

5.1 某指令系统指令长 16 位, 每个操作数的地址码长 6 位, 指令分为无操作数、单操作数和双操作数 3 类。若双操作数指令有  $K$  条, 无操作数指令有  $L$  条, 问单操作数指令最多可能有多少条?

地址 指令  
6 位 无/单/双  
L K

双操作数  $K$  条  $\underbrace{\quad 4 \quad 6 \quad 6 \quad}_{0000-1110}$   $2^4$  条  
单操作数  $X$  条  $\underbrace{\quad 10 \quad 6 \quad}_{1111, 0000, 00-1111, 1111, 10}$   
无  $L$  条  $\underbrace{\quad 16 \quad}_{1111, 1111, 11, 0000, 00-1111, 11}$   
$$X_{max} = (2^4 - K) \times 2^6 - \lfloor \frac{L}{2^6} \rfloor$$

## 5.11 讨论 RISC 和 CISC 在指令系统方面的主要区别。

5.12 现有两个 5 位数  $X=00101$   $Y=01010$ , 最高位为符号位, 求  $X \perp$

RISC (精简指令集计算) 和 CISC (复杂指令集计算) 是两种不同的处理器架构, 它们在指令系统方面有一些主要区别。

### 1. 指令集大小:

RISC: 精简指令集计算采用较小的指令集, 这意味着处理器需要执行的指令数量较少。这使得处理器设计更简单, 更容易实现。

CISC: 复杂指令集计算采用较大的指令集, 包含许多复杂的指令。这使得处理器能够执行更多的操作, 但也增加了处理器设计的复杂性。

### 2. 指令复杂性:

RISC: RISC 架构中的指令通常较为简单, 每个指令执行的操作较少。这使得指令执行速度更快, 从而提高了处理器的性能。

CISC: CISC 架构中的指令可能非常复杂, 一个指令可能执行多个操作。这导致指令执行速度较慢, 可能影响处理器性能。

### 3. 指令执行时间:

RISC: 由于 RISC 架构中的指令较为简单, 大多数指令可以在一个时钟周期内完成执行。这有助于提高处理器的性能。

CISC: CISC 架构中的复杂指令可能需要多个时钟周期才能完成执行。这可能导致处理器性能降低。

### 4. 寄存器使用:

RISC: RISC 架构通常使用更多的寄存器, 以减少对内存的访问。这有助于提高处理器的性能, 因为寄存器访问速度比内存访问速度快。

CISC: CISC 架构通常使用较少的寄存器, 这可能导致更频繁的内存访问。这可能影响处理器性能, 因为内存访问速度较慢。

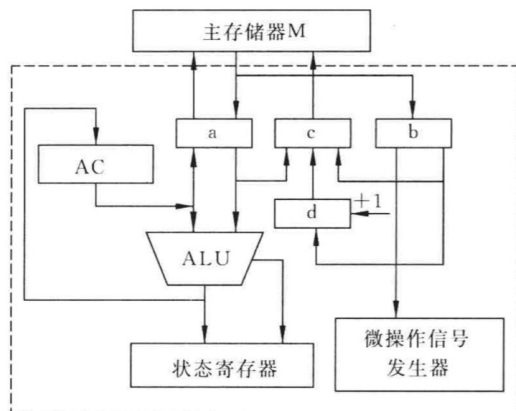
### 5. 编译器优化:

RISC: RISC 架构依赖于编译器进行优化, 以生成高效的指令序列。这需要更高级的编译器技术, 但有助于提高处理器性能。

CISC: CISC 架构的指令复杂性使得编译器优化较为困难。这可能导致生成的指令序列不够高效, 从而影响处理器性能。

6.1 CPU 结构如下图所示,其中有一个累加寄存器 AC、一个状态条件寄存器和其他 4 个寄存器,各部分之间的连线表示数据通路,箭头表示信息传送方向。要求:

- (1) 标明图中 a、b、c、d 这 4 个寄存器的名称。
- (2) 简述指令从主存取出到产生控制信号的数据通路。
- (3) 简述数据在运算器和主存之间进行存/取访问的数据通路。



(1)

a: 数据缓冲寄存器 DR

b: 指令寄存器 IR

c: 主存地址寄存器 AR

d: 程序计数器 PC

(2)  $M \rightarrow IR \rightarrow$  微操作信号发生器

(3) 读:  $M \rightarrow DR \rightarrow ALU \rightarrow AC$

写:  $AC \rightarrow DR \rightarrow M$

6.4 从供选择的答案中,选出正确答案填入□中。

微指令分成水平型微指令和 A 两类。B 可同时执行若干个微操作,所以执行指令的速度比 C 快。

在实现微程序时,取下一条微指令和执行本条微指令一般是 D 进行的,而微指令之间是 E 执行的。

实现机器指令的微程序一般是存放在 F 中的,而用户可写的控制存储器则由 G 组成。

供选择的答案如下。

A~C: ① 微指令;② 微操作;③ 水平型微指令;④ 垂直型微指令。

D, E: ① 顺序;② 重叠。

F, G: ① 随机存储器(RAM);② 只读存储器(ROM)。

6.5 某机有 8 条微指令 I1~I8, 每条微指令所包含的微命令控制信号如下表所示。

微指令	微命令信号									
	a	b	c	d	e	f	g	h	i	j
I1	✓	✓	✓	✓	✓					
I2	✓			✓		✓	✓			
I3		✓						✓		
I4			✓							
I5			✓		✓		✓		✓	
I6	✓							✓		✓
I7			✓	✓				✓		
I8	✓	✓						✓		

a~j 分别对应 10 种不同性质的微命令信号。假设一条微指令的控制字段为 8 位, 请安排微指令的控制字段格式。

① 直接控制法

$10 > 8$  不满足

② 字段直接 + 直接

$10 > 9$  不满足

$\boxed{2 \mid 6}$

$3 + 6$

③ 字段直接 + 直接

$\boxed{3 \mid 5}$

$7 + 5$

(15)

$10 < 12$

$10 = 10$  不满足互斥

④ 字段直接 + 字段直接 + 直接

$\boxed{2 \mid 2 \mid 4}$

$3 + 3 + 4$

$10 = 10$  满足

$\begin{bmatrix} 01-e \\ 10-f \\ 11-h \end{bmatrix} \begin{bmatrix} 01-b \\ 10-i \\ 11-j \end{bmatrix} [acd g]$

6.13 从供选择的答案中选出正确答案,填入□中。

微机 A 和 B 是采用不同主频的 CPU 芯片,片内逻辑电路完全相同。若 A 机的 CPU 主频为 8MHz, B 机为 12MHz。则 A 机的 CPU 主振周期为   $\mu\text{s}$ 。如 A 机的平均指令执行速度为 0.4MIPS, 那么 A 机的平均指令周期为   $\mu\text{s}$ , B 机的平均指令执行为  MIPS。

供选择的答案如下。

A~C: ①0.125; ②0.25; ③0.5; ④0.6; ⑤1.25; ⑥1.6; ⑦2.5。

6.15 造成流水线阻塞的因素有多个。试列举 3 个造成流水线阻塞的因素,并给出其中两个的化解措施。

流水线阻塞是指在处理器的流水线执行过程中,因为某些原因导致其无法连续顺畅地执行指令。以下是三个常见的造成流水线阻塞的因素:

1. 数据冒险 (Data Hazard): 当一个指令需要等待前面指令执行完成并产生结果后才能执行时,就会发生数据冒险。这通常是因为指令之间存在数据依赖关系。
2. 控制冒险 (Control Hazard): 当一个指令需要改变程序的控制流(如跳转、分支等)时,可能会发生控制冒险。这是因为在流水线中,指令的执行顺序是预先确定的,而跳转或分支可能会改变这个顺序。
3. 结构冒险 (Structural Hazard): 当处理器的某个硬件资源在同一时间被多个指令所需时,就会发生结构冒险。这通常是因为硬件资源的限制,例如访问同一个存储器或寄存器。

以下是两个化解流水线阻塞的措施:

1. 数据冒险的化解: 可以使用旁路技术 (Bypassing, 又称 Forwarding) 来解决数据冒险。通过在流水线寄存器之间添加旁路,可以将上一个指令的结果直接传递给下一个指令,而不必等待上一个指令完成。这样,数据依赖关系的指令可以在不等待前一个指令执行完成的情况下继续执行。
2. 控制冒险的化解: 可以使用分支预测技术 (Branch Prediction) 来解决控制冒险。分支预测器会预测分支指令的结果,从而提前确定下一个要执行的指令。如果预测正确,流水线可以连续顺畅地执行指令;如果预测错误,流水线需要清空并重新填充正确的指令,但总体上仍然能提高流水线的执行效率。



## 6.17 机器加电后第一条执行的指令地址是怎样形成的?

由 reset 信号设置 PC 的初始值.

7.1 计算机存储系统分哪几个层次? 每一层次主要采用什么存储介质? 其存储容量和存取速度的相对值如何变化?

存储系统层次: 寄存器组 — cache — 主存 — 辅存  
 ↓ ↓ ↓ ↓  
 存储介质: 电路 SRAM DRAM 磁表面存储  
 容量由小到大, 速度由高到低

7.3 设某流水线计算机有一个指令和数据合一的 cache, 已知 cache 的读/写时间为 10ns, 主存的读/写时间为 100ns, 取指的命中率为 98%, 数据的命中率为 95%, 在执行程序时, 约有 1/5 指令需要存/取一个操作数, 为简化起见, 假设指令流水线在任何时候都不阻塞。问设置 cache 后, 与无 cache 比较, 计算机的运算速度可提高多少倍?

有 cache:

$$[10 \times 0.98 + (10 + 100) \times 0.02] + [10 \times 0.95 + (10 + 100) \times 0.05] \times \frac{1}{5}$$

$$= 15 \text{ ns}$$

无 cache:

$$100 \times 1 + 100 \times \frac{1}{5} = 120 \text{ ns}$$

提高 8 倍

7.7 一个组相联 cache 由  $2^6$  个存储块组成, 每组包含  $2^2$  个存储块, 主存由 8192 个存储块组成, 每块由  $2^5$  字组成, 访存地址为字地址。问:

$$2^{10} \times 2^3 = 2^{13}, m=13$$

(1) 主存和 cache 地址各多少位? 地址映像是几路组相联?

块内地址 (2) 在主存地址格式中, 区号、组号、块号和块内地址各多少位?

(1)  $2^6 = 64, 2^2 = 4$  主存 13 位, cache 11 位, 4 路组相联  
 $C=6$   $r=2$  块  $n=m+b$   $c+b$   
 主存  $n=m+b=13$   
 cache  $c+b=6+5=11$   
 (2) 区号  $C+1$  7 位, 组号  $r$  4 位, 块号  $r$  2 位, 块内地址  $b$  5 位

- 7.11 设某虚存有如下快表放在相联存储器中,其容量为 8 个存储单元。问:  
按如下 3 个虚拟地址访问主存,主存的实际地址码各是多少?(设地址均为十六进制)

• 180 •

页号	本页在主存起始地址
33	42000
25	38000
7	96000
6	60000
4	40000
15	80000
5	50000
30	70000

页号      页内地址

1	15	0324
---	----	------

80324

2	7	0128
---	---	------

96128

3	48	0516
---	----	------

去主存查找

可能需要重新分配

- 7.12 某程序对页面要求的序列为  $P_3P_4P_2P_6P_4P_3P_7P_4P_3P_6P_3P_4P_8P_4P_6$ 。(1)设主存容量为 3 个页面,求分别采用 FIFO 和 LRU 替换算法时各自的命中率(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?(3)程序运行时,CPU 访问主存的命中率会增加还是减少?

(1) FIFO  $\frac{3}{15} = 20\%$

LRU  $\frac{6}{15} = 40\%$

(2) FIFO  $\frac{6}{15} = 40\%$

LRU  $\frac{9}{15} = 60\%$

(3) 增加