0.1 Implementazione

Sulle fpga Artix7 il DCM[1] è stato sostiuito dal CMT [2], tale componente include all'interno un MMCM [3] e un PLL [4].

Per poter sintetizzare un generatore di clock con una frequenza minore o multipla di 100 Mhz limitando lo scew, possiamo utilizzare il CMT. Per utilizzare tale componente, Xilinx fornisce un IP-core che deve essere opportunamente personalizzato,mediante il clocking-wizard , al fine di poter generare un modulo vhdl che utilizza/ configura correttamente il CMT sulla board al fine di fornire un clokc alla frequenza.

Durante la personalizzazione del componente il clocking wizard ci permette di scegliere se avere o meno un segnale, LOCKED, che ci indica quando CMT è riuscito ad agganciare la fase del clock principale e di conseguenza possiamo iniziare a utilizzare correttamente tutti i dispositivi che usano il clock in uscita al CMT.

Nella figura .1 si vede come abbiamo configurato il nostro componente che utilizza il CMT, in particolare abbiamo deciso di utilizzare tre uscite di tale componente configurate in modo tale da fornire segnali di clock con frequenza pari alla metà, un quarto e un decimo della frequenza in ingresso [5].

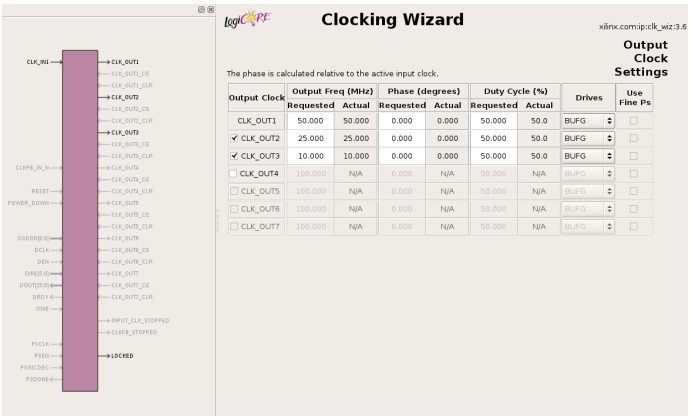

Figure 1: DClocking_wizard

Una volta conclusa la procedura il Wizard genera un file vhd. Per poter simulare e verificare che il componente funzionasse correttamente, abbiamo deciso di realizzare un piccolo componente che utilizza il componente generato e un left shifter register che esegue un operazione di shift ad ogni colpo di clock. La top level entity presenta la seguente interfaccia:

[1] DCM : Digital Clock Manager
[2] Clock Management Tile
[3] mixed-mode clock manager
[4] phase-locked loop
[5] La board Nexys4 è dotata di un oscillatore con frequenza pari a 100 Mhz

[language=VHDL,caption=clock$_t ester.vhd$][...]$entityclk_testerisGENERIC(N$: $integer := 8); Port(clock_in : inSTD_LOGIC; enable : inSTD_LOGIC; reset_n$: $inSTD_LOGIC; d_in : inSTD_LOGIC; q_out : outSTD_LOGIC; Q : outSTD_LOGIC_VECTOR(N-$ $1downto0); half_clock : outSTD_LOGIC; quarter_clock : outSTD_LOGIC; tenth_clock$: $outSTD_LOGIC; locked : outSTD_LOGIC); endclk_tester;$

[...]

C'è da precisare, che tale interfaccia ha come uscita i segnali di clock soltanto per rendere più agevole la fase di analisi della simulazione.

Mentre il collegamento tra i componenti è stato realizzato come di seguito :

[language=VHDL,caption=clock$_t ester.vhd$][...]

signal enable$_i nt : STD_LOGIC :=' 1';$

begin half$_c lock <= half_clock_int; quarter_clock <= quarter_clock_int; tenth_clock <=$ $tenth_clock_int; locked <= enable_int;$

clock$_I nts : my_clockportmap(CLK_IN1 => clock_in, CLK_OUT1 => half_clock_int, CLK_OUT2 =>$ $quarter_clock_int, CLK_OUT3 => tenth_clock_int, LOCKED => enable_int);$

shifter$_r egister_i nst : shifterRegistergenericmap(N => N)portmap(clock =>$ $half_clock_int, enable => enable_int, reset_n => reset_n, left => left, d_in =>$ $d_in, q_out => q_out, Q => Q);$ [...]

In particolare lo shifter registrer è abilato dal segnale di locked del generatore di clock, pertanto appena il componente ci segnalerà che ha aggangiato correttamente la fase del segnale in ingresso lo shifter register inizierà a funzionare, e funzionerà ad una frequenza pari alla metà di quella del segnale in ingresso al componente che usa il CMT.

Di seguito analizziamo i risultati della simulazione, 2, di tale componente.

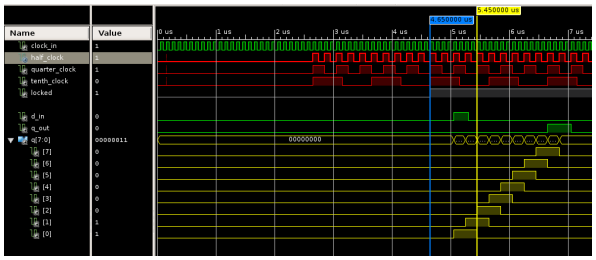

Figure 2: Risultati simulazione

Dopo circa 40 cicli di clock, il segnale locked va alto,marker blu, e il lo shifter register inizia a lavorare, e non appena riceve un valore in ingresso alto, osserviamo che ad ogni qualvolta il segnale half_clock va alto il bit viene shiftato verso sinistra. Inoltre osserviamo che poichè d_in resta alto per circa un periodo e mezzo, il valore in ogni cella dello shifter register viene mantunuto alto per due periodi di half_clock, dopo il terzo periodo il valore ritorna basso perchè si inizia a propagare il valore zero da d_in in tutte le celle a partire dall'instante segnalato dal marker giallo.

L'implementazione completa è consultabile qui: run:./esercizio04/design/clock$_t ester.vhdclock_tester.vhdru$