

Quartus II e la simulazione con ModelSim

Voglio scrivere questo tutorial per facilitare il primo approccio con **ModelSim**, programma che ho trovato con un'interfaccia utente poco intuitiva e capace di spiazzare un utente alle prime armi.

Dedico questo tutorial agli utenti del forum del sito web <http://www.delucagiovanni.com/> e a **Giovanni** in particolare. Insieme mi hanno permesso di fare notevoli passi avanti nella comprensione delle logiche programmabili e consiglio a chiunque interessato all'argomento di frequentare il forum per eliminare i propri dubbi o semplicemente apprendere nuove informazioni.

La guida è stata scritta utilizzando Quartus II 11.1 SP2 (Web Edition) e ModelSim Starter 10.0c ma ampiamente usufruibile anche dagli utilizzatori delle versioni precedenti.

Con ModelSim è possibile effettuare alcuni tipi di simulazioni per verificare i nostri progetti:

- RTL (Register-Transfer Level) simulation
- Gate level simulation

La prima permette di capire a livello logico se il nostro hardware funziona come desiderato, la seconda permette inoltre di simulare le tempistiche e i ritardi reali delle celle sulla FPGA.

Per fare un esempio una semplice porta XOR con la simulazione RTL funziona correttamente anche a 100 GHz mentre con la simulazione a livello di porta (Gate Level) si scopre che funziona correttamente solamente fino a 120 MHz (su una determinata FPGA impostata tramite Quartus II) e rappresenta dunque meglio la realtà.

Vediamo ora come è possibile simulare un semplice modulo VHDL che divide il Clock di ingresso di 50MHz della scheda di sviluppo DE0-Nano in un clock prossimo a 9600Hz necessario ad esempio per una comunicazione UART seriale

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

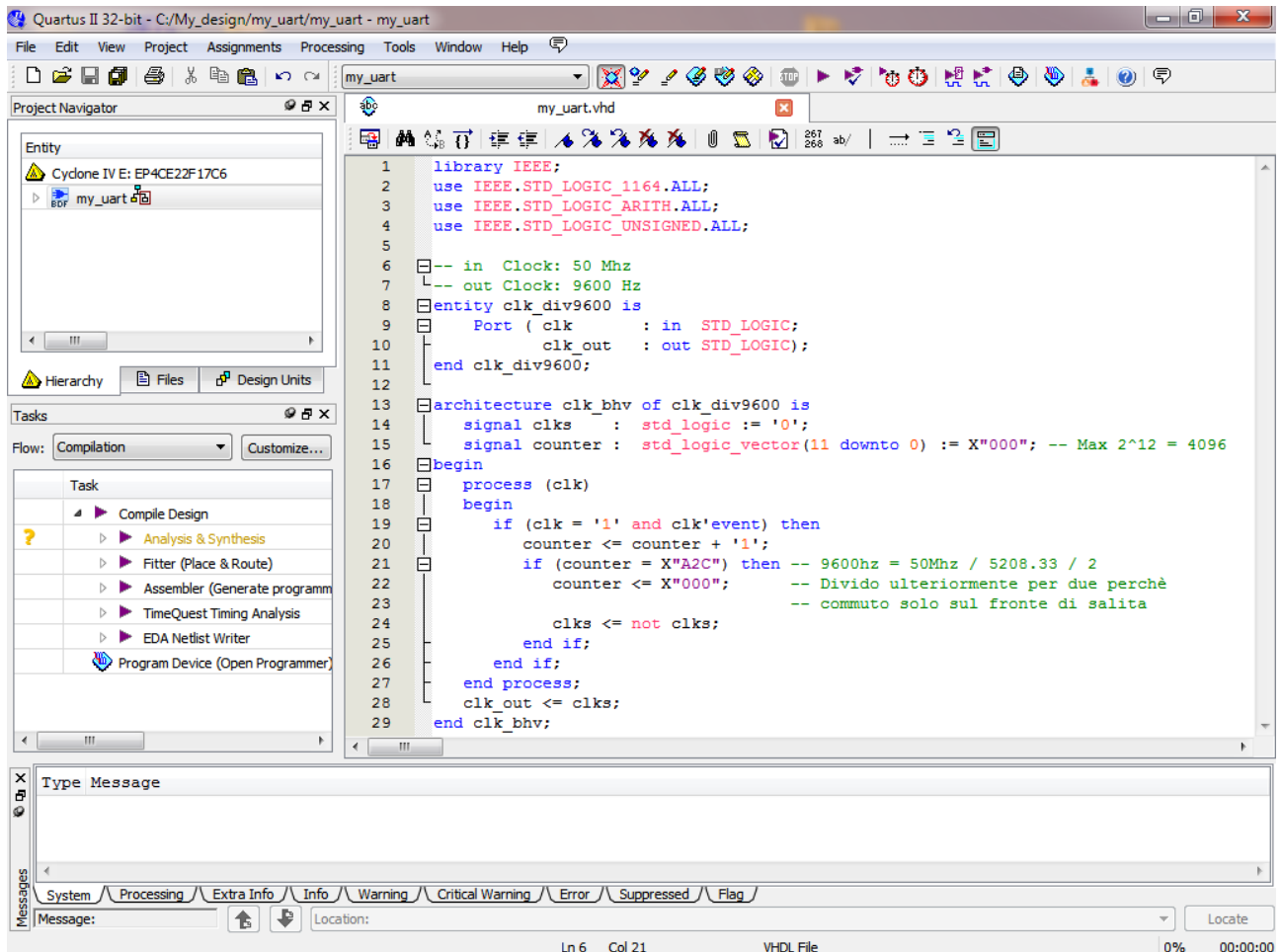
-- in      Clock:      50 Mhz
-- out     Clock:      9600 Hz
entity clk_div9600 is
    Port ( clk          : in  STD_LOGIC;
           clk_out       : out STD_LOGIC);
end clk_div9600;

architecture clk_bhv of clk_div9600 is
    signal clks          : std_logic := '0';
    signal counter       : std_logic_vector(11 downto 0) := x"000";
begin
    process (clk)
    begin
        if (clk = '1' and clk'event) then
            counter <= counter + '1';
        end if;
    end process;
end clk_bhv;
```

```

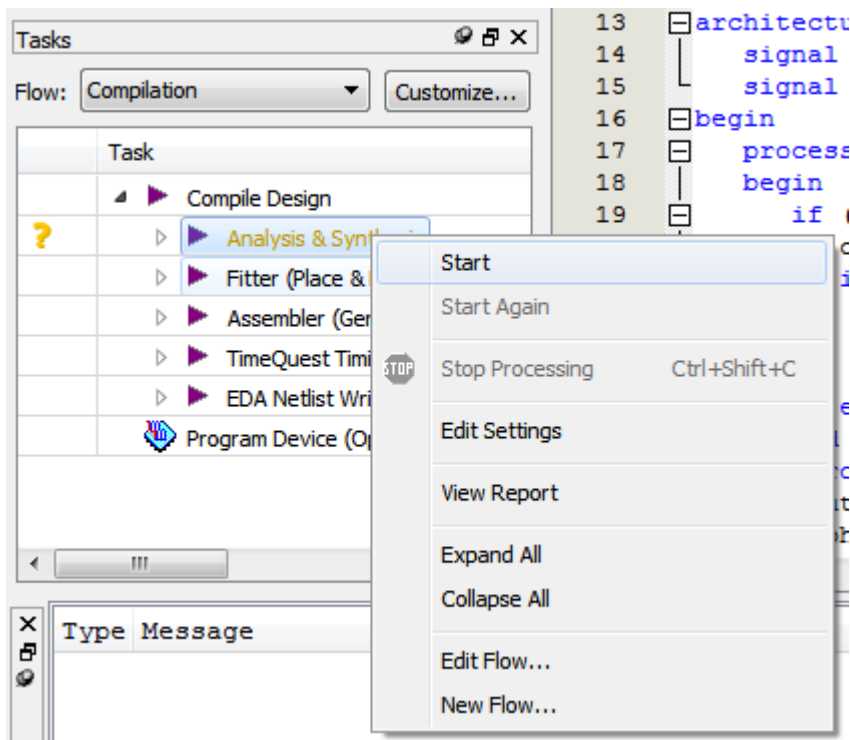
        if (counter = X"A2C") then
            counter <= X"000";
            clks <= not clks;
        end if;
    end if;
end process;
clk_out <= clks;
end clk_bhv;

```



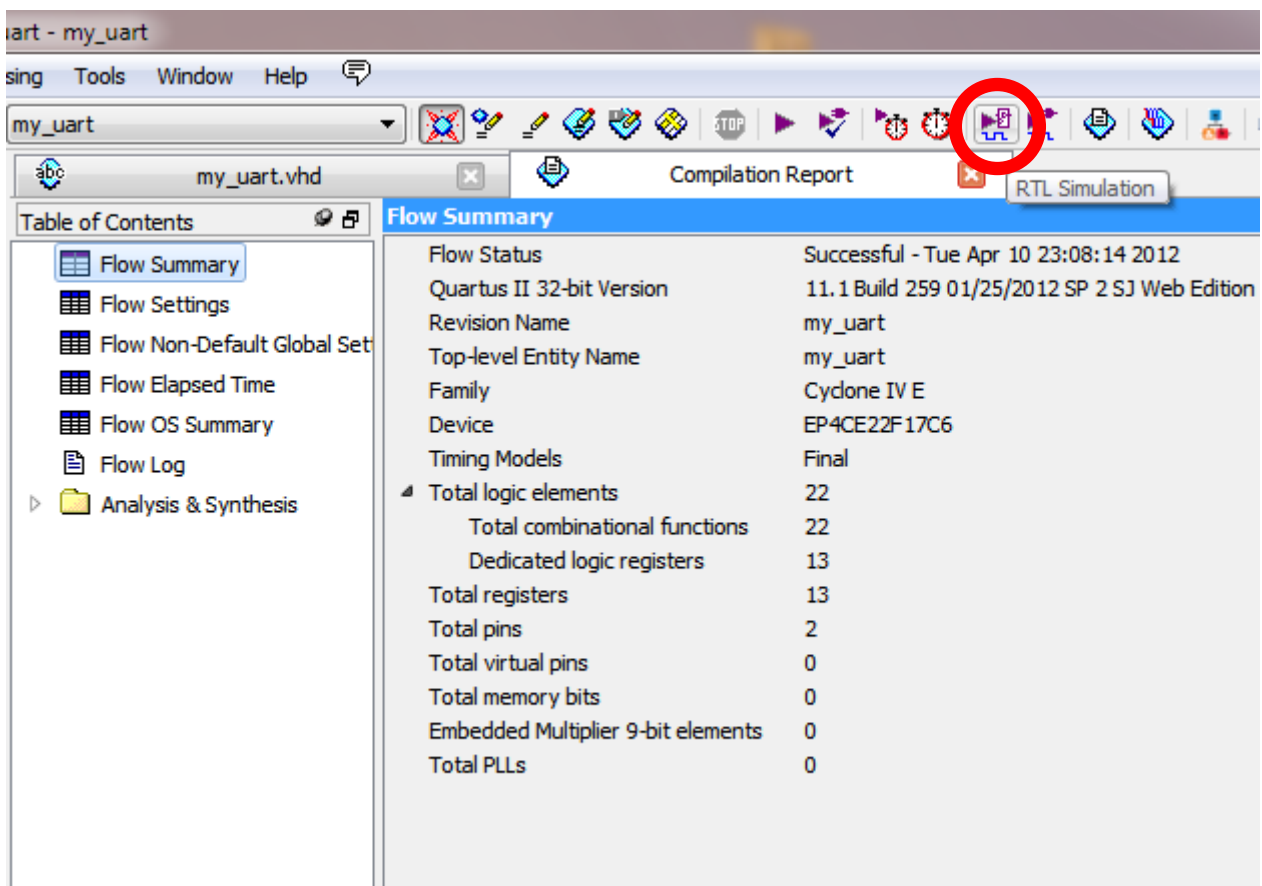
Dopo aver creato un nuovo progetto, scritto ed impostato come entità principale il nostro modulo, il primo passo necessario per la simulazione è l'analisi e la sintesi del modulo.

Facciamo Click destro su **Analysis & Synthesis** e scegliamo **Start** per avviare il processo. E' anche possibile compilare interamente il progetto ma per risparmiare tempo è preferibile eseguire la sola analisi e sintesi.

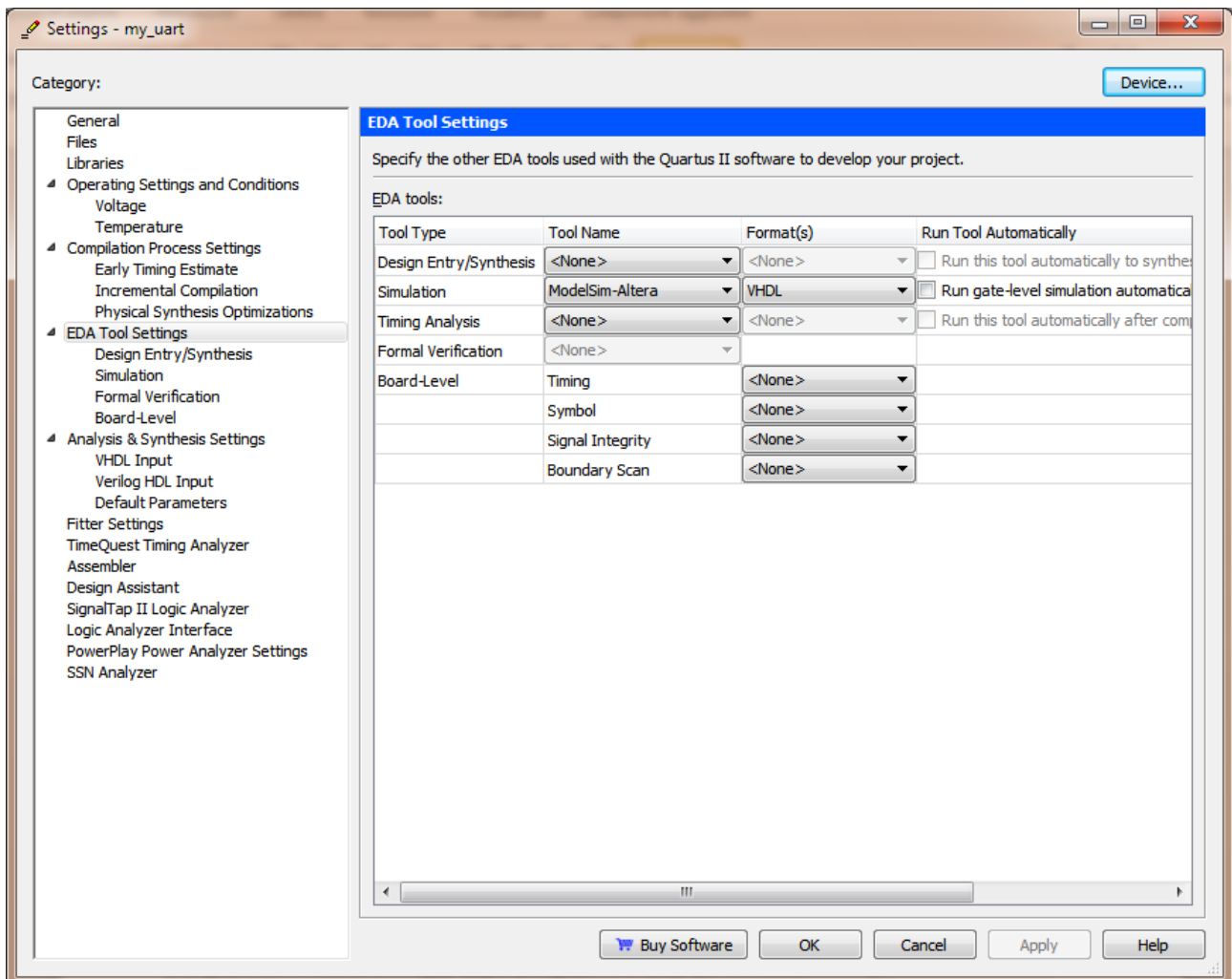


Dopo qualche istante il procedimento sarà completato e un report ci riassumerà le risorse utilizzate sulla nostra FPGA.

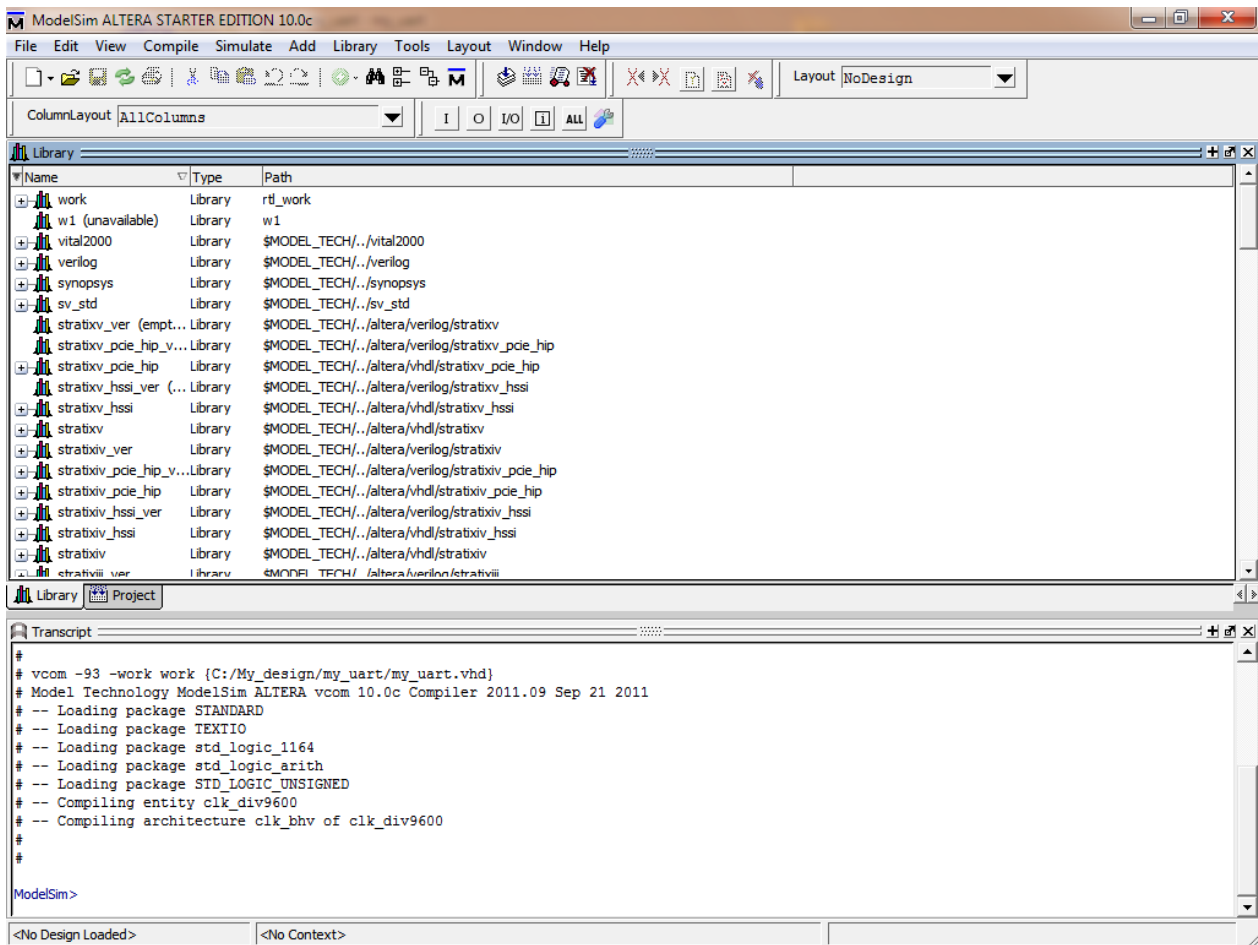
Procediamo scegliendo **RTL Simulation** dal menù **Tools/Run Simulation Tool** o l'icona nella barra degli strumenti.



Verrà lanciato il programma **ModelSim**, se il programma non dovesse apparire potrebbe essere necessario configurarlo nelle impostazioni del progetto tramite il menù **Assignments/Settings** nella categoria **EDA Tool Settings**

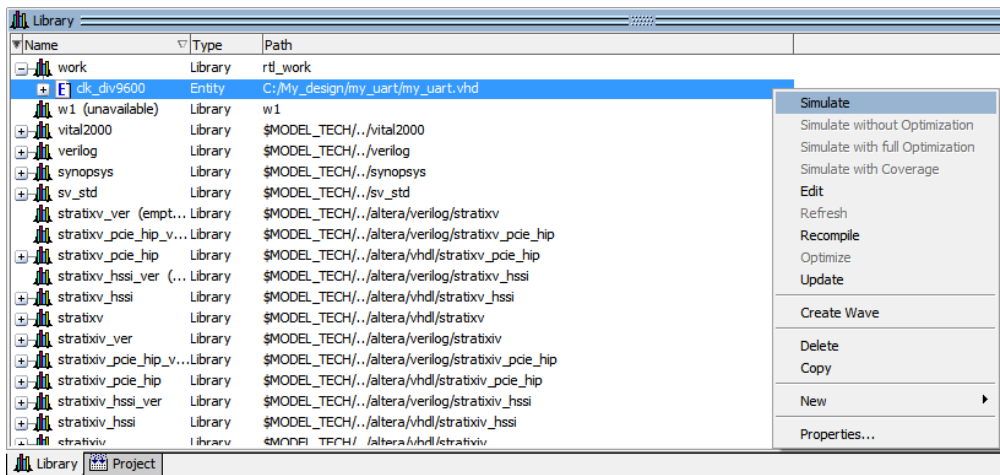


Una volta aperto ModelSim si presenterà una finestra **Library** nella parte superiore ed una finestra **Transcript** nella parte inferiore.



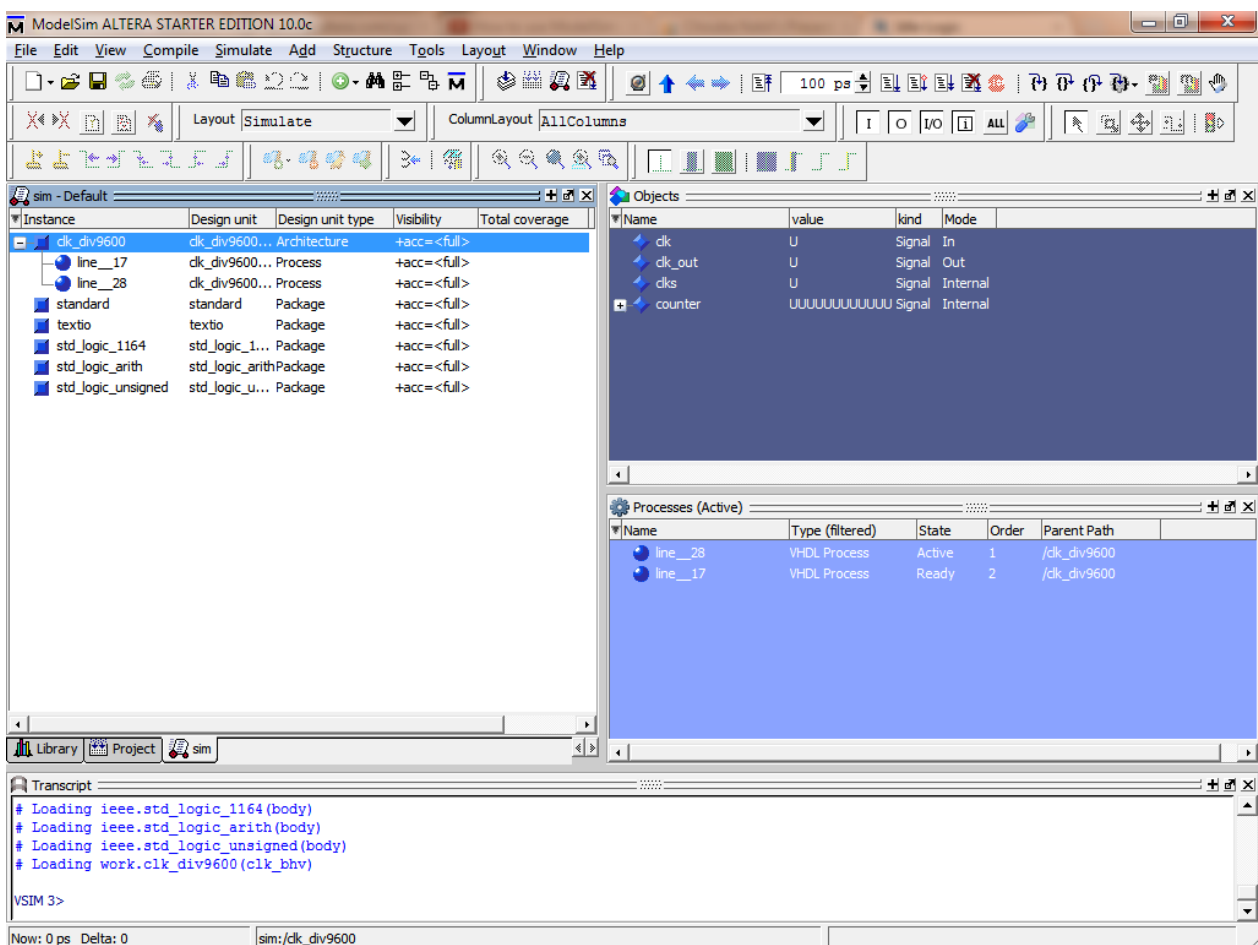
La finestra **Transcript** permette di inserire direttamente dei comandi evitando di utilizzare l'interfaccia grafica, i comandi eseguiti tramite l'interfaccia grafica vengono inoltre riportati in questa finestra, che risulta utile per capire la sintassi dei comandi stessi oltre che per automatizzare alcune operazioni.

Nella finestra **Library** notiamo che Quartus II ha creato una libreria chiamata **work** che contiene le nostre entità. Per iniziare la simulazione facciamo click col tasto destro e scegliamo **Simulate**.

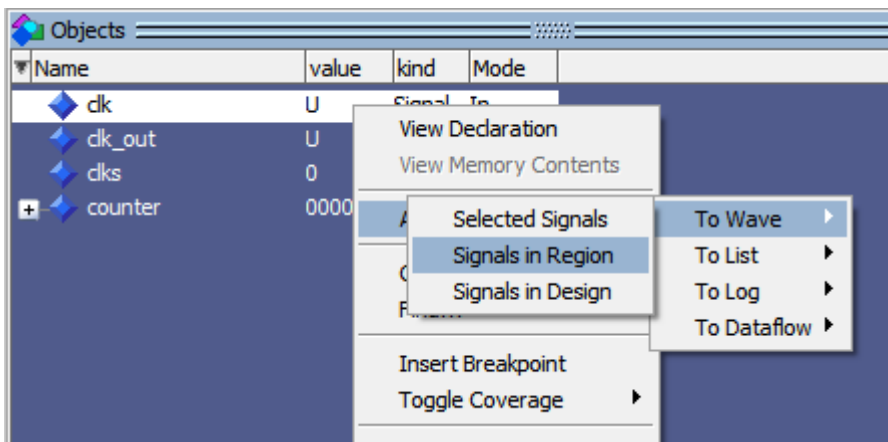


Verranno aperte le finestre **Sim**, **Objects** e **Processes (Active)**.

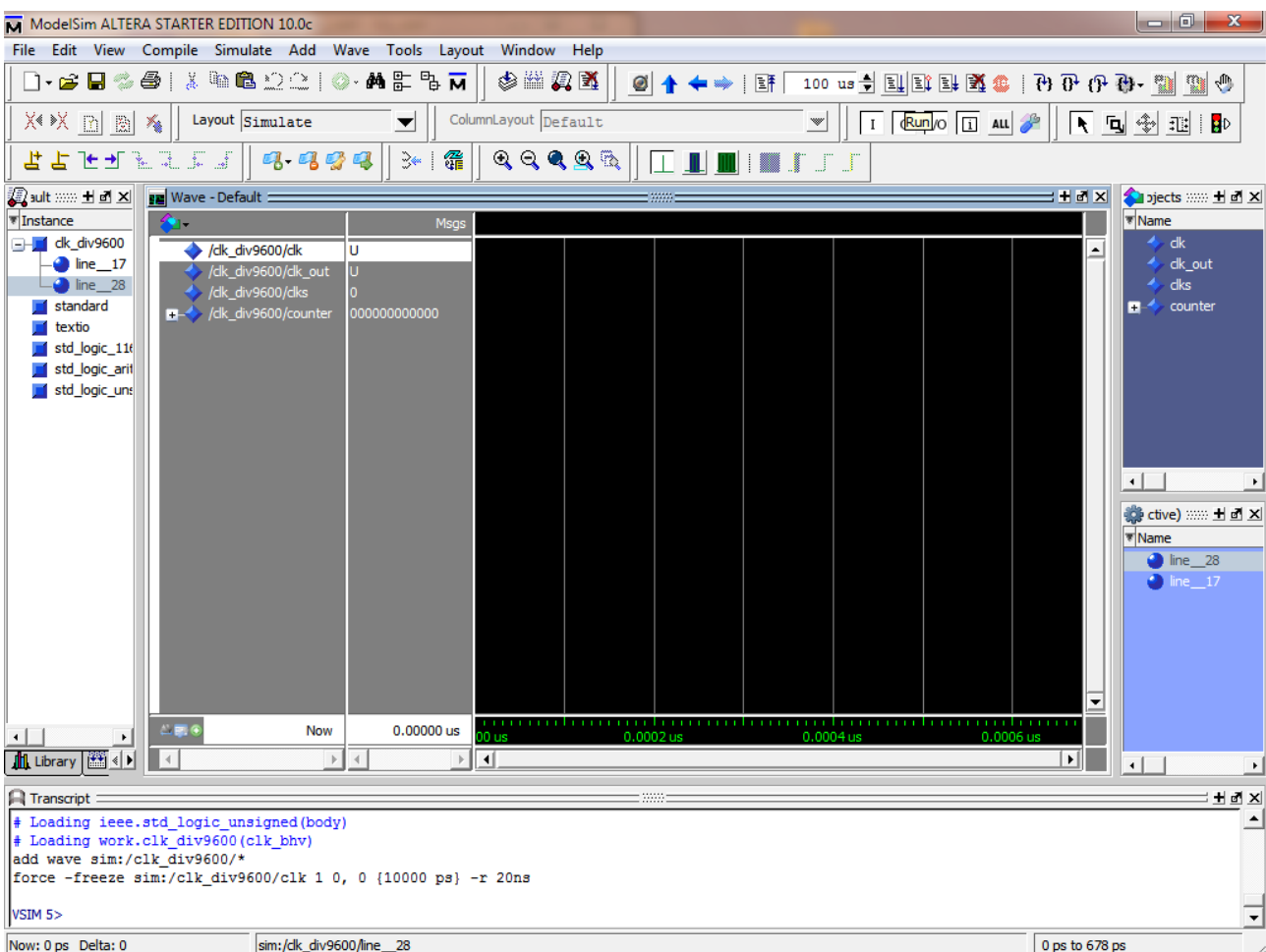
All'interno della finestra **Objects** possiamo trovare i segnali sia interni che esterni del nostro modulo.



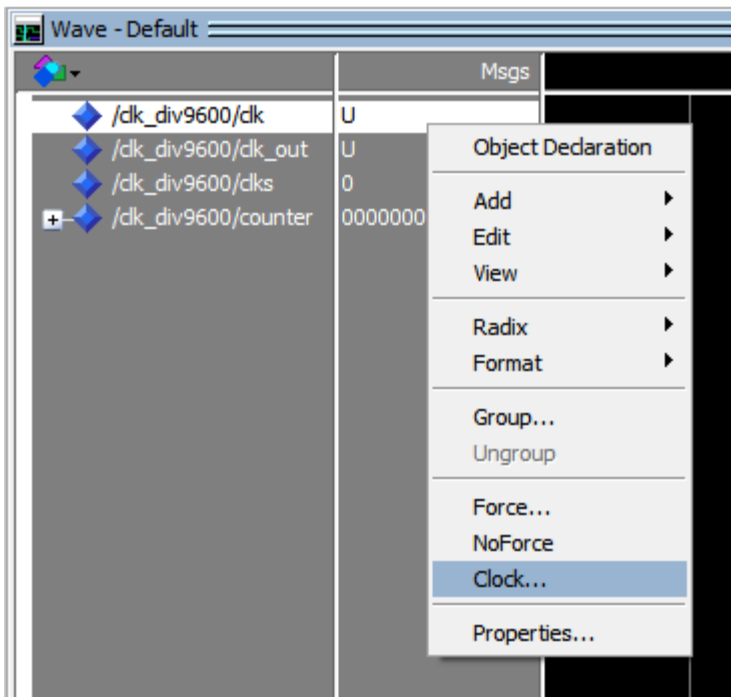
Facendo click destro sul primo oggetto scegliamo **Add/To Wave/Signals in Region** per aggiungere tutti i segnali della regione alla schermata delle onde. Aiutandoci col tasto Control (Ctrl) per selezionare i segnali e scegliendo **Selected Signals** avremmo potuto aggiungere solamente i segnali di nostro interesse.



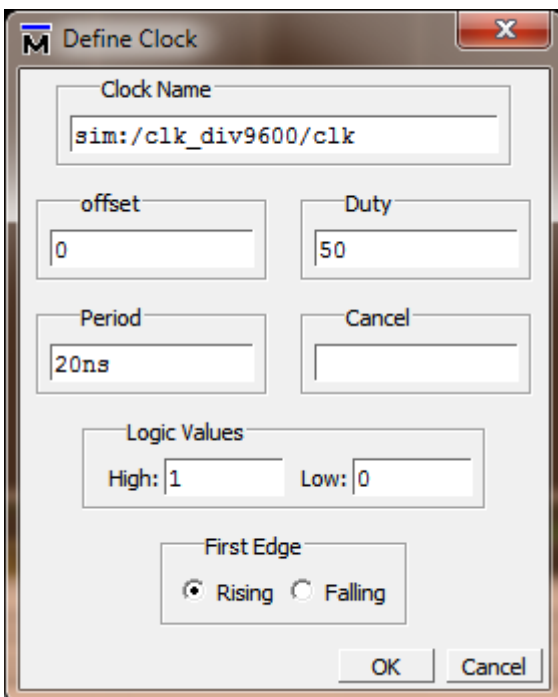
Una volta aggiunti i segnali apparirà la finestra **Wave**, se così non fosse è sempre possibile richiamarla dal menù **View/Wave**. Una volta aperta la finestra ridimensioniamo leggermente le altre per visualizzarla meglio.



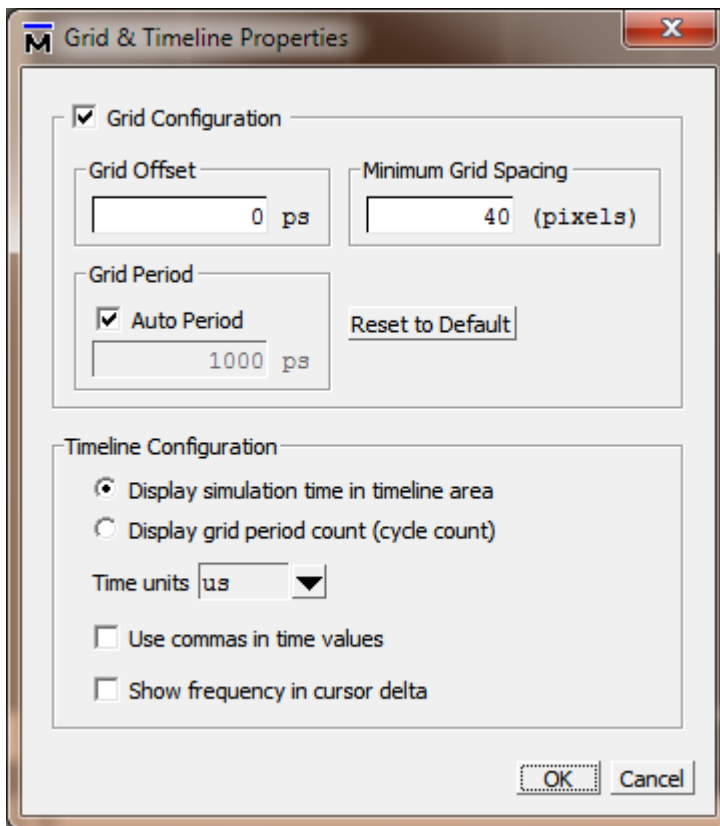
Selezioniamo ora il segnale **clk** nella finestra **Wave** e facendo click col tasto destro scegliamo **Clock** per forzare in ingresso un'onda quadra periodica.



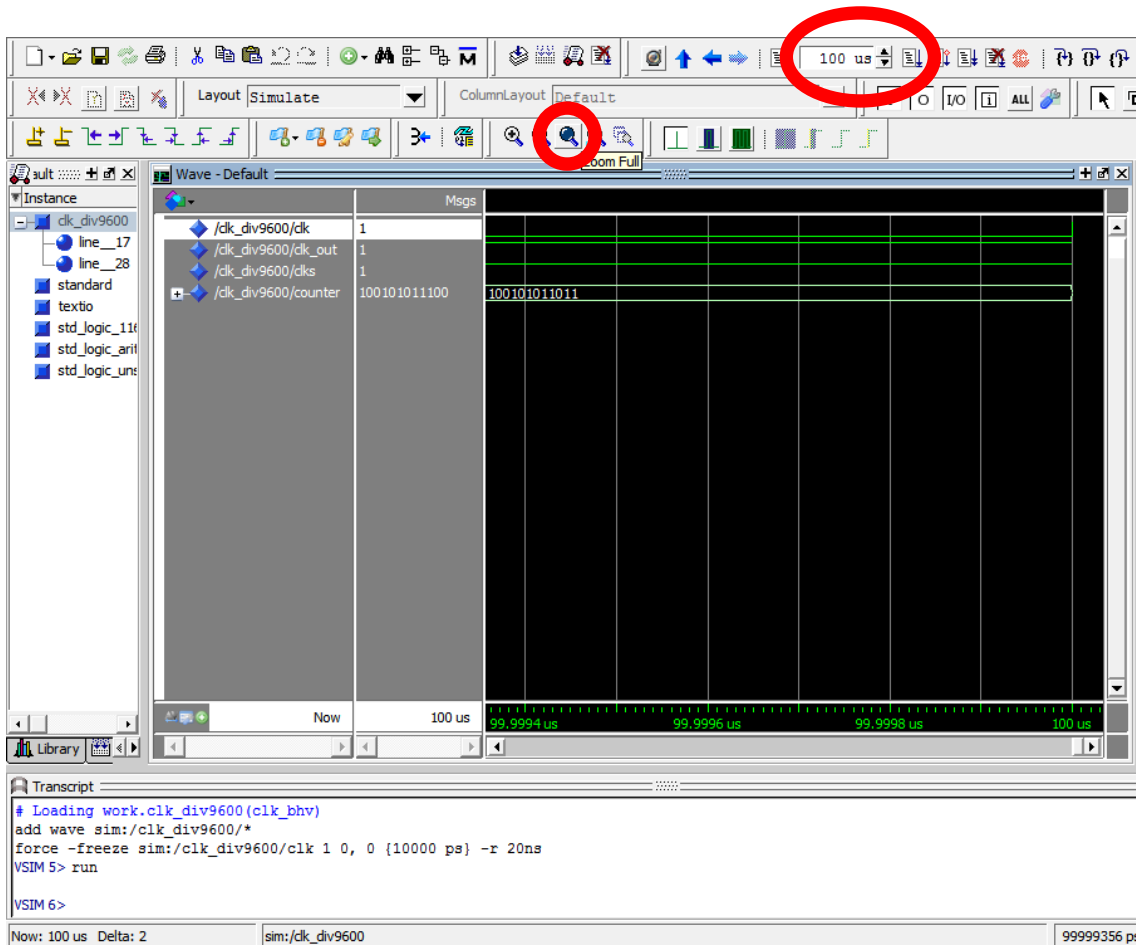
Nella finestra **Define Clock** che apparirà impostiamo un periodo di **20ns** per avere una frequenza di 50 Mhz.



Facendo click sull'ascissa dei tempi col tasto destro e scegliendo **Grid & Timeline Properties** è possibile scegliere come unità del tempo i microsecondi. Saranno utili per verificare la frequenza di uscita di 9600Hz equivalenti a 104.16 μ s



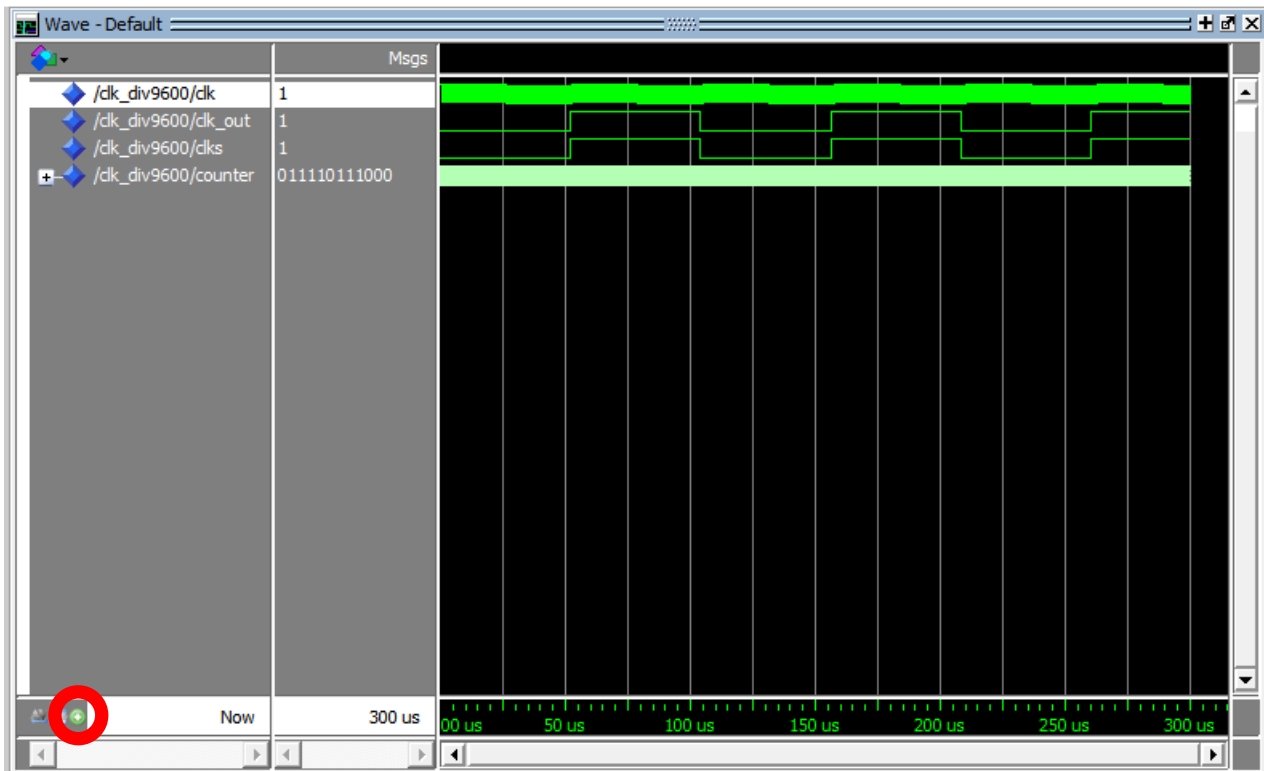
Per impostare la durata della simulazione scriviamo **100us** nella casella di testo nella barra degli strumenti e clicchiamo il pulsante **Run**, il grafico assumerà l'aspetto in figura e mostrerà gli ultimi microsecondi della simulazione.



E' possibile scorrere il grafico tramite le barre di scorrimento ed ingrandire e rimpicciolire il grafico tramite i pulsanti della barra degli strumenti.

Notiamo i comandi che sono stati chiamati nella finestra **Transcript**.

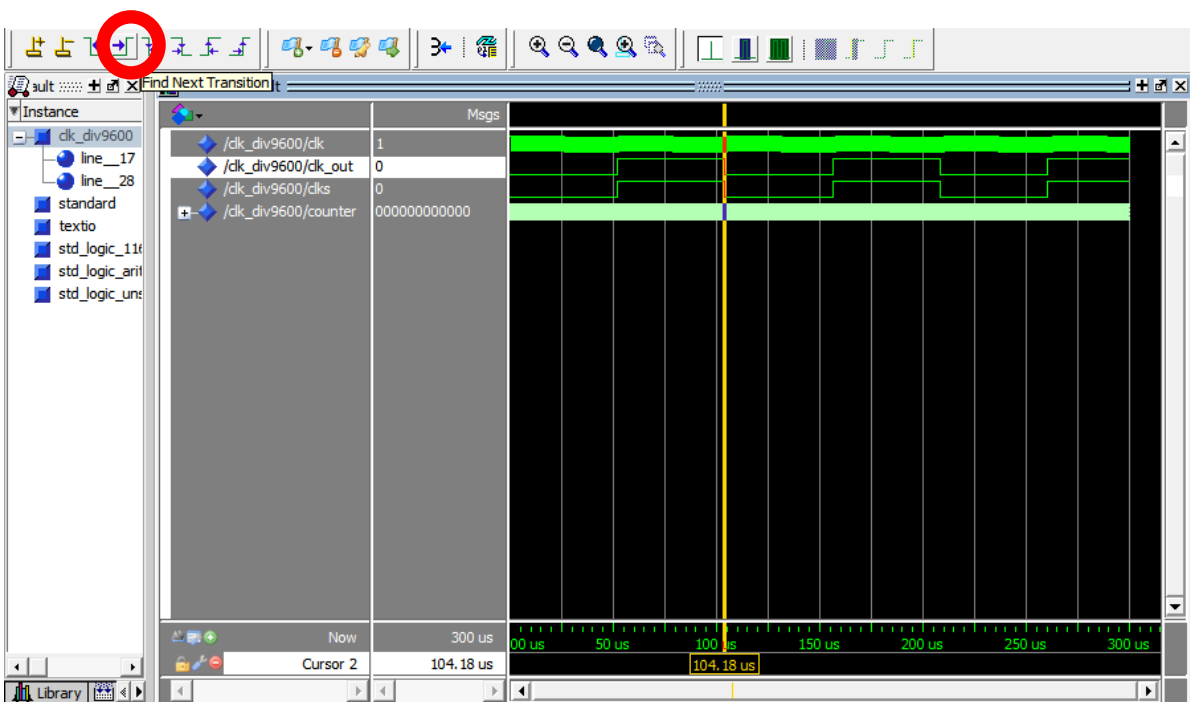
Clicchiamo altre due volte su **Run** per simulare in tutto 300uS e su **Zoom Full** (immagine precedente) per visualizzare l'intera simulazione



Come aspettato il segnale di uscita **clk_out** genera un frequenza molto inferiore al segnale **clk**.

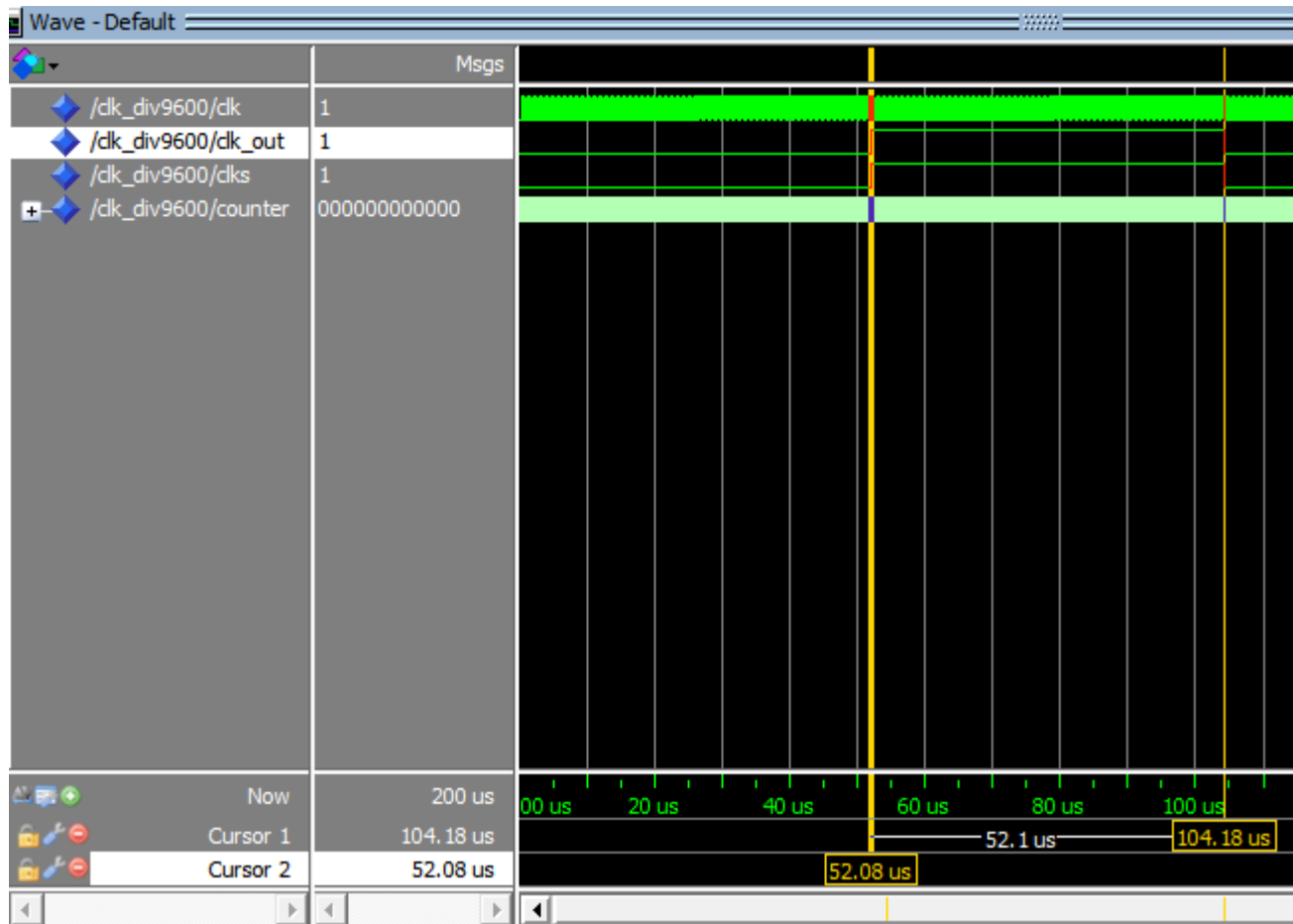
È possibile misurare con precisione la frequenza del segnale aggiungendo un nuovo cursore facendo click sul pulsante **Insert Cursor** evidenziato con un cerchio rosso in figura.

Apparirà un nuovo cursore chiamato **Cursor 2**, selezioniamo il segnale **clk_out** e facciamo click due volte su **Find Next Transition** nella barra degli strumenti per spostare il cursore sul fronte di discesa del segnale.



Una linea indicante il tempo della transizione ci aiuterà a trovare la frequenza del segnale. Nell'immagine la seconda transizione che conclude il primo ciclo del clock di uscita è a 104.18 us, un risultato molto buono considerando che un segnale a 9600 Hz ha cicli di 104.166 us.

Inserendo più cursori è inoltre possibile visualizzare in automatico la differenza di tempo, funzionalità che potrà tornare senz'altro molto utile in diverse situazioni.



La simulazione RTL non tiene conto dei ritardi dovuti ai tempi di propagazione dei segnali ma è un utile strumento per verificare a livello logico il proprio progetto.

Nel prossimo tutorial eseguiremo una simulazione a livello di gate per scoprire un comportamento più vicino alla realtà che tuttavia dipende anche dal PCB e quindi l'ultima parola è sempre data dall'oscilloscopio.

Alla prossima

Leonardo (Flz47655)