



中山大學
SUN YAT-SEN UNIVERSITY



计算机组成原理

第三章：计算机中的运算

中山大学计算机学院
陈刚

2022年秋季

Condition Codes

One often wants 4 other bits of information from an arith unit:

□ Z (zero): result is = 0

□ big NOR gate

□ N (negative): result is < 0

□ S_{N-1}

□ C (carry): indicates that most significant position produced a carry, e. g., “1 + (-1)” (表示借位)

□ Carry from last FA

□ V (overflow): indicates answer doesn't fit

□ precisely:
$$V = A_{i-1}B_{i-1}\bar{N} + \bar{A}_{i-1}\bar{B}_{i-1}N$$

-or-

$$V = CO_{i-1} \oplus CI_{i-1}$$

To compare A and B, perform A-B and use condition codes:

Signed comparison:

LT $N \oplus V$

LE $Z + (N \oplus V)$

EQ Z

NE $\sim Z$

GE $\sim (N \oplus V)$

GT $\sim (Z + (N \oplus V))$

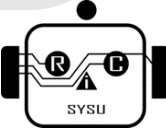
Unsigned comparison:

LTU C

LEU $C + Z$

GEU $\sim C$

GTU $\sim (C + Z)$



N位无符号数比较

□ $A - B \quad (A > 0, B > 0)$

□ $B + (-B)_{\text{补}} = 2^{(N+1)} \quad (\text{补码公式})$

□ $A - B = A - (2^{(N+1)} - (-B)_{\text{补}}) = A + (-B)_{\text{补}} - 2^{(N+1)}$

□ 如果 $A + (-B)_{\text{补}}$ 有最高位进位, 说明不需要借位

□ 则 $A + (-B)_{\text{补}} - 2^{(N+1)} > 0$

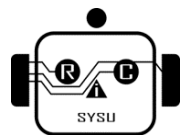
□ $A > B$

□ 如果 $A + (-B)_{\text{补}}$ 有最高位没有进位, 说明需要借位

□ 则 $A + (-B)_{\text{补}} - 2^{(N+1)} < 0$

□ $A < B$

□ 实际中借位和进位是相反的



举个例子

□ 考虑4位无符号数A和B

□ $A=0x0001$, $B=0x0111$;

□ 正常计算里面

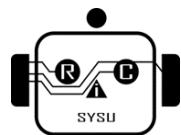
□ $A-B=0x0001+0x1001=0x1001$ (没有进位, $Z=0$, 所以 $A<B$)

□ 如果 $A=0x0111$, $B=0x0001$

□ $A-B=0x0111+0x1111=0x0110$ (有进位, $Z=0$, 所以 $A>B$)

□ 如果 $A=0x001$, $B=0x0001$

□ $A-B=0x0001+0x1111=0x0000$ (有进位, $Z=1$, 所以 $A=B$)



联系方式

□ Acknowledgements:

□ This slides contains materials from following lectures:

- Computer Architecture (ETH, NUDT, USTC)

□ Research Area:

- 计算机视觉与机器人应用计算加速,
- 人工智能和深度学习芯片及智能计算机

□ Contact:

- 中山大学计算机学院
- 管理学院D101 (图书馆右侧)
- 机器人与智能计算实验室
- cheng83@mail.sysu.edu.cn

