1. 计算机组成原理概论

冯·诺依曼计算机中区分指令和数据的依据是：指令周期的不同阶段

机器字长（计算机位数）：计算机一次能处理的二进制位数，等于寄存器位数（数据）

操作系统位数：操作系统可寻址的位数（地址）

高级语言翻译成机器级目标代码的程序：编译程序

解释程序：不生成目标代码，直接执行

寄存器的设置对汇编语言不透明

系列机的基本特性是指令系统向后兼容

软件和硬件在逻辑上是等效的

计算机硬件不能直接执行HDL

MIPS衡量标量机，MFLOPS衡量向量机

吞吐量衡量批处理性能

IPS = 主频/CPI

程序执行时间 = CPI\*指令条数/f

1. 存储系统

RAM和ROM均属于主存，均为随机存取器

磁带和光盘均为顺序存储器

磁盘为直接存取器，先随机后顺序

存储器的存取周期指存储器进行连续读或写操作允许的最短时间间隔

设机器字长为32位，按半字寻址，容量为16MB的存储器，可寻址单元数为2的23次方

注意字节和位的区别

多级存储系统是为了降低存储成本

虚拟存储器中主存和辅存的数据调度对系统程序员不透明，对应用程序员透明

分散刷新的读写周期翻倍，不存在死时间

SRAM（除电源和接地）的引脚包括：地址线，数据线，片选线1根，读写控制线1或2根

SRAM的地址引脚数量正常计算，4M\*8的芯片需要22根地址线

DRAM的地址引脚数量为正常计算的一半，如4M\*8的芯片需要11根地址线

操作系统在硬盘中的计算机，内存需要RAM和ROM

SDRAM：Synchronous DRAM，同步动态随机存储器（与CPU时钟同步）

MAR的位数 = 主存地址空间的位数 ≠ 当前主存储器的实际容量

采用指令Cache和数据Cache分离的目的是：减少指令流水线的资源冲突

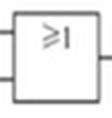
1.假设用若干个2k\*4位的芯片组成8k\*8位的存储器，地址为081FH所在芯片的最小地址为？

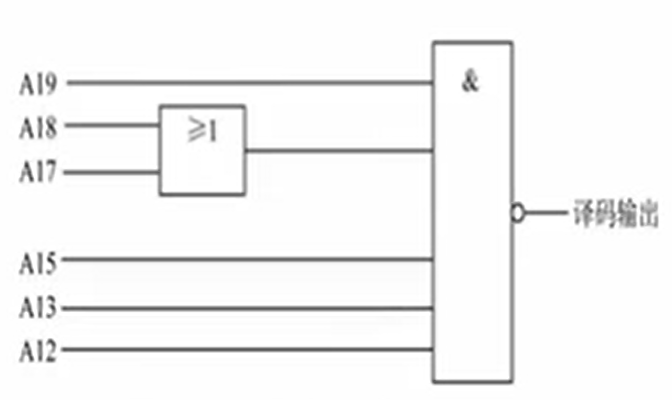
081FH：0000 1011 0001 1111 可见，该地址为第二组芯片内的地址，固答案为0800H

2.某存储器容量为64KB，按字节编址，地址4000-5FFFH为ROM区，其余为RAM区，若采用8k\*4的SRAM芯片进行设计，需要该芯片的数量是？

5FFFH – 4000H + 1 = 2000H = 0010 0000 0000 0000 = 2的13次方 = 8KB

固答案为：（64KB-8KB）/8K\*0.5B = 56/4 = 14

此为或门



该译码器的数学表达式为：



3.地址总线为A0~A15，用4K\*4位的存储芯片组成16KB存储器，则产生片选信号的24译码器的输入地址线为：

原芯片地址线为12根，数据线为4根。16KB可拆为32K\*4位，即使用8张芯片

但译码器为2-4译码器，固应当使用4张芯片，改拆为16K\*8位。此时地址线为14根

此时A12,A13为用作译码器输入的地址线。

4.某计算机Cache共有16块，采用二路组相联方式。每个主存块为32字节，按字节编址，主存129号单元所在的主存块应当装入的Cache组号是？（2009.14）

Cache共分为8组，主存块按照模8法进入不同的Cache组中。

如0，8，16号主存块进入第一组Cache，即Cache 0组。

则129号主存单元属于第5号主存块，进入第5组Cache，即Cache 4组。

5.某存储系统中，主存容量是Cache容量的4096倍，Cache被分为64个块，当主存地址和Cache地址采用直接映像时，地址映射表的大小为（不考虑替换算法位）

主存字块标记位数：主存容量与Cache容量的比值，64\*（12+1+1）bit

6.主存容量为1MB，Cache容量为16KB，每个字块有8个字，每个字为32位，采用直接映像方式，若主存地址为35301H，且CPU访问Cache命中，则在Cache的哪个字块中？

直接映射的地址结构：主存字块标记|Cache字块地址|字块内地址

主存字块标记位数 = 20-14 = 6

每个字块有32字节，故字块内地址位数 = 5

Cache字块地址位数 = 14 – 5 = 9

给出主存地址为3501H，即 0011 0101 0011 0000 0001 块号为128+16+8 = 152

7.有如下C语言段：

for（k=0;k<1000;k++）

a[k] = a[k] + 32;

若数组a及变量k均为int类型，int型数据占4字节，Cache采用直接映射，大小为1KB，每个Cache块大小为16B，则程序执行过程中Cache的缺失率为？

执行a[k] = a[k] + 32 时，需要两次访问数组a。因一个Cache块可放4个数据，取出a[0]时连同a[1],a[2],a[3]一并取出，缺失率为1/8.

8.假设主存地址为32位，按字节编址，主存与Cache之间采用直接映射，主存块大小为4字，1字32位，采用回写方式，则能存放4K字数据的Cache的总容量位数至少是？

主存字块标记 = 主存地址位数 – Cache地址位数 = 32 – 14 = 18

Cache一块的位数至少为：18 + 1 + 1 + 4\*32 = 148，一共有1k行，故总位数至少为148K

1. 指令系统

1.一个计算机系统采用32位单字长指令，地址码为12位，如果定义了250条二地址指令，则还可以有几条一地址指令？

2的8次方 – 250 = 6，6\*2的12次方 = 24K

2.某计算机按字节编址，指令字长固定且只有两种指令格式，其中三地址指令29条，二地址指令107条，每个地址字段6位，则指令字长至少为？

计算机按字节编址，则指令位数需要是8的整数倍。

三地址有29种指令，则操作码至少有5位，说明三地址指令有3种操作码未使用

则二地址指令可以有至多3\*64 = 192种，题目中只使用了107种，说明最少只需要5位操作码，亦即当前指令字长为23位。因按字节编址，故至少应为24位。

进位/借位标志：CF 零标志：ZF

无符号整数比较大于时转移： ~（CF + ZF）= 1 / ~CF \* ~ZF = 1

16位补码的表示范围是：-32768 ~ 32767

原码：将一个整数，转换成二进制，就是其原码。

如单字节的5的原码为：0000 0101；-5的原码为1000 0101。

反码：正数的反码就是其原码；负数的反码是将原码中，除符号位以外，每一位取反。

如单字节的5的反码为：0000 0101；-5的反码为1111 1010。

补码：正数的补码就是其原码；负数的反码+1就是补码。

如单字节的5的补码为：0000 0101；-5的原码为1111 1011。

3.设相对寻址的转移指令占3字节，第一字节为操作码，第二三字节为相对位移量（补码表示）。CPU从存储器中每取出一个字节，自动完成(PC) + 1 -> PC，若当前PC为240，要求转移到290，则二三字节的机器代码为？若当前PC为240，要求转移到200，则二三字节的机器代码为？

取出当前PC为240的指令时，PC为243，则相对位移量为47。

47的补码为0010 1111，即2F。正数高位需要补0，即二三字节为：2FH 00H。

取出当前PC为240的指令时，PC为243，则相对位移量为-43。

-43的原码为1010 1011，反码为1101 0100，则补码为1101 0101，即D5。

负数高位需要补1，即二三字节为：D5H FFH。

4.某计算机主存空间为4GB，字长为32位，按字节编址，若指令按字边界对齐存放，则PC位数至少是？

因按字节编址，则主存地址块有2的32次方。但指令按字边界对齐存放，则只需要30位PC即可定位一条指令。类似有，如果一条指令分别存在前后两个字的后两个地址块和前两个地址块中，则需要31位PC才够。

1. 中央处理器

条件转移指令执行时所依据的条件来自标志寄存器（程序状态字寄存器）

PSW属于运算器，包含指令执行结果的状态信息和控制信息（如允许中断）

每个指令周期一定大于等于CPU时钟周期

空操作指令的指令周期中并不是任何寄存器的内容都不会改变（PC）

指令周期由机器周期组成，机器周期通常由存取周期决定

指令字长和机器字长没有关系，一般等于存储字长

单总线CPU中，ALU的一个输入端和输出端都需要暂存器与总线相连

采用CPU内部总线的数据通路：结构简单，容易实现，性能较低，存在较多的冲突

不采用CPU内部总线（专用数据通路）：结构复杂，硬件量大，性能高，基本无数据冲突

硬布线控制器指令执行速度快，指令功能修改扩展难

硬布线控制器就是组合逻辑控制器，延迟主要来自电路延迟

微程序控制器的优点：方便修改和扩展，延迟主要来自访存延迟

微程序控制器：一条机器指令-->一个微程序-->一组微指令 一条微指令-->一组微命令

机器指令的操作码字段形成微程序入口地址，微指令的微地址码指出下条微指令所在的位置

微程序控制器的控制存储器由ROM制成，μPC决定微指令执行顺序，不决定指令执行顺序

水平型微指令：

一位对应一个控制信号（并行） 微指令长，编写微程序麻烦，微程序短，执行速度快

垂直型微指令：

一条指令对应一种操作（串行） 微指令短，编写微程序容易，微程序长，执行速度慢

一条包含33个微命令的微指令，构成五个互斥类，分别包含7，3，12，5，6个微命令，则微指令的操作控制字段至少有几位？

log8+log4+log13+log6+log7 = 3+2+4+3+3 = 15

采用断定法（下地址字段法）确定下条微指令地址的微程序控制器计算机中，微指令中下地址字段位数 = log[Σ(μInstruction in μProgram) + μInstruction in μIF\_Program] (2014.18)

控制存储器在CPU控制器的控制单元内，按地址访问，存储微指令，不存储数据

流水线采用定长机器周期

超标量流水线不缩短流水线功能段的处理时间，能在一个时钟周期同时发射多条指令，可以结合动态调度技术（编译优化）提高指令执行并行性

流水线吞吐率 = n\*f/(n+k-1) = n/(n+k-1)Δt

流水线加速比 = n\*k/(n+k-1)

RAW：数据冲突

WAR：乱序发射（转变为了RAW）

WAW：多个功能部件（ALU）加减法比乘除法算的快

指令流水线数据通路包含暂存计算结果的通用寄存器组，不包含控制单元

五．总线

中断类型号指出中断向量的地址，中断向量指出中断服务程序的地址

中断类型号在数据线上传输

单周期处理器中所有指令执行周期为一个时钟周期，不能采用单总线结构的数据通路，处理器的时钟频率较低，在指令执行过程中控制信号不变，每条指令的CPI均为1

CPU通过控制单元识别地址和数据

多总线系统中主存通过总线类型识别地址和数据

有效数据传输率指传输数据内存在校验位等其他信息时，单纯传输数据的速率

存储器总线可支持突发传送方式

采用突发传输方式可提高总线数据传输率

采用分离通信方式可提高总线利用率（需要通道）

打印机按照打字原理可分为击打式和非击打式，按能否打出汉字分为点阵式和活字式

键盘和鼠标都是通过中断实现输入操作的

VGA (Video Graphics Array)，又称D-sub端口，用于传输模拟信号 适合低分辨率

数字信号 -> 模拟信号 -> VGA -> CRT显示器

数字信号 -> 模拟信号 -> VGA -> 数字信号 -> LCD液晶显示器

DVI(Digital Visual Interface)，用于传输数字信号 适合高分辨率

HDMI(High Definition Multimedia Interface) 传输音频和视频信号

显存用于刷新屏幕，带宽 = 分辨率 \* 色深 \* 帧频

RAID0：无冗余无校验的磁盘阵列，利用条带化加快访存速度

六．I/O系统

I/O指令实现的数据传输发生在通用寄存器和I/O端口之间

I/O接口中CPU可访问的寄存器称为I/O端口

状态端口和控制端口可以用同一个寄存器

I/O总线的数据线上传输的信息包括I/O接口中的命令字，状态字和中断类型号

时钟中断属于外中断，浮点数下溢和Cache都不会使系统产生中断

中断I/O方式请求的是CPU时间，DMA方式请求的是总线控制权

DMA响应发生在一个总线事务完成后/总线空闲时每个机器周期的结束均可

七．数字系统

（101 001）BCD = 29 （52）8 = （101 010）2 （233）16 = （0010 0011 0011）2

进制数越低表示小数的能力就越弱 二进制无法表示任何十进制小数

码距：两个码字对应位上数字不同的个数

校验码的码距必须大于1，码字越大，检错纠错能力越强

增加奇偶校验码的位数不会提高校验能力

海明码： 2的k次方 ≥ n + k + 1

CRC中校验位位数 = 生成多项式位数 – 1

char =1字节，short = 2字节，int = float = 4字节，long = double = 8字节

多位数据加少位数据时会对少位数据自动类型转换，负数补1，正数补0

负数的补码，表面看起来越小的数，其绝对值越大

负数的补码，每增大1，其绝对值就会减少1

某计算机存储器按字节编址，采用小端方式存放数据，数据按边界对齐存储，某段C语言程序如下：

struct{

int a；

char b；

short c；

}record；

record.a = 273;

若record变量的首地址是0xC008，则地址0xC008和record.c所在的地址分别是？

0x11 0xC00E

273 = 256 + 16 + 1 = 0x 00 00 01 11

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 地址 | 0xC008 | 0xC009 | 0xC00A | 0xC00B |
| 内容 | record.a(0x11) | record.a(0x01) | record.a(0x00) | record.a(0x00) |
| 地址 | 0xC00C | 0xC00D | 0xC00E | 0xC00F |
| 内容 | record.b |  | record.c | record.c |

对于x0.x1x2…xn

无符号定点小数的取值范围是：【0,1-2的-n次方】

原码和反码定点小数的取值范围是：【-1+2的-n次方,1-2的-n次方】

补码和移码定点小数的取值范围是：【-1,1-2的-n次方】

对于xn+1xn…x2x1

无符号定点整数的取值范围是：【0，2的n+1次方-1】

原码和反码定点整数的取值范围是：【1-2的n次方,2的n次方-1】

补码和移码定点整数的取值范围是：【-2的n次方,2的n次方-1】

同为N位的p进制小数和q进制小数，p进制小数能表示的数的个数/q进制小数能表示的数的个数 = (p/q)的N次方 N位m进制小数可以表示m的N次方个数

补码定点小数1.00… = -1，即补码小数能表示的最小的数

算术移位操作中，正数补0，原码负数补0，反码负数补1，补码负数左添0右添1

定点运算器中，加法电路和溢出判断电路用异或实现，移位电路用触发器实现

编码电路和译码电路一般用与非门来实现

每个模四补码存储时只需1个符号位，计算时调整为2位

模四补码更容易检查加减运算的溢出问题

判断加减法溢出时，判断进位的方式中符号位的进位是C0,最高位的进位是C1

C0\*~C1 + C1\*~C0 = 1时就发生了加法溢出，其中~C0\*C1是上溢，C0\*~C1是下溢

对于二进制数乘或除2的次方可以直接移位

原码一位乘法中符号位不参与运算，补码Booth乘法中符号位参与运算

Ci = Ai\*Bi +(Ai⊕Bi)\*Ci-1 = Gi + PiCi-1 Gi称为进位生成信号 Pi称为进位传递信号

四块74181和一块74182可以组成组内并行进位，组间串行进位的16位ALU