原码: 0/1 ______ Binary digit ______

数符　　　尾数

定点数: { 整数 / 小数　0/1 . ______ Binary digit ______

表示正负　　32位 → 1位, 31位

整:

原码　1 1111111… ~ 0 11111…

$-(2^{31}-1) \sim 2^{31}-1$

补码　1 00000… ~ 011111

$-2^{31} \sim 2^{31}-1$

小: 原: 1.1111111 ~ 0.1111111..

$-(1-2^{-31})$　　$1-2^{-31}$

条件码

CF → unsigned　ZF　SF　OF → signed

但不都一定

leaq 不改变 其他都会设置 (jge 不改变 (+)

inc V

sar/shl X

计算磁盘容量

磁道密度 单位为自之 每inch多少磁道

位密度 bits/inch

旋转速度 RPM

总容量

eg. 内径/0 inches 外径20 inches 直径

磁盘 25/inches 位密 2000 bits/inch 磁道 5000 RPM 最内圈

总容量: $\frac{(20-10)}{2} \cdot 25 \cdot (2000 \times 10\pi)$

数据传输率: 一转 $\frac{60}{5000}$ s $\frac{3}{250}$ s

$2000 \times 10\pi$ bits

$\frac{3}{250}$ s

$= \frac{2000 \times 200}{3}$ bits/s

750

高速缓存 有关当

直写 当回 写分配 非写分配

知道 访问数据

直写+非写分配 / 写回+写分配

写技术

高速缓存的一致性保持技术

条件传送与条件分支.

cmovge %rdx, %rax

ret

练习题 3.21 C 代码开始的形式如下：

```
long test(long x, long y) {
    long val = __8*x__;
    if (__y>0__) {
        if (__x>y__)
            val = __x+y__;
        else
            val = __y-x__;
    } else if (__y≤-2__)
        val = __x+y__;
    return val;
}
```

GCC 会产生如下汇编代码：

```
long test(long x, long y)
x in %rdi, y in %rsi
test:
    leaq    0(,%rdi,8), %rax
    testq   %rsi, %rsi
    jle     .L2          y≤0
    movq    %rsi, %rax   y
    subq    %rdi, %rax   y-x
    movq    %rdi, %rdx   x
    andq    %rsi, %rdx   x+y
    cmpq    %rsi, %rdi   y  x-y>0
    cmovge  %rdx, %rax
    ret                  val= x+y
.L2:
    addq    %rsi, %rdi   x+y
    cmpq    $-2, %rsi    y+2≤0
    cmovle  %rdi, %rax
    ret
```

else

2路组相联

2-Way Set Associative Cache Simulation

t=2	s=1	b=1
xx	x	x

M=16 byte addresses, B=2 bytes/block,
S=2 sets, E=2 blocks/set

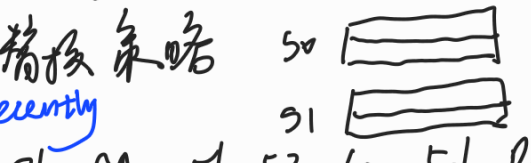
Address trace (reads, one byte per read):

0	[0000₂],	miss
1	[0001₂],	hit
7	[0111₂],	miss
8	[1000₂],	miss
0	[0000₂]	hit

	v	Tag	Block
Set 0	1	00	M[0-1]
	1	10	M[8-9]
Set 1	1	01	M[6-7]
	0		

Eg. 设 Cache 二路组相联，总容量 32 bytes，每块 8 bytes

每次访问 1 byte 使用 LRU 替换策略（→ recently）



设顺序访问主存的地址流：0x0b, 0a, 56, 52, 1c, 5d, 0a, 53

	访问地址	Tag	Index	Offset	是否命中	Cache 内容（T(i)表示 index 中的 Tag）
1011	0x0b	0	1	3	F	T(1)=0
1010	0a	0	1	2	F	T(1)=0/0
0110	56	5	0	6	F	T(1)=0/0 T(0)=5
0010	52	5	0	2	T	T(1)=0/0 T(0)=5
1100	1c	1	1	4	F	T(1)=1/0 T(0)=5
1101	5d	5	1	5	F	T(1)=1/5 T(0)=5
1010	0a	0	1	2	F	T(1)=0/5 T(0)=5
0011	53	5	0	3	T	T(1)=0/5 T(0)=5

53 → Tag；Index+Offset

Eg. 某虚拟存储系统，主存有 4 Pages，采用 LRU 页替换算法

设页面访问序列为 3, 2, 1, 6, 1, 2, 5, 7, 3, 1.

会发生的缺页次数且多少？

3 2 1 6 1 2 5 1 3 1

用一种DS　　访问→加入　重复：删除再加入

满了 弹出，加入

例. 某页存储系统共6 pages，分配给程序使用的主存有4 pages，分别用 FIFO/LRU（FIFO：队列，进入最久的；LRU：访问最久的，改造的队列），程序中访问的页面如下（页面调度）　访入 替换 命中

页地址流	P1	P3	P4	P1	P6	P2	P3	P5	P1	P3	页命中率
FIFO	1	1	1	1	1#	2	2	2	2	2#	20%
		3	3	3	3	3#	3#	5	5	5	
			4	4	4	4	4	4#	1	1	
					6	6	6	6	1#	3	
	入	入	入	中	入	换	中	换	换	换	
LRU	1	1	1	1	1	1	1#	5	5	5	20%
		3	3	3	3#	2	2	2	2#	2#	
			4	4	4	4#	3	3	3	3	
					6	6	6	6#	1	1	
	入	入	入	中	入	换	换	换	换	中	