

计算机组成原理期末复习题

1. 世界第一台通用电子计算机ENIAC， 1 9 4 6
2. 第一台存储程序式计算机—EDSAC， 1 9 4 9
3. 数据校验码主要有奇偶校验码、海明校验码和循环冗余校验码
4. 磁表面存储器记录信息是利用磁性材料的磁滞回归线特性（
5. 计算机的字长取决于运算器一次运算二进制数的位数
6. 模 m 交叉存储器有 m 个存储模块，它们有各自的地址寄存器和数据缓冲寄存器
7. 使用虚拟存储器时，由操作系统完成地址转换
8. 若显示器灰度级为 32，则每像素应用 5 位表示
9. RAM 芯片位扩展可以增加存储器字长
10. 输入输出指令的功能是 CPU 与外设的数据传送
11. 假设校验位的个数为 r ， $k=2^{19}-1-r$ 个信息能用于纠正被传送数据的位数，需满足： $2^{r+1} \geq k+r+1$
12. 某计算机字长是 16 位，它的存储容量是 1MB，按字编址，它的寻址范围是 $0-2^{19}-1$
13. 操作数在寄存器中的寻址方式称为寄存器寻址；操作数在指令中的寻址方式称为立即寻址，相对寻址中，指令地址码给出一个偏移量(带符号数)，基准地址隐含由 PC 给出。
14. 操作码的编码有两种方式：定长操作码法，扩展操作码编法
15. 串行传输时数据位的低位在前，高位在后。
16. DMA 控制器与 CPU 分时使用内存常采用以下三种方法：停止 CPU 访内，周期挪用和交替访问。
17. 流水线中的主要相关问题指资源相关、数据相关和控制相关。
18. 双端口存储器中一个存储器具有两组相互独立的读写控制电路，可进行并行的独立操作。
19. 多模块交叉存储器中每个模块都有自己的读写控制电路，地址寄存器和数据寄存器；CPU 同时访问四个模块，由存储器控制电路控制它们分时使用数据总线。
20. 磁表面信息的存取是利用电—磁变换实现的。
21. 指令系统中采用不同的寻址方式的主要目的是缩短指令长度，扩大寻址范围
22. 常用的虚拟存储系统由主存—辅存两级存储器组成。
23. 存放欲执行指令的寄存器是 IR
24. 在单级中断系统中，CPU 一旦响应中断，则立即关闭中断允许标志，以防止本次中断 服务结束前同级的其他中断源产生另一次中断进行干扰。
25. 微指令格式分为水平型和垂直型，水平型微指令的位数较多，用它编写的微程序较短。
26. 所谓三总线结构的计算机是指 I/O 总线、主存总线和 DMA 总线三组传输线，微型机系统中，主机和高速硬盘进行数据交换一般采用 DMA 方式。
27. 程序员编程所用的地址叫做逻辑地址。

28. 在主存和 CPU 之间增加 cache 存储器是为了解决 CPU 和内存速度不匹配的问题;引入虚拟存储器则是为了解决主存容量不够大的问题。为使主存的平均读出时间接近于 Cache 的读出时间,必须让 Cache 的命中率接近于 1。

29. 磁盘上访问信息的最小物理单位是扇区。

30. 相联存储器是按内容访问的存储器。

31. 如果Cache的容量为128块,在直接映象下,主存中第*i*块映象到缓存第*i mod 128* 块。

32. CPU 能直接访问 cache 和主存,但不能直接访问磁盘和光盘。

33. 补码运算的特点是符号位与数值位一起计算

34. 若浮点数用补码表示,则判断运算结果是否为规格化数的方法是 C 数符与尾数小数点后第一位数字相异为规格化数。

35. 某计算机有五级中断 L4~L0,中断屏蔽字为 M4M3M2M1M0, $M_i=1$ ($0 \leq i \leq 4$) 表示对 Li 级中断进行屏蔽。若中断响应优先级从高到低的顺序是 L4→L0→L2→L1→L3,则 L1 的中断处理程序中设置的中断屏蔽字是 01010

36. 什么是 ISA?

答: ISA 指 Instruction Set Architecture,即指令集体系结构;ISA 是一种规约 (Specification),它规定了如何使用硬件,不同 ISA 规定的指令集不同,如,IA-32、MIPS、ARM 等;ISA 在计算机系统中是必不可少的一个抽象层。

37. 什么是数据通路?它主要由哪两类元件组成?控制器的主要功能是什么?

答: CPU 主要由数据通路和控制器组成,指令执行过程中,数据所经过的路径,包括路径中的部件称之为数据通路,它是指令的执行部件。数据通路主要由组合逻辑元件(也称操作元件)和时序逻辑元件组成。控制器负责对指令进行译码,生成指令对应的控制信号,控制数据通路的动作,是指令的控制部件。

38. 什么是 CPI? CPU 执行时间与哪几个因素有关?

CPI: Cycles Per Instruction,每条指令执行所需的时钟周期数,对一个程序或机器来说,CPI 是所有指令执行所需的平均时钟周期数。 $\text{CPU 执行时间} = \text{CPI} \times \text{程序总指令条数} \times \text{时钟周期}$

39. 用二进制编码的原因:

答: 制造二个稳定态的物理器件容易;二进制编码、计数、运算规则简单;正好与逻辑命题对应,便于逻辑运算,并可方便地用逻辑电路实现算术运算

40. 先行进位解决的是 ALU 的运算速度问题;

其基本思想使串行进位的产生变成并行进位的产生变成并行进位的产生,从而使各个进位在一个时钟周期内完成,提高加法运算速度。

41. 什么是刷新?为什么要刷新?有哪几种常用的刷新方式?

动态存储元主要由电容构成,由于电容存在的漏电现象而使其存储的信息不稳定,故 DRAM 芯片需要定时刷新。常用的刷新方式有集中式刷新、分布式刷新。

42. 中断与子程序

(1) 中断服务程序与中断时 CPU 正在运行的程序是相互独立的,它们之间没有确定的关系。子程序调用时转入的子程序与 CPU 正在执行的程序段是同一程序的两部分。

(2)除了软中断通常中断产生都是随机的,而子程序调用是由 CALL 指令(子程序调用指令)

引起的。

(3) 中断服务程序的入口地址可以通过硬件向量法产生向量地址，再由向量地址找到入口地址。子程序调用的子程序入口地址是由CALL 指令中的地址码给出的。

(4) 调用中断服务程序和子程序都需保护程序断点，前者由中断隐指令完成，后者由CALL 指令本身完成。

(5) 处理中断服务程序时，对多个同时发生的中断需进行裁决，而调用子程序时一般没有这种操作。

(6) 在中断服务程序和所调用的子程序中都有保护寄存器内容的操作。

43. CISC 与 RISC

CISC：复杂指令系统计算机（Complex Instruction Set Computer）其特点是：

指令格式不固定，指令可长可短，操作数可多可少；

寻址方式复杂多样，操作数可来自寄存器也可来自存储器；

采用微程序控制，执行每条指令均需完成一个微指令序列；

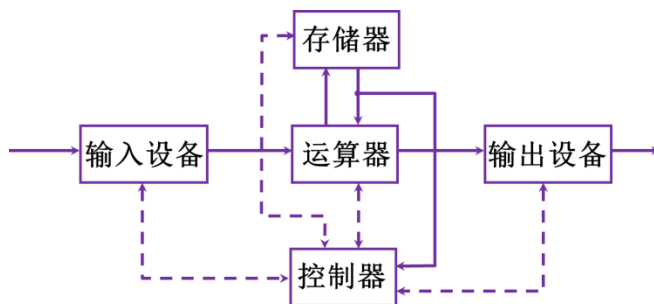
指令越复杂，CPI 越大。指令系统庞大，研制周期长，系统效率低。

RISC：简单指令系统计算机（Reduced Instruction Set Computer）其特点是：

只保留功能简单的指令。指令系统尽可能简单，尽是减少指令的执行时间以提高效率。

多用寄存器指令，少用访内指令，指令格式一致，寻址方式简单，

44. 画出冯·诺依曼计算机的硬件组成框图，并简要说明各部分主要功能和工作原理。



控制器：是整个计算机的指挥中心，控制计算机从内存中取出解题步骤（即指令）加以分析，然后执行某种操作。

存储器：存放程序和数据的部件，是一个记忆装置，也是计算机能够实现“存储程序控制”的基础。

输入设备：是输入用户提供的原始信息，并且将其转换为计算机能识别的信息。

输出设备：是将计算机的处理结果以能被人们接受的或能为其他计算机所接受的形式输出。

运算器：以二进制为基础完成对信息的加工、处理，其核心是算术逻辑运算部件 ALU。是一种执行部件。

总线：计算机通过总线将五大部件连接起来，总线是它们相互通信的公共通路，主要任务是在各部件之间传送地址信息、数据信息和控制信息。

工作原理：

采用存储程序工作方式：

① 事先编制程序(根据问题找算法编程序)

② 将程序存储于计算机的存储器中

③ 计算机在运行时自动地、连续地从存储器中依次取出指令加以执行

以控制流(指令)驱动程序执行, 信息流(数据流)被动地被调用处理。

45. 什么是校验码? 其基本原理是什么? 常用的数据校验码有哪些?

答: 数据校验码是一种常用的带有发现某些错误或自动改错能力的数据编码方法。它的实现原理, 是加进一些冗余码, 使合法数据编码出现某些错误时, 就成为非法编码。这样, 就可以通过检测编码的合法性来达到发现错误的目的。常用的数据校验码主要有奇偶校验码、海明校验码和循环冗余校验码。

46. 什么是“寻址方式”? 常用的操作数的寻址方式有哪些?

答: 指令或操作数地址的指定方式。即: 根据地址找到指令或操作数的方法。操作数的寻址通常有立即 / 直接 / 间接 / 寄存器 / 寄存器间接 / 偏移 / 堆栈等,

47. Cache 有哪些特点?

Cache 具有如下特点:

位于 CPU 与主存之间, 是存储器层次结构中级别最高的一级。容量比主存小, 目前一般有数 KB 到数 MB。

速度一般比主存快 5~10 倍, 通常由存储速度高的双极型三极管或 SRAM 组成。

其容量是主存的部分副本。

可用来存放指令, 也可用来存放数据。

快存的功能全部由硬件实现, 并对程序员透明。

48. 试从记录密度, 存储容量, 存取速度方面来说明磁表面存储器的相关技术指标。

记录密度

道密度(TPI): 磁盘半径方向单位长度包含的磁道数, 单位: 道/英寸 (TPI) 或道/毫米 (TPM)

位密度(BPI): 在每一个磁道内单位长度内所能记录的二进制信息数。

存储容量

存储容量是指整个磁表面存储器所能存储的二进制信息的总量, 一般用位或字节为单位表示, 它与存储介质尺寸和记录密度直接相关。

非格式化容量是指磁记录表面上可全部利用的磁化单元数

格式化容量是指用户实际可以使用的存储容量。

平均寻址时间 T_a

当磁头接到读/写命令, 从原来的位置移动到指定位置, 并完成读/写操作的时间叫存取时间。平均找道时间 t_{sa} +平均等待时间 t_{wa}

数据传输率: 磁表面存储器在单位时间内与主机之间传送数据的位数或字节数。

49. 在异步串行传输系统中, 若每个数据帧包含 1 个起始位, 8 个数据位, 1 个奇校验位, 1 个结束位, 比特率 160bit/s, 画出传输字符 “C” (ASCII 码为 43H) 的波形图。

50. 一个典型程序的转换过程:

经典的“hello.c”C-源程序

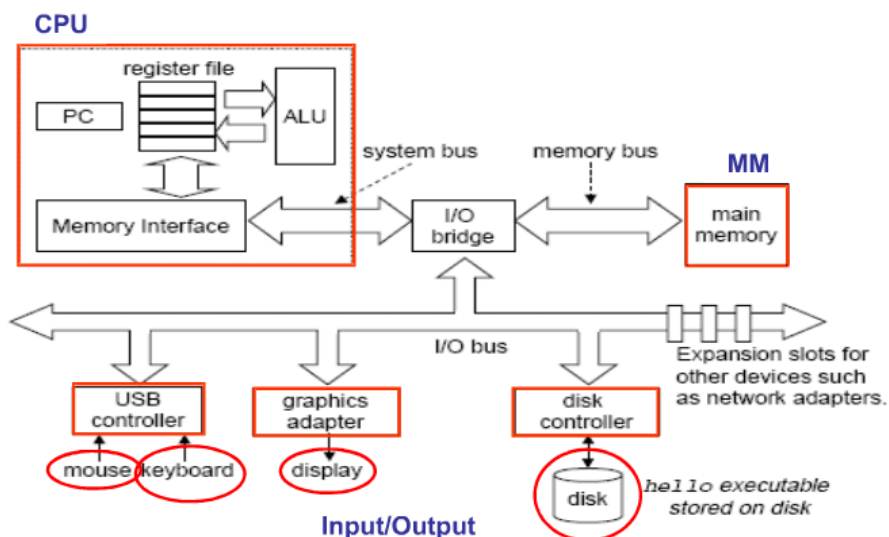
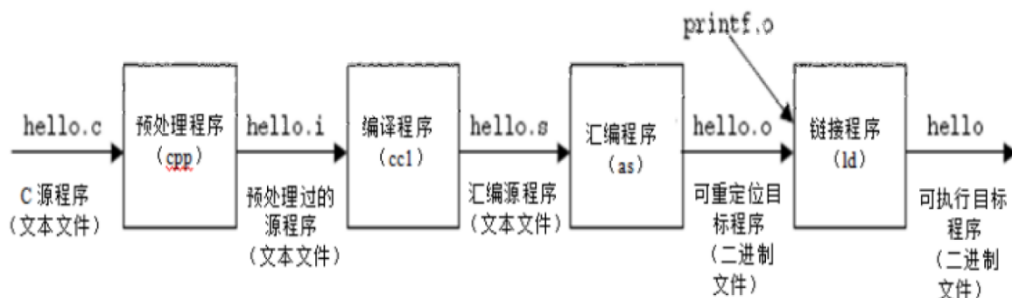
```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("hello, world\n");
6 }
```

程序的功能是：

输出“hello,world”

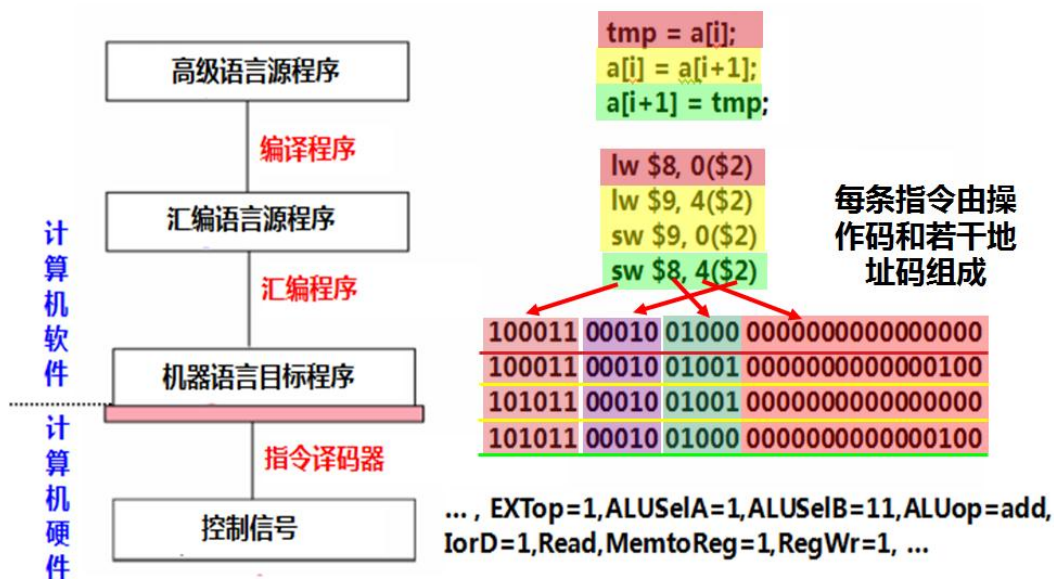
hello.c的ASCII文本表示

```
# i n c l u d e < s p > < s t d i o .
35 105 110 99 108 117 100 101 32 60 115 116 100 105 111 46
h > \n \n i n t < s p > m a i n ( ) \n {
104 62 10 10 105 110 116 32 109 97 105 110 40 41 10 123
\n < s p > < s p > < s p > < s p > p r i n t f ( " h e l
10 32 32 32 32 112 114 105 110 116 102 40 34 104 101 108
l o , < s p > w o r l d \n " ) ; \n }
108 111 44 32 119 111 114 108 100 92 110 34 41 59 10 125
```



PC：程序计数器；ALU：算术/逻辑单元；USB：通用串行总线

51. 不同层次语言之间的等价转换



52. 假如某程序 P 编译后生成的代码由 ALU、Load, Store, Branch 四类指令组成, 它们在程序中所占比例如图所示, 已知它们的 CPI 分别为 1, 2, 2, 2, 现重新对程序进行编译优化, 生成的新目标代码中 ALU 指令减少了 50%, 其他类指令条数不变。

(1) 编译优化前后程序的 CPI 各是多少?

(2) 如果机器的时钟周期为 20ns(50MHz 时钟频率), 则优化前后的 MIPS 各是多少?

Op	Freq	Cycle
ALU	43%	1
Load	21%	2
Store	12%	2
Branch	24%	2

解: 优化后各指令所占的新的比例为:

$$21.5 / (21.5 + 21 + 12 + 24) = 27\%$$

$$21 / (21.5 + 21 + 12 + 24) = 27\%$$

$$12 / (21.5 + 21 + 12 + 24) = 15\%$$

$$24 / (21.5 + 21 + 12 + 24) = 31\%$$

$$\text{优化前的 CPI} = 1 \times 43\% + 2 \times 21\% + 2 \times 12\% + 2 \times 24\% = 1.57$$

$$\text{优化后的 CPI} = 1 \times 27\% + 2 \times 27\% + 2 \times 15\% + 2 \times 31\% = 1.73$$

$$\text{优化前的 MIPS} = 50\text{M} / 1.57 = 31.8\text{MIPS}$$

$$\text{优化后的 MIPS} = 50\text{M} / 1.73 = 28.9\text{MIPS}$$

Figure 1.10 shows the internal logic of the MIPS ALU. It takes two 32-bit operands, $a[31:0]$ and $b[31:0]$, and a 5-bit function code, $aluc[4:0]$. The function code selects one of eight operations: ADD, SUB, AND, OR, XOR, LUI, SLL, SRL, or SRA. The ALU is composed of several blocks: **addsub32** (32-bit adder/subtractor), **and** (2-input AND), **or** (2-input OR), **shift** (logical shift left), and a **right/arith** block (right shift or arithmetic shift). The result of the operation is stored in a 32-bit register z . The function code is also used to select the appropriate operation for the right/arith block.

```

module alu_main (a,b,aluc,r,z);
    input  [31:0] a,b;
    input  [3:0] aluc;
    output [31:0] r;
    output z;
    wire [31:0] d_and = a & b;
    wire [31:0] d_or  = a | b;
    wire [31:0] d_xor = a ^ b;
    wire [31:0] d_lui = {b[15:0],16'h0};
    wire [31:0] d_and_or  = aluc[2]? d_or  : d_and;
    wire [31:0] d_xor_lui = aluc[2]? d_lui : d_xor;
    wire [31:0] d_as,d_sh;
    addsub32 as32 (a,b,aluc[2],d_as);
    shift shifter (b,a[4:0],aluc[2],aluc[3],d_sh);
    mux4x32 select (d_as,d_and_or,d_xor_lui,d_sh,aluc[1:0],r);
    assign z = ~|r;
endmodule

```

- 结合图示和代码分析该ALU实现的功能
- 写出其中多路选择器的实现代码

```

module mux4x32 (a0, a1, a2, a3, s, y);
    input  [31:0] a0, a1, a2, a3;
    input  [1:0]  s;
    output [31:0] y;

    function [31:0] select;
        input [31:0] a0, a1, a2, a3;
        input [1:0]  s;
        case (s)
            2'b00: select = a0;
            2'b01: select = a1;
            2'b10: select = a2;
            2'b11: select = a3;
        endcase
    endfunction
endmodule

```

```

    endcase
    endfunction
    assign y = select(a0, a1, a2, a3, s);
endmodule

```

54. 结合表中MIPS指令格式，将下列MIPS汇编语言转换为对应的机器指令，并说明寄存器PC, r4, r1, r5中数据的变化情况。

指令	[31:26]	[25:21]	[20:16]	[15:11]	[10:6]	[5:0]	意 义
addi	001000	rs	rt	immediate			立即数加
andi	001100	rs	rt	immediate			立即数与
ori	001101	rs	rt	immediate			立即数或
lui	001111	00000	rt	immediate			设置高位
sw	101011	rs	rt	offset			存整数数据字
jal	000011	address					调用

```

(00) main: lui r1,0
(04)      ori r4,r1,80
(08)      addi r5,r0, 4
(0c)call: jal sum
(10)      sw r2,0(r4)

```

```

32'h3c010000; // (00)
32'h34240050; // (04)
32'h20050004; // (08)
32'h0c000018; // (0c)
32'hac820000; // (10)

```

55. 假定小端机器中指令: mov AX, 0x12345(BX)

其中操作码 mov 为 40H, 寄存器 AX 和 BX 的编号分别为 0001B 和 0010B, 立即数占 32 位, 则存放顺序为:

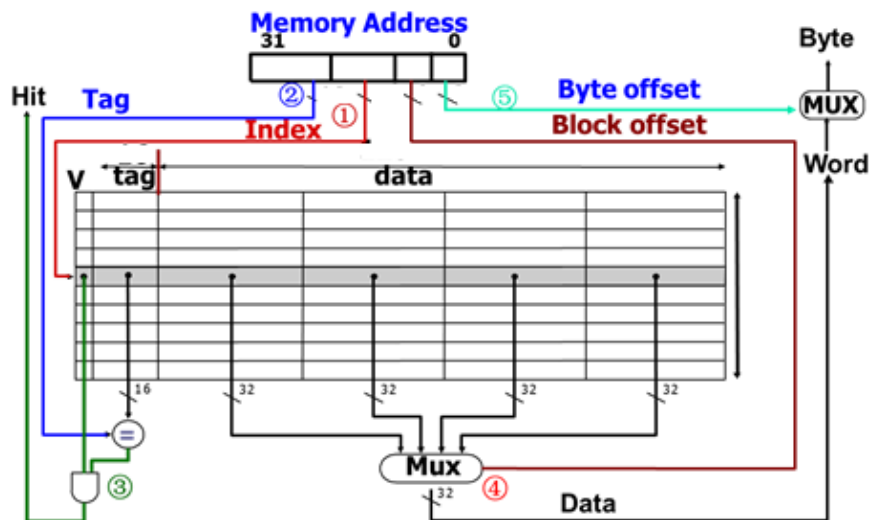
40	1	2	45 23 01 00
----	---	---	-------------

若在大端机器上, 则存放顺序如何?

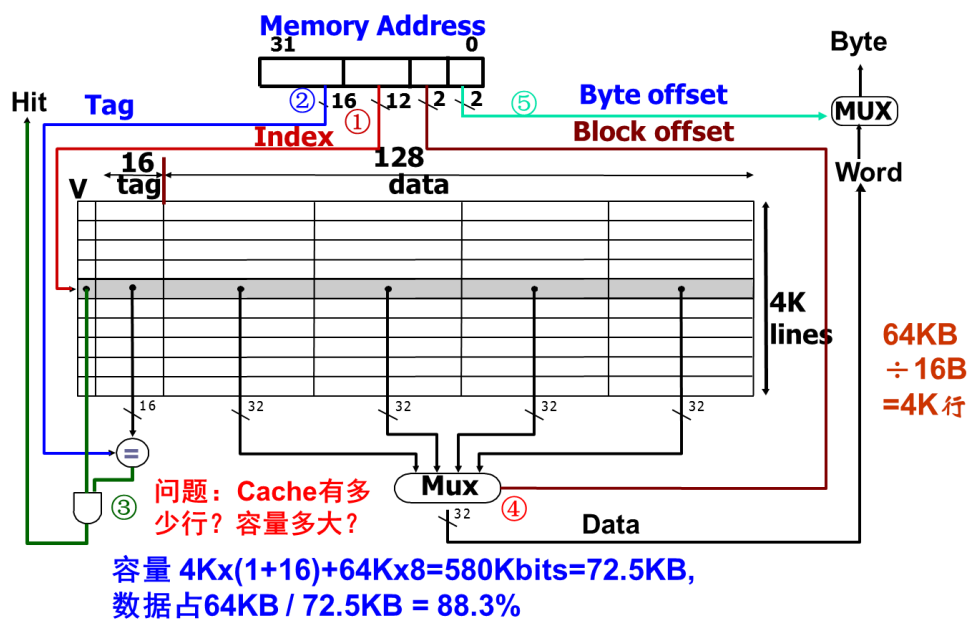
40	1	2	00 01 23 45
----	---	---	-------------

(左边为低地址端)

56. 主存和 Cache 之间直接映射, 块大小为 16B。Cache 的数据区容量为 64KB, 主存地址为 32 位, 按字节编址。要求: 1、说明主存地址如何划分和访存过程。2、Cache 有多少行? 容量多大?



答案：



57. 假定计算机系统主存空间大小为 $32K \times 16$ 位，且有一个 $4K$ 字的 4 路组相联 Cache，主存和 Cache 之间的数据交换块的大小为 64 字。假定 Cache 开始为空，处理器顺序地从存储单元 $0, 1, \dots, 4351$ 中取数，一共重复 10 次。设 Cache 比主存快 10 倍。采用 LRU 算法。试分析 Cache 的结构和主存地址的划分。说明采用 Cache 后速度提高了多少？采用 MRU 算法后呢？

答：假定主存按字编址。每字 16 位。

主存： $32K$ 字= 512 块 $\times 64$ 字 / 块

Cache: 4K 字=16 组 x 4 行 / 组 x 64 字 / 行

标志位	组号	字号
5	4	6

主存地址划分为:

4352/64=68, 所以访问过程实际上是对前 68 块连续访问 10 次。

58. 设某指令系统指令字是 16 位, 每个地址码为 6 位。若二地址指令 15 条, 一地址指令 34 条, 则剩下零地址指令最多有多少条?

解: 操作码按短到长进行扩展编码

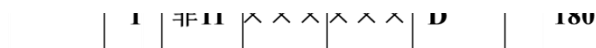
二地址指令: (0000 ~ 1110)

一地址指令: 11110 (00000 ~ 11111); 11111 (00000 ~ 00001)

零地址指令: 11111 (00010 ~ 11111) (000000 ~ 111111)

故零地址指令最多有 $30 \times 2^6 = 15 \times 2^7$ 种

59. 某指令系统的指令字长 12 位, 每个地址码长 3 位, 试提出一种分配方案, 使该指令系统有 4 条三地址指令, 8 条二地址指令, 180 条单地址指令。若二地址指令仅有 7 条, 单地址指令最多可有多少条? (10 分)



若二地址仅 7 条, 则多出: 11 111 000 ~ 111 共 8 条

∴ 单地址最多可有 $((23-4) \cdot 23-7) \cdot 23=200$ 条

60. CPU 的地址总线 16 根 (A15~A0, A0 为低位), 双向数据总线 8 根 (D7~D0), 控制总线中与主存有关的信号有 MREQ~ (允许访存), R/ W~。主存地址空间分配如下: 0 - 8191 为系统程序区, 由只读存储芯片组成, 8192 - 32767 为用户程序区; 最后 (最大地址) 2K 地址空间为系统程序工作区。按字节编址。现有如下芯片:

EPROM : 8K * 8 位 (控制端仅有 CS~)

SRAM : 16K * 1 位, 2K * 8 位, 4K * 8 位, 8K * 8 位

试设计该计算机主存储器。

解:

由题意, 主存地址空间分布如右图:

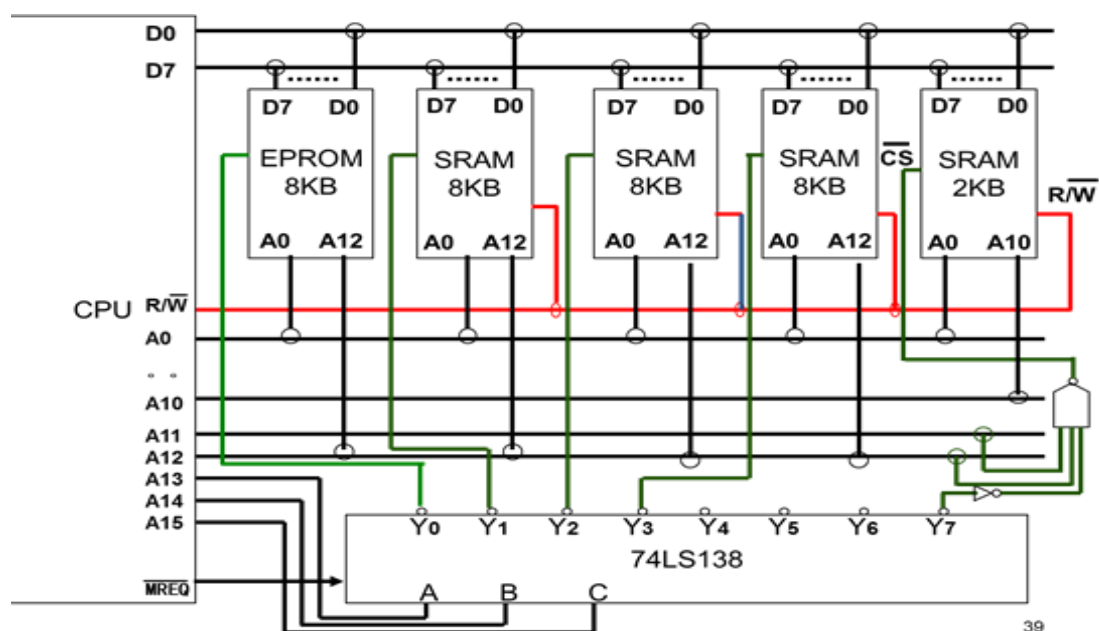
故: 低 8K 范围可选用: 一片 8K * 8 位的 EPROM

24K 用户空间可选用: 3 片 8K * 8 位的 SRAM

高 2K 空间: 只需用 1 片 2K * 8 位的 SRAM 即可。

其连接图如下:

0	
8191	8K (EPROM)
8192	24K (SRAM)
32767	30K (未用)
63487	2K (SRAM)
65535	



61. 设某机主存容量为 4MB, Cache 容量为 16KB, 每块包含 8 个字, 每字 32 位, 设计一个四路组相联映像 (即 Cache 每组内共有四个块) 的 Cache 组织, 要求:

画出主存地址字段中各段的位数。 设 Cache 的初态为空，CPU 需依次从主存第 0，1，2，……，9) 号单元读出 100 个字（主存一次读出一个字。），并重复按此次序读 8 次，问命中率是多少？

〔解〕

(1) 主存容量为 4MB，按字编址，所以主存地址为 20 位，地址格式如下所示：

区号 (8 位) 组号 (7 位) 组内块号 (2 位) 块内地址 (5 位)

(2) Cache 的地址格式为

组号 (7 位) 组内块号 (2 位) 块内地址 (5 位)

主存第 0, 1, 2, …, 99 号单元在第 0 区第 0~12 组中, 被映射到 Cache 的第 0~12 组中, 不存在映射时的组冲突。

由于 Cache 起始为空，所以第一遍读时每一块中的第一个单元没命中（此时要把该块的 8 个字都调入 Cache，这样 CPU 之后再读该块的其它单元就会命中（未考虑块调入过程）），而后面七遍读取时每个单元均可以命中。

$$\text{命中率} = N_c / (N_c + N_m) = (100 - 13 + 7 \times 100) / (8 \times 100) = 98.4\%$$

62. 磁盘组有 6 片磁盘，每片有两个记录面，最上最下两面不用。存储区域内直径 22cm，

外直径 33cm, 道密度为 40 道/cm, 内层位密度 400 位/cm, 转速 2400 转/分, 问: (分)

- 1)、共有多少柱面?
- 2)、盘组总存储容量是多少?
- 3)、数据传输率多少?
- 4)、如果某文件长度超过一个磁道的容量, 应将它记录在同一个存储面上, 还是记录在同一个柱面上?

解:

1、有效存储区域= $33/2 - 22/2 = 16.5 - 11 = 5.5\text{cm}$

柱面数=道密度*有效区域=40道/cm*5.5cm=220道, 即220个圆柱面。

2、内道周长= $3.14 * 22 = 69.08\text{cm}$

道信息量 $N = 400 \text{ 位/cm} * 69.08\text{cm} = 27632 \text{ 位} = 3454\text{B}$

面信息量= $3454\text{B} * 220 = 759880\text{B}$

盘组容量= $759880\text{B} * 10 = 7598800\text{B} \approx 7.25\text{MB}$

3、磁盘数据传输率=转数 r * 道容量 $N = (2400 \text{ 转/60 秒}) * 3454\text{B} = 40 * 3454 = 13816\text{B/S} = 13.492\text{KB/S}$

4、记录在同一个柱面上, 因为不需要重新找道, 数据读写速度快。

63. 以 IEEE754 单精度浮点数格式表示十进制数 $-1/8$,

$-1/8 = -0.125 = -0.001\text{B} = -1.0 \times 2^{-3}$, 阶码为 $-3+127 = 01111100\text{B}$, 数符为1, 尾数为 $1.0\cdots0$, 所以 $-1/8$ 表示为 $1\ 01111100\ 000\ 0000\ 0000\ 0000\ 0000\ 0000$, 用十六进制表示为 BE000000H 。

64. 在 IEEE 754 浮点数运算中, 如何判断浮点运算的结果是否溢出?

浮点运算结果是否溢出, 并不以尾数溢出来判断, 而主要看阶码是否溢出。尾数溢出时, 可通过右规操作进行纠正。阶码上溢时, 说明结果的数值太大, 无法表示; 阶码下溢时, 说明结果数值太小, 可以把结果近似为0。

在进行对阶、规格化、舍入和浮点数的乘/除运算等过程中, 都需要对阶码进行加、减运算, 可能会发生阶码上溢或阶码下溢, 因此, 必须对阶码进行溢出判断。

65. 假定一个存储器系统支持4体交叉存取, 某程序执行过程中访问地址序列为3, 9, 17,

2, 51, 37, 13, 4, 8, 41, 67, 10, 则哪些地址访问会发生体冲突?

解:

对于4体交叉访问的存储系统, 每个存储模块的地址分布为:

Bank0: 0、4、8、12、16

Bank1: 1、5、9、13、17...37...41...

Bank2: 2、6、10、14、18

Bank3: 3、7、11、15、19...51...67

如果给定的访存地址在相邻的4次访问中出现在同一个Bank内, 就会发生访存冲突。所以, 17和9、37和17、13和37、8和4发生冲突。

66. 假定某机主存空间大小1GB, 按字节编址。cache的数据区(即不包括标记、有效位等

存储区)有64KB, 块大小为128字节, 采用直接映射和全写(write-through)方式。

请问:

(1) 主存地址如何划分? 要求说明每个字段的含义、位数和在主存地址中的位置。

(2) cache的总容量为多少位?

参考答案:

(1) 主存空间大小为1GB, 按字节编址, 说明主存地址为30位。cache共有 $64\text{KB}/128\text{B} = 512$ 行, 因此, 行索引(行号)为9位; 块大小128字节, 说明块内地址为7位。因此, 30位主存

地址中，高14位为标志（Tag）；中间9位为行索引；低7位为块内地址。

(2) 因为采用直接映射，所以 cache 中无需替换算法所需控制位，全写方式下也无需修改（dirty）位，而标志位和有效位总是必须有的，所以，cache 总容量为 $512 \times (128 \times 8 + 14 + 1) = 519.5\text{K}$ 位。

67. 某计算机字长 16 位，每次存储器访问宽度 16 位，CPU 中有 8 个 16 位通用寄存器。现为该机设计指令系统，要求指令长度为字长的整数倍，至多支持 64 种不同操作，每个操作数都支持 4 种寻址方式：立即（I）、寄存器直接（R）、寄存器间接（S）和变址（X），存储器地址位数和立即数均为 16 位，任何一个通用寄存器都可作变址寄存器，支持以下 7 种二地址指令格式（R、I、S、X 代表上述四种寻址方式）：RR 型、RI 型、RS 型、RX 型、XI 型、SI 型、SS 型。请设计该指令系统的 7 种指令格式，给出每种格式的指令长度、各字段所占位数和含义，并说明每种格式指令需要几次存储器访问？

指令格式可以有很多种，只要满足以下的要求即可。
操作码字段：6位；寄存器编号：3位；直接地址和立即数：16位；变址寄存器编号：3位；总位数是8的倍数。

指令格式例 1

RR 型	0000	OP (6 位)	R t (3 位)	Rs (3 位)		
RI 型	0010	OP (6 位)	Rt (3 位)	000	Imm16 (16 位)	
RS 型	0100	OP (6 位)	R t (3 位)	Rs (3 位)		
RX 型	0110	OP (6 位)	Rt (3 位)	Rx (3 位)	Offset16 (16 位)	
XI 型	1000	OP (6 位)	Rx (3 位)	000	Offset16 (16 位)	Imm16 (16 位)
SI 型	1010	OP (6 位)	R t (3 位)	000	Imm16 (16 位)	
SS 型	1100	OP (6 位)	Rt (3 位)	Rs (3 位)		

指令格式 2：

RR 型	OP (6 位)	01	Rt (3 位)	01	Rs (3 位)	
RI 型	OP (6 位)	01	Rt (3 位)	00	Imm16 (16 位)	000
RS 型	OP (6 位)	01	Rt (3 位)	10	Rs (3 位)	
RX 型	OP (6 位)	01	Rt (3 位)	11	Rx (3 位)	Offset16 (16 位)
XI 型	OP (6 位)	11	Rx (3 位)	Offset16 (16 位)	00	Imm16 (16 位)
SI 型	OP (6 位)	10	Rt (3 位)	00	Imm16 (16 位)	000
SS 型	OP (6 位)	10	Rt (3 位)	10	Rs (3 位)	

寻址方式字段(2 位)---00: 立即; 01: 寄直; 10: 寄间; 11-变址

68. 假定一个程序重复完成将磁盘上一个 4KB 的数据块读出, 进行相应处理后, 写回到磁盘的另外一个数据区。各数据块内信息在磁盘上连续存放, 并随机地位于磁盘的一个磁道上。磁盘转速为 7200RPM, 平均寻道时间为 10ms, 磁盘最大数据传输率为 40MBps, 磁盘控制器的开销为 2ms, 没有其他程序使用磁盘和处理器, 并且磁盘读写操作和磁盘数据的处理时间不重叠。若程序对磁盘数据的处理需要 20000 个时钟周期, 处理器时钟频率为 500MHz, 则该程序完成一次数据块“读出-处理-写回”操作所需的时间为多少? 每秒秒钟可以完成多少次这样的数据块操作?

参考答案:

平均旋转等待时间: $(1s / (7200/60)) / 2 \approx 8.33/2 \approx 4.17ms$

因为块内信息连续存放, 所以数据传输时间: $4KB / 40MBps \approx 0.1ms$

平均存取时间T : 寻道时间+ 旋转等待时间+ 数据传输时间

$= 10ms + 4.17ms + 0.1ms = 14.27ms$

读出时间(写回时间): $14.27ms + 2ms = 16.27ms$

数据块的处理时间: $20000 / 500MHz \approx 0.04ms$

因为数据块随机存放在某个磁道上, 所以, 每个数据块的“读出-处理-写回”操作时间都是相同的, 所以完成一次操作时间: $16.27ms \times 2 + 0.04ms = 32.58ms$

每秒中可以完成这样的数据块操作次数: $1s / 32.58ms \approx 30$ 次

69. 假定某计算机的 CPU 主频为 500MHz, 所连接的某个外设的最大数据传输率为 20KBps, 该外设接口中有一个 16 位的数据缓存器, 相应的中断服务程序的执行时间为 500 个时钟周期, 则是否可以用中断方式进行该外设的输入输出? 假定该外设的最大数据传输率改为 2MBps, 则是否可以用中断方式进行该外设的输入输出?

参考答案:

(1) 外设最大传输率为 20KBps

每传输完 16 位进行一次中断处理, 因此一秒钟内的中断次数为: $20K/2=10K$ 次

中断响应过程就是执行一条隐指令的过程，所用时间相对于中断处理时间（即执行中断服务程序的时间）而言，几乎可以忽略不计，因而整个中断响应并处理的时间大约： $(1s / 500MHz) \times 500 = 1\mu s$

一秒钟内的中断服务程序执行要用去 $10K \times 1\mu s = 10ms \ll 1s$

所以可以用中断方式进行该外设的输入输出。

(2) 外设最大传输率为2Mbps

每传输完16位进行一次中断处理，因此一秒钟内的中断次数为： $2M/2=1M$ 次

一秒钟内的中断服务程序执行要用去 $1M \times 1\mu s = 1s!$

所以不可以用中断方式进行该外设的输入输出。

70. 若某计算机有5级中断，中断响应优先级为 $1>2>3>4>5$ ，而中断处理优先级为 $1>4>5>2>3$ 。

要求：

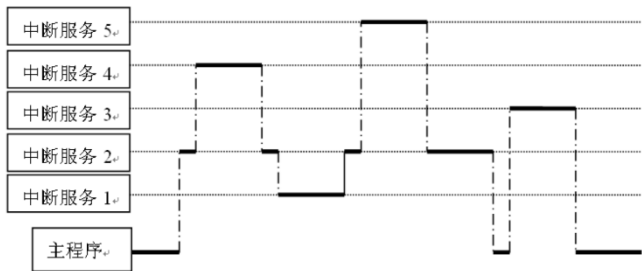
- (1) 设计各级中断处理程序的中断屏蔽位(假设1为屏蔽，0为开放)；
- (2) 若在运行主程序时，同时出现第2、4级中断请求，而在处理第2级中断过程中，又同时出现1、3、5级中断请求，试画出此程序运行过程示意图。

参考答案：

(1) 1级中断的处理优先级最高，说明1级中断对其他所有中断都屏蔽，其屏蔽字为全1；3级中断的处理优先级最低，所以除了3级中断本身之外，对其他中断全都开放，其屏蔽字为00100。以此类推，得到所有中断屏蔽字：

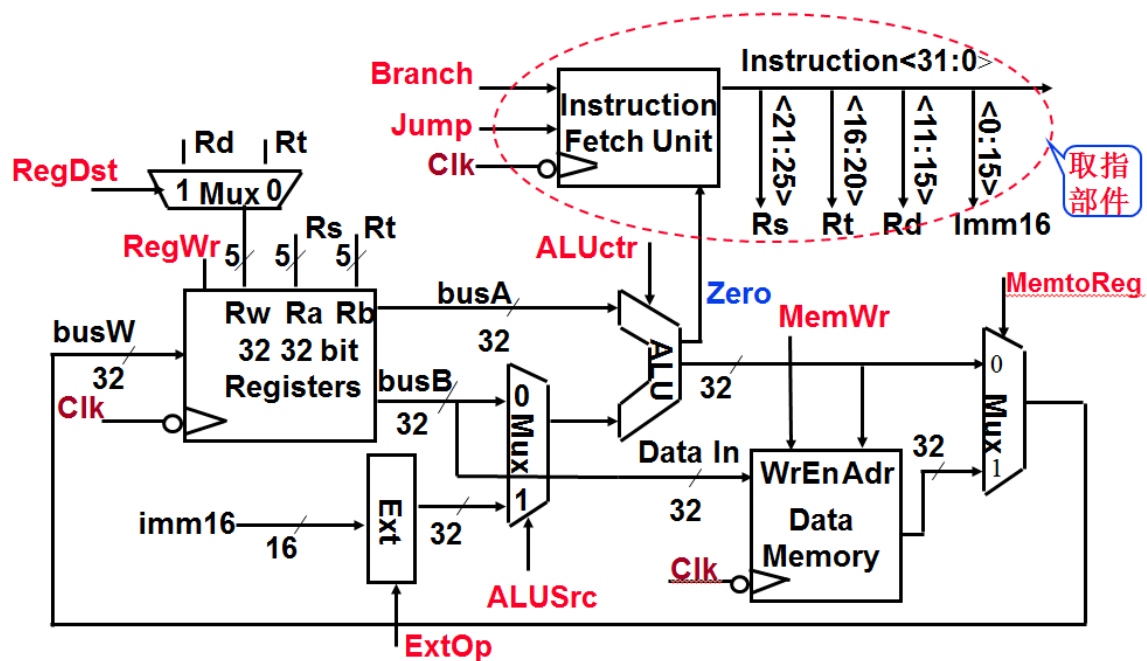
中断程序级别	中断屏蔽字				
	1级	2级	3级	4级	5级
第1级	1	1	1	1	1
第2级	0	1	1	0	0
第3级	0	0	1	0	0
第4级	0	1	1	1	1
第5级	0	1	1	0	1

(2) 程序运行过程示意图



71. 下图为教材中进行单周期处理器设计时使用的数据通路图。

- (1) add 指令的功能为 $R[rd] \leftarrow R[rs] + R[rt]$ ，请描述其执行过程，并将控制信号取值写入表中。
- (2) load 指令的功能为 $R[rt] \leftarrow \text{Data Memory} \{R[rs] + \text{SignExt}[\text{imm16}]\}$ ，请描述其执行过程，并将控制信号取值写入表中。



- (1) 操作数从寄存器 rs 和 rt 描述的寄存器从输出，进入 ALU 完成加法运算，计算结果送入 rd 所描述的寄存器中。
 - (2) Load 指令先对立即数 imm16 进行符号扩展，然后和寄存器 Rs 的内容相加得到访存地址，最后从该地址读取一个 32 位数送到寄存器 Rt 中。
- 控制信号取值如下表：

控制信号	(1)题取值	(2)题取值
RegDst	1	0
ALUSrc	0	1
MemtoReg	0	1
RegWr	1	1
MemWr	0	0
Branch	0	0
Jump	0	0
ExtOp	x	1