汇编语言

第二次实验

吴昊

Tip 1：信息存储

- 大多数计算机中使用8bits的块（字节byte）来作为最小的可寻址单位。
- 1byte = 8bits，值域是00000000_2~11111111_2。也就是十进制0_{10} ~ 255_{10}。
- 十进制与位模式转换很麻烦，因此我们引入了十六进制（hex）0 ~ 9、A ~ F来表示16个可能的值，以0x开头标示。
- 每台计算机都有一个字长word，来指明指针数据的大小。对于一个字长为w的机器来说，寻址范围就是0 ~ 2^w-1。

C声明	32-bit	64-bit	C声明	32-bit	64-bit
[unsigned] char	1	1	char *	4	8
[unsigned] short	1	1	float	4	4
[unsigned] int	4	4	double	8	8
[unsigned] long	4	8			

Tip 2：C语言移位运算

操作	值
参数 x	[01100011] [10010101]
x << 4	[0011***0000***] [0101***0000***]
x >> 4 (逻辑右移)	[***0000***0110] [***0000***1001]
x >> 4 (算数右移)	[***0000***0110] [***1111***1001]

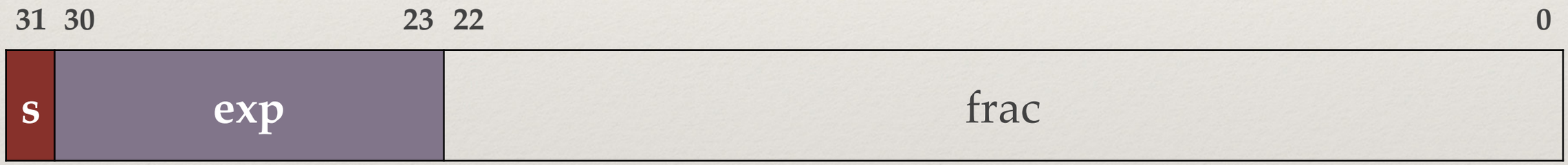
1. C语言没有明确定义有符号数用哪种类型的右移。
2. 约定俗称的规则，有符号数用算数右移，无符号数右移必须是逻辑的。
3. 如果要是移动k很大怎么破，比如int32位，移动32bit？（答：k mod w）

Tip 3：浮点数表述

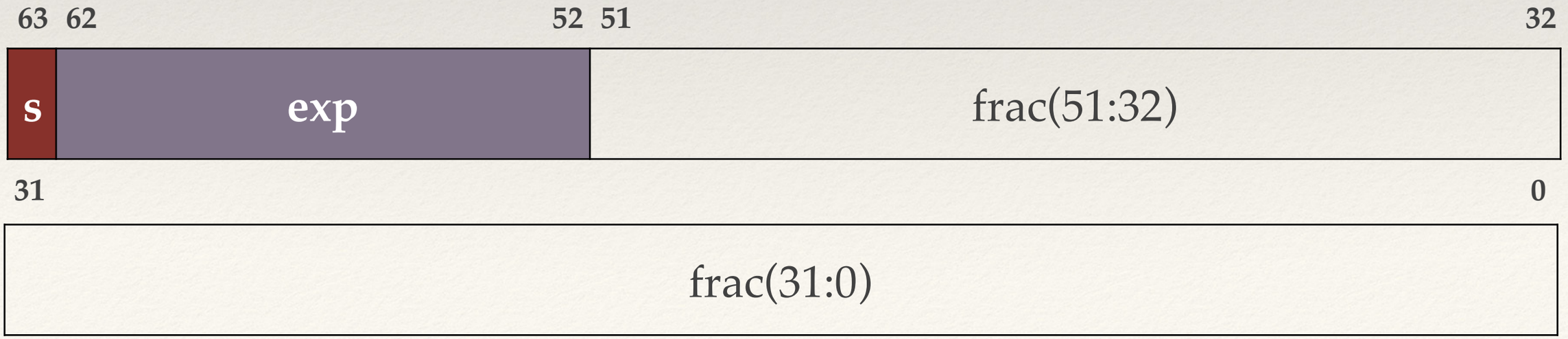
浮点数$V = (-1)^s \times M \times 2^E$

s决定正负；M（尾数）是一个二进制小数；E（阶码）浮点数的权重

单精度 float



双精度 double



Tip 4：条件码

除了整数寄存器，CPU维护着一组**单个位**的**条件码**（conditioncode）寄存器，用于描述最近的算数或者逻辑运算的属性，常见的有：

- CF: 进位标志。最近的操作使最高位产生了进位。可用来检查无符号操作的溢出。
- ZF: 零标志。最近的操作得出的结果为0.
- SF: 符号标志。最近的操作得到的结果为负数。
- OF: 溢出标志。最近的操作导致一个补码溢出—正溢出或负溢出。

例： t = a + b

CF	(unsigned) t < (unsigned) a	无符号溢出
ZF	(t == 0)	零
SF	(t < 0)	负数
OF	(a<0 == b<0)&&(t<0 != a<0)	有符号溢出

Tip 5： gdb

1. DGB是一个debugger，debugger 能够在程序运行时检查甚至修改变量。

2. GDB基本使用方法参见讲义，如： r/c/s/si/n/ni/b/d/p/i/q等。

(GDB cheatsheet： https://gist.github.com/jez/53a86b4e94da6edfdf97eaa4c12de2bb)

(For more: https://sourceware.org/gdb/onlinedocs/gdb/index.html#SEC_Contents)

Tip 6： gdb examine memory

- **x 以指定格式显示给定内存地址的内容，语法如下：**

  x [Address expression]

  x /[Format] [Address expression]

  x /[Length][Format][Address expression]

Address expression: 指定内存地址，含寄存器($eip)，伪寄存器($pc)，如果没指定地址，command就会继续显示上次x指令运行后紧跟着的地址。

Format: o/x/d/u/f/a/c/s/i/b/h/w/g

Length: 要显示的元素（由Format决定）个数。

Tip 6： gdb TUI

基本命令 **layout [name]** 提供多个text显示窗口

next 显示下一个layout

prev 显示前一个layout

src　显示源码和命令窗口

asm　显示汇编和命令窗口

split 显示源码，汇编和命令窗口

regs　显示源码，汇编和命令窗口。若目前在asm或split layout中，同时显示寄存器，汇编，命令窗口。