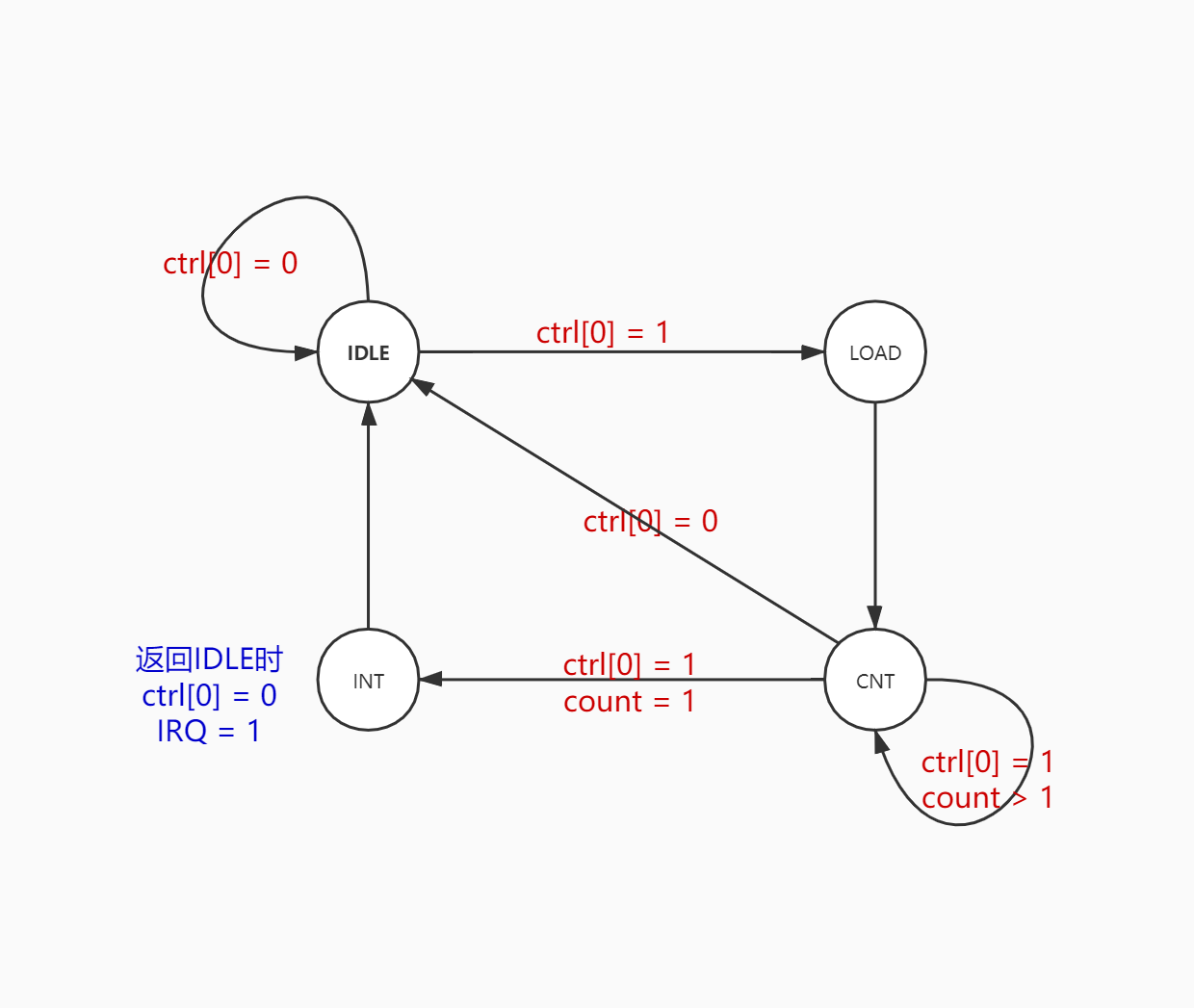
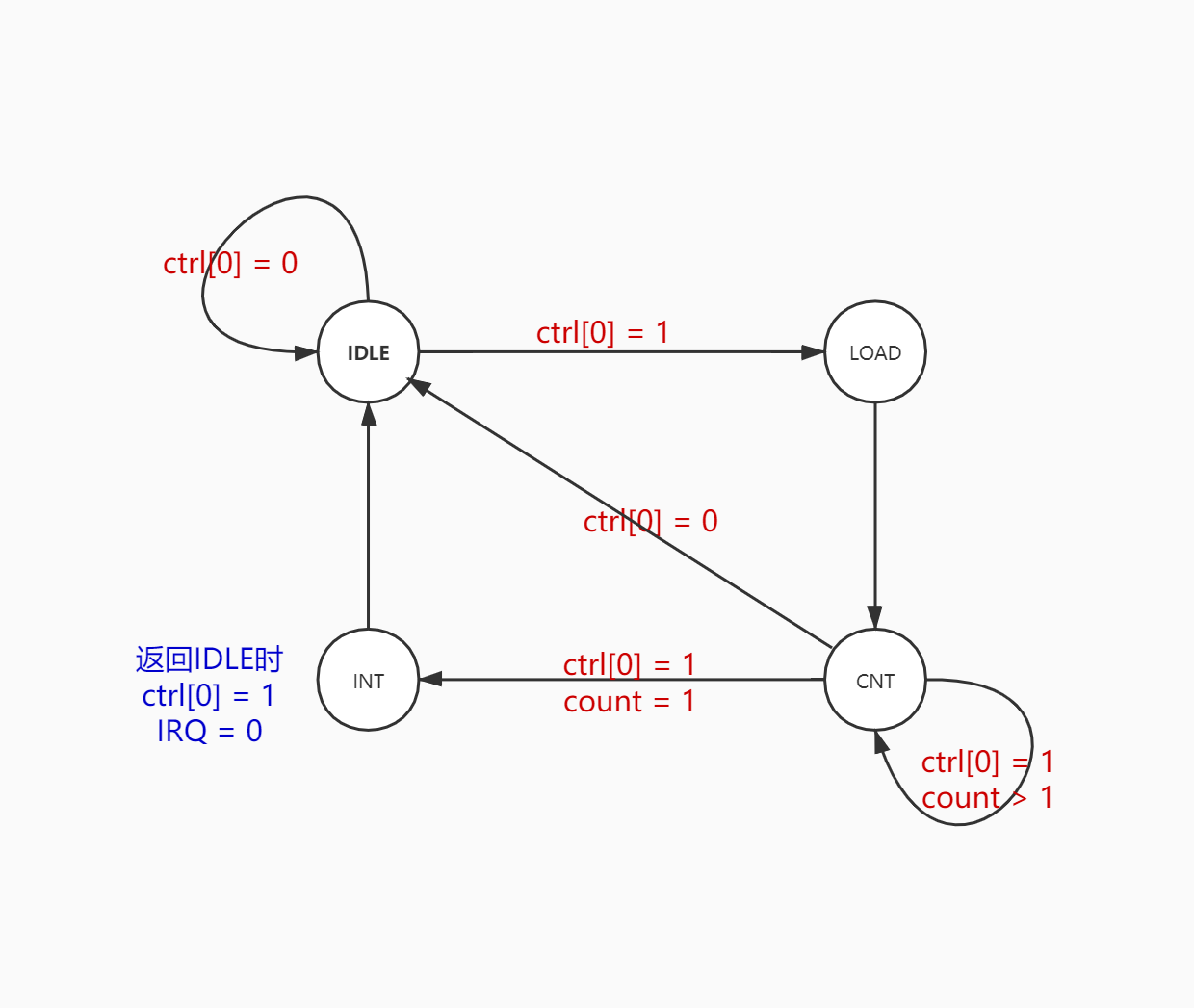
COCO定时器使用说明

1. 状态转移图
2. 模式0下定时器状态转移图



1. 模式1下定时器状态转移图



1. 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。 |

1. 计时器主要行为
2. 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**
2. 闲置状态（IDLE）：ctrl[0] = 1进入加载状态，中断请求恢复为0；ctrl[0] = 0停留在闲置状态，不进行其他操作。
3. 加载状态（LOAD）：PRESET寄存器数据存入COUNT寄存器，进入计数状态。
4. 计数状态（CNT）：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；ctrl[0] = 0停止计数，返回至闲置状态。
5. 中断状态（INT）：ctrl[0]清零，返回至闲置状态，但此时中断请求保持为1。
6. **模式1**
7. 闲置状态（IDLE）：ctrl[0] = 1进入加载状态，中断请求恢复为0；ctrl[0] = 0停留在闲置状态，不进行其他操作。
8. 加载状态（LOAD）：PRESET寄存器数据存入COUNT寄存器，进入计数状态。
9. 计数状态（CNT）：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；ctrl[0] = 0停止计数，返回至闲置状态。
10. 中断状态（INT）：中断请求清零，返回至闲置状态，但此时ctrl[0]保持为1，可重新计数。
11. 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。

1. 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。

1. 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。

1. 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
2. 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。
3. 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。
4. 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。
5. 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。
6. 误操作
7. 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。
8. 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。
9. 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。
10. 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。