

16-17-2学期计算机组成原理期末复习

学习资料

公式整理

- 计算机性能指标

$$CPU\text{时间}T_{CPU} = \left[\sum_{i=1}^n (I_i * CPI_i) \right] * T_c$$

- I_i : 第*i*种机器指令
- CPI_i : 执行第*i*种指令所需要的时钟周期数
- T_c : CPU时钟周期, $T_c = 1/f_c$
- f_c : CPU主频, 节拍频率

$$\text{每秒百万条指令数} MIPS = \frac{n\text{个程序指令总}}{n\text{个程序运行总秒数}} \times 10^{-6}$$

$$\text{每秒百万次浮点数操作} MFLOPS = \frac{n\text{个程序浮点数操作总次数}}{n\text{个程序运行总秒数}} \times 10^{-6}$$

- 定点整数

$$[X]_{\text{原}} = \begin{cases} X, & X \in [+0, 2^{n-1} - 1] \\ 2^{n-1} - X, & X \in [-2^{n-1} + 1, -0] \end{cases}$$

$$[X]_{\text{补}} = \begin{cases} X, & X \in [0, 2^{n-1} - 1] \\ 2^n + X, & X \in [-2^{n-1}, -1] \end{cases}$$

- 定点小数

$$[X]_{\text{原}} = \begin{cases} X, & X \in [+0, 1) \\ 1 - X, & X \in (-1, -0] \end{cases}$$

$$[X]_{\text{补}} = \begin{cases} X, & X \in [0, 1) \\ 2^n + X, & X \in [-1, 0) \end{cases}$$

- 海明校验码[SEC]

$$n + k + 1 \leq 2^k$$

- Cache

$$\text{命中率 } P = \frac{N_C}{N_C + N_m}$$

$$\text{平均访存时间 } T_a = PT_C + (1 - P)T_m$$

$$\text{访问效率 } e = \frac{T_C}{T_a}$$

- 流水线

$$\text{吞吐率 } T_P = \frac{n}{T_{\text{流水}}}$$

$$\text{加速比 } S_P = \frac{T_{\text{串行}}}{T_{\text{流水}}}$$

$$\text{效率 } E = \frac{nm\Delta t}{m(m + n - 1)\Delta t}$$

概论

- 冯·诺依曼计算机特点

1. 由运算器、存储器、控制器、输入设备、输出设备组成
2. 指令和数据以同等地位存储在存储器内，按地址访问
3. 指令和数据都用二进制表示
4. 指令由操作码和地址码组成
5. 机器以运算器为中心（现代计算机改进：以存储器为中心）
6. 以存储程序原理为核心

- 存储程序原理

程序和数据预先存放在存储器中，机器工作时自动按照程序的逻辑顺序从存储器中逐条取出指令并执行

寄存器

- 状态条件寄存器[Program Status Word, PSW]

保存由算术指令和逻辑指令运行或测试的结果建立的各种条件码内容

- 运算结果为零标志(Z)
- 运算结果进位标志(C)
- 运算结果正负标志(S)
- 运算结果溢出标志(V)
- 中断标志(I)
- 方向标志(D)

存储器

- 动态存储器刷新方式

1. 集中式刷新
前一段时间用于读写，后一段时间集中刷新所有行
2. 分散式刷新
每次读写后刷新一行

3. 异步刷新

若干次读写后刷新一行，保证在在一个间隔内能刷新全部行数

4. 透明刷新

在取指译码阶段刷新

● Cache原理

1. 程序访存的时间局部性和空间局部性原理

2. CPU访问Cache时按照主存地址访问

3. 主存地址长度与CPU最大主存容量有关，与实际主存空间大小无关

● Cache替换算法(只有全相联映像和组相联映像需要)

1. 随机法[RAND]

2. 先进先出法[FIFO]

3. 最近最少使用法[LRU]

● Cache写策略

1. 写直达法[WT]

写Cache的同时也写主存

2. 写回法[WB]

只有在Cache块被替换时才更新主存信息

指令系统/控制器

● 指令集

1. 复杂指令集[CISC] 变长编码

2. 精简指令集[RISC] 定长编码

● 指令周期

1. 取指周期

2. 执行指令周期

● 微指令格式

1. 水平型微指令格式

同时定义并执行多个微命令的微指令格式

2. 垂直型微指令格式

同时定义并执行一个或几个微命令的微指令格式

- 微指令编码方式

1. 直接编码方式

编码中每一位对应一个微命令

2. 字段直接编码方式

每个子字段中的每个编码表示一组互斥的微命令中的一个微命令

- 微指令排序

1. 计数器法：由 μ PC产生下一条指令地址

2. 下地址字段法：在指令编码中加入下地址字段

3. 测试网络法/断定法：

- 硬布线控制器/组合逻辑控制器

- 输入信号：

- 来自指令译码器的输出I
- 来自执行部件的反馈信息B
- 来自时序发生器的时序信号T

- 寻址方式

1. 立即寻址

操作数包含在指令内，不寻址

2. 直接寻址

$EA=A$

3. 间接寻址

$EA=(A)$

4. 寄存器直接寻址

$EA=R_i$

5. 寄存器间接寻址

$EA=(R_i)$

6. 相对寻址

$EA=A+(PC)$

7. 基址寻址

$EA=A+(R_{\text{基址}})$ ，R不变，A可变，用于存储空间分配

8. 变址寻址

$EA=A+(R_{\text{变址}})$ ，A不变，R可变，用于数组访问

$EA=(SE)+(R_{\text{变址}})$

$$EA = (IR_{[地址段]}) + (R_{变址})$$

9. 隐含寻址

EA=(PC)、EA=SP等

- 采用不同寻址方式的目的

1. 缩短指令长度
2. 扩大寻址空间
3. 提高编程的灵活性

- 指令流水停顿

1. 结构相关：不同指令在同一拍要使用同一条数据通路

- 增设存储器
- 增设数据存储缓冲器
- 增设指令预取缓冲器
- 采用多体交叉存储器

2. 数据相关：写后读/写后写（数据过期）

- 后推法：读操作延迟至写完成后
- 重定向法：读操作数据地址重新定向至写操作来源部件
- 乱序流动法：先执行后续的无关操作

3. 控制相关：遇到转移指令，在转移指令未执行完前无法知道地址

- 转移延迟：将与转移无关的指令重新排序至转移指令后的延迟槽内先执行
- 转移预测：静态转移预测/动态转移预测
- 冻结法：暂停转移指令下一条的取址操作

总线

- 总线事务

1. 总线请求和仲裁
2. 寻址
3. 数据传送
4. 结束

- 集中式总线仲裁

- 1. 菊花链式查询方式
- 2. 计数器定时查询方式
- 3. 独立请求方式
- 分布式总线仲裁
 - 1. 自举式

I/O接口

- 信息传送控制方式
 - 1. 程序查询方式

CPU通过程序不断查询I/O设备的状态位来判断是否进行下一步信息交换
 - 2. 程序中断方式

I/O设备通过发送中断请求来通知CPU准备就绪

 - 中断请求：表示有亟待处理事件的信号
 - 中断源：能够产生中断请求的部件
 - 中断服务程序：处理中断请求的程序
 - 中断响应：CPU从当前程序转入中断服务程序的过程
 - 条件
 - 1. CPU处于开中断状态
 - 2. 无DMA请求
 - 3. 当前正在执行的指令执行完
 - 4. 有中断请求
 - 步骤
 - 1. 关中断
 - 2. 保护断点信息
 - 3. 识别中断源
 - 中断服务：CPU执行中断服务程序的过程
 - 中断返回：CPU从中断服务程序转回当前程序的过程
 - 中断处理：中断服务+中断返回
 - 步骤
 - 1. 保护现场及旧屏蔽字
 - 2. 设置新屏蔽字

3. 开中断
4. 中断服务
5. 关中断
6. 恢复现场和旧屏蔽字
7. 清除中断请求
8. 开中断
9. 返回

■ 中断过程：中断响应+中断服务+中断返回

3. DMA方式

直接存储器存取方式

■ 过程

1. 外设发出DMA请求
2. DMAC发出总线请求
3. CPU总线响应，设置DMAC各项参数
4. CPU脱离总线，DMAC接管总线
5. 外设准备数据
6. DMAC传输数据
7. DMAC发出DMA结束信号，进行后续处理（数据校验等）
8. DMAC脱离总线，CPU接管总线

■ CPU/DMAC访存冲突

1. CPU停止法：DMA传输时CPU脱离总线直至DMA结束
2. 周期挪用/窃取法：DMA传输时CPU让出1-2个总线事务周期给DMAC
3. 交替分时访问法：每个存储周期分为两个时间片，一个给CPU访存，一个给DMA

■ 情境

1. CPU不需要访存，不冲突
2. CPU正在访存，DMA需要等到CPU访存结束后才能进行
3. CPU和I/O设备同时需要访存，I/O访存优先级高于CPU访存，故CPU要延迟1-2个存取周期，等DMA完成后再访存