

# 1 考试

## 1.1 题型

选择 \*20 解答综合

## 1.2 大纲

1. 运算器 (2)
2. 控制器 (4, 5)
3. 存储器 (3)
4. 总线 (6)
5. I/O (8)

### 1.2.1 概论

补充图

## 2 概论

### 2.1 性能指标

**吞吐量** 一台计算机在某一时间间隔类能够处理的信息量

**响应时间** 从输入有效到系统产生响应之间的时间度量

**利用率** 实际使用的时间占比

**处理机字长** 运算器中一次能够完成二进制数运算的位数 (32, 64...)

**总线宽度** 数据总线二进制位数

**存储器容量** 存储器中所有存储单元的总数目 (KB, MB, GB, TB)

**存储器带宽**

**主频** 2GHz

**周期** 1ns

**CPU 执行时间** s

**CPI** 每条指令周期数

**MIPS** Million Instruction Per Second, 每秒百万条定点数指令数

**FLOPS** 每秒浮点数指令数

**MFLOPS** 每秒百万浮点数指令数

### 2.2 计算机硬件组成

**冯诺依曼架构** 存储程序并按地址顺序执行. 组成部分: 运算器, 控制器, 存储器, 输入设备, 输出设备

#### 2.2.1 控制器

**取值周期**

**执行周期**

**指令流** 取值周期从内存读出的信息流

**数据流** 执行周期从内存读出的信息流

## 3 运算器

### 3.1 定点数

表示方法  $x_n$  (符号),  $x_{n-1} \dots x_0$  (尾值, 尾数)

纯小数  $\pm.x_{n-1} \dots x_0$

纯整数  $\pm x_{n-1} \dots x_0.$

原码 反 (补码) + 1

补码 反码 + 1,  $[-128]_{\text{补}} = 1000\ 0000$

#### 3.1.1 补码加减法溢出检测 (双符号位)

00: 正, 11: 负, 01: 正溢出, 10: 负溢出

$x = -1100, y = -1000$

$110100 + 111000 = 101100$ , 负溢出

#### 3.1.2 乘法

“加法-移位”类似人工运算过程

不带符号的阵列乘法器 并行操作

带符号的阵列乘法器  $|x| * |y|$

#### 3.1.3 除法

加减交替

### 3.2 浮点数

$$N = 2^e \cdot M$$

早期计算机  $E_s, E_{m-1} \dots E_0, M_s, M_{n-1} \dots M_0$

IEEE754(32bit)  $S_{31}, E_{30-23}, M_{22-0}, E = e + 127$

IEEE754(64bit)  $S_{63}, E_{62-52}, M_{51-0}, E = e + 1023$

IEEE754 规格化表示  $(-1)^S \times (1.M) \times 2^e$

### 3.2.1 加减法

1. 对阶
2. 尾数求和
3. 浮点数规格化
4. 舍入问题
5. 检查阶码是否溢出

# 4 存储器

## 4.1 分类

- 存储介质
  1. 半导体存储器
  2. 磁表面存储器
- 存取方式
  1. 随机存储器
  2. 顺序存储器 (磁带)
- 内容可变性
  1. ROM
  2. RAM
- 信息易失性
  1. 易失性
  2. 非易失性
- 系统中的作用
  - 内存
    - \* 主存
    - \* 告诉缓存
    - \* 控制存储器
  - 外存
    - \* 辅存

## 4.2 技术指标

**字存储单元** 存放一个机器字的存储单元，相应的单元地址叫字地址

**字节存储单元** 存放一个字节的单元，相应的地址称为字节地址

**存储容量** 指一个存储器中可以容纳的存储单元总数

**存取时间 (存储器访问时间)** 指一次读操作命令发出到该操作完成，将数据读出到数据总线上所经历的时间。通常取写操作时间等于读操作时间。

**存储周期** 指连续启动两次读操作所需间隔的最小时间

**存储器带宽** 单位时间里存储器所存取的信息量

## 4.3 RAM

优点：存储速度快，存储体积小，价格低

缺点：属于易失型存储器，断电后不能保存信息

### 4.3.1 SRAM

静态存储元阵列

存储体，读写电路，地址译码电路，驱动电路，控制电路，读写周期

### 4.3.2 DRAM

靠电容存储电荷的原理来寄存信息的

#### DRAM 的刷新

刷新周期 = 存储周期

按行刷新

- 集中刷新
- 分散刷新: 刷新周期绑定存储周期
- 异步刷新: 定时刷新

## 4.4 扩展容量

### 4.4.1 物理扩展

位扩展，字扩展，字位同时扩展

### 4.4.2 虚拟存储器

## 4.5 加速存储

### 4.5.1 并行存储

1. 可以并行操作的双端口存储器: 冲突
2. 多模块交叉存储器
  1. 顺序方式在一个模块内, 程序是从低位地址连续存放; 易扩充容量; 故障局部性
  2. 交叉方式对连续单元存取, 则多个模块并行工作; 不易扩展; 故障全局性

### 4.5.2 Cache

命中率  $h$   $h = \frac{N_c}{N_c + N_m}$

cache 访问时间  $t_c$

主存访问时间  $t_m$

平均访问时间  $t_a$

访问效率  $e = \frac{t_c}{t_a}$

$r$   $r = \frac{t_m}{t_c}$

#### 4.5.2.1 地址映射

1. 全相连
2. 直接方式
3. 组相联方式

标记 (tag)  $s - r$

行  $r$

字地址  $w$

#### 4.5.2.2 替换方式

1. LFU
2. LRU, 命中至 0, 其余加 1
3. 随机替换

# 5 指令系统

## 5.1 指令格式

**操作码字段** 表征指令的操作特性与功能

**地址码字段** 指示操作数的地址

地址码分类:

- 三地址
  - SS
  - RR
  - RS
- 二地址
- 单地址
- 零地址

**等长指令** 各种指令字长度是相等的，指令字结构简单，且指令字长度是不变的

**非等长指令** 各种指令字长度随指令功能而异，结构灵活，能充分利用指令长度，但指令的控制较复杂

## 5.2 寻址方式

### 5.2.1 指令寻址

1. 顺序寻址
2. 跳跃寻址 (JMP)

### 5.2.2 操作数寻址

- 在指令中
  - 立即 (数) 寻址
- 在寄存器中



- 寄存器寻址
- 在主存中
  - 直接寻址
  - 间接寻址
    - \* 寄存器间接
    - \* 存储器间接
  - 偏移寻址
    - \* 相对寻址:  $(PC) + A$
    - \* 基址寻址:  $(R) + A$
    - \* 变址寻址:  $A + (R++)$
    - \* 基址加变址寻址?

# 6 CPU

## 6.1 功能

- 指令控制: 保证机器按程序规定的顺序取出执行
- 操作控制: CPU 产生每条指令所对应的操作信号, 并把各种操作信号送往相应的部件, 从而控制这些部件按指令的要求进行动作
- 时间控制: 对各种操作的实施时间进行定时
- 数据加工: 对数据进行算术运算和逻辑运算处理

## 6.2 主要寄存器

数据缓冲寄存器 (DR) 暂时存放由内存读出或写入的指令或数据字

指令寄存器 (IR) 保存当前正在执行的一条指令

程序计数器 (PC) 确定下一条指令的地址

地址寄存器 (AR) 保存当前 CPU 所访问的内存单元的地址

通用寄存器 (R0 ~ R3)

状态条件寄存器 (PSW) 保存由算术和逻辑指令的结果建立的各种条件码

## 6.3 指令系统

指令 = 微程序 (ADD) > n 条微指令 (PC -> AR) > m 条微命令 (PC, G, AR)  
= 微操作 (PC, G, AR)

### 6.3.1 指令周期

指令周期 > CPU 周期 > 时钟周期

### 6.3.2 控制方式

- 同步控制方式 (指令的机器周期和时钟周期数不变)
  - 完全统一的机器周期执行各种不同的指令
  - 采用不定长机器周期

- 中央控制于局部控制的结合
- 异步控制方式
  - 每条指令需要多长时间就占多长时间
- 联合控制方式
  - 大部分指令在固定的周期内完成，少数难以确定的操作采用异步方式
  - 机器周期的节拍脉冲固定，但是各指令的机器周期数不固定（微程序控制器采用）

### 6.3.3 微程序

采用存储程序原理，需要执行多条微指令，速度较慢

**硬布线控制器** 用布尔代数写出逻辑表达式，然后用门电路、触发器等器件实现，速度快，不容易修改新添功能

### 6.3.4 微指令

#### 6.3.4.1 微地址转移

**计数器方式** PC 自加 1, 寄存器-PC

顺序控制字段短，机制简单，功能弱，速度慢，灵活性差

**多路转移方式** 一条微指令具有转移多个转移分支的能力

按顺序控制字段的“判别测试”标志和“状态条件”信息来选择其中一个微地址

#### 6.3.4.2 转换微命令

**水平型微指令** [控制字段, 判别测试字段, 下地址字段]

指一次能定义并能并行执行多个微命令的微指令

**垂直形微指令** [操作码字段、源地址字段、目的地址字段]

同时操作的微命令种类和数目加以适当的限制

## 控制字段编码方式

- 直接编码
- 编码表示
- 混合表示

## 6.4 流水 CPU

### 6.4.1 并行技术

1. 时间并行
2. 空间并行
3. 时间并行 + 空间并行

### 6.4.2 结构

1. 指令部件
2. 指令队列
3. 执行部件

### 6.4.3 问题

1. 资源相关
2. 数据相关: RAW, WAR, WAW
3. 控制相关: (控制 PC 的分支指令)

## 6.5 RISC, CISC

RISC 指令系统是 CISC 指令系统的改进，它的最大特点是：

1. 指令条数少
2. 指令长度固定，指令格式和寻址方式种类少
3. 只有取数/存数指令访问存储器，其余指令的操作均在寄存器之间进行
4. 大量通用寄存器
5. 流水线优化

- 奔腾 CPU 是流水 CPU 的 CISC 机器

# 7 系统总线

总线的一次信息传送过程，大致可分为如下五个阶段：请求总线，总线仲裁，寻址（目的地址），信息传送，状态返回（或错误报告）

## 7.1 连接方式

- 单总线

使用一条总线来连接 CPU、主存和 I/O 设备，结构简单，成本低廉，但运行效率差，总线负荷重。

- 多总线

体现了高速、中速、低速设备连接到不同的总线上同时进行工作，以提高总线的效率和吞吐量，而且处理器结构的变化不影响高速总线。

通过桥 CPU 总线、系统总线和高速总线彼此相连。桥实质上是一种具有缓冲、转换、控制功能的逻辑电路。

## 7.2 结构

### 7.2.1 早期

CPU 是唯一的主控者，虽然后来加了 DMA，但仍然不能满足多 CPU 的环境

1. 地址总线

一般为单向三态，用来传输地址信息，地址线的位数决定了微机系统的寻址范围。主存和 IO 设备接口电路的地址

2. 数据总线

一般为双向三态，用来传输数据，数据总线的宽度（位数）反映了总线传输数据的速率

3. 控制总线

用来传输控制或状态信号。它根据使用条件不同，有的为单向，有的为双向传输，有的是三态，有的是非三态。控制总线代表了总线的控制能力

### 7.2.2 现代

1. 数据传送总线 (地址, 数据, 控制线)
2. 总裁总线
3. 中断和同步总线
4. 共用线

## 7.3 传输方式

- 串行传送
- 并行传送
- 分时传送

## 7.4 总裁

1. 集中式仲裁
  1. 链式查询方式: 线路简单, 易扩充, 对故障敏感, 不灵活
  2. 计数器定时查询方式: 可程序设置
  3. 独立请求方式: 速度快, 灵活可程序设置, 线路复杂
2. 分布式总裁

## 7.5 定时

- 同步定时: 事件出现在总线上的时刻由总线时钟信号来确定, 总线周期的长度是固定的
- 异步定时: 后一事件出现在总线上的时刻取决于前一事件的出现, 建立在应答式或互锁机制基础上, 不需要统一的公共时钟信号。在异步定时中, 总线周期的长度是可变的

## 7.6 实现

### 7.6.1 PCI

与处理器无关的**高速外围总线**，采用**同步时序协议**和**集中式总裁策略**，具有**自动配置**的能力

猝发式传送。利用桥实现总线间的猝发式传送，使所有存取按需出现在总线上。允许多条总线并行工作。

低成本小系统，微型机

### 7.6.2 InfiniBand 标准

瞄准了高端服务器市场的最新 **I/O 规范**，它是一种基于开关的体系结构，可连接多达 **64000** 个服务器、存储系统、网络设备，能替代当前服务器中的 PCI 总线，数据传输率高达 **30GB/s**。因此适合于高成本的**较大规模计算机系统**。



## 8 IO 系统

### 8.1 程序查询方式

简单, CPU 效率低, CPU、各外设要串行

### 8.2 程序中断方式

并行工作, CPU 利用效率略高, CPU、各外设可并行

准备就绪标志 RD 中断触发源

允许中断触发器 EI 0 不允许

中断请求触发器 IR 1 发出中断请求

中断屏蔽触发器 IM 1 屏蔽

#### 8.2.1 多级中断

### 8.3 直接内存访问 (DMA) 方式

并行运行, 外设直接访问内存, CPU 利用效率更高, 速度快, 硬件线路复杂

中断向量地址 中断服务程序入口地址

#### 8.3.1 传送方式

每传送一个数据

1. 停止 CPU 访问内存, 简单, 但内存利用率低 (IO 设备慢)
2. 周期挪用, 适用于 IO 设备读写周期大于内存存储周期的情况, 申请建立归还控制权需要占用周期
3. DMA 与 CPU 交替访问, 不需要协议过程, 透明 DMA, 硬件复杂

### 8.3.2 分类

1. 选择型 DMA: 物理上可以连接多个设备, 而在逻辑上只允许连接一个设备. 高速设备
2. 多路型 DMA: 允许这些外围设备同时工作, 各设备以字节交叉方式进行数据传输. 慢速设备

## 8.4 通道方式 (IOP)

实现对外设的统一管理和外设与内存之间的数据传送

组织外围设备和内存进行数据传输; 控制外围设备

通道是一个特殊功能的处理器, 它有自己的指令和程序专门负责数据输入输出的传输控制, 而 CPU 将“传输控制”的功能下放给通道后只负责“数据处理”功能

### 8.4.1 通道类型

- 选择通道
- 多路通道
  - 数组多路通道, 传输数据独占, 控制性操作可以断开, 高速设备
  - 字节多路通道, 大量低速设备

### 8.4.2 实例

#### 8.4.2.1 IEEE1394

串行 I/O 标准接口, 与 SCSI 并行 I/O 标准接口相比, 更高的数据传输速率, 更强的数据传输实时性, 体积小, 连接方便

#### 8.4.2.2 SCSI

系统级接口, 主适配器和智能设备控制器之间的并行 I/O 接口. SCSI-3 允许连接 16 个 (8 个), 可支持 16 位数据传输