Progetto di reti Logiche

Programmable Windowed Watchdog Counter

Marco Filippini

Sommario

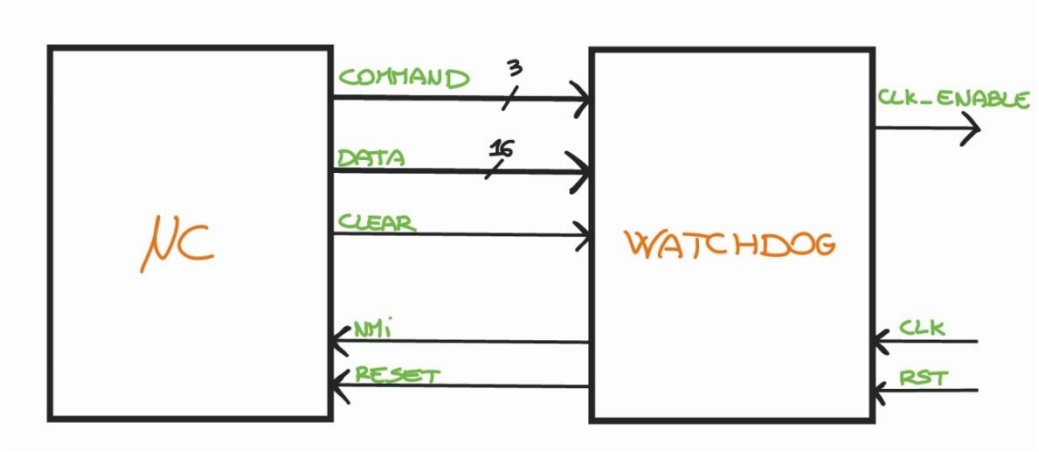
Introduzione

Il componente è un watchdog counter a 16 bit. L'aggettivo "windowed" che lo descrive indica che è realizzato in modo tale da segnalare, tramite un segnale in uscita, il funzionamento non corretto del microprocessore che ne fa utilizzo. Si tratta inoltre di un dispositivo programmabile: ciò significa che, prima di avviare il conteggio, può (e deve) essere configurato dal microprocessore stesso per impostare determinati valori minimo e massimo della finestra di conteggio. Inoltre presenta una funzionalità di pre-allarme. È infatti possibile configurare un terzo parametro che rappresenta una soglia di pre-allarme nei confronti del microprocessore: se il conteggio raggiunge la soglia rappresentata da tale parametro, significa che si è prossimi al limite superiore di conteggio stabilito dalla finestra. Infine, il dispositivo è dotato di un prescaler: esso serve per aumentare la dimensione della finestra di conteggio oltre i limiti imposti dai 16 bit del dispositivo e dalla frequenza del segnale di clock che lo alimenta.

Specifica

L'obiettivo imposto è quello di separare la logica di gestione della configurazione dei parametri del watchdog dal conteggio, che rappresenta il cuore del dispositivo, e ancora dalla gestione dell'evoluzione dei segnali di uscita.

Interfaccia del sistema



Segnali di ingresso

- *COMMAND[]* → segnale a 3 bit usato per specificare il tipo di comando che il microprocessore invia al watchdog in base alla seguente codifica:
  - "000" → SET PRESCALER;
  - "001" → SET TWMIN (soglia inferiore della finestra di conteggio);
  - "010" → SET TWMAX (soglia superiore della finestra di conteggio);
  - "011" → SET TNMI (soglia di pre-allarme);
  - "100" → START.

  N.B.: le combinazioni "101", "110" e "111" vengono ignorate dal watchdog e non hanno alcun effetto sullo stesso.

- *DATA[]* → segnale a 16 bit attraverso cui il watchdog riceve un valore da impostare per il parametro specificato dal tipo di comando. Se si riceve il comando START qualsiasi valore del

segnale DATA viene ignorato e il watchdog diventa insensibile ad ogni altro ulteriore comando finché non viene effettuato un reset da parte del microprocessore.

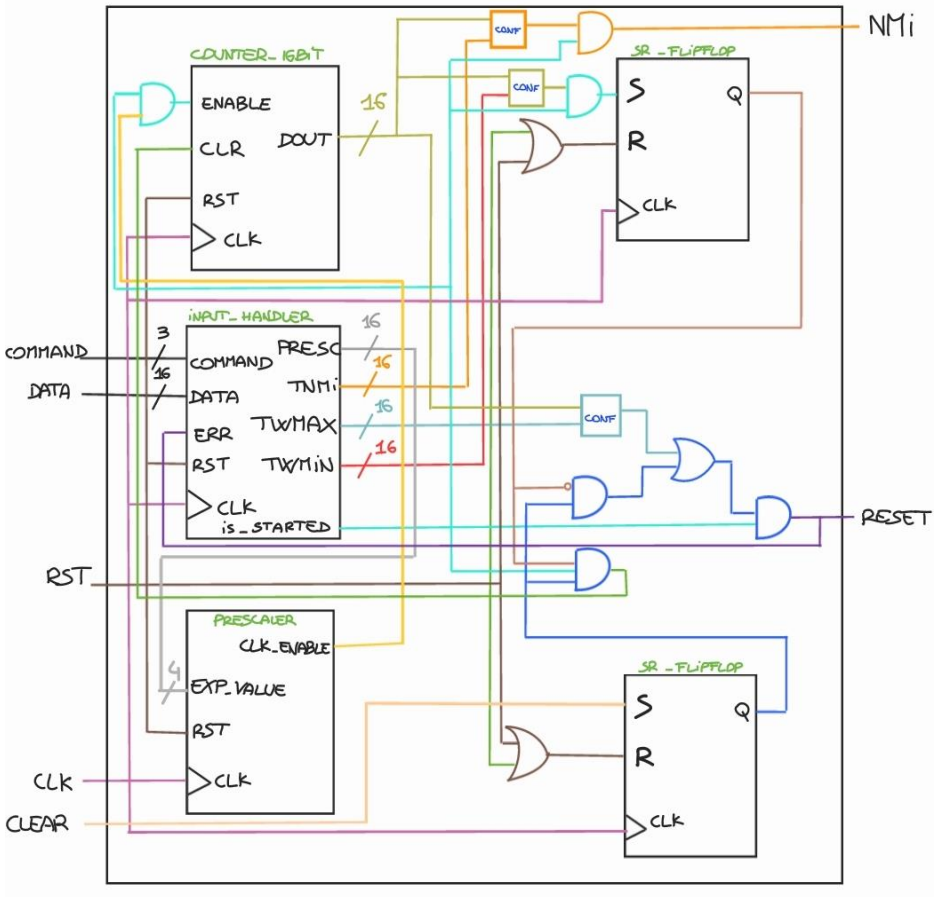
- *CLK* → segnale a 1 bit. Rappresenta il clock con cui il dispositivo è alimentato. Tramite prescaler e l'opportuna impostazione di un parametro, il watchdog è in grado di funzionare alla frequenza di clock di tale segnale oppure quest'ultimo può essere diviso per un multiplo di 2 (da $2^1 = 2$ a $2^{10} = 1024$).
- *RST* → segnale a 1 bit. Viene usato per portare il watchdog in uno stato di funzionamento noto all'inizio di una simulazione;
- *CLEAR* → segnale a 1 bit. Quando assume valore 1(uno), il conteggio riparte da zero.

Segnali di uscita

- *NMI* → segnale di 1 bit normalmente posto a 0 (zero). Tale segnale assume valore 1 (uno) per un ciclo di clock nel momento in cui il conteggio assume valore pari al parametro di pre-allarme *TNMI*.
- *RESET* → segnale di 1 bit normalmente posto a 0 (zero). Assume valore 1 (uno) se il conteggio raggiunge il limite superiore della finestra, espresso dal parametro *TWMAX*, oppure se il microprocessore invia al watchdog un segnale di *CLEAR* prima che il conteggio abbia raggiunto il limite inferiore della finestra, espresso dal parametro *TWMIN*;
- *CLK_ENABLE* -> segnale di abilitazione ad operare per le componenti interne del watchdog. Viene generato dal prescaler in base alla frequenza di clock selezionata.

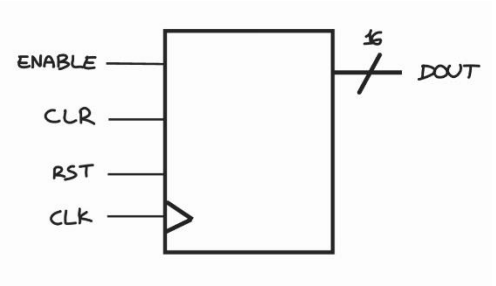
Architettura del sistema

Il watchdog counter è costituito da quattro moduli principali: *COUNTER_16BIT, INPUT_HANDLER, PRESCALER, SR_FLIPFLOP*. Sono inoltre presenti ulteriori porte logiche atte a gestire correttamente i valori assunti dalle uscite del dispositivo in base ai valori generati in uscita da ciascun modulo. La struttura interna generale è riportata nella figura a fianco:



Counter_16bit

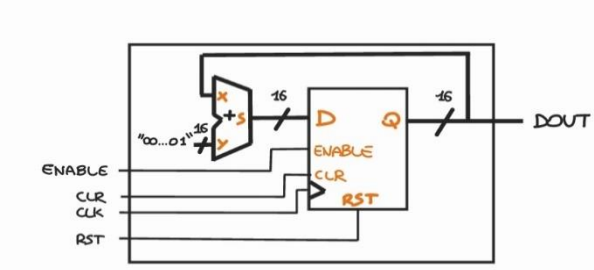
Si tratta di un contatore modulo 2^{16}: il conteggio può assumere quindi tutti i valori tra 0 e 65535 inclusi.

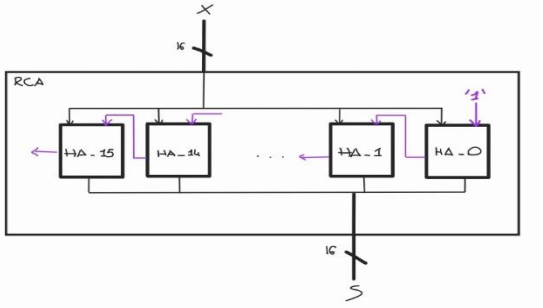


I segnali (in ingresso e uscita) rappresentano:

- *DOUT* → valore attuale del conteggio (16 bit);
- *ENABLE* → bit che, quando posto a 1, abilita il dispositivo al conteggio;
- *RST* → bit che, se posto a 1, riporta il contatore allo stato iniziale, espresso dal valore DOUT = "0000000000000000";
- *CLK* → clock di alimentazione del dispositivo;
- *CLR* → segnale di un bit che, quando posto ad 1, procede a far ripartire da 0 (zero) il contatore.

Nello specifico, il contatore è realizzato tramite il corretto collegamento dei moduli *D_FLIPFLOP_WITH_ENA* e *RCA_16BIT*. Il primo differisce da un classico flipflop di tipo D per la sola presenza del segnale d'ingresso *ENABLE* che, quando posto ad 1, abilita il dispositivo al campionamento dell'ingresso sui fronti di salita del segnale di clock e, contrariamente, rimane

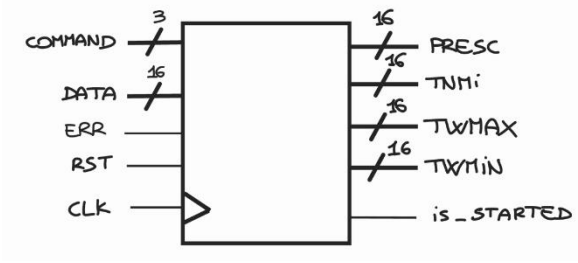




insensibile a qualsiasi cambiamento (anche del clock stesso) se posto a 0. Il secondo è un Ripple Carry Adder costituito da 16 half-adder in cascata. Per quest'ultimo, non è necessaria l'adozione di full-adder in quanto il secondo addendo (*Y*), che rimane fisso, ha costantemente i primi 15 di 16 bit posti a 0 il che consente, dal secondo passo della somma in poi, di sommare all'i-esimo bit del primo addendo (*X*) il riporto generato dalla somma al passo precedente. In altre parole, per ognuno dei 16 bit, l'operazione si riduce alla somma di due soli bit.

Input_Handler

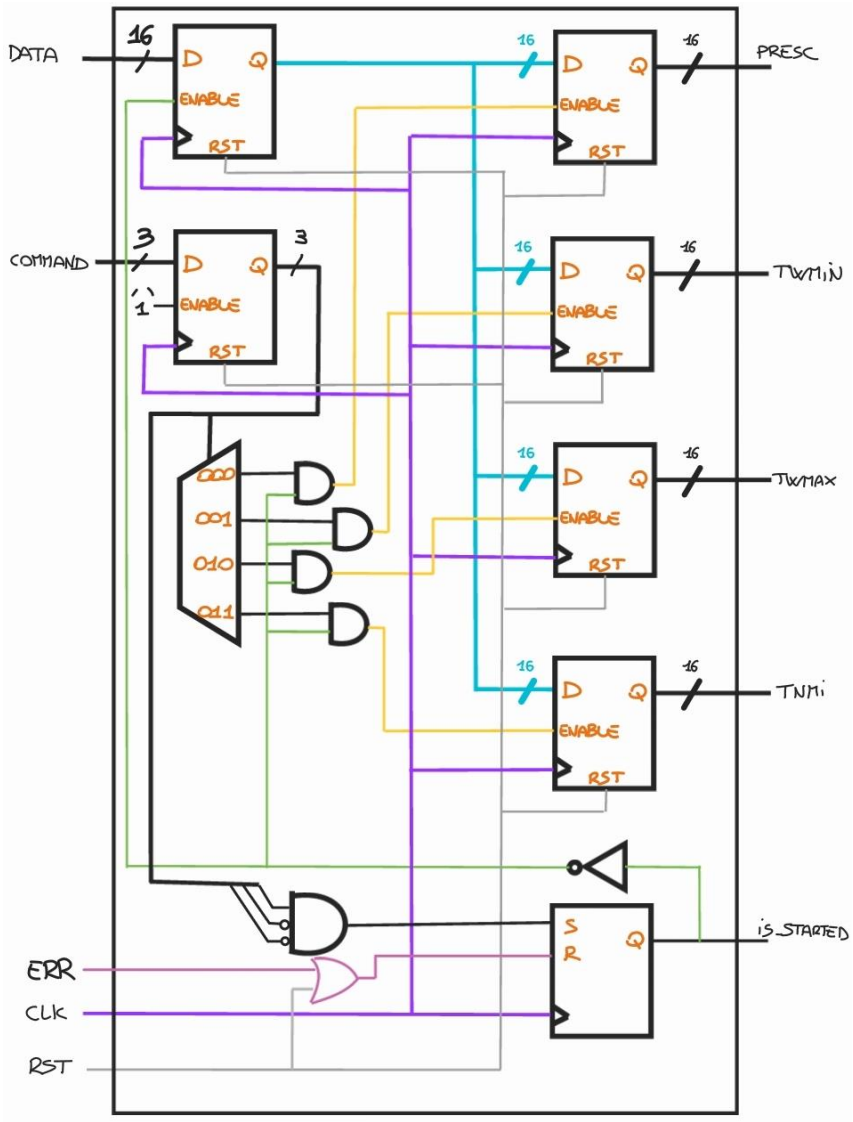
Questo modulo ha il compito di gestire i comandi e i dati in ingresso che il watchdog riceve dal microprocessore nonché di salvare tali dati in opportuni registri in modo tale che durante il conteggio siano sempre disponibili. La sua interfaccia presenta i seguenti segnali (in ingresso e in uscita):



- *COMMAND* → segnale a 3 bit direttamente collegato all'ingresso *COMMAND del watchdog;*
- *DATA* → segnale a 16 bit direttamente collegato all'ingresso *DATA* del watchdog;
- *RST* → segnale di 1 un bit usato per portare a 0 (zero) tutti i bit dei segnali di uscita del modulo;
- *CLK* → clock di alimentazione del dispositivo;
- *ERR* → segnale di 1 bit che, quando posto ad 1, fa sì che il watchdog torni ad essere sensibile ai comandi di ingresso e che venga posta a 0 (zero) l'uscita *IS_STARTED* in modo che il conteggio si interrompa;

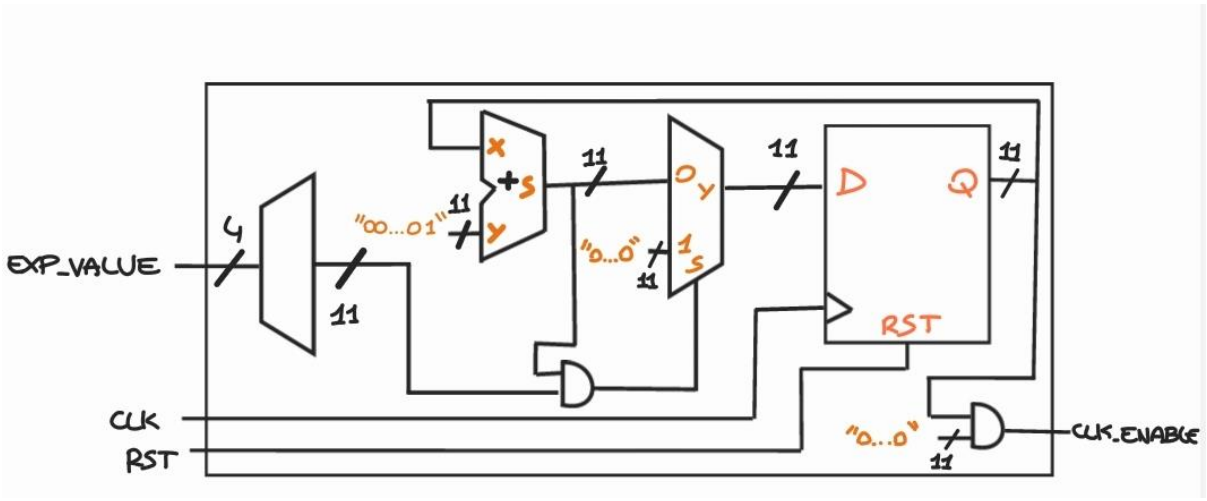
- *PRESC* → esprime il valore precedentemente impostato per il prescaler. Più precisamente, esso esprime l'esponente della potenza di 2 scelta per la divisione del clock. Il valore è espresso su 16 bit ma quelli utili sono i quattro meno significativi;
- *TNMI* → segnale a 16 bit che rappresenta il valore della soglia di pre-allarme;
- *TWMAX* → segnale a 16 bit che esprime il limite superiore della finestra di conteggio, raggiunto il quale viene rilevata una condizione di errore nell'esecuzione fatta dal microprocessore;
- *TWMIN* -> segnale a 16 bit che esprime il limite inferiore della finestra di conteggio. Se il watchdog riceve il segnale di *CLEAR* dal microprocessore prima che il conteggio abbia raggiunto il valore espresso da questo parametro, viene rilevata una condizione di errore nell'esecuzione del microprocessore stesso;
- *IS-STARTED* → segnale di un bit che viene posto ad 1 (uno) nel momento in cui il watchdog riceve il comando di start.

Per ognuno dei segnali in uscita, internamente al modulo, è presente un registro che ne mantiene il valore: per i primi quattro (*PRRESC, TNMI, TWMAX, TWMIN)* vengono utilizzati altrettanti registri di tipo di D (rappresentati dai moduli *D_FLIFLOP_WITH_ENA*) mentre per il segnale *IS_STARTED* è previsto un registro di tipo SR (rappresentato dal modulo *SR_FLIPFLOP*). Ognuno di essi, fatta eccezione per il flipflop SR, oltre alle caratteristiche di un normale registro, prevede un ingresso classificato come *ENABLE* che ne abilita la lettura dell'ingresso sui fronti di salita del segnale di clock. Inoltre, il modulo *INPUT_*HANDLER, presenta al suo interno un decoder (modulo *DECODER_3BIT* controllato dall'ingresso *COMMAND*: esso ha il compito di abilitare la modifica di un solo registro. Riguardo al registro di tipo SR, la sua uscita viene posta a 1 (uno) quando in ingresso si riceve *COMMAND* = "100", che corrisponde al comando di *START*. Questo viene usato per impedire che il successivo inserimento di un comando differente abiliti nuovamente la modifica di un altro registro. Lo schema interno del modulo è riportato nella figura accanto.



Prescaler

Il prescaler è un modulo il cui compito è selezionare la giusta frequenza del segnale di clock per il modulo *COUNTER_16BIT.* La scelta è tra il clock stesso o una sua variazione ottenuta dividendone la frequenza per un multiplo di 2 (da 2^1=2 a 2^{10} = 1024). Questo serve per poter aumentare il range di conteggio e superare i limiti imposti dalla frequenza del clock "base" e dal numero di bit del *COUNTER_16BIT*: poiché quest'ultimo gestisce un conteggio espresso su 16 bit, il limite superiore raggiungibile dal dispositivo è $2^{16} - 1 = 65535$ e, visto che l'incremento del contatore è di 1 per ogni fronte di salita del segnale di clock, ciò corrisponde ad altrettanti cicli di clock. Se la necessità fosse quella di contare oltre questo valore, si procede all'impiego del prescaler grazie al quale si ottiene un intervallo maggiore di conteggio, rinunciando tuttavia a un minimo di sensibilità nello stesso (Si noti che il *COUNTER_16BIT* gestisce sempre e solo valori compresi tra 0 e 65535 inclusi, a variare è la frequenza con cui lo stesso effettua l'incremento unitario del conteggio).
Questo modulo è costituito da un contatore modulo 2^{11}: quando il valore del conteggio raggiunge la potenza del 2 selezionata, esso viene immediatamente riportato a 0 cosicché il ciclo di conteggio possa riprendere. Per la selezione della potenza corretta, è previsto l'impiego di un decoder con ingresso a 4 bit e uscita a 11 bit: in base al valore dell'ingresso (proveniente dall'uscita PRESC del modulo INPUT_HANDLER, della quale vengono considerati solo i 4 bit meno significativi), viene conseguentemente modificato il comportamento del segnale di controllo del multiplexer, in modo da ripristinare il conteggio se e solo se è necessario. Il segnale di uscita *CLK_ENABLE*, assume valore 1(uno) solo quando il valore in uscita dal flipflop D è pari a 0 (zero espresso su 11 bit). L'interfaccia, con annessi segnali di ingresso e uscita, e lo schema interno del modulo sono presentati sotto:



- *RST* → segnale a un bit che, se posto ad 1 (uno), forza il contatore del prescaler a ripartire da zero nel conteggio;
- *CLK* → clock di alimentazione derivante direttamente dall'ingresso *CLK* del watchdog;
- *EXP_VALUE* → valore a quattro bit dell'esponente scelto e mantenuto all'interno del modulo *INPUT_HANDLER*;
- *CLK-ENABLE* →segnale a un bit che funge da abilitatore di clock per i moduli che ne fanno uso.

Verifica

Data la specifica del problema, è evidente che, una volta che il watchdog ha ricevuto il comando di *START*, si possano verificare tre differenti situazioni:

- il microprocessore non invia il segnale di *CLEAR* prima che il conteggio abbia raggiunto il valore *TWMAX*;
- Il microprocessore invia il segnale di *CLEAR* quando il conteggio ha raggiunto un valore compreso tra *TWMIN* e *TWMAX*;
- Il microprocessore invia il segnale di *CLEAR* prima che il conteggio abbia raggiunto il valore espresso da TWMIN.

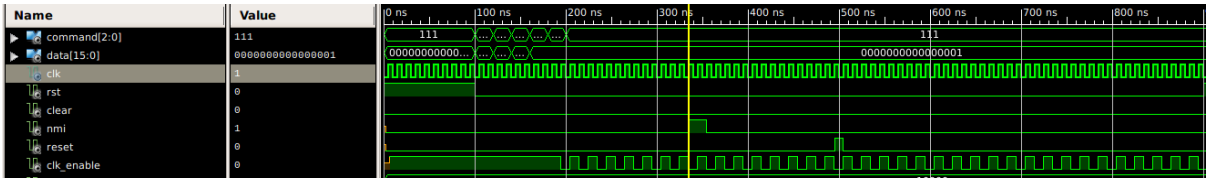
Per ogni simulazione il watchdog necessita di essere configurato. Nello specifico, è necessario impostare i valori *PRESC*, *TWMIN*, *TNMI*, *TWMAX*. Inoltre, tra due test consecutivi, il dispositivo viene ripristinato attraverso il segnale di ingresso *RST* posto pari ad 1 (uno) per 100ns.

Test-bench

Si riportano di seguito i test eseguiti sottolineando che la configurazione dei parametri *PRESC*, *TWMIN*, *TWNMI*, *TWMAX* è la stessa per ciascun test.

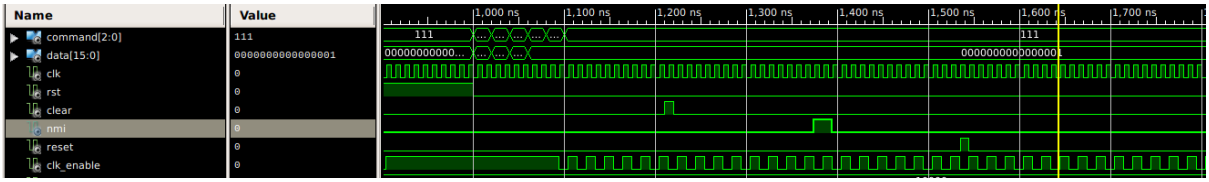
Prima simulazione

Nel caso in cui il microprocessore non invii il segnale di *CLEAR* prima che il conteggio abbia raggiunto il valore stabilito da *TWMAX*, il segnale di uscita *RESET* del watchdog viene posto ad 1 (uno).



Seconda simulazione

In questo secondo caso, il segnale di *CLEAR* da parte del microprocessore arriva nella finestra temporale stabilita da *TWMIN* e *TWMAX*. Ne consegue un ripristino del conteggio al valore 0 (zero) mantenendo attivo il segnale che funge da counter_enable cosicché il conteggio possa riprendere.

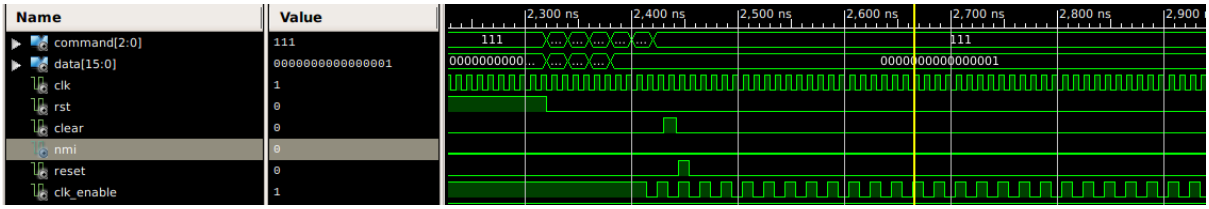


Ricordando che la configurazione del watchdog, e quindi del prescaler, è la stessa per tutti i test, risulta evidente che in seguito alla ricezione del segnale di *CLEAR* il comportamento del watchdog è il medesimo del test precedente. Analizzando l'intervallo di tempo che intercorre tra la ricezione del comando *START* e l'impulso di *RESET* in uscita, si nota che, in effetti, la ricezione del segnale di *CLEAR* ha agito riportando il conteggio al valore zero, senza apportare altre modifiche al dispositivo.

Terza simulazione

Si consideri ora il caso in cui il segnale di *CLEAR* da parte del microprocessore arrivi prima che il conteggio abbia raggiunto il valore espresso da *TWMIN*. Come conseguenza, il watchdog arresta il

conteggio e invia un impulso di *RESET* in uscita per segnalare al microprocessore una situazione anomala.



Casi d'uso

Il dispositivo funzione con segnale di clock avente periodo pari a T_{CLK} = 10ns.