



PS: 基本互连函数:

① 恒等置换: $r(x_{n-1} x_{n-2} \dots x_1 x_0) = x_{n-1} x_{n-2} \dots x_1 x_0$

② 反位置换: $h(x_{n-1} x_{n-2} \dots x_1 x_0) = x_{n-1} x_{n-2} \dots x_0 \overline{x_1}$

③ 方体置换:

方体置换是实现用二进制编码的地址号的第 i 位不同的输入端与输出端的连接, 其表式式为:

$$C_i(x_{n-1} x_{n-2} \dots x_1 x_0) = x_{n-1} x_{n-2} \dots \overline{x_i} \dots x_1 x_0 \quad (9-3)$$

根据式 (9-3), 对于 N 个输入输出的情况, 应有 C_0, C_1, \dots, C_{n-1} , 共有 $n = \log_2 N$ 种方体置换。当 $N=8$ 时, 则有

$$C_0(x_2 x_1 x_0) = x_2 x_1 \overline{x_0}$$

$$C_1(x_2 x_1 x_0) = x_2 \overline{x_1} x_0$$

$$C_2(x_2 x_1 x_0) = \overline{x_2} x_1 x_0$$

(9-4)

④ 均匀洗牌置换

$$\begin{cases} \text{正: } G(x_{n-1} x_{n-2} \dots x_1 x_0) = x_{n-2} \dots x_1 x_0 x_{n-1} \\ \text{逆: } G(x_{n-1} x_{n-2} \dots x_1 x_0) = x_0 x_{n-1} x_{n-2} \dots x_1 \end{cases}$$

⑤ 加減 2^i 置换: $pm_{2+i}(x) = (x + 2^i) \bmod N$

$$pm_{2-i}(x) = (x - 2^i) \bmod N$$

⑥ 蝶形置换: $\beta(x_{n-1} x_{n-2} \dots x_1 x_0) = x_0 x_{n-2} \dots x_1 x_{n-1}$

SMP 特点与优点

对称多处理机系统具有如下特点:

1. 这样的系统是由两个以上的多个相同的处理机构成。
2. 多个处理机通过总线或其他互连方式连接在一起。图 9.31 中, 是利用系统总线将这些处理机连在一起。
3. 多个处理机共享同一主存储器。并且, 每一个处理机访问主存储器的时间是相同的, 也就是一致的。这或许就是一致性存储器访问计算机(UMA)的来历。
4. 所有的处理机通过相同的通道或不同的通道共享 IO 设备。
5. 每一处理机都能完成相同的功能, 这或许是对称多处理机中对称的由来。
6. 整个对称多处理机系统是在一个集中的操作系统统一管理下工作。操作系统能够为每一处理机按排进程或线程, 对各处理机的工作进行统一地调度与控制。

对称多处理机系统相对于单个处理机的计算机系统, 有如下突出的优点: \leftarrow

1. SMP 包含多个处理机, 而这些处理机可以并行工作。因此, SMP 具有比单处理机构成的系统更高的性能。 \leftarrow
2. SMP 具有高的可靠性和高的可用性。由于系统中包含多处理机, 当某一处理机出现故障时, 其他处理机仍可以工作。 \leftarrow
3. SMP 系统具有很好的扩展性能, 用户可以比较容易地在系统中增加一个处理机, 只要将相同的处理机连接在系统总线即可达到目的。从而较方便地增强系统的性能。 \leftarrow
4. 可以灵活地按照用户的需求构成不同数量处理机的 SMP 系统。 \leftarrow

MPP 特点 & 优点

大规模并行处理(MPP)系统的特点如下:

- (1) MPP 是由大量的结点构成, 结点数目少则数百多达上万。
- (2) MPP 的结点可由一个或多个带有 Cache 的处理机构成。具体构成形式有以下三种:

- ① 每个结点只包含一个处理器。
- ② 每一个结点包含一台 SMP。
- ③ 每一个结点由 CC-NUMA 系统构成。

- (3) 构成 MPP 的结点可以是同构的也可以是异构的。

(4) MPP 各结点的存储器是各自独立的, 每个结点只能直接存取自己结点的本地存储器, 不能直接访问非本地结点的存储器。若需访问非本地结点的存储器, 则需使用消息传递的方式来达到目的。这使得编程困难而且增加通信开销。

(5) 每个结点都有自己的操作系统, 各结点可以有各不相同的操作系统。用户可将任务提交给任务管理系统, 由该系统负责任务的调度, 以便使系统负载平衡。在 MPP 系统中也允许用户登录到某一特定结点上或指定在某一结点上运行某任务。

4. Cluster 系统与 MPP 系统的不同

由图 9.38 可以看到, Cluster 系统与 MPP 系统十分类似, 但存在一些差别。两者的主要差别如下所述。

集群是一种分布存储的并行系统, 属于 NURMA 结构, 各结点通信主要使用消息传递方式。集群与 MPP 的主要区别是:

集群的每个结点都是一个完整的计算机系统, 包括 CPU、内存、硬盘, 但可能没有显示器、键盘、鼠标等外围设备, 这样的结点称为“无头工作站”; MPP 的每个结点内不一定有硬盘。集群的结点间通常使用低成本的商品化网络相连, 如以太网、ATM、Myrinet 等, 而 MPP 使用专门定制的网络, 这个特征被认为是集群与 MPP 最主要的区别。集群结点与系统级网络的网络接口是连接到结点内的 I/O 总线上的, 属于松耦合; 而 MPP 的网络接口是连到结点内的存储总线上的, 属于紧耦合。集群的每个结点上驻留有完整的操作系统; 而 MPP 的结点内通常只有操作系统的微内核。

(1) 超标量处理机的相对性能最高，其次是超标量超流水线处理机，超流水线处理机的相对性能最低。主要有三方面原因：

- 超标量处理机在每个时钟周期的开始就同时发射多条指令，而超流水线处理机则要把一个时钟周期平均分成多个流水线周期，每个流水线周期发射一条指令，指令之间的启动延时比超标量处理机大；
- 条件转移造成的损失，超流水线处理机比超标量处理机大；
- 在指令执行过程中的每一个功能段，超标量处理机都设置多个相同的操作部件，而超流水线处理机只是把同一条指令执行部件分解为多个流水级。因此，超标量处理机指令执行部件的冲突比超流水线处理机小。

解：① 超标流水线：在指令译码段检测到分支指令后，向目标地址确定前取出所有紧跟分支指令之后的所有指令；当分支指令从执行段退出，确定新PC值后，流水线依据新PC值填充流水线。

② 预取分支目标：当分支指令被识别后，除了顺序指令，分支目标也被预取，并保存到分支预取缓冲。

③ 循环缓冲器：循环缓冲器是一种容量小，速度高的存储器，保存最近几条顺序预取的指令。若分支发生，硬件首先检查缓冲器，若循环缓冲器中有分支目标，则下一条指令从缓冲器获取。

④ 分支预测：

1. 静态分支预测：在编译阶段完成。预测方法包括：

1.1 预测分支不会发生 { 预测成功：分支失败，流水线正常执行
预测失败：分支成功，需操作代替已取指令，
1.2 预测分支总会发生 { 预测成功：立刻从目标地址处取指令

1.3 由编译器预测：由编译器认为转移发生时，将指令中某些位置

1.4 预测法。

2. 动态分支预测：在执行阶段完成。通过记录分支指令近期运行历史信息，作为预测依据，提高分支预测准确度。

⑤ 延迟分支：把分支指令延迟为长度为 n 的分支指令，后跟 $n-1$ 延迟槽。流水线遇到分支指令时，按正常方式处理，同时执行延迟槽中指令。

6.7 流水 CPU 中的主要问题是 结构 相关, 数据 相关和 控制 相关; 为此需要采用相应的技术对策, 才能保证流水畅通而不断流。

定义:

- (1) 结构相关。当硬件资源满足不了流水线同时重叠执行指令的要求, 而发生资源冲突时, 出现结构相关。
- (2) 数据相关。当一条指令需要用到前面某条指令的结果, 而前面的指令尚在流水还未产生结果, 因而不能重叠执行时, 发生数据相关。 3种
- (3) 控制相关。当流水线遇到分支等转移类指令或其他能够改变 PC 值的指令时, 产生控制相关。

解决方法:

① 结构相关:

1. 增加资源副本

可以设计数据存储器 and 指令存储器, 流水线的取指段与数据访存段就可以通过两个独立的通路同时访问两个独立的存储器。

2. 延迟 (或暂停) 流水线

通过延迟 (或暂停) 流水线的冲突段或在冲突段插入流水线气泡 (气泡在流水线中只占资源不做实际操作), 使各段“轮流”使用资源。

3. 改变资源以便它们能并发的使用

在相邻近的指令间尽可能不使用相同的资源。例如, 在相邻的 m (流水线的段数) 条指令中, 不相关的数据尽量使用不同的寄存器; 或通过程序再设计或寄存器重命名技术来改变寄存器资源。

② 数据相关

1. 采用定向技术

定向通道是指在某些流水线段之间设置的直接连接通路。当定向硬件检测到前面某条指令的结果操作数就是当前指令的源操作数时, 控制逻辑会将前面那条指令的结果从其产生的地方直接连接到当前指令所处的位置。

2. 增加专用硬件

加流水线互锁硬件。互锁硬件先要检测流水线中指令的数据相关性, 当互锁硬件发现数据相关时, 使流水线工作停顿下来, 直到相关消失为止。

3. 利用编译器

某些系统的编译器可以对指令重新排序或插入空操作指令, 使得加载任何冲突数据的操作被延迟, 但对程序逻辑或输出不受影响, 这种技术称为流水线调度或指令调度。

PCI总线特点:

- ① 高性能: PCI总线 数据传输速率高, 时钟频率 33MHz/66MHz.
可进行64位传输 \Rightarrow 528 MB/s, 同时兼容32位与64位电路板
- ② 总线设备工作与CPU相对独立: PCI总线控制器可以缓冲数据
因此PCI工作与CPU不同步, 可以支持不同型号CPU, 具有更长生命周期
- ③ 即插即用: 在PCI总线上的电路板, 插在PCI总线上便可以工作。
- ④ 支持多主控设备: 在PCI总线上的设备均可提出总线请求,
通过PCI仲裁器允许该设备成为主控设备, 实现主控设备
与从属设备点对点数据传输。
- ⑤ 错误检测及报告: PCI可以对传送地址与数据进行奇偶
校验, 报告错误发生。
- ⑥ 两种电压环境: 5V或3.3V均可工作。
- ⑦ 两种兼容卡槽: 长卡, 短卡 (分别64位, 32位)
分类: PCI-X, PCI-E, Mini PCI, Low Profile PCI

PS:

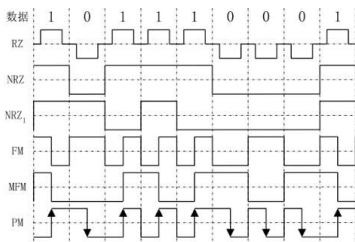


图8.12 磁记录波形 (写入电流或磁化强度)

$$R_{RZ} = 1$$

$$R_{NRZ} = \frac{1}{\infty} = 0$$

$$R_{NRZi} = \frac{1}{\infty} = 0$$

$$R_{FM} = \frac{0.5}{1} = 0.5$$

$$R_{MFM} = \frac{1}{2} = 0.5 \text{ (101时)}$$

$$R_{PM} = \frac{0.5}{1} = 0.5$$

$$\eta_{RZ} = \frac{1}{2} = 0.5$$

$$\eta_{NRZ} = \frac{1}{1} = 1$$

$$\eta_{NRZi} = \frac{1}{1} = 1$$

$$\eta_{FM} = \frac{1}{2} = 0.5$$

$$\eta_{MFM} = \frac{1}{1} = 1$$

$$\eta_{PM} = \frac{1}{2} = 0.5$$

-归零制(RZ): 0为负、1为正, 两位信息之间电流归零

-不归零制(NRZ): 只有0变1和1变0是电流改变

-见1就变不归零制(NRZ~1~): 只有记录1时电流改变

-调频制(FM): 每位改变一次, 记录1时中间翻转一次

-改进调频制(MFM): 记录1时只中间改变一次, 记录2个及以上0时每位改变一次

-调相制(PM): 用不同电流方向表示0或1

自同步能力: 从单个磁道读出的脉冲序列中提取同步时钟脉冲的难易程度。

R = 最小磁化翻转间隔 / 最大磁化翻转间隔

编码效率 (记录密度): 位密度与最大磁化翻转密度之比。

η = 位密度 / 最大磁化翻转密度

统一编址优缺点:

✓ 优点:

- ✚ 可以用访问存储器的指令来访问 IO 端口, 访问存储器的指令功能比较齐全, 可以实现直接对 IO 端口内的数据进行处理;
- ✚ 可以将 CPU 中的 I/O 操作与访问存储器操作统一设计为一套控制逻辑, 简化内部结构, 同时减少 CPU 的引脚数目

✓ 缺点:

- ✚ 由于 IO 端口占用了一部分存储器地址空间, 因而使存储地址空间减小;
- ✚ 由于利用访问存储器的指令来进行 IO 操作, 指令的长度通常比单独 IO 指令要长, 因而执行的时间也较长

分开编址的优缺点:

✓ 优点:

- ✚ IO 端口不占用存储器地址, 不会减少用户的存储器地址空间
- ✚ IO 指令编码短, 执行速度快
- ✚ IO 指令的地址码较短。地址译码方便
- ✚ 采用单独的 I/O 指令, 使程序中 I/O 操作和其他操作层次清晰, 便于理解

✓ 缺点: 缺

8.28 说明中断的一般过程, 若某中断一次向外设传送一个字节, 从请求到返回整个中断处理过程平均需要 $50\ \mu\text{s}$ 。试决定利用中断可向外设传送的最大速率为多少?

中断过程是指从中断源提出中断请求, 到中断被 CPU 响应并被处理, 最后返回到被中断的程序的全过程。经历的处理步骤包括:

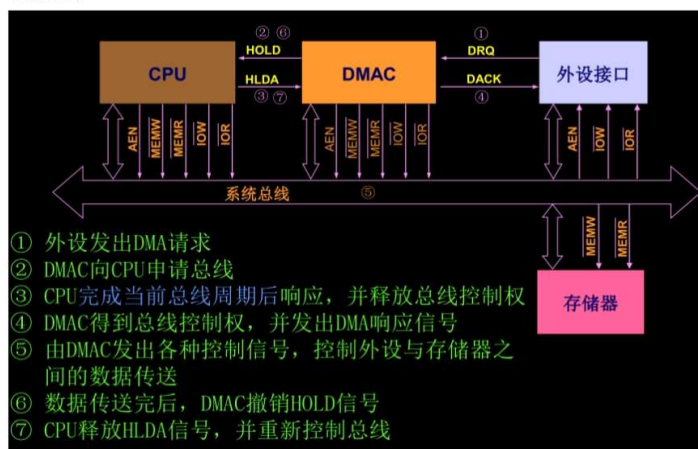
- (1) 中断请求。需要服务的中断源向 CPU 发出中断请求;
- (2) 中断判优。由中断优先权判决电路识别中断请求的中断源, 当同时有多个中断源提出请求时, 按一定的优先级顺序选择响应中断;
- (3) 中断响应。当满足 CPU 响应中断的条件时, CPU 获取响应的中断信息, 即中断类型码, 并自动保护某些现场信息, 从而转入中断服务处理程序的过程;
- (4) 中断处理。执行中断服务程序, 为发出中断请求的中断源服务;
- (5) 中断返回。恢复现场信息, 返回被中断的程序。

向量

8.30 说明中断优先级控制所要解决的两个主要问题。

- 1、当不同优先级的多个中断源同时提出中断请求时, CPU 首先响应最高优先级的中断源;
- 2、当 CPU 正在对某一中断源服务时, 有比它优先级更高的中断源提出中断请求, CPU 能够在正在执行的该中断源服务程序而去对优先级更高的中断源进行服务, 服务结束后再返回原优先级较低的中断源服务程序继续执行。

区别	中断方式	DMA方式
请求对象	程序控制权	总线使用权
响应时刻	指令周期结束	CPU周期结束
CPU的作用	全程控制，硬件控制逻辑和中断处理程序的结合	仅参与DMA的启动和结束
处理异常的能力	具有处理异常能力	仅进行数据传送
现场的保护	需要保护CPU现场	不会改变现场
优先级	比DMA低	比中断高
外设与主存的连接方式	需经过CPU	数据通路直接连接



4. 若 8088 系统中断向量表的地址 00120H~00123 内容如下图所示, 则其对应的中断类型(向量)码、中断向量分别是多少?

00120H	12H	} IP
00121H	34H	
00122H	56H	} CS
00123H	78H	

高 15 2 字节
低 16 2 字节

中断向量码为: $\frac{120H}{4} = 48H$

中断向量 CS:IP = 7856H:3412H

7. 中断控制器 8259 的固定优先级是如何安排的? 若 8259 中 IR1 的中断向量号为 49H, 则 IR6 的中断向量号是多少?

固定优先级: $IR0 > IR1 > \dots > IR7$
49H

IR1: 49H

IR6: 4EH

6. SRAM 和 DRAM 对比

(1) SRAM:

- 用稳态电路存信息, 不需刷新;
- 相对于 DRAM, 速度快、集成度低、功耗大、成本高

(2) DRAM:

- 用暂态电路(电容)存信息, 需刷新;
- 地址采用复用技术, 分行列两次加载, RAS、CAS
- 相对于 SRAM, 速度慢、集成度高、功耗小、成本低

- 刷新(再生)方法: 集中式、分散式、异步式
- 刷新控制电路: 刷新计数器、刷新/访存裁决、刷新控制逻辑, 是 CPU 和 DRAM 的接口

可擦除：通过熔丝的击穿与连接进行模拟。EPROM具有一种特殊的场效应管，其浮栅对外连接。在场效管的栅级之间加高压，则瞬间击穿场效应管，高压电子突破绝缘层进入浮栅，此时，外界高压去除，电子存放在浮栅中，模拟熔丝连接。通过紫外线照射，电子浮栅中的电子获得大量能量，突破绝缘层，实现放电，模拟熔丝击穿。

