



南京大學
NANJING UNIVERSITY



从C表达式到逻辑电路

南京大学

计算机科学与技术系

袁春风

email: cfyuan@nju.edu.cn

2015.6

开场白

上一讲我们介绍了计算机中最基本的运算电路，我们知道，计算机中的所有运算都是由相应的运算电路完成的，而这些运算电路是由基本的逻辑门电路实现的。

那么，计算机是如何知道在运算电路中该执行什么操作？该对什么样的操作数进行运算的呢？

本讲主要介绍高级语言程序中的表达式、运算类指令和运算电路之间的关系。

C语言支持的基本数据类型

C语言声明	操作数类型	存储长度（位）
(unsigned) char	整数 / 字节	8
(unsigned) short	整数 / 字	16
(unsigned) int	整数 / 双字	32
(unsigned) long int	整数 / 双字	32
(unsigned) long long int	-	2×32
char *	整数 / 双字	32
float	单精度浮点数	32
double	双精度浮点数	64
long double	扩展精度浮点数	80 / 96

整数类型分：无符号整数和带符号整数

从C表达式到运算类指令

C语言程序中的基本数据类型、基本运算类型

–基本数据类型

- 无符号数(二进制位串)、带符号整数(补码)
- 浮点数(IEEE 754标准)
- 位串、字符(串)(ASCII码)

–基本运算类型

- 算术(+ - * / % > < >= <= == !=)
- 按位(| & ~ ^)
- 逻辑(|| && !)
- 移位(<< >>)
- 扩展和截断

例如：对于C赋值语句 $y=(x>>2)+k$ ，如何在计算机中实现呢？

转换为指令序列，通过执行运算类指令来实现！

从运算类指令到运算电路

计算机如何实现高级语言程序中的运算？

—将各类表达式编译（转换）为指令序列

例如： $y = (x \gg 2) + k$ 转换为以下指令序列：

`sarw $2, %ax ; x >> 2`

`addw %bx, %ax ; (x >> 2) + k`

—计算机直接执行指令来完成运算

控制器对指令进行译码，产生控制信号送运算电路


—操作数在运算电路中运算

`sarw $2, %ax`：将操作数“2”和“R[ax]”送移位器运算

`addw %bx, %ax`：将R[ax]和R[bx]送整数加减器中运算

移位器和整数加减运算器都是由逻辑门电路构成的！

数据的运算

- 高级语言程序中涉及的运算（以C语言为例）
 - 整数算术运算、浮点数算术运算
 - 按位、逻辑、移位、位扩展和位截断等运算
 - 指令集中涉及到的运算
 - 涉及到的定点数运算
 - 算术运算
 - 带符号整数：取负 / 符号扩展 / 加 / 减 / 乘 / 除 / 算术移位
 - 无符号整数：0扩展 / 加 / 减 / 乘 / 除 / 逻辑左移 / 逻辑右移
 - 逻辑运算
 - 逻辑操作：与 / 或 / 非 / ...
 - 涉及到的浮点数运算：加、减、乘、除
 - 指令中的运算操作在运算电路中进行
 - 基本运算部件ALU、通用寄存器组，以及其他部件
- 
- The diagram consists of two large, light blue curved arrows. The first arrow starts at the top right of the first bullet point ('高级语言程序中涉及的运算') and points down to the second bullet point ('指令集中涉及到的运算'). The second arrow starts at the top right of the second bullet point and points down to the third bullet point ('指令中的运算操作在运算电路中进行').