（科目：　　）**数　学　作　业　纸**

编号：　　班级：　　姓名：　　第 1 页

计算机组成原理

一. 计算机系统概述

1. 计算机的组成

① 存储器

地址：以字节/字为单位，存储单元序号的二进制码.

字节：8个二进制位，计算机按字节编址

字：不同计算机不同字长.

字长：CPU一次可直接处理的二进制位数.

取决于通用寄存器位数和数据总线宽度.

对准字：字存于偶数地址单元 } 一个总线周期.

对准双字：双字存于被4整除地址单元 }

小端格式：低位字节存低/高地址

(字符串按顺序存).

X86：小端；风：大端；MIPS：两种.

② 运算器 ALU

③ 数据通路.

CPU的数据通路：ALU、寄存器及连接它们的内部总线 ⇒

④ 控制器.

基本组成：操作控制器OC，指令译码器ID

指令寄存器IR、程序计数器PC，

内存地址寄存器MAR、内存数据寄存器MDR. ⇒

计算机操作的核心：取指-译码-执行循环

⑤ 总线

总线：连接多个功能部件的一组公共信号线

地址总线AB → 寻址能力

数据总线DB → 一次传数据宽度.

控制总线CB

2. 计算机系统

硬件：组机所需物理部件与设备. 功能：执行指令.

软件：控制机器工作的程序和数据.

机器语言：全部二进制机器指令.

汇编语言：助记符形式的指令的集合.

汇编器：汇编语言→机器语言.

编译器：高级语言→机器语言.

解释器：顺序检查每句高级语言，直接执行等价的机器语言指令序列.

高级语言在运行二进制机器指令的机器上运行的两种途径：编译、解释.

3. 机器性能评价.

响应时间：提交到完成总耗时 → 个人.

吞吐量：单位时间工作量. → 企业级

执行时间：执行特定程序的时间 -1.

(包括等待I/O操作等).

CPU执行时间 = (特定程序的)指令数 · CPI · 时钟周期.

CPU时钟周期数 = (程序)指令数 · 每指令时钟周期CPI

$CPI = \sum_{i=1}^{n} CPI_i \cdot P_i$

优化编译技术：指令数↓

快速电路：时钟周期↓

流水线、超标量：CPI↓

CISC：指令数↓、CPI↑，时钟周期↑

RISC：指令数↑，CPI↓，时钟周期↓

SPEC性能测试指数 = 参考机执行时间 / 测试机执行时间

$MIPS = \frac{\text{指令数}}{\text{执行时间} \times 10^6}$.

不能用MIPS比较指令集不同的计算机，因同一程序的指令数

4. 计算机内信息的表示

机器数：机内使用，各位各为0/1都数值化.

原码：$-(2^{n-1}-1) \sim 2^{n-1}-1$

反码：$-(2^{n-1}-1) \sim 2^{n-1}-1$

补码：$-2^{n-1} \sim 2^{n-1}-1$

模、基数、n位数.

进位：加进减借　无符号运算叫×

溢出：同加得反　有符号运算叫×.

浮点数的表示

规格化处理；阶码偏置($+2^{k-1}-1$).

表数范围：$\pm[1+(1-2^{-m})] \times 2^{(2^k-1)-(2^{k-1}-1)-1}$

表数精度：$2^{1-(2^{k-1}-1)}$

k：阶码位数；m：尾码位数.

压缩BCD：4位二进制；非压…：4 dup(0)+4位二进制

ASCII：7位；Unicode：16位.

二. 指令集.

1. 概述

指令：规定计算机执行特定操作的命令.

指令集/指令系统ISA：all指令.

指令集体系结构ISA：

寄存器组织，存储器组织和寻址方式，I/O系统结构，数据类型及表示，指令集，中断机制，机器工作状态的定义及切换，保护机制 ⇒ 编译器或解释器设计者可见.

向下兼容：低；向前兼容：前.

2. ISA

① 堆栈型：逆波兰.

- 栈位于内存：Push/pop：2　ALU：3
- 栈位于寄存器：Push/pop：1　ALU：0.

② 累加器型：load/store：1　ALU：1

③ 寄存器-存储器型：MOV/ALU：1 or 0.

④ 寄存器-寄存器型：load/store：1　ALU：0.

(装入-存储型)：RISC.

3. 基本指令集.

① 数据传送类：

② ALU

③ 程序控制类：程序调用和返回、中断、循环、跳转分支

④ I/O

⑤ 特权.

4. 寻址方式

形式地址：指令中操作数地址.

有效地址 EA：结合形式地址算物理地址 ⇒.

三. 指令格式设计.

1. 操作码编码.

熵法：$H = -\sum_{i=1}^{n} P_i \log_2 P_i$

冗余：$R = 1 + H/$某种编码平均码长.

① 固定长：$\log_2 n$

② Huffman：不定长前缀.

③ 扩展：定长前缀.

2. 地址码编码

① 直址寻址：指令中只有偏移.

② 寄存器间接寻址：地址存入寄存器.

3. 地址码与操作码的结合.

操作码扩展编码法.

4. CISC 与 RISC.

CISC：复杂指令集计算机.：指令数↓.

多时钟同期；微指令；多指令分存；

有专用寄存器；难以优化编译.

RISC：精简指令集计算机：CPI↓

除访存外单同期；load/store 访存.

较多通用寄存器；硬布线控制逻辑；优化编译.

四. ALU 设计.

1. ALU 概述

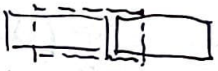
2. 逻辑运算器.

3. 移位器

MSB 决定 算术/逻辑/循环移位.

① 2-1 MUX 阵：0/1 - 0/2 - 0/4 …

② 4-1 MUX 阵：0/4/8/12 - 0/1/2/3 …

③ 漏斗 funnel： 

④ 桶式 barrel：左：左漏斗；下：右漏斗；控制位选列

4. 加减法器

① 行波进位加法器 RCA. 2T/全加器

② 超前进位加法器 CLA.

1) 进位生成 $g_i = a_i \cdot b_i$ } 1T/g.p.

2) 进位传播 $p_i = a_i \oplus b_i$ } 2T/cin

3) 进位 $cin_{i+1} = g_i + p_i \cdot cin_i = g_i + g_{i-1} \cdot p_i + \cdots + cin_0 \cdot \prod_{j=0}^{i} p_j$

4) 和 $sum_i = a_i \oplus b_i \oplus cin_i = p_i \oplus cin_i$ 1T/sum

③ 组内并行、组间串行进位加法器 3T+2T+…+2T+T

④ 组内并行、组间并行进位加法器. cin pipeline

1) 组间进位生成 = 组内进位 - $cin_j \prod p_i$ } 3T/g.p.

2) 组间进位传播 = $\prod$ 组内进位传播 } G.P

3) 组间进位 $_{i+1} = G_i + P_i \cdot$ 组间进位 $_i$ 2T/CIN

4) 和 sum_i 所由各组分别产生. +2T/cin, 1T/sum

⑤ 减法器：$a - b = a + \bar{b} + 1$：$cin_0 = 1$.

五. 数据通路设计

1. 基本概念

主机系统：内存、ALU、控制器.

CPU：运算器、控制器.

数据通路：CPU内各部件间信息传送通路.

数据通路(广义)：CPU、内存、I/O设备间…

数据通路组成：CPU内连线、辅助寄存器

数据通路结构：总线/专用通路.

六. 控制器设计

1. 控制器概述.

① 控制器的主要功能

- 从内存取指，计算下址.
- 指令译码，产生控制信号
- 控制执行步骤，控制数据流动方向.
- 异常处理.

② 控制器组成：控制信号=f(指令、时序、条件码…).

指令周期：指令从取指到完成所时.

机器周期：微操作用时.

时钟周期：组成机器周期的时间单元.

时序控制部件：一节拍~机器周期.

③ 硬连线控制器：

快(组合逻辑)；复杂；难扩充新指令.

④ 微程序控制器：

慢；规整，易扩充新指令.

机器指令→微指令→微操作控制信号.

七. 指令级并行处理 ILP.

1. 指令级并行处理概述.

- 流水线：指令分解 pipeline
- 超标量：指令并行 superscale
- 超长指令字：多字段指令 VLIW.

2. 流水线的基本概念

取指、译码、执行、访存、写回.

瓶颈：流水线中执行时间长的子功能段.

装入时间：第一个任务双输入到输出.

排空时间：最后的任务从输入到输出 maybe hazard

3. 流水线的性能指标：n个任务k级流水.

① 吞吐率：单位时间指令数.

$$TP=\frac{n}{(k+n-1)\Delta t}$$

$$TP_{max}=\lim_{n\to\infty} TP=\frac{1}{\Delta t}$$

$$TP=\frac{n}{\sum_{i=1}^{k}\Delta t_i+(n-1)\max\{\Delta t_i\}}$$

$$TP_{max}=\frac{1}{\max\{\Delta t_i\}}$$

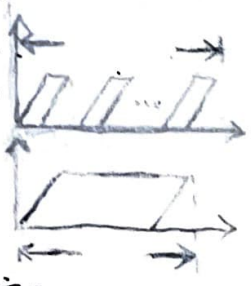
② 加速比：不用流水线用时/流水线用时

$$S=\frac{k\cdot n\cdot\Delta t}{(k+n-1)\Delta t}=\frac{k\cdot n}{k+n-1}$$

$$S_{max}=k.$$

$$S=\frac{n\cdot\sum_{i=1}^{k}\Delta t_i}{\sum_{i=1}^{k}\Delta t_i+(n-1)\max\{\Delta t_i\}}$$

$$S_{max}=\sum_{i=1}^{k}\Delta t_i/\max\{\Delta t_i\}.$$



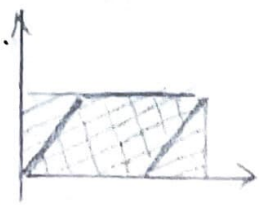
③ 效率：任务时空区/总时空区.

$$E=\frac{k\cdot n\cdot\Delta t}{k\cdot(k+n-1)\Delta t}=\frac{n}{k+n-1}.$$

$$E_{max}=1.$$

$$E=\frac{n\sum_{i=1}^{k}\Delta t_i}{k\left[\sum_{i=1}^{k}\Delta t_i+(n-1)\max\{\Delta t_i\}\right]}$$

$$E_{max}=\sum_{i=1}^{k}\Delta t_i/(k\cdot\max\{\Delta t_i\}).$$



Read after Write：Write → read

（科目：　　）**数　学　作　业　纸**

2^k 组 → 2^{c-k} 路

编号：　　班级：　　姓名：　　第 5 页

4. 流水线中的冒险问题.

① 结构冒险（资源冒险）

现象：多指令同时访问部件

解决：增加运算部件、用哈佛...

② 数据冒险（数据相关）.

现象：多指令访问数据：RAW、WAR、WAW(多ALU)

解决：1）编译检测：编译时遇冒险加nop.

2）流水停顿：取锁部件，遇冒险加气泡.

3）冒险专用通路：前传旁路，减停流水.

4）指令调度：寄存器重命名消WAR、WAW.

③ 控制冒险.

现象：转移指令引起冒险&转移前条件码冒险

解决：转移延迟槽、分支预测...

八. 存储系统(1).

1. 基本概念

访问时间 T_A：申请→读出.

存储周期 T_M：访问→访问.

带宽 B_M = 数据总线宽度(Byte)/T_M.

SRAM：快、贵、贵.

DRAM：慢、密、麻.

局部性原理：

时间局部性：← 循环.

空间局部性：→ 顺序执行；数据聚集存放.

存储系统的层次结构：根据局部性原理，

寄存器 - Cache - 主存(内存) - 辅存(磁盘).

命中：在本层；缺失：在下层；命中率 H.

平均访问时间 $T_A = T_{AM} + (1-H)T_{AL}$，访问相对比 $r = \frac{T_M}{T_{AL}}$，访问效率 $e = \frac{T_M}{T_A}$.

2. Cache 技术

Cache：小而快的存储器，保存内存数据最近副本

统一Cache：自动调整 Cache 中指令数据比

哈佛Cache：load 和 store 同时执行.

Cache 的基本结构：Cache 存储体、地址映射变换机构、Cache 替换机构.

① Cache 和主存的地址格式

主存：m 位块号，b 位块内地址，共 m+b=n 位.

Cache：c 位行号，b 位行内地址，c<m.

② Cache 存储体的结构

数据区：存 Cache 行的数据.

标签：指明 Cache 行和主存块的对应 } Cache 目录.

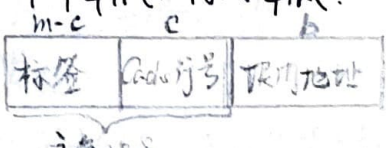
有效位：行是否有效.

Cache 目录容量 = (标签位数+有效位位数)×2^行数.

③ 地址映射与变换

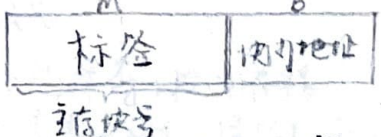
1）直接映射：命中率低，利用率低.

主存地址：



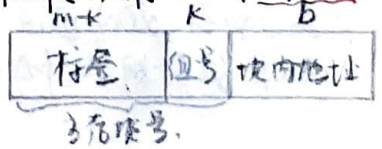
2）全相联映射：命中率高，利用率高，代价高.

主存地址：



3）2^{c-k} 路组相联映射.（2^k：相联度）

主存地址：



查块：1路组相联，2^c 组；全相联：2^c 路组相联，1 组.

④ 替换策略.

当新数据要写入Cache时，若组中有空行，则选空行存入，否则，需替换一行.

1) FIFO法：先进先出

不符合局部性原理.

2) LRU法：替换最近最少使用的行.

比FIFO合理.

- 每Cache行一计数器，替换时选计数最大的行；
- 装入/替换的行计数清0，其它行+1；
- 命中的行计数清0，小于命中行原值的行+1，否则加.

3) 随机法.

⑤ 更新策略.

CPU访问Cache命中时的写主存策略.

1) 通写：写入Cache同时写入主存.

与主存一致；替换时无需写主存；降低速度.

2) 带缓冲的通写：写主存的数据先入write buffer.

不影响CPU速度，write buffer以分时间写回存.

3) 回写：写命中的行被置dirty但不写回主存

直到该行要被替换.

减短写操作时间；主存与Cache在替换前不一致.

4) 写无效：写命中时直接写主存，并使Cache行无效.

(不可读写，直到被替换).

增加缺失率.

⑥ 分配策略

CPU访问Cache缺失时的写主存策略.

1) 写分配：写缺失时，若有空行，写入并置dirty

若无空行，用替换策略替换后置该行dirty，与回写共用.

2) 写不分配：只写主存，不写Cache. 一般与通写共用.

九. 存储系统(2).

1. 存储器管理概述.

进程(任务)：有独立功能的程序在某数据集合上的一次活动.

逻辑地址空间：编译程序时由物理地址对应出的属于该程序的地址号空间.

存储管理单元MMU：CPU执行程序时将逻辑地址转换为物理地址的部件.

段式管理则根据段表转换；页式一页表一段表/页表在内存中.每进程一个.

2. 段式存储管理.

段：将内存划分成一系列不定长的块，每进程占若干块.

段表：段表基址寄存器给出其在内存中的首地址.

每项为基址，段长.

- 逻辑地址：段选择符；段内偏移
- 物理地址：[段表基址+段选择符]+段内偏移

外碎片：段间空闲已小，操作系统可以低效率将内存中的各段拼成一个大片.

3. 页式存储器管理.

页：将内存划分成一致的等长的块，每进程占若干块.

页表：页表基址寄存器给出其在内存中的首地址

每项为页框号（物理页号），映射整个内存.

逻辑地址：逻辑页号；页内偏移.

物理地址：[页表基址+逻辑页号]+页内偏移.

内碎片：页中的空闲区域.

优点：内存利用率高，地址转换快，表项简单.

缺点：程序存储的模块化性不佳；页表>段表（页<段）.

二级页表：页目录 — 页表.

页目录只指出属于进程的页表，其它指向相应

页表的指针为NULL.

转换后备缓冲器TLB（快表）：最近用过的

页转换的Cache：

表项：逻辑（虚拟）页号；物理页号（页框号）.

功能：CPU访存时，将逻辑地址中的逻辑

页号和TLB中的逻辑页号相联比较.

命中：转换为物理地址访存

缺失：在页表/二级页表中按逻辑页号找.

4. 虚拟存储器.（页式管理）.

内存分配溢出时，操作系统将页移入辅存，

并置页表中相应项有效位为0.

页表项：有效位；修改位；页框号/外存地址.

有效位1：页框号

有效位0：外存地址.（原页为其它进程所用）

修改位1：页内容dirty，在替换时写回外存.

修改位0：页内容自调入来未修改.

①调入策略.

从外存调页至内存的时机.

1）请求调页：内存缺页（有效位为0）则调入

1/0次参与.

2）预调页：一次调入相邻n/1页.

可能预测不准；用于程序从外存装入内存的调页.

②替换策略.

从外存向内存调页时，若有空闲页框，则

选空闲页框调入，否则，需替换一页框.

FIFO、LRU…

③清除策略

将内存中修改过的页刷新到外存.

1）请求清除：页将被替换时调出，清除dirty.

2）预清除：页被替换前调出

总结：程序访存过程.

向MMU给出逻辑（虚拟）地址→MMU用之查TLB

命中：TLB将逻辑（虚拟）地址转为物理地址.

缺失：查内存中页表的相应项（该进程的页表）.

有效位1：MMU将其转为物理地址.

有效位0：缺页中断：CPU执行操作系统

提供的缺页中断处理程序，

MMU将该地址存于缺页故障

寄存器，缺页中断处理程序由该

寄存器中的虚拟地址得页表项地址，

进而查得外存地址，从外存中将该页

调入内存.

同时存在MMU、TLB和Cache时，分两种情况：

虚拟寻址Cache：Cache在MMU前，用虚拟地址。

CPU → Cache $\xrightarrow[\text{虚拟地址}]{\text{缺失}}$ MMU（同前）。

{ 优点：Cache命中则不用激活MMU。
缺点：不同虚拟地址可能对应同一页框，Cache不一致。

物理寻址Cache：Cache在MMU后，用物理地址。

CPU → MMU $\xrightarrow[\text{物理地址}]{\text{命中}}$ Cache。

{ 优点：避免Cache不一致。
缺点：Cache访问前MMU地址转换带来延时。

汇编语言程序设计

十. x86汇编语言程序设计

1. 实模式软件体系结构

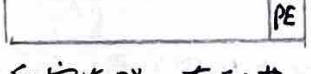
小端格式，字节为最小单位，不要求更大对齐。

8位：字节；16位：字；32位：双字。

20条地址线：1MB；实模式下16位：64KB。

① 寄存器

1) 控制寄存器



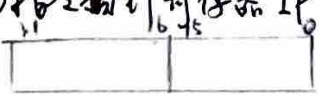
CR0：模式切换。

2) 段寄存器：存段基址：2^{20-16} 份备数。

需将其中的地址×16+相应偏移量(16位)→物理地址。

CS → IP
DS → BX、SI、DI、[]
SS → SP、BP
ES → DI

3) 指令指针寄存器 IP：不可直接修改。



4). 通用数据寄存器 32

a. 累加寄存器 AX：

算术、逻辑运算；IN、OUT中存数据。

b. 基址寄存器 BX

作基址指针

c. 计数寄存器 CX

LOOP前、串处理指令中，自动减计数。

d. 数据寄存器 DX：

乘除：AX低16/商，DX高16/余数；DX↔[DX]Mem/IO

e. 指针寄存器

SP：栈顶偏移量。

BP：栈中数据区基址的偏移量。

f. 变址寄存器：

SI：源操作数偏移量。

DI：目的操作数偏移量。

5) 标志寄存器 16

状态标志位6：CF（加减Neg和乘除超8/16时）

PF奇偶（低8位1的个）、AF（低4位进位）、ZF、

SF（最高位）、OF（同加得反、乘除超8/16时）。

控制标志位3

② 物理地址与逻辑地址

逻辑地址（存的地址EA）：段基址：偏移量。

物理地址：段基址×16+偏移量

I/O地址：独立内存，16位64KB，IN、OUT使用

段基址：操作系统将程序从外存调入内存时设置。

中断向量表：00000~003FF，32位段外指向中断服务程序入口。

△ ③ 堆栈：≤64KB，栈底高地址(SS)，SP±2(or 4)(Pop/Push)

2. 常用指令的寻址方式：

① 立即寻址

② 寄存器寻址

③ 存储器寻址：

1) 直接存储器寻址：DS:[EA].

2) 寄存器间接寻址：DS:[SI/DI/BX], SS:[BP]

3) 基址寻址：DS:[BX+偏移], SS:[BP+偏移].

4) 变址寻址：DS:[SI/DI+偏移], ~~SS:~~.

5) 基址加变址寻址：DS:[BX+SI/DI+偏移], SS:[BP+...].

6) 段超越前缀：eg MOV AX, ES:[3000H].

3. 常用指令的功能：与机器指令对应. CPU

① 数据传送

1) MOV.

目标不为立即数、CS，为段寄存器时源不为immediate数；

目标、源不同时为内存，类型须匹配.

2) PUSH/POP.

不以CS为目标.

3) XCHG.

目标、源不同时为内存.

② 算术运算.

1) ADD/SUB.

影响所有标志位.

2) INC/DEC

不影响CF.

3) MUL/IMUL/DIV/IDIV

操作码(乘数、除数)不为立即数，除法出现0型中断

③ AND/OR/XOR/NOT

④ 移位指令. OP 目标，计数(CL).

SHL: shift left　SHR: shift right

SAL: shift arithmetic left (同SHL)

SAR: shift arithmetic right (符号扩展).

ROL: rotate left　ROR: rotate right

RCL: rotate with carry left (进位入低位)

RCR: rotate with carry right (进位入高位).

⑤ TEST　改变ZF.

CMP　目标-源：影响所有标志位.

JMP

JA JAE JB JBE　JG JGE JL JLE

JNZ JNC JNO JNP JNS

JCXZ: CX.

条件转移可转范围：-128~+127 B.

⑥ LOOP　前CX-1，减到0跳出.

4. 常用伪指令的功能：不与机器指令对应，汇编器

① DB: byte; DW: word; DD: dword.

符号地址 DB/DW/DD 初值表.

② DUP.

重复次数 DUP(重复参数表).

③ PTR.

byte/word/dword... PTR 符号地址.

④ $

到段基址的距离(byte);

在指令中，表示本条指令的地址.

⑤ 段定义：

段名 segment

…

段名 ends

⑥ 段约定、定义的名变量有相应段的基址.

assume 段寄存器：段名，…

⑦ 程序结束.

end 程序的首指令(非伪指令)地址(start)

⑧ 过程定义

过程名 proc near/far; push IP/CS,IP.

…

ret (n); n为返回距离，缺省2.

过程名 endp.

5. IC编语言程序设计的基本方法.

① 分支：单分支/双分支.

循环：do-while/while-do.

② 过程调用

近程：call 过程名.（push IP）

远程：call far ptr 过程名（push CS、IP）.

ret：sp←sp+2

ret n：sp←sp+n+2

参数传递：通过寄存器、内存、栈.

调用者保护栈平衡：call前push call后pop.

被调者保护栈平衡：call前push子程序中ret 2n

6. 常用DOS及BIOS功能的调用方法.

操作系统以中断服务程序的形式

为IC编程序提供系统服务：int 0x25

eg DOS：mov AH，功能号

入口参数设置.

int 21H

出口参数分析.

计算机输入输出系统.

十一. 输入输出系统

1. I/O设备与接口

I/O设备：机械部分，电子部分（设备控制器/适配器）

设备控制器：实现设备与主机通过系统总线通信.

接口寄存器：⇔数据总线.

- 控制寄存器：存外设的控制字/命令字.
- 状态寄存器：存外设的状态字.
- 数据寄存器.

2. I/O端口的编址方式

① I/O独立编址

每个接口寄存器一个I/O端口，独立于内存地址.

eg：X86.

接口寄存器选择⇔地址总线.

② 内存映射编址.

I/O端与内存统一编址.

优点：访存指令可访外设；操作系统易管理

外设地址映射到虚拟地址空间.

缺点：程序查询状态Cache I/O状态，总线影响性能

解决：硬件防I/O页被Cache，主机-系统-I/O三级总线

3. I/O端口地址译码技术

高位地址线与控制信号：寻址接口控制器.
低位地址线：寻址片内寄存器.

固定式译码：eg 74LS138
可选式译码：eg DIP+比较器+74LS138.

即插即用PnP：自动为外设分配不冲突的中断和I/O地址.

4. I/O端口的读写控制

输入缓冲：三态门；输出锁存：锁存器.

5. I/O数据的传送方式：外设⇌内存

① 程序控制方式 PIO
1) 无条件传送：无状态寄存器
2) 程序查询方式传送：CPU不断查状态寄存器.

② 中断传送方式 → 反应时间长而不定的外设.
CPU执行完一条指令检查是否有中断请求；
外设向CPU请求中断.

③ 直接存储器访问 DMA → 快速外设成批数据.
CPU在一个总线周期结束后响应DMAC请求；
外设向DMAC请求传送 ⇒

十二. 总线技术.

1. 概述

总线：设备（部件）与设备（部件）间传送数据的一组公用信号线，公用性.

电源线、地线.
控制总线：总线主控（主设备）→ 从控（从设备）.
双向/单向；三态/非三态.
地址总线：译码选中外设.
单向；三态
数据总线：数据、控制字、命令字、状态字.
双向；三态.

总线性能指标：
总线宽度：数据总线根数./位.
寻址能力：$2^{\text{地址总线宽度}}$
总线带宽（传输率）：Q=寻址能力/时钟周期数·总线时钟周期
负载能力：可接设备数.
正常传送：地址→数据→下地址→…
突发传送：首地址→连续数据.

总线仲裁：总线控制器决定主设备.
优先级/公平策略.

同步总线：read, addr →(clk) data →(…clk) end
异步总线：read, addr → 主握手 → data, 从握手 → 撤消主握手 → 撤消从握手.
半同步总线：read, addr →(clk)
{ data →(…clk) end
{ wait ——(clk)——↑

十三. 中断技术.

1. 中断的基本概念.

中断 interrupt：外部中断、硬件中断、异步、非屏蔽、可禁止.
异常 exception：软件中断、异常、同步、可重现、不可禁止.
eg. 内存访问错误，除0，系统服务调用，缺页…

数　学　作　业　纸

中断：随机，与现行程序无关，可多中断源请求。
子程序：安排，受主程序控制，单子程序调用

中断：每指令执行后响应，改CPU现场，软~方式I/O，开始
DMA：每总线周期后响应，不改CPU现场，硬~方式I/O，结束。

中断 → 获得中断服务程序ISP入口。
中断向量表：每个中断有一个中断服务程序
中断原因(优先)寄存器，通用中断服务程序。
→保护CS、IP、FLAGS、… ⇄中断授权返回套。

十四．DMA技术
1．DMA技术概述。
①第三方DMA(标准DMA)。
- 地址寄存器：CPU 数据总线 → 写初值++。
- 数据数量计数器：CPU 数据总线 → 写初值--。
- 数据缓冲寄存器
- 控制/状态寄存器
- 中断机构：传完则向CPU发中断请求。

1)传送前预处理(CPU)：软。
2)数据传送(DMAC)：硬。
a．外设 DREQ DMAC HRQ CPU bus ⇔ 内存．HRQ。
b．外设 DACK DMAC HLDA CPU HLDA，HRQ
↕bus 内存。 (DMAC成为主设备)。
c．外设 IOR/IOW 内存地址 DMAC MEMR/MEMW 内存．HLDA，HRQ
(循环，传数)。
d．外设 EOP DMAC 中断请求 CPU。
3)传送后处理(CPU)：软。
②第二方DMA(总线主控)：外设上有DMA电路。

十五．串行接口与定时计数技术。
1．软件定时：延时子程序，占CPU时间
硬件定时：定时计数器：不占CPU时间。
可编程定时计数器8253：无状态寄存器。

- CLKi：in 时钟/外部事件　i=0~2
- GATEi：in 计数器是否工作。
- OUTi：out 减至0时输出规定信号。

A1~A0：in 40~43H，A1A0选四个寄存器 ⇔ 8086 bus

- 初值寄存器CR：写控制字→写初值 $N=\frac{f_{CLK}}{f_{OUT}}$ 先
- 输出锁存器OL：写控制字(锁存)→读计数值 低后高

D7~D0：⇔ data bus。

- 方式2：分频器：计到1
- 方式3：方波发生器：计到$\lfloor \frac{N}{2} \rfloor$。

自动从CR取得原初值。

2．串行通信基础。
- 外设↔接口电路：串/并行
- CPU↔接口电路：并行。

波特率：每秒传送的二进制位数．bps。
波特率因子：接收/发送时钟频率/波特率 ≥1
单工→；全双工⇌；半双工→ ←。
数据通讯设备DCE：传输信号 eg Modem。
数据终端设备DTE：接收、发送 eg 8250。
异步通信：一位一位传输：帧-空闲位…
同步通信：一组一组传输：同步字符-数据块-
同步快于异步。

十六. 并行接口与模拟量I/O.

1. 并行接口概述.

优点：快，对CPU无需并转换，协议简单.
缺点：时钟偏移，串扰，直流偏置.

输入接口：片选&读有效. eg 74LS244
输出接口：片选&写有效. eg 74LS373.

2. DAC0832

双缓冲方式：两个寄存器各选一次：$\overline{CS}\to\overline{XFER}$.（第一次AL中为待转换数字，第二次AL中的值无意义）
单缓冲方式：只选一次：$\overline{XFER}=\overline{WR_2}=0$；$\overline{CS}$.

数据输入寄存器

ILE=1，$\overline{CS}=\overline{WR_1}=0$：直通 LE1=1
$\overline{WR_1}=1$，锁存. LE1=0.

DAC寄存器.

$\overline{XFER}=\overline{WR_2}=0$：直通 LE2=1
$\overline{WR_2}=1$，锁存，LE2=0.

3. ADC0809

通道选择（通道连续接地址码低三位）.

```
MOV DX, 220H+n ; n=0~7
out DX, AL     ; AL值无关.
```

读取转换后的数字.

```
MOV DX, 220H   ; 选中ADC，任一端口可.
IN  AL, DX.
```

EOC：转换结束信号，高有效

接中断请求引脚，AD完成引发中断信号.
接DB，CPU查询三态数据缓冲器知AD是否完成.

△