

## P133-137

- 特殊的算术操作

指令	效果	描述
imulq S	$R[\%rdx]: R[\%rax] \leftarrow S \times R[\%rax]$	有符号全乘法
mulq S	$R[\%rdx]: R[\%rax] \leftarrow S \times R[\%rax]$	无符号全乘法
cltq	$R[\%rdx]: R[\%rax] \leftarrow \text{符号扩展}(R[\%rax])$	转换为八字
idivq S	$R[\%rdx] \leftarrow R[\%rdx]: R[\%rax] \bmod S$ $R[\%rdx] \leftarrow R[\%rdx]: R[\%rax] \div S$	有符号除法
divq S	$R[\%rdx] \leftarrow R[\%rdx]: R[\%rax] \bmod S$ $R[\%rdx] \leftarrow R[\%rdx]: R[\%rax] \div S$	无符号除法

图 3-12 特殊的算术操作。这些操作提供了有符号和无符号数的全 128 位乘法和除法。  
一对寄存器 `%rdx` 和 `%rax` 组成一个 128 位的八字

### 1. 条件控制

#### a. 条件码

- 除了整数寄存器，cpu 还维护了一组条件码寄存器，用于描述算术或逻辑操作的属性。

CF：进位标志

ZF：零标志

SF：符号标志

OF：溢出标志

#### b. 访问条件码，三种方式

- 根据条件码某种组合，将一个字节设置为 0 或 1
- 条件跳转到程序某个其他部分
- 有条件地传送数据

#### c. 第一种方式其实就是 SET 指令：

一条 `set` 指令的目的操作数是低位单字节寄存器元素之一，当然，高位需要清零。

指令	同义名	效果	设置条件
sete <i>D</i>	setz	$D \leftarrow ZF$	相等/零
setne <i>D</i>	setnz	$D \leftarrow \sim ZF$	不等/非零
sets <i>D</i>		$D \leftarrow SF$	负数
setns <i>D</i>		$D \leftarrow \sim SF$	非负数
setg <i>D</i>	setnle	$D \leftarrow \sim(SF \wedge OF) \wedge \sim ZF$	大于（有符号>）
setge <i>D</i>	setnl	$D \leftarrow \sim(SF \wedge OF)$	大于等于（有符号>=）
setl <i>D</i>	setnge	$D \leftarrow SF \wedge OF$	小于（有符号<）
setle <i>D</i>	setng	$D \leftarrow (SF \wedge OF) \vee ZF$	小于等于（有符号<=）
seta <i>D</i>	setnbe	$D \leftarrow \sim CF \wedge \sim ZF$	超过（无符号>）
setae <i>D</i>	setnb	$D \leftarrow \sim CF$	超过或相等（无符号>=）
setb <i>D</i>	setnae	$D \leftarrow CF$	低于（无符号<）
setbe <i>D</i>	setna	$D \leftarrow CF \vee ZF$	低于或相等（无符号<=）

图 3-14 SET 指令。每条指令根据条件码的某种组合，将一个字节设置为 0 或者 1。  
有些指令有“同义名”，也就是同一条机器指令有别的名字