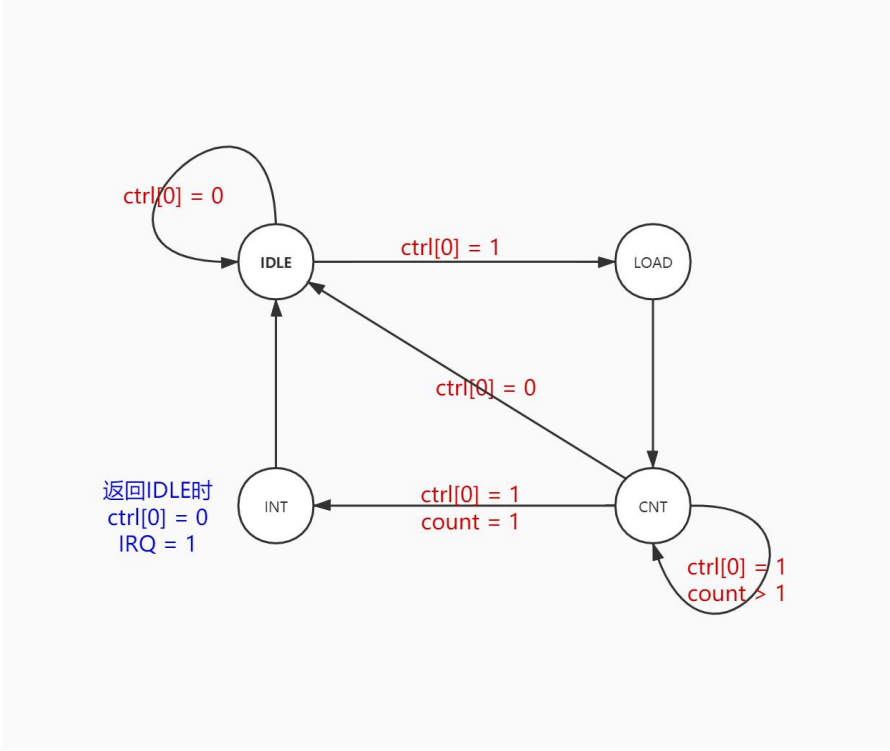


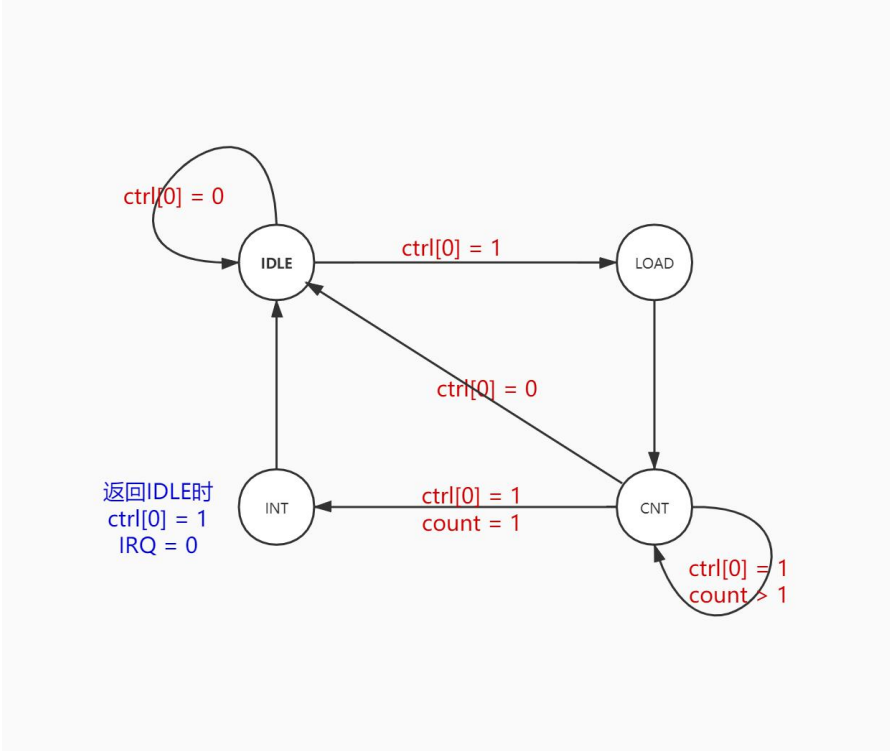
COCO 定时器使用说明

一、状态转移图

1. 模式 0 下定时器状态转移图



2. 模式 1 下定时器状态转移图



二、计时器端口与结构

信号名	方向	描述
clk	I	时钟信号
reset	I	复位信号
Addr[31:2]	I	30 位输入地址，按字存取
TimerWrite	I	1 位输入 Timer 写使能信号
Din[31:0]	I	32 位输入存入 Timer 的数据
Dout[31:0]	O	32 位输出 Timer 读取数据
IRQ	O	1 位输出 Timer 产生的中断请求信号

计时器中包含三个 32 位寄存器，对应存储空间连续的 12 个字节；一个 2 位状态寄存器，不占用存储部分空间。

寄存器名称	寄存器功能
CTRL[31:0]	控制寄存器，支持读写。[3]号位用于中断屏蔽，0 禁止中断，1 允许中断；[2:1]号位用于模式选择；[0]号位为计数器使能信号，0 停止计数，1 允许计数
PRESET[31:0]	初值寄存器，支持读写。用于记录将用于倒计时的初始值，倒计时从此数据开始倒数至 0。
COUNT[31:0]	计数值寄存器，仅支持读取，不支持写入。用于倒计时，计数状态下每周期减一至 0。
state[1:0]	状态寄存器，不支持读写。用于记录当前计时器所处状态。

三、计时器主要行为

（1）计数行为

计数是计时器的最主要功能，根据不同模式会有不同的计时行为，但不同计时模式共用相似的流程，即从闲置状态开始，之后进入加载状态，再进入计数状态，计数结束后到达中断状态产生中断请求，之后回到闲置状态。

① 模式 0

- a. 闲置状态 (IDLE): $\text{ctrl}[0] = 1$ 进入加载状态, 中断请求恢复为 0; $\text{ctrl}[0] = 0$ 停留在闲置状态, 不进行其他操作。
- b. 加载状态 (LOAD): PRESET 寄存器数据存入 COUNT 寄存器, 进入计数状态。
- c. 计数状态 (CNT): $\text{ctrl}[0] = 1$ 允许计数, 若 COUNT 寄存器值大于一, 时钟上升沿该值减一, 状态停留于计数状态; 若 COUNT 寄存器值小于等于 1, 时钟上升沿 COUNT 寄存器值减为 0, 中断请求信号置 1, 进入中断状态; $\text{ctrl}[0] = 0$ 停止计数, 返回至闲置状态。
- d. 中断状态 (INT): $\text{ctrl}[0]$ 清零, 返回至闲置状态, 但此时中断请求保持为 1。

② 模式 1

- a. 闲置状态 (IDLE): $\text{ctrl}[0] = 1$ 进入加载状态, 中断请求恢复为 0; $\text{ctrl}[0] = 0$ 停留在闲置状态, 不进行其他操作。
- b. 加载状态 (LOAD): PRESET 寄存器数据存入 COUNT 寄存器, 进入计数状态。
- c. 计数状态 (CNT): $\text{ctrl}[0] = 1$ 允许计数, 若 COUNT 寄存器值大于一, 时钟上升沿该值减一, 状态停留于计数状态; 若 COUNT 寄存器值小于等于 1, 时钟上升沿 COUNT 寄存器值减为 0, 中断请求信号置 1, 进入中断状态; $\text{ctrl}[0] = 0$ 停止计数, 返回至闲置状态。
- d. 中断状态 (INT): 中断请求清零, 返回至闲置状态, 但此时 $\text{ctrl}[0]$ 保持为 1, 可重新计数。

(2) 存数行为

仅支持 sw 指令按字向 CTRL 和 PRESET 寄存器存储数据, 即该部分存储空间的前两个字, 要求 TimerWrite 为 1 时存入指定地址。其中 TimerWrite 信号经桥来自 CPU, 由存储指令和地址信息共同控制。

(3) 取数行为

仅支持 lw 指令按字读取三个内部寄存器的值, 通过桥接入 CPU 以存入 CPU 内部寄存器堆中的指定寄存器。

(4) 复位行为

受顶层 reset 信号控制，当 reset 信号为 1 时对内部寄存器清零，即 CTRL、PRESET、COUNT 清零，状态寄存器 state 清零，返回闲置状态。

(5) 产生中断信号行为

同时中断请求与 CTRL 寄存器中的屏蔽信号，输出总中断信号（屏蔽信号为 1 且中断请求 IRQ 为 1 时才能输出中断信号 1）。

四、用户操作与误操作

1. 各状态下用户操作

(1) 闲置状态：最优先支持 reset 操作；其次支持写入操作，当需要写入时写入但不进行状态转移；任何情况下支持读取操作。

(2) 加载状态：优先支持 reset 操作，全部寄存器清零，返回闲置状态；其次若有写入操作，优先执行写入操作，但不进行状态转移，PRESET 不存入 COUNT，计数不开始；任何情况下支持读取操作。

(3) 计数状态：优先支持 reset 操作，全部寄存器清零，返回闲置状态；其次若有写入操作，优先执行写入操作，但不进行状态转移，计数器不减，尤其可以通过写入 CTRL 控制计数停止；任何情况下支持读取操作。

(4) 中断状态：优先支持 reset 操作，全部寄存器清零，返回闲置状态；其次若有写入操作，优先执行写入操作，但不进行状态转移；任何情况下支持读取操作。

2. 误操作

(1) lh、lhu、lb、lbu 读取计时器内部寄存器：理论上产生整字输出，但实际由 CPU 内部进行异常处理跳转，CPU 忽略此时 Timer 的输出。

(2) load 指令对齐错误：理论上产生整字输出，但实际由 CPU 内部进行异常处理跳转，CPU 忽略此时 Timer 的输出。

(3) sh、sb 向 Timer 内部寄存器存储数据：TimerWrite 信号为 0，不存入数据，又 CPU 进行异常处理跳转。

(4) sw 向 Timer 内部 COUNT 寄存器存储数据：TimerWrite 信号为 0，不存入数据，由 CPU 进行异常处理跳转。