

一、填空题（每空 2 分，共 30 分）

题号	空【1】答案	空【2】答案
1	先进后出	/
2	缓冲器	信息
3	速率	定时
4	CPU 性能指标	I/O 吞吐率
5	状态条件	布尔
6	补码	先行进位
7	存储密度	平均存取
8	多级	内存

二、选择题（每题 1 分，共 10 分）

题号	答案	题号	答案	题号	答案	题号	答案	题号	答案
1	D	2	C	3	C	4	A	5	C
6	A	7	B	8	C	9	C	10	C

三、综合题（共 60 分）

1、解：（6 分）

- (1) 操作码字段为 6 位，可指定 $2^6 = 64$ 种操作，即 64 条指令。（2 分）
- (2) 单字长（32）二地址指令，用于访问存储器。（2 分）
- (3) 一个操作数在源寄存器（共 16 个），另一个操作数在存储器中（由变址寄存器内容 + 偏移量决定），所以是 RS 型指令。（2 分）

2、解：（6 分）

这是属于独立请求总线仲裁方式，其工作原理如下：

每一个共享总线的设备均有一对“总线请求”（BR）和“总线授权”（BG）线。当设备要求使用总线时，便发出“总线请求”信号，总线控制部件中一般有一个排队电路，根据一定的优先次序决定首先响应哪个设备的请求，当请求的设备排上队，便收到“总线授权”（BG）信号，从而可以使用总线。（2 分）

优点：响应时时映，对优先次序的控制也是相当灵活的，它可以预先固定，也可以通过程序来改变优先次序，并且可以在必要时屏蔽某些设备的请求。（2 分）

缺点：控制线数量多，为控制 n 个设备，必须有 $2n$ 根“总线请求”和“总线授权”线，相比之下，链式查询方式只需 2 根，计数器定时查询方式只需约 $\log_2 n$ 根；另外，总线仲裁器也要复杂得多。（2 分）

3、解：（6 分）

中断屏蔽触发器（IM）：CPU 是否受理中断或批准中断的标志。IM 标志为“0”时，CPU 可受理外界中断请求。（2 分）

中断请求触发器（IR）：暂存中断请求线上由设备发出的中断请求信号，IR 标志为“1”时，表示设备发出了中断请求。（2 分）

允许中断触发器（EI）：用程序指令来置位，控制是否允许某设备发出中断请求。IE 为“1”时，某设备可以向 CPU 发出请求。（2 分）

4、解：（14 分）

（1）微指令格式如图 1 所示。（4 分）

10个微命令	7个微命令	15个微命令	3个微命令	5个微命令	P1~P4	
直接控制	编码控制	编码控制	编码控制	编码控制	判别字段	下地址字段
10位	3位	4位	2位	3位	3位	7位

图 1 某机微指令格式

（2）微指令字长 32 位，控制字段共 22 位，判别字段 3 位，因此下地址字段为 $32 - (22 + 3) = 7$ 位，控制存储器容量为 $2^7 = 128$ 个单元。（2 分）

（3）微程序控制器结构框图如图 2 所示。（8 分）

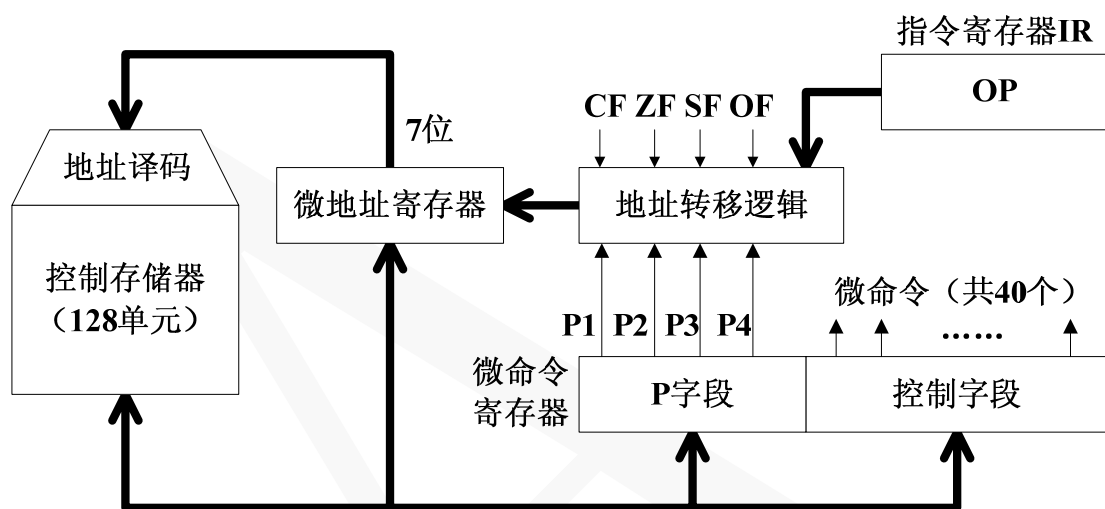


图 2 某机微程序控制器结构框图

5、解：（14 分）

（1）因为： $t_a = t_c / e$ 所以：

$$t_c = t_a \times e = 50 \times 0.8 = 40\text{ns} \quad (2 \text{ 分})$$

$$t_m = t_c \times r = 40 \times 5 = 200\text{ns} \quad (2 \text{ 分})$$

因为： $e = 1 / [r + (1 - r)h]$ ，所以：

$$h = (r - 1/e) / (r - 1) = 3.75 / 4 = 0.9375 \quad (2 \text{ 分})$$

（2）全相联映射的优缺点：

优点是可使主存的一个块直接拷贝到 cache 中的任意一行上，非常灵活，不易产生冲突。（2 分）

缺点是比较器电路复杂，难于设计和实现，因此只适合于小容量 cache 采用。（2 分）

直接映射的优缺点：

优点是硬件简单，成本低。因此适合大容量 cache 采用。（2 分）

缺点是每个主存块只有一个固定的行位置可存放，容易产生冲突。（2 分）

6、解：（14 分）

$$[x]_{\text{原}} = 1.01111 \quad (1 \text{ 分}) \quad \left[\frac{1}{2}x\right]_{\text{补}} = 1.001111 \quad (1 \text{ 分})$$

$$[y]_{\text{补}} = 0.11001 \quad (1 \text{ 分}) \quad \left[\frac{1}{4}y\right]_{\text{补}} = 0.0011001 \quad (1 \text{ 分})$$

$$[-y]_{\text{补}} = 1.00111 \quad (1 \text{ 分})$$

$$\begin{array}{r} [x]_{\text{补}} \quad 11.10001 \\ + [y]_{\text{补}} \quad 00.11001 \\ \hline 00.01010 \quad (1 \text{ 分}) \end{array}$$

因为符号位相同，所以无溢出， $x+y = +0.01010$ (1 分)

$$\begin{array}{r} [x]_{\text{补}} \quad 11.10001 \\ + [-y]_{\text{补}} \quad 11.00111 \\ \hline 10.11000 \quad (1 \text{ 分}) \end{array}$$

因为符号位相异，所以 $x-y$ 的结果发生溢出 (1 分)

(2) 一位全加器 FA 的逻辑结构如图 3 所示。(5 分)

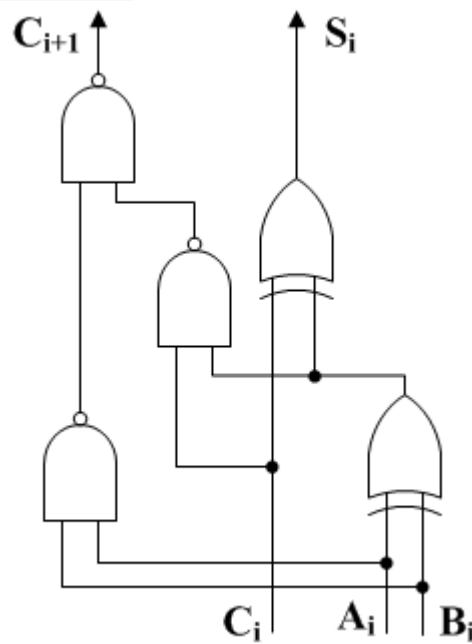


图 3 一位全加器 FA 的逻辑结构图