

第一章WD 模块

11.1 概述

系统在工作时，由于程序编制不当、受到外界干扰等一些原因会导致程序跑飞，进而陷入死循环状态，程序的正常运行被打断，系统无法继续正常工作。这时，看门狗的引入可以起到实时监测的作用，防止程序的跑飞。

看门狗，又叫 watchdog，本质上是一个计数电路，一般有一个寄存器用来喂狗(kicking the dog or service the dog)，一个寄存器用来写入初值，一个输出到PMU，处理器正常工作的时候，每隔一段时间就写一个值到喂狗的寄存器内，这样可以使得计数器回到初始值；如果超过规定的时间不喂狗，(一般在程序跑飞时)，看门狗计数超时，就会给出一个复位信号，使程序复位。看门狗的作用就是防止程序发生死循环，或者说程序跑飞。

11.2 功能特性

11.2.1 单次超时复位模式

该模式下，每次看门狗由于程序不及时喂狗，导致计数超时后，即在复位输出端发出一个复位信号。

11.2.2 一次超时中断，二次超时复位

该模式下，每次看门狗由于程序不及时喂狗，导致计数第一次超时后，即在中断输出端给出中断信号（拉高），直到处理器接收到中断信号做出处理行为或者发生可消除中断的情况，则中断输出端信号复位（拉低）；否则，第二次计数超时后，复位输出端将发出一个复位信号，同时，中断输出端复位（拉低）。

11.2.3 接口信号

- Sysenable 来自于 top.v 中，一直为 1
- Time_out---->watchdog_reset 给 PMU，则就会有 p_reset
- Intout 中断

11.3 编程接口

11.3.1 寄存器列表

模块的基址为：40038000

寄存器名称	偏移地址	复位值	描述
reg_StartValue	0x00	0xFFFFFFFF	设置看门狗计数初始值。
reg_feeddog	0x08	—	往该寄存器写入数值起到喂狗作用。
reg_mode	0x10	0x0	选择看门狗的工作模式。

11.3.2 寄存器描述

11.3.2.1 数据寄存器

数据位	名称	类型	复位值	描述
31:0	StartValue	R/W	0xFF	始值设置位： 往寄存器里面写入 32 位的一个值， 该值代表了看门狗计数初始值。

每次往该寄存器里面写入一个数值后，计数的模都会实时地发生变化。

11.3.2.2 喂狗信号寄存器

数 据 位	名称	类型	复位值	描述
31:0	RESERVED	WO	-	往寄存器里面写入任意一个值，则 flag 置 1， 表示处理器有进行喂狗操作。重置计数值。

只有往喂狗寄存器里面写入数据，喂狗信号 flag 才有效，此时，计数器里面的计数值才会复位至计数初始值，否则，计数值一直递减，直到计数超时，并复位至计数初始值。

11.3.2.3 模式选择寄存器

数据位	名称	类型	复位值	描述
31:2	RESERVED	-	-	-
1	ENABLE	RW	0x1	看门狗工作使能位： 1：看门狗工作使能，看门狗工作；

				0: 看门不使能，看门狗不工作；
0	MODE	RW	0x0	看门狗工作模式选择位： 0: 单次超时复位模式； 1: 一次超时中断，二次超时复位模式

11.3.3 编程实例

```
int main()
{
    int i,j;
    int_enable(24);
    reg_StartValue=10000000;
    //RTC_INT_EN=0x22;
    WD_mode = 0x0;
    for(i=0;i<=2;i++)
    {
        if(GPIO_PORTA_DAT & (1<<20))
        {
            GPIO_PORTA_SEL|=(1<<20);
            GPIO_PORTA_DAT&=~(1<<20);
            GPIO_PORTA_DIR&=~(1<<20);
        }
        else
        {
            GPIO_PORTA_SEL|=(1<<20);
            GPIO_PORTA_DAT|=(1<<20);
            GPIO_PORTA_DIR&=~(1<<20);
        }
        for(j=0;j<1000000;j++);
    }
    WD_mode = 0x2;
    for(;;)
    {
        reg_feeddog=0x1;
    }
    while(1);
    for(;;);
}
```

