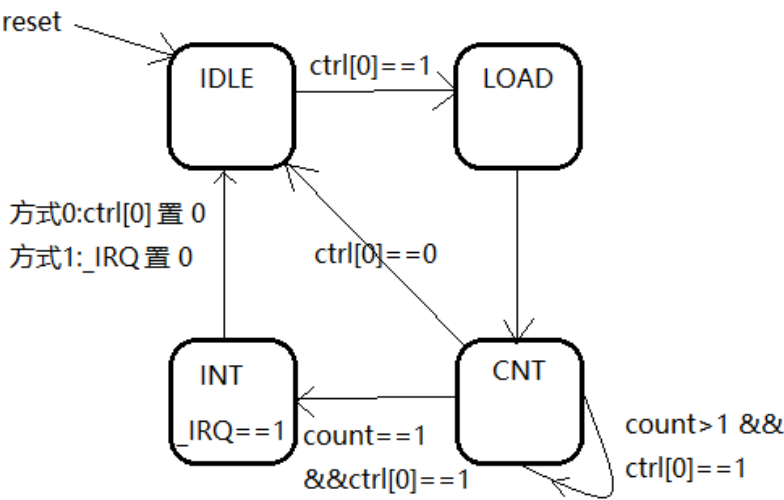


# 计时器说明

端口定义：

名称	方向	说明
clk	I	时钟信号
reset	I	重置信号
Addr	I	输入地址
WE	I	是否写计时器
Din	I	数据输入
Dout	O	数据输出
IRQ	O	模拟的中断信号

状态转移图：



说明：

内置主要有 3 个寄存器：ctrl，present，count，对应 mem[0]，mem[1]，mem[2]。此外还有 \_IRQ 寄存器，输出的中断定义为 \_IRQ & ctrl[3]

一共有 4 个状态：IDLE、LOAD、CNT、INT。

时钟上跳沿时，如果有写使能，优先响应写寄存器的操作。如果不需要

写，就按照状态来转移。

IDLE 是初始态，如果 `ctrl[0]==0` 就转到 LOAD，把 `present` 寄存器的值保存到 `count` 寄存器，方便倒数。到 CNT 时，如果 `ctrl[0]==0`，直接回 IDLE。不为 0 就倒数。倒数结束以后，转到 INT。此时 `_IRQ` 置 1，就是说如果 `ctrl[3]` 也为 1，就有中断信号了。在 INT 阶段，如果 `ctrl[2:1]==0`（方式 0），就把 `ctrl[0]` 置 0；否则把 `_IRQ` 置 0（方式 1），此时没有中断。方式 0 会保持中断，由于 `ctrl[0]` 为 0，需要外界操作（写寄存器）来改变状态，否则就一直在这个状态。方式 1 如果没有外界操作，由于没有改动 `ctrl[0]`，会继续到下一个状态。

### 注意：

用户只能改 `ctrl`（控制状态转移），`present`（控制倒数的时间）寄存器，而不能改 `count` 寄存器。如果不慎改动 `count`，会使得定时错乱。此时需要外界的中断异常处理来避免这种情况。

任意时钟上沿，用户的改变的优先级高，内部的循环优先级低。