(物理地址高12位)，故访问该地址时Cache不命中。



指令系统—扩展技术

- 例：已知指令字长是**16bit**，要求操作码段和三个地址码段都各为**4bit**，如下图所示。试设计一个**16位**的扩展操作码，能对下列所有指令都能进行译码：（10分）

- (1) 7条三地址指令。
- (2) 255条单地址指令。
- (3) 16条零地址指令。
- (4) 求操作码的平均长度



● 解:

三地址指令: 操作码 (OP) 4bit, $2^4=16$, 取其中1~7表明7条。

当操作码OP=0000时, 表明**为单地址指令**。此时将A1, A2扩展为操作码, $2^8=256$, 取1~255表明255条单地址指令。

当OP=0000, 且A1 A2 为全“0”时, 表明为**零地址指令**, 此时A3扩展为操作码, $2^4=16$, 故可以表示16条零地址指令。扩展操作码如图所示。

4	4	4	4	
OP	A ₁	A ₂	A ₃	
0001	A ₁	A ₂	A ₃	三地址
0010	A ₁	A ₂	A ₃	
...	
0111	A ₁	A ₂	A ₃	
0000	0000	0001	A	单地址
0000	0000	0010	A	
...	A	
0000	1111	1111	A	
0000	0000	0000	0000	零地址
0000	0000	0000	0001	
...	
0000	0000	0000	1111	

单地址标识符

零地址标识符

(4) 操作码的平均长度

$$(7*4+255*12+16*16)/(7+255+16)=12$$



5.17 某机字长**16**位，直接寻址空间为**128**字，变址时的位移量是-**64**~**+63**，**16**个通用寄存器都可以作为变址寄存器，设计一套指令系统，满足下列寻址类型的要求。

- ①直接寻址的二地址指令**3**条；
- ②变址寻址的一地址指令**6**条；
- ③寄存器寻址的二地址指令**8**条；
- ④直接寻址的一地址指令**12**条；
- ⑤零地址指令**32**条。⑥求操作符的平均长度

解答：



操作码可为**00、01、10**，共**3**条



操作码可为**11000—11101**，共**6**条



操作码可为**11110000—11110111**，共**8**条

④

9bits

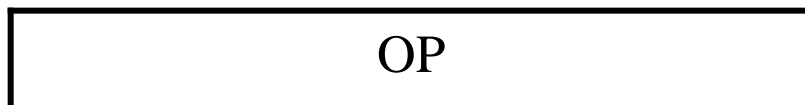
7bits



操作码可为111110000—111111011，共12条

⑤

16bits



操作码可为1111111100000000—1111111000011111，共32条

⑥操作符的平均长度

$$(2*3+5*6+8*8+9*12+16*32) / (3+6+8+12+32) = 11.8$$

3
1

例4：某32位机共有微操作控制信号52个，构成5个相斥类的微命令组，各组分别包含4个、5个、8个、15个和20个微命令。已知可判定的外部条件有CY和ZF两个，微指令字长29位。

(1) 给出采用断定方式的水平型微指令格式。

(2) 控制存储器的容量应为多少位？

(注意各控制字段中应包含一种不发出命令的情况，条件测试字段包含一种不转移的情况。)

解：（1）微指令的格式如下：

D_{28}	$D_{26} D_{25}$	$D_{23} D_{22}$	$D_{19} D_{18}$	$D_{15} D_{14}$	$D_{10} D_9$	$D_8 D_7$	D_0
4个微命令	5个微命令	8个微命令	15个微命令	20个微命令	条件测试字段	下一地址字段	
3位	3位	4位	4位	5位	2位	8位	

（2）控存容量为 $2^8 \times 29 = 256 \times 29$

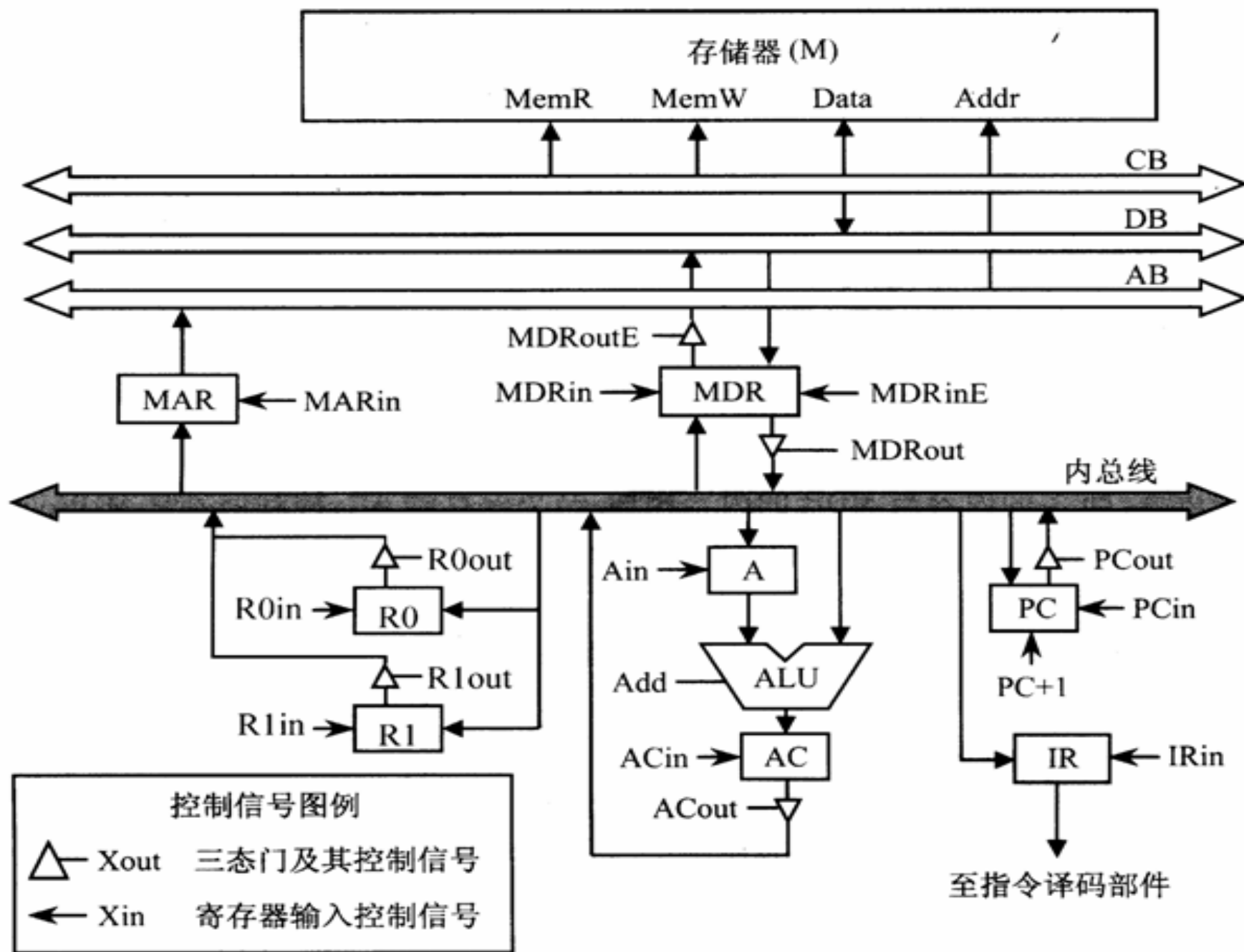
19. 假定不采用Cache和指令预取技术，且机器处于“开中断”状态，则在下列有关指令执行的叙述中，错误的是：

- A. 每个指令周期中CPU都至少访问内存一次
- B. 每个指令周期一定大于或等于一个CPU时钟周期
- C. 空操作指令的指令周期中任何寄存器的内容都不会被改变
- D. 当前程序在每条指令执行结束时都可能被外部中断打断

解： C
每条指令都要涉及到PC+1。

44. 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下图所示，图中所有控制信号为1时表示有效、为0时表示无效，例如控制信号**MDRinE**为1表示允许数据从**DB**打入**MDR**，**MDRin**为1表示允许数据从内总线打入**MDR**。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 **$(R0) + ((R1)) \rightarrow (R1)$** ，即将**R0**中的数据与**R1**的内容所指主存单元的数据相加，并将结果送入**R1**的内容所指主存单元中保存。下表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，请按表中描述方式用表格列出指令执行阶段每个节拍的功能和有效控制信号。

时钟	功能	有效控制信号
C1	$MAR \leftarrow (PC)$	PCout, MARin
C2	$MDR \leftarrow M(MAR), PC \leftarrow (PC) + 1$	MemR, MDRinE, PC+1
C3	$IR \leftarrow (MDR)$	MDRout, IRin
C4	指令译码	无



参考答案一:

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	$M(MAR) \leftarrow (MDR)$	$MDR_{outE}, MemW$

“ $A \leftarrow (R0)$ ”也可在 C7:“ $AC \leftarrow (MDR) + (A)$ ”之前单列的一个时钟周期内执行。

参考答案二:

时钟	功能	有效控制信号
C5	$MAR \leftarrow (R1)$	$R1_{out}, MAR_{in}$
C6	$MDR \leftarrow M(MAR)$	$MemR, MDR_{inE}$
C7	$A \leftarrow (MDR)$	MDR_{out}, A_{in}
C8	$AC \leftarrow (A) + (R0)$	$R0_{out}, Add, AC_{in}$
C9	$MDR \leftarrow (AC)$	AC_{out}, MDR_{in}
C10	$M(MAR) \leftarrow (MDR)$	$MDR_{outE}, MemW$

12.某计算机主频为**1.2GHz**，其指令分为**4**类，它们在基准程序中所占比例及**CPI**如下表所示。

指令系统	所占比例	CPI
A	50%	2
B	20%	3
C	10%	4
D	20%	5

该机的**MIPS**数是

A. 100 B.200 C.400 D.600

解析：执行各程序所占的**CPI**数量：

$0.5 \times 2 + 0.2 \times 3 + 0.1 \times 4 + 0.2 \times 5 = 3$ ，计算机的主频为**1.2GHz**，为**1 200MHz**，该机器的是**MIPS**为 **$1\ 200/3=400$** 。