华中科技大学

HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

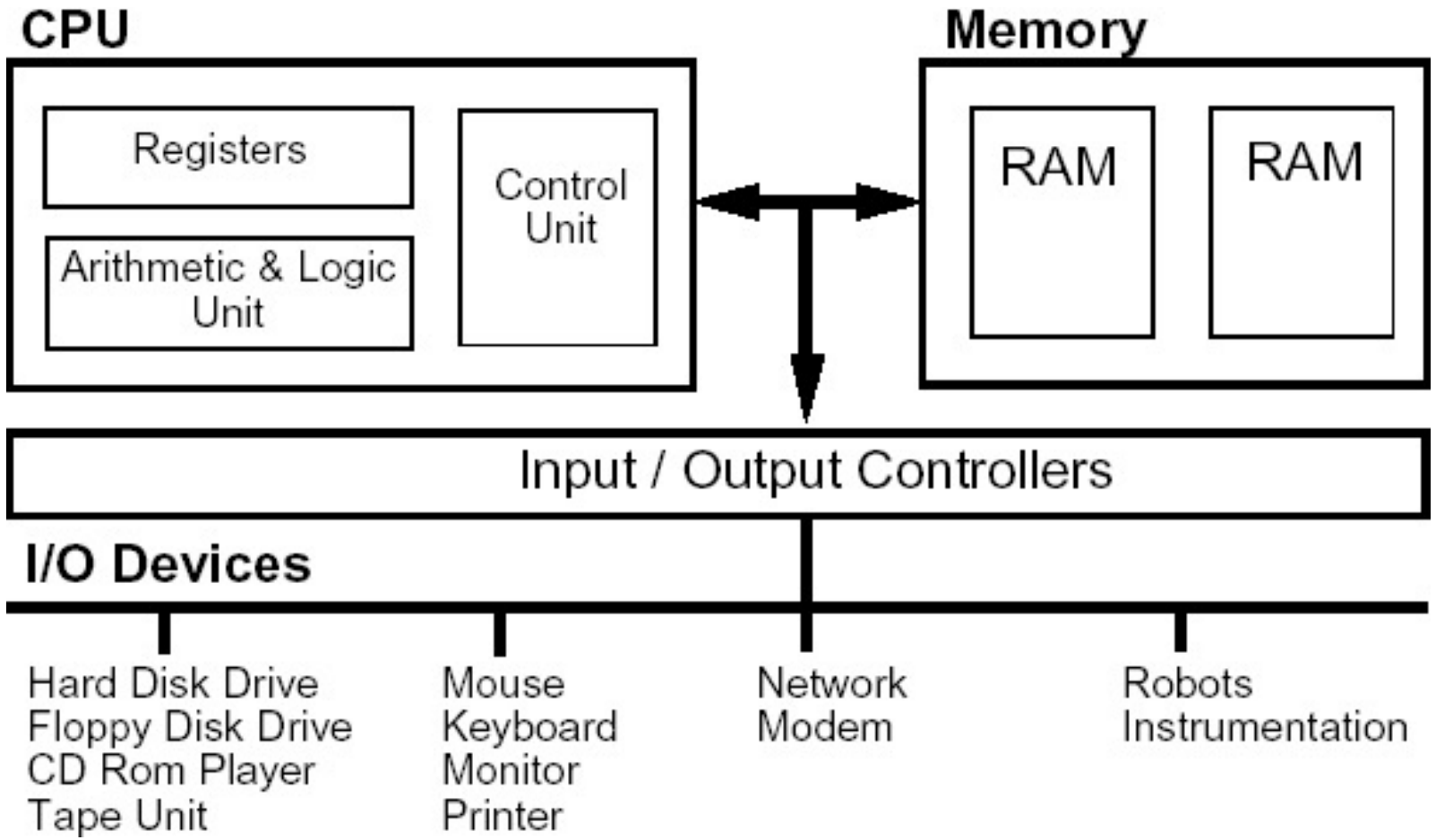
网络空间安全学院

2.1 x86 汇编语言基础

网络空间安全学院　慕冬亮

Email : dzm91@hust.edu.cn

计算机体系架构


CPU
Registers
Arithmetic & Logic Unit
Control Unit
Memory
RAM
RAM
Input / Output Controllers
I/O Devices
Hard Disk Drive
Floppy Disk Drive
CD Rom Player
Tape Unit
Mouse
Keyboard
Monitor
Printer
Network
Modem
Robots
Instrumentation

x86 汇编基础知识回顾

汇编语言是最简单的编程语言，因为需要CPU来理解并执行
汇编指令由操作码和操作数组成，如 add rax, 0x10

- 操作码由助记符来说明具体功能
- 操作数有三种形式：
  - 立即数，0x12345678
  - 寄存器, al/ah/ax/eax/rax
  - 内存, [0x32], [rax], [rax+4], [rax+rbx], [rax+rbx*2+0x4]

此处关键是大家有能力阅读汇编代码及编写部分汇编片段

x86重要汇编指令

传送指令

- mov/xchg/lea
- push/pop

算数运算类指令

- add/sub/mul/div/
- inc/dec/neg/cmp

位操作指令

- not/and/or/test
- ror/rol

字符串操作指令

- movs/cmps/scas

控制转移类指令

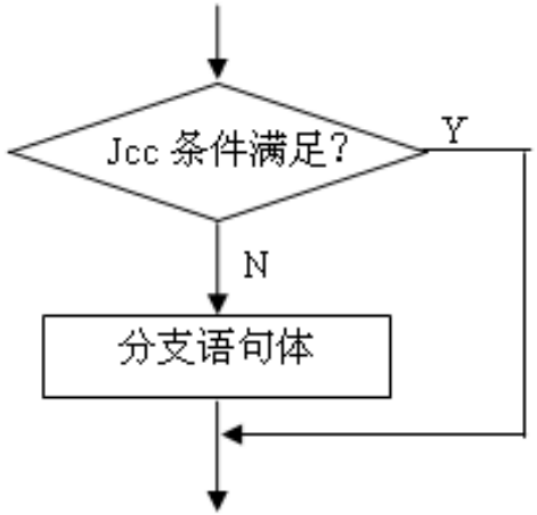
- jmp/jcc
- call/ret
- loop

处理器控制类指令

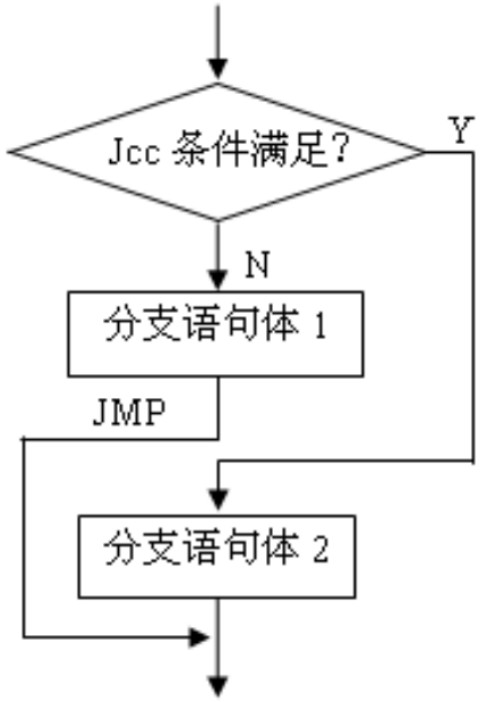
- cli/cld/std/sti

.....

汇编程序设计之分支结构



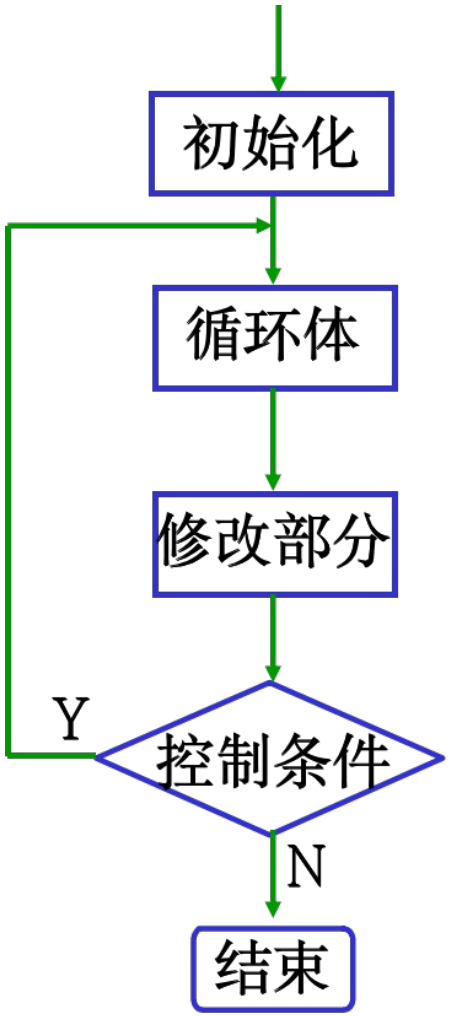
（a） 单分支结构



（b） 双分支结构

```
;  计算AX的绝对值
        cmp ax,0    ;注意cmp指令影响的符号位
        jns nonneg      ;分支条件：AX≥0
        neg ax          ;条件不满足，求补
nonneg: mov result,ax ;条件满足
```

汇编程序设计之循环结构




```
sum     dw ?
        ;代码段
        xor ax,ax          ;被加数AX清0
        mov cx,100
again:  add ax,cx
        ;从100,99,...,2,1倒序累加
        loop again
        mov sum,ax         ;累加和送入指定单元
```

x86_64 vs x86_32

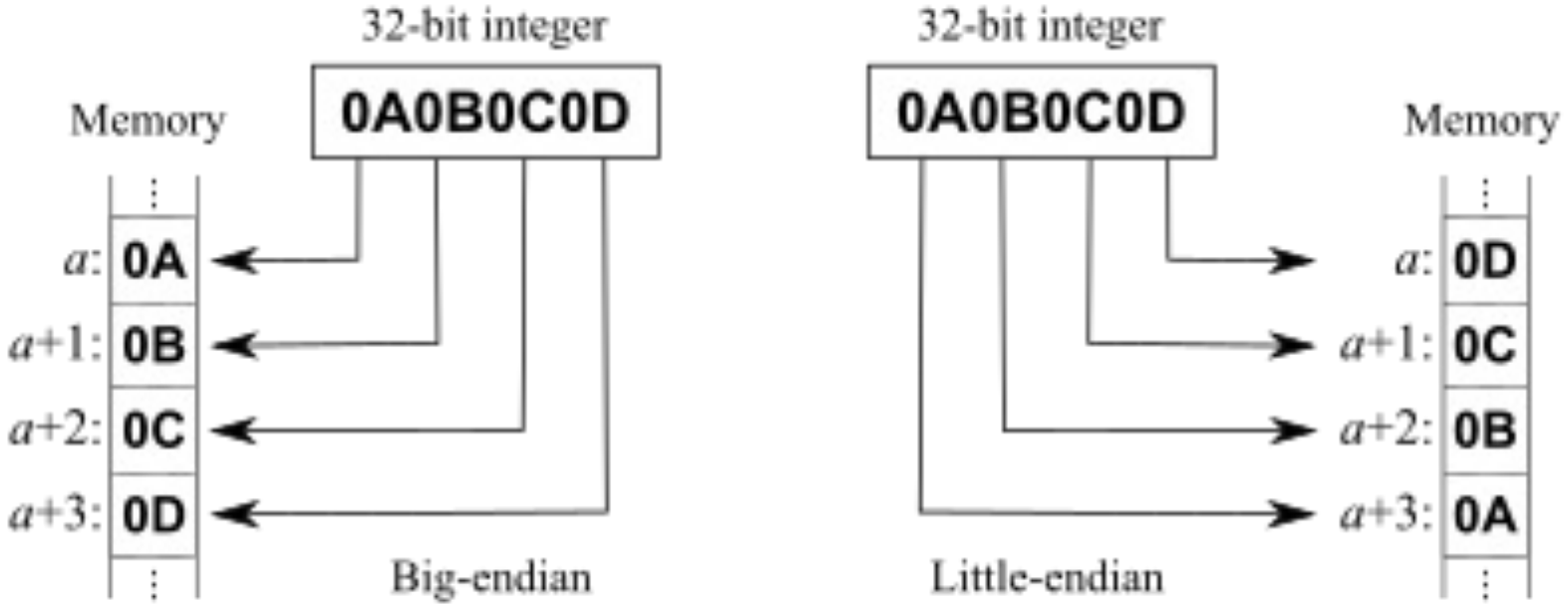
- x86_64 拥有更多的通用寄存器
  - RAX - RDX vs EAX - EDX
  - RSI/RDI vs ESI/EDI
  - RBP/RSP vs EBP/ESP
  - R8 - R15 (x86_64 独有)
- x86_32 调用规范
  - 函数参数按照从右向左压到栈中
- x86_64 调用规范
  - 前六个参数： RDI, RSI, RDX, RCX, R8, R9
  - 与 x86_32 相同，后面的参数按照自右向左依次压到栈中

[1] https://wiki.osdev.org/Calling_Conventions

x86_64 寄存器

64位寄存器	32位子寄存器	16位子寄存器	8位子寄存器
r8	r8d	r8w	r8b
r9	r9d	r9w	r9b
r10	r10d	r10w	r10b
r11	r11d	r11w	r11b
r12	r12d	r12w	r12b
r13	r13d	r13w	r13b
r14	r14d	r14w	r14b
r15	r15d	r15w	r15b

大端存储 vs 小端存储



- 大端：指数据的低位保存在内存的高地址中，数据的高位保存在 内存的低地址中
- 小端：指数据的低位保存在内存的低地址中，数据的高位保存在 内存的高地址中

x86处理器的工作模式

- x86处理器支持3种工作模式：实模式、保护模式和虚拟8086模式。
  - 实模式和虚拟8086模式是为了向下兼容8086处理器的程序而设计。

实模式

- 80x86处理器在复位或加电时是以实模式启动的。
- 寻址方式：20位寻址（段＋偏移），1M空间。
- 不能对内存进行分页管理。
- 不支持优先级，所有的指令相当于工作在特权级(优先级0)。
- 切换到保护模式：通过在实模式下初始化控制寄存器，GDTR，LDTR等管理寄存器以及页表，然后再置位CR0寄存器的保护模式使能位（PE：Protected-Mode Enable，第0位）。

保护模式

- 是80x86处理器的常态工作模式；
- 32位处理器支持32位寻址，物理寻址空间达4G。
- 支持内存分页机制，提供了对虚拟内存的良好支持；
- 支持优先级机制，根据任务特性进行了运行环境隔离；
- 切换到实模式：通过修改控制寄存器CR0的PE位（Protected-Mode Enable，第0位），切换到实模式。

虚拟8086模式

- 为了在保护模式下兼容8086程序而设置的。
- 虚拟8086模式是以任务的形式在保护模式上执行的，在80X86上可以同时支持多个真正的80X86任务和虚拟8086模式构成的任务。
- 支持任务切换和内存分页。
  - 操作系统用分页机制将不同的虚拟8086任务的地址空间映射到不同的物理地址上面去，使得每个虚拟8086任务看来都认为自己在使用0～1MB的地址空间。

三种工作模式关系

Intel80X86处理器三种工作模式关系：

实模式、保护模式和虚拟86模式