

0.1 Clock divisor e contatore modulo 2^n

0.1.1 Contatore modulo 2^n

Il contatore è un componente che conta il numero di impulsi applicati in ingresso (sul fronte di salita del clock). Oltre al *clock*, in ingresso c'è un segnale di abilitazione (1-attivo) e un segnale di *reset_n* per resettare il conteggio. In uscita, il segnale *counter* riporta il valore di conteggio corrente, mentre *counter_hit* diventa 1 solamente se il valore del conteggio è costituito da tutti 1 (valore massimo). L'implementazione, effettuata tramite descrizione behavioural, è consultabile qui: `run:./esercizio03/design/counter_mod2n.vhdcounter_mod2n.vhd`.

0.1.2 Clock divisor

Prima di utilizzarlo, il segnale di clock in ingresso al componente viene filtrato tramite un clock divisor, che si occupa di filtrare i fronti del clock ad una frequenza *clock_frequency_in* per averli ad una frequenza più bassa *clock_frequency_out*. Il funzionamento di tale componente è del tutto analogo a quello di un contatore modulo 2^n , dove *clock_frequency_out* non è altro che il *counter_hit*, ossia un valore che diventa alto solamente quando il contatore ha raggiunto il suo valore massimo (calcolabile come $\text{clock_frequency_in} / \text{clock_frequency_out} - 1$). Ovviamente sappiamo che il clock generato con questo metodo non è esente da problemi di sincronizzazione e di scue, ma per i nostri scopi e per il fatto che avevamo bisogno di frequenze minori di 5 Mhz, tale scelta risulta più adatta.

L'implementazione, effettuata tramite descrizione behavioural, è consultabile qui: `run:./esercizio03/design/clock_divisor.vhdclock_divisor.vhd`.