**Progetto di reti Logiche**

Windowed WatchdDog Counter Programmable Multi-Stage

Mariani Michele

Bianchi Matteo

# Introduzione

Un Windowed WatchDog è un componente elettronico utilizzato per rilevare malfunzionamenti software o hardware e in grado quindi di resettare il sistema. È utilizzato in sistemi embedded o in sistemi remoti automatizzati non facilmente raggiungibili e riavviabili direttamente dall’uomo.

Questo componente può essere interessante anche in varie applicazioni che richiedono livelli più elevati di supervisione di microprocessori a basso costo. I circuiti automobilistici, ad esempio, spesso implementano una sicurezza aggiuntiva con microprocessori ridondanti che si controllano a vicenda. Nel controllo della stabilità del veicolo, ad esempio, gli input dei sensori di imbardata e accelerazione vengono elaborati per determinare come il sistema dovrebbe intervenire per applicare i freni in una situazione critica per la quale il conducente non ha controllo.

Nella fattispecie, un Windowed WatchDog Programmable Multi-Stage è un WatchDog avente tre livelli differenti di tempi di trigger:

-TNMI è una soglia di trigger programmabile che se raggiunta genera un segnale di NMI e nel sistema senza bisogno del segnale di Clear.

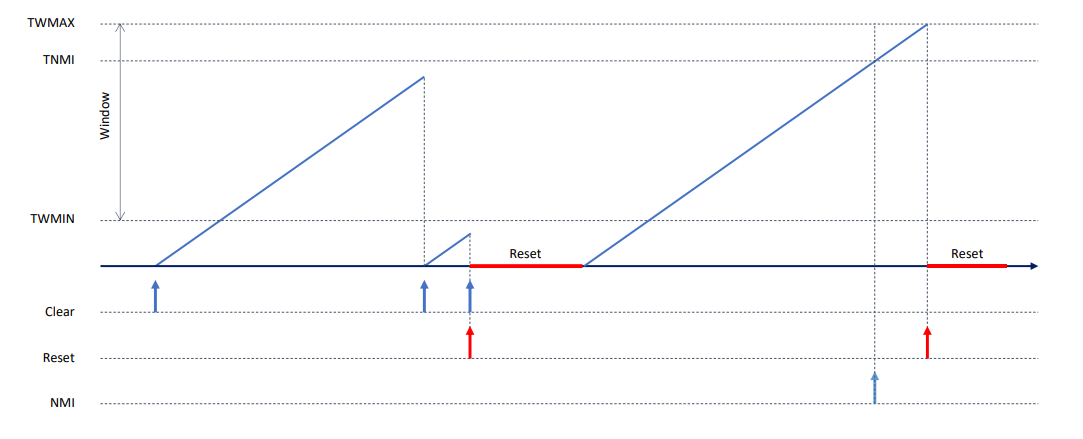
-TMin e TMax sono gli estremi della finestra del counter e stabiliscono la finestra entro il quale il counter può lavorare. Se un segnale di Clear viene attivato al di fuori di questa finestra il watchdog genera un segnale di reset che resetta il sistema e il counter interno.

I contatori necessitano di un componente di tipo Prescaler, in grado di suddividere la frequenza del clock programmando l’ampiezza della finestra di lavoro del watchdog.

Le assunzioni si basano sulla grandezza dei segnali di ingresso:

-Il Prescaler può dividere la frequenza di clock da un minimo di 1 fino a 128 volte.

- I valori di TMin, TMax e TNMI programmabili sono rappresentabili su 16 bit.



# Specifica

Il Windowed watchdog counter è composto da due watchdog principali:

-Un WatchDogNMI che si occupa di generare il segnale NMI nel sistema quando viene raggiunta la soglia TNMI

-Un WatchDogRST che si occupa invece di generare il segnale di reset se il segnale in ingresso di clear è fuori dalla finestra di TMin e TMax o se viene raggiunta la soglia di TMax senza segnale di clear.

-Un CommandSelect che si occupa di programmare i vari componenti attraverso coppie comando e dati.

Ogni watchdog è composto da:

-Un Prescaler programmabile che divide la frequenza del clock in ingresso di un fattore pari a 2^K, con K = [0…7].

-Un counter in grado di contare su parole di lunghezza di 16 bit.

Il Counter è costruito a partire da:

- Incrementer: un circuito in grado di generare in uscita il valore in ingresso incrementato di uno, dato in ingresso un valore di 16 bit.

-Un registro: in grado di salvare il valore del counter per un colpo di clock.

Il Prescaler incrementa un contatore di 8 bit, ha in ingresso un valore k che identifica di quanto il clock viene diviso. In uscita produce un segnale di clock enable (LOW\_CLK) che rende il clock 2^k volte più lento rispetto al clock originario.

## Interfaccia del sistema

Il sistema è composto dai seguenti segnali:

INPUT:

-RST: segnale di Reset per eliminare tutti i valori salvati nel watchdog e resettare i contatori (1 bit)

-COMMAND: utilizzato per settare i valori di prescaling e di soglia (3 bit)

-DATA: identifica i valori di prescaling e di soglia (16 bit)

-CLEAR: utilizzato per resettare il valore del counter del WatchDog (1 bit)

-CLK: segnale di clock dato dal sistema (1 bit)

OUTPUT:

-RESET: segnale di errore che resetta il sistema andato in fault (1 bit)

-NMI: segnale di allarme lanciato dal sistema (1 bit)

Per utilizzare il sistema bisogna:

1. Resettare il sistema attraverso il segnale di RST per un ciclo di clock
2. Porre in ingresso i valori COMMAND e DATA in sequenza per un ciclo di clock per tutti i dati presenti nel WatchDog.
3. Presentare il comando di START per un ciclo di clock per far partire il circuito
4. Il sistema inizierà a contare
5. Potrà essere attivato in ogni momento il segnale di CLEAR con durata un ciclo di clock

Una volta avviato non è più possibile resettare né interrompere il Watchdog.

|  |  |
| --- | --- |
| COMMAND | FUNZIONE |
| 000 | START |
| 001 | Prescaler\_RST |
| 010 | Prescaler\_NMI |
| 011 | TM1 |
| 100 | TM2 |
| 101 | TNMI |

## Architettura del sistema

Il Windowed WatchDog counter programmable multi-stage è formato da:

-Un CommandSelect 🡪 per gestire i dati in ingresso bus

-Due Prescaler 🡪 abbassano la frequenza del clock del sistema per i due WatchDog

-Un WatchDogRST 🡪 per contare la finestra di lavoro del sistema

-Un WatchDogNMI 🡪 per segnalare la soglia di guardia

Il segnale di CLEAR (CLS) agisce solamente sui WatchDog resettando il contatore interno e generando (se necessario) il segnale di RESET o di NMI.

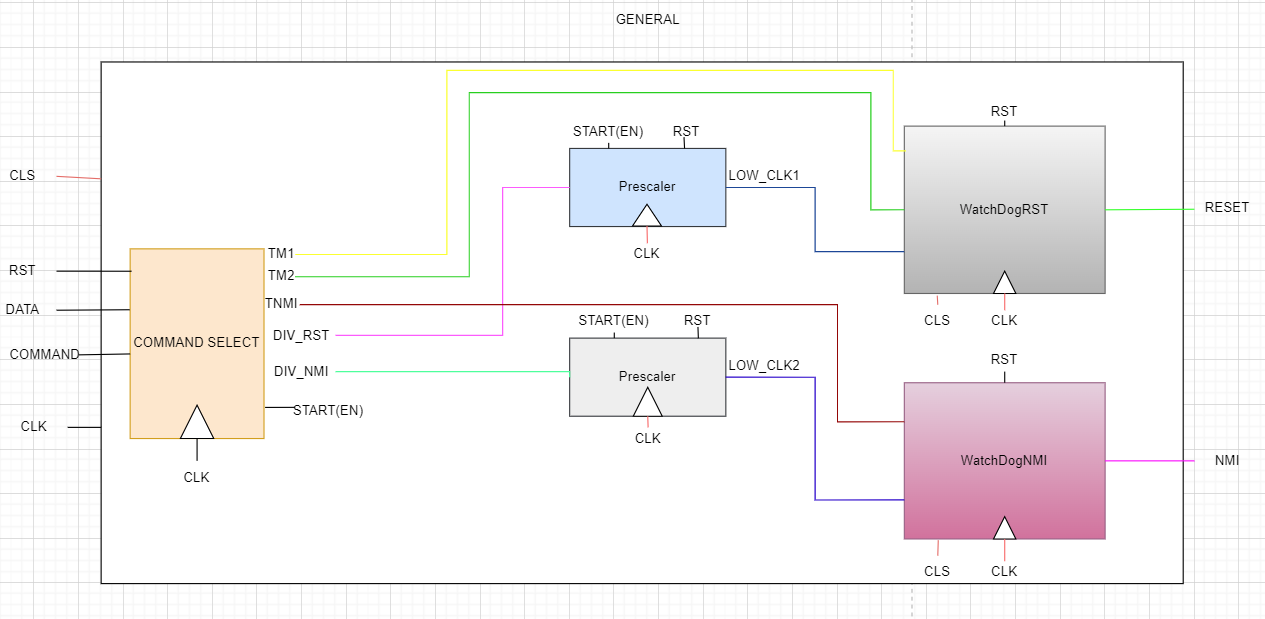
Il segnale di Reset (RST) è utilizzato per resettare il sistema e porre tutti i segnali, i contatori e i valori salvati a zero. Dopo la fase di Reset deve sempre seguire una fase di caricamento dei parametri nel sistema.

Alcuni dei moduli in uso vengono riutilizzati:

-Il Counter, essendo un componente generico, viene istanziato più volte con dimensioni diverse.

-I FlipFlop Register Parallel vengono riutilizzati più volte nel Command Select e nei Counter per salvare il dato appena incrementato, oltre che nel WatchDogRst e WatchDogNmi per evitare glitch.

-Incrementer viene utilizzato con grandezza variabile da tutti i Counter



## Command Select

Il componente è utilizzato per prelevare i dati dal bus in ingresso e settare WatchDog e i Prescaler con i valori di soglia TM1, TM2, TNMI e di prescaling DIV\_RST e DIV\_NMI.

INPUT:

-RST std\_logic 1 bit

-DATA std\_logic\_vector 16 bit

-COMMAND std\_logic\_vector 3 bit

-CLK std\_logic 1 bit

OUTPUT:

-TM1 std\_logic\_vector 16 bit

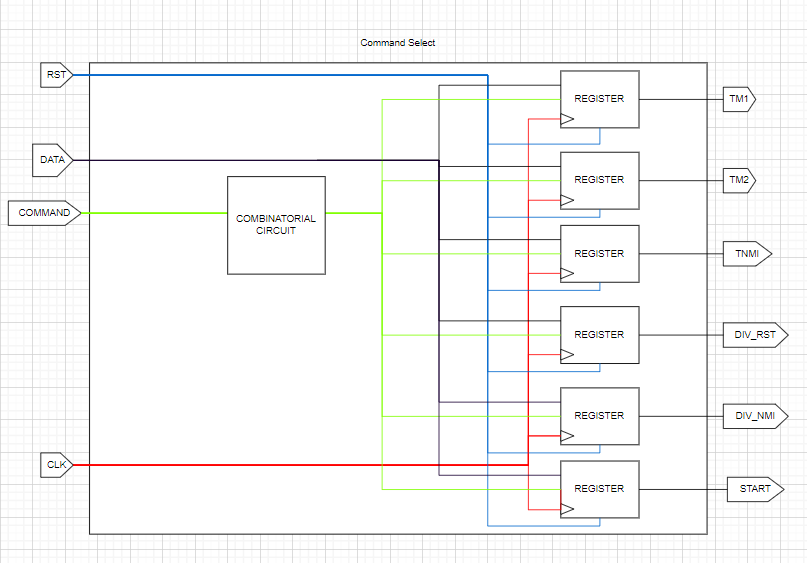
-TM2 std\_logic\_vector 16 bit

-TNMI std\_logic\_vector 16 bit

-DIV\_RST std\_logic\_vector 3 bit

-DIV\_NMI std\_logic\_vector 3 bit

-START std\_logic 1 bit



Command entra in un circuito combinatorio (“selettore”) e funziona da clk enable per i registri.

Abbiamo un segnale CS\_EN in uscita dal circuito combinatorio per ogni registro che memorizza le soglie.

Ogni comando per il segnale COMMAND è associato ad un parametro da settare attraverso il segnale DATA.

Quando viene presentato il comando ‘000’ viene asserito il segnale di START per far partire il WatchDog.

Se il segnale di RST viene attivato, vengono eliminati tutti i parametri salvati e il segnale di START viene abbassato.

Tutti i segnali di ingresso (DATA, COMMAND, RST e CLEAR) devono essere mantenuti per un ciclo di clock in modo da essere correttamente campionati.

Prescaler

Riduce la frequenza del clock in ingresso da alta a bassa frequenza

INPUT:

-K std\_logic\_vector a 3 bit (DIV1/DIV2)

-EN std\_logic 1 bit

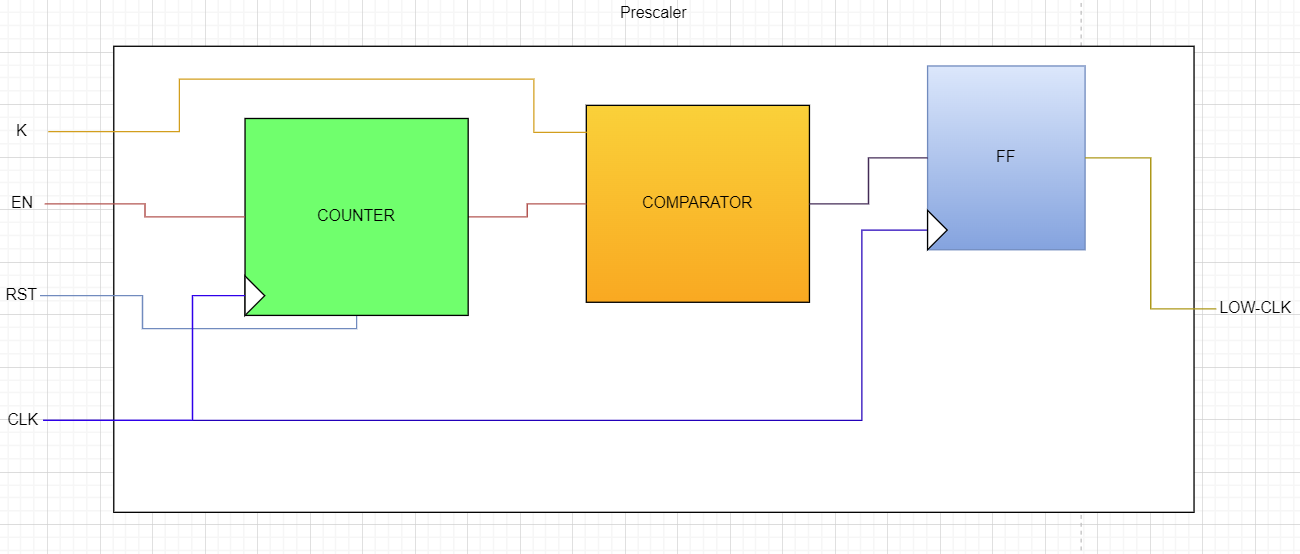
-RST std\_logic 1 bit

-CLK std\_logic ad 1 bit

OUTPUT:

-LOW\_CLK std\_logic ad 1 bit

In base al valore di prescaling K viene generato un segnale di LOW\_CLK per abbassare la frequenza di clock.  
Il segnale di EN è utilizzato invece per mostrare al counter il CLK e cominciare così a contare.



WatchDog:

1)WatchDogRST

Definisce la dimensione della finestra attraverso i segnali di TMin e TMax

INPUT:

-START std\_logic 1 bit

-TM1 std\_logic\_vector 16 bit

-TM2 std\_logic\_vector 16 bit

-RST std\_logic 1 bit

-CLEAR std\_logic 1 bit

-LOW\_CLK std\_logic 1 bit

-CLK std\_logic 1 bit

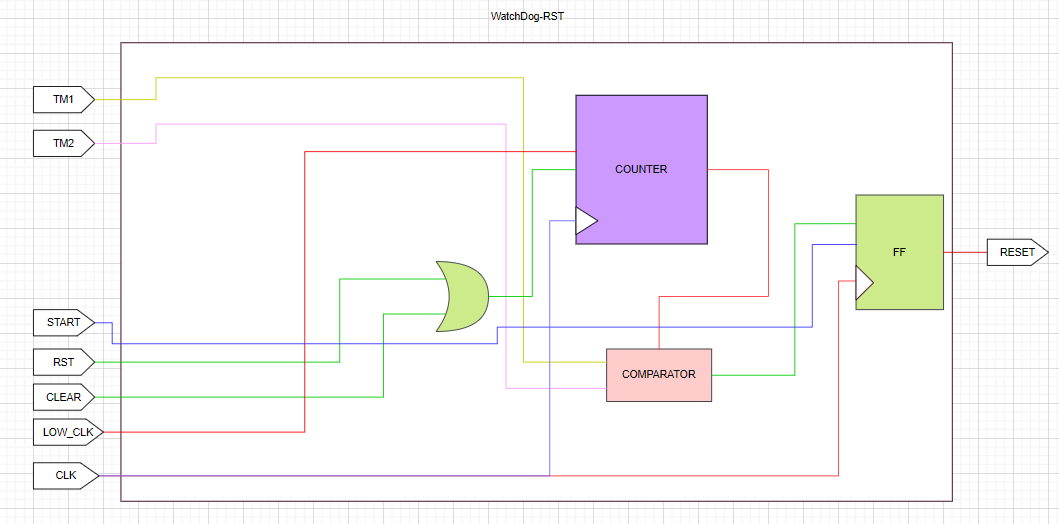
OUTPUT:

-RESET std\_logic 1 bit

Il WatchDogRst istanzia un Counter a 16 bit e crea la finestra di lavoro. Il segnale in ingresso di CLEAR resetta solamente il contatore lasciando inalterata la finestra di lavoro. Se CLEAR venisse attivato mentre il counter si trova al di fuori della finestra di lavoro, il WatchDog genererà in uscita un segnale di RESET.  
Il segnale RST è utilizzato per resettare il valore della finestra di lavoro del WatchDog.

Il segnale START(EN) permette al Counter di vedere il CLK e cominciare così a contare.   
Il segnale LOW\_CLK è utilizzato per prelevare il clock abbassato proveniente dal Prescaler.

Il registro in uscita serve per sincronizzare l’uscita e non avere glitch.



2)WatchDogNMI

Definisce la soglia di NMI attraverso il segnale di TNMI

INPUT:

-TNMI std\_logic\_vector 16 bit

-START std\_logic 1 bit

-RST std\_logic 1 bit

-CLEAR std\_logic 1 bit

-LOW\_CLK std\_logic 1 bit

-CLK std\_logic 1 bit

OUTPUT:

-NMI std\_logic 1 bit

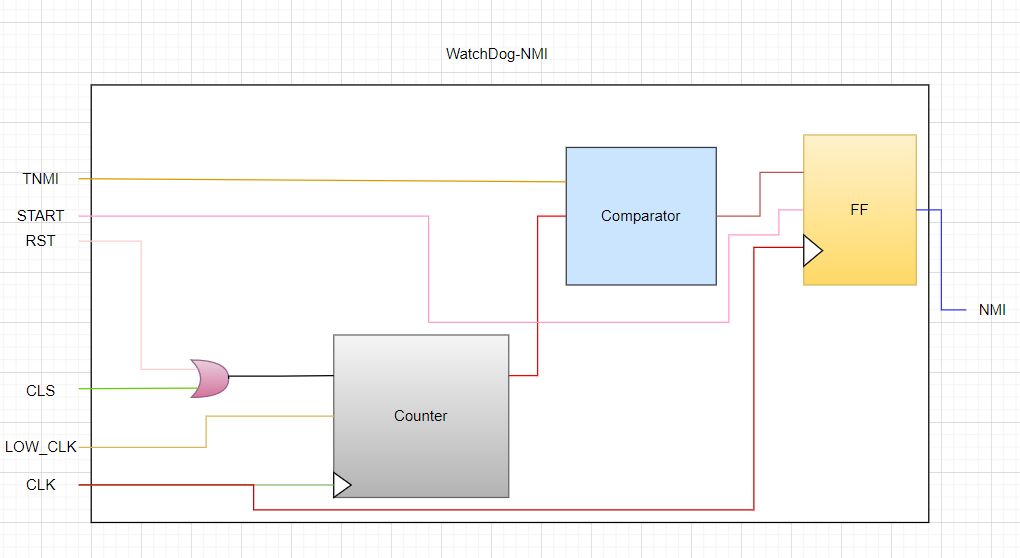
Il WatchDogNMI istanzia un Counter a 16 bit e crea la soglia di NMI. Il segnale di ingresso Clear (CLEAR) resetta solamente il contatore lasciando inalterata la soglia di NMI.

Se il contatore raggiunge quindi la soglia di NMI prima che venga attivato il segnale di Clear allora il WatchDogNMI genererà un segnale di NMI.

Il segnale RST è utilizzato per resettare il valore della soglia di NMI e farla ripartire.

Il segnale LOW\_CLK è utilizzato per prelevare solo il segnale di CLK quando LOW\_CLK è alto.

Il registro in uscita serve per sincronizzare l’uscita e non avere glitch.



# Verifica

Verrà testato ogni singolo modulo che forma il Windowed WatchDog Programmable Multi-Stage in modo indipendente dagli altri.

## Test-bench

La verifica del test bench avviene attraverso una combinazione di stimoli dei segnali di ingresso dell’architettura: COMMAND, DATA, CLEAR e RESET.

## Casi d’uso

I test coprono ogni singolo componente in modo da verificare il loro corretto funzionamento all’interno dell’architettura, per quanto riguarda il modulo principale:

-Selezione dei valori di soglia TM1, TM2 e TNMI e dei valori di Prescaling e attivazione di START.

-Attivazione del segnale di NMI da parte dell’architettura quando viene superata la soglia di TNMI.

-Stimolazione dell’architettura da parte del segnale di Clear (CLEAR) il quale resetta i valori del counter interni ai due WatchDog all’interno della finestra di lavoro e quindi senza la generazione del segnale di RESET

-Stimolazione dell’architettura attraverso il segnale di Clear (CLEAR) quando si è al di fuori della zona di lavoro e/o al di sotto della soglia minima (TM1) generando così il segnale di RESET da parte dell’architettura

-Attivazione del segnale di RESET da parte dell’architettura quando viene superata la soglia massima della finestra di lavoro (TM2)