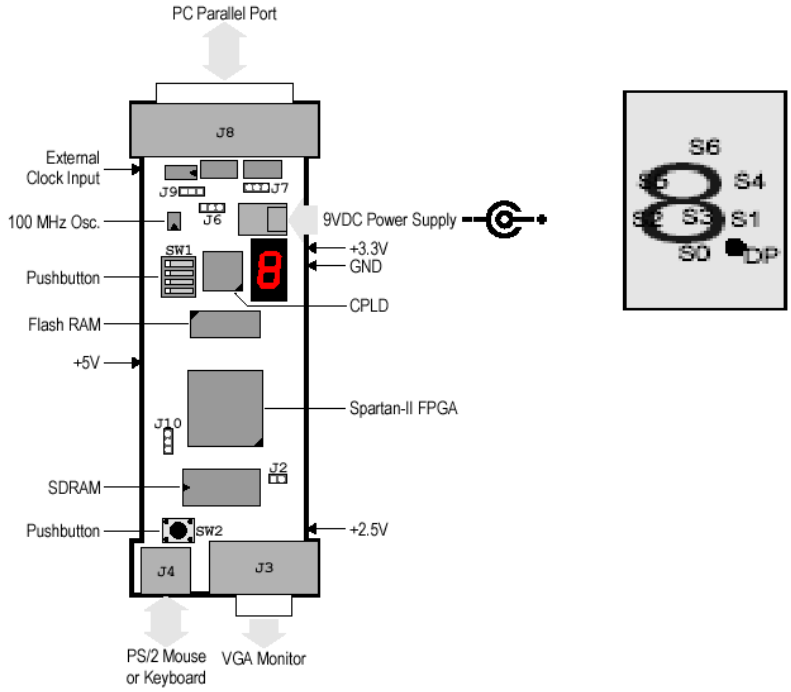
Esercitazione n° 4

Lo scopo di questa esercitazione è quello di introdurre alla creazione di un disegno di progetto utilizzando il tool StateCAD al fine di realizzare una Finite State Machine (FSM) da cui poi verrà generato il codice VHDL.

Problema

Realizzare mediante un file sorgente costituito da un diagramma di stato (State Diagram), un dispositivo che, una volta premuto il pulsante SW2, riconosca se si è tenuto premuto a lungo il pulsante (‘click lungo’) o se si è tenuto premuto per un intervallo breve (‘click breve’). Se si ottiene un click lungo sul display a 7 segmenti si dovrà accendere il led S0, altrimenti si accenderà il led S1.
Per semplificare il progetto, si faccia in modo che, una volta riconosciuto il tipo di click, per rieffettuare l’operazione si debba azionare un tasto (anche lo stesso scelto per il reset asincrono), ad esempio dipsw1.



- Valgono le seguenti specifiche:

  ~0.35s < ‘click breve’ < ~1s
  ‘click lungo’ > ~1s

Suggerimenti

- Anche utilizzando il massimo divisore di clock presente sulla XSA Board, si ha comunque un clock a 48.7 kHz. Pertanto sarà necessario portare il clock a una velocità umana, in modo da riuscire effettivamente a discriminare i due casi (‘click lungo’ e ‘click breve’). Bisognerà progettare un **divisore di clock**. Si tenga conto del fatto che se si ottiene un clock troppo lento, il dispositivo non funzionerà ugualmente.

Soluzione

Si inizi come sempre creando un nuovo progetto dal nome ‘click2’ per esempio.
Seguendo quanto suggerito, procediamo con la realizzazione del divisore di clock.

Realizzazione del divisore di clock

Esistono vari modi per progettare il disegno di un divisore di clock. Di base è necessario ci sia un dispositivo che conti. Noi realizzeremo il tutto utilizzando lo StateCAD. Si tenga conto che nel realizzare un diagramma di stato è necessario inserire un reset, sincrono o asincrono. Nel nostro caso assoceremo al diagramma un reset asincrono posizionato sul dipsw2 (location ‘p64’).

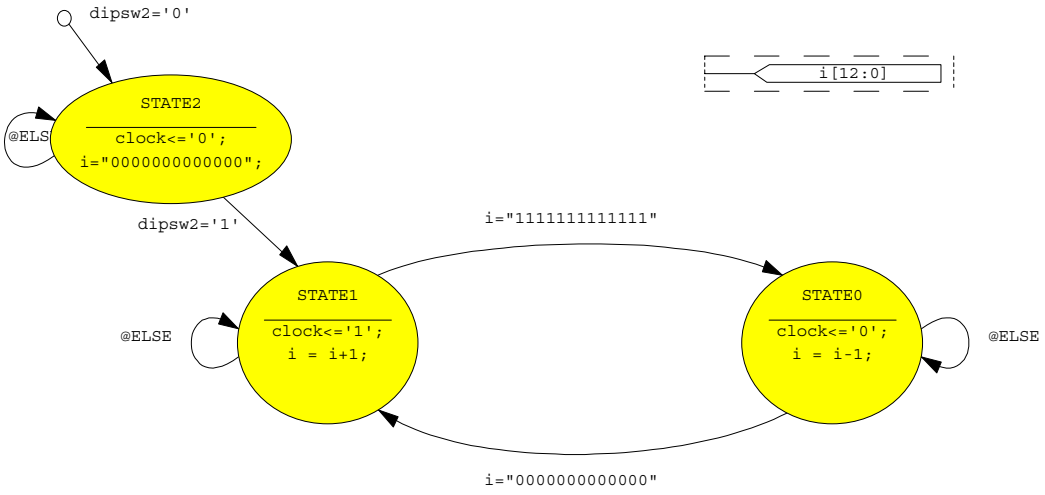
- Dal Project Navigator si aggiunga un file sorgente del tipo ‘State Diagram’ (Project → New Source → State Diagram) dal nome ‘div_clk’, per esempio. Si aprirà la finestra dello StateCAD.

- A questo punto analizziamo il comportamento del divisore di clock. Deve ridurre il clock della XSA Board, che imposteremo a 48.7 kHz. Faremo in modo che il clock finale sia di circa 3Hz. Dividendo 48700 per 3 si ottiene circa 16233. poniamo che un ciclo di clock sia composto da mezzo ciclo in cui il clock vale 1 e mezzo ciclo in cui il clock vale 0. Quindi il segnale di clock in uscita dal divisore di clock varierà ogni 16233/2 colpi del clock di macchina, cioè circa ogni 8120 colpi. Poiché il contatore sarà contenuto in un registro binario vediamo quale potenza di 2 si avvicina maggiormente a 8120:

  $$2^{13} = 8192$$

  quindi opteremo per un registro di conteggio ‘i’ a 13 posizioni del tipo [12:0], per cui il numero binario “1111111111111” corrisponde a 8191 in base 10.
  Il disegno del diagramma di stato sarà costituito da due stati fondamentali, uno associato all’uscita clock<=‘1’ (STATE1), l’altro all’uscita clock<= ‘0’ (STATE0): il nuovo clock sarà scandito dall’uscita ‘clock’; in più inseriremo uno stato (STATE2) per il reset asincrono del dispositivo allo scopo di azzerare il contatore. Nello STATE1 (clock<= ‘1’), il contatore viene incrementato di 1 (i=i+1) unità finchè i=16383 (in codifica binaria ‘1111111111111’); in questo momento avviene il passaggio allo STATE0 (clock<= ‘0’); nello STATE0 (clock<= ‘0’) il contatore viene decrementato di 1 unità (i=i-1). La permanenza nello STATE0 si ha finchè i=0 (cioè i= “0000000000000”); quando i= “0000000000000” si ritorna allo STATE1.
  La macchina lavora in questa maniera a meno che non si attivi il meccanismo di reset. In tal caso si passa allo STATE2 e vi si permane fintanto che dipsw2 è ON (cioè il circuito è messo a terra. Vedi XSA Board User Manual a pag. 24).
  Il diagramma da realizzare è il seguente:

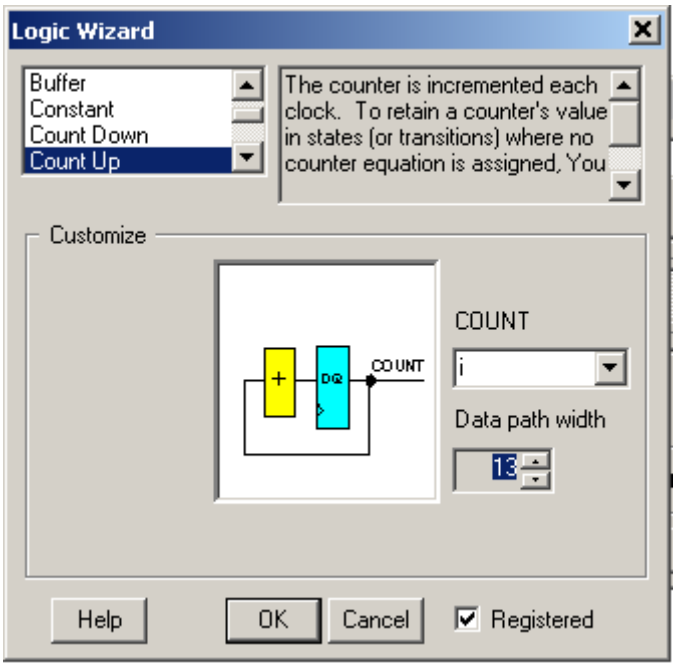
  Il divisore di clock così realizzato, anche se intuitivo da un punto di vista algoritmico, risulta piuttosto complesso specialmente considerando le risorse impiegate. Sarebbe molto piu' semplice realizzare un semplice contatore ed utilizzare come "clock a bassa frequenza" il bit piu' significativo dell'uscita.



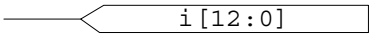
- A partire dalla finestra dello StateCAD, per creare un nuovo stato premere il pulsante  , poi posizionare il puntatore sull'area di lavoro e fare click.

  Per collegare i vari stati premere il pulsante  , poi clickare in serie (prima su uno, poi sull'altro) i due stati da collegare.

  Per assegnare il reset premere il pulsante  , poi clickare in serie un punto sull'area bianca vicino allo stato di reset e lo stato di reset. A questo punto verrà chiesto se il reset deve essere sincrono o asincrono. Nel nostro caso sarà, come già detto, asincrono.
  Ora bisogna specificare le uscite e le operazioni per ciascuno stato, e le condizioni di passaggio di stato. Per accedere al menu di configurazione di questi parametri, basta premere il pulsante  e fare doppio click sullo stato prescelto o sulla freccia che segna il passaggio.
  Per programmare i contatori per gli stati STATE0 e STATE1 ci serviremo dell' 'Output Wizard'.
  Per lo STATE1. Doppio click su STATE1 → Output Wizard. Poi come segue:



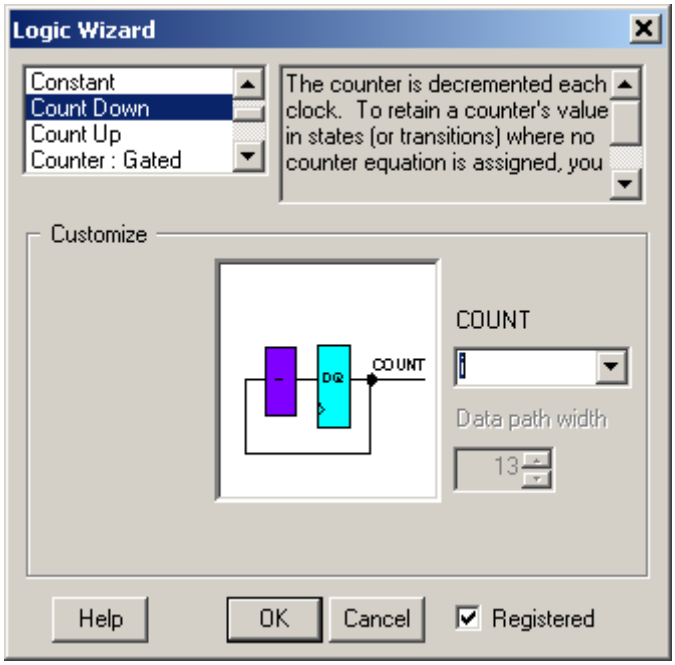
  Poi premere 'OK'. Nell'area 'Outputs' verrà generata l'istruzione i=i+1. Verrà inoltre creato il registro i[12:0]:



  Per lo STATE1 resta ancora da specificare il valore della variabile d'uscita 'clock'.
  L'istruzione: clock<='1';

va scritta nell'area 'Outputs' della finestra 'Edit State' relativa allo STATE1.

Per lo STATE0 utilizzeremo un contatore che agisca sempre sul registro 'i', ma sottraendo: si proceda come per lo STATE1: selezionando l' 'Output Wizard', poi:



Per completare l'istruzione: clock<='0';
va scritta nell'area 'Outputs' della finestra 'Edit State' relativa allo STATE0.

Per quanto concerne lo STATE2, le istruzioni:
clock<='0';
i="0000000000000";
vanno scritte nell'area 'Outputs' della finestra 'Edit State' relativa a STATE2.

Analogamente per le condizioni di passaggio di stato.

L'istruzione: i="1111111111111"
va scritta nell'area 'Condition' della finestra 'Edit Condition' relativa alla freccia di passaggio dallo STATE1 allo STATE0.

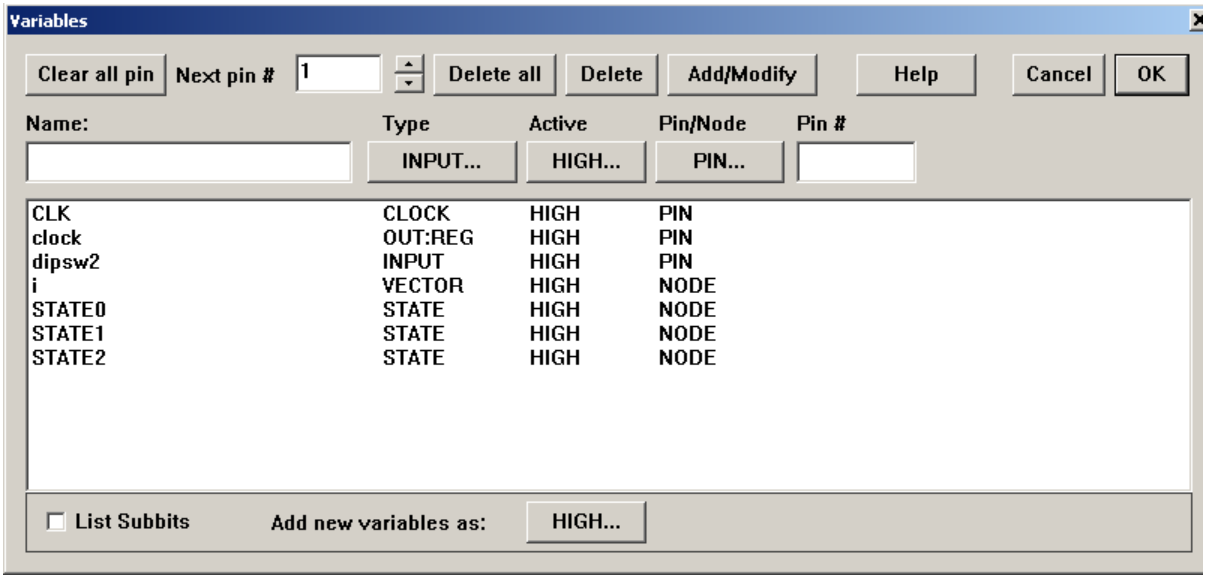
L'istruzione: i="0000000000000"
va scritta nell'area 'Condition' della finestra 'Edit Condition' relativa alla freccia di passaggio dallo STATE0 allo STATE1.

L'istruzione: dipsw2='1'
va scritta nell'area 'Condition' della finestra 'Edit Condition' relativa alla freccia di passaggio dallo STATE2 allo STATE1.

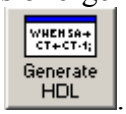
L'istruzione: dipsw2='0'
va scritta nell'area 'Condition' della finestra 'Edit Condition' relativa alla freccia del reset.

Le frecce che ritornano sullo stato sono marcate automaticamente con @ELSE.

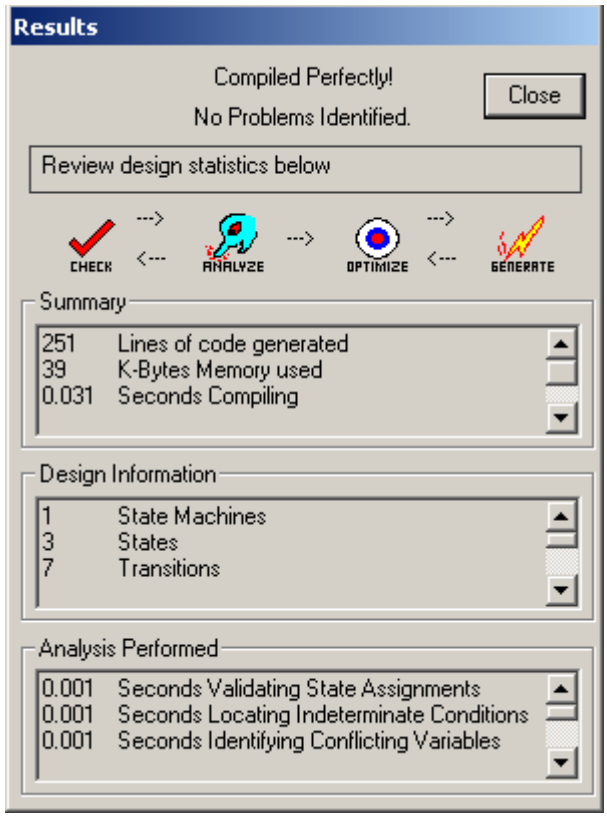
- A questo punto è possibile controllare le variabili in uso: Options → Variable:



Si faccia attenzione alla sezione 'Active', che può essere 'HIGH' o 'LOW': se 'HIGH', quando la variabile vale '1' l'onda relativa si trova allo stato alto; se 'LOW', quando la variabile vale '1', l'onda relativa trova allo stato basso. Si vedrà meglio in simulazione.
Salviamo il file.

- È quindi possibile generare il codice VHDL corrispondente al disegno appena completato premendo su 

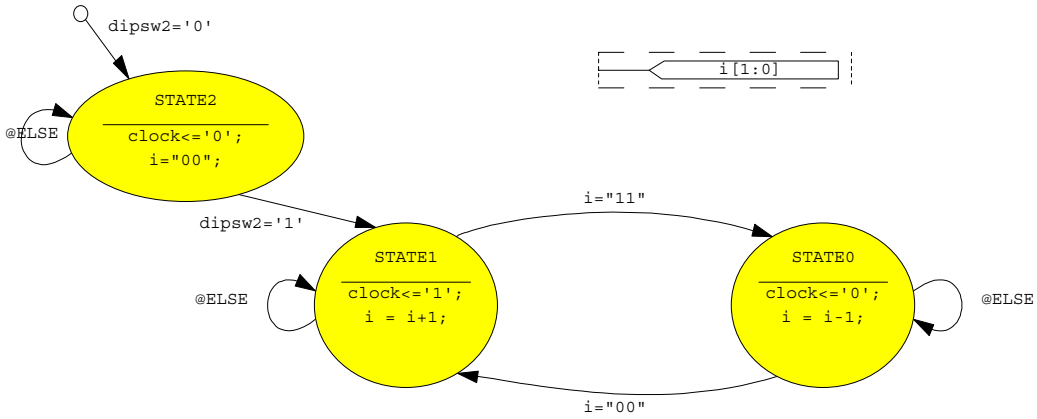
.
Comparirà la finestra 'Optimize Outputs for Speed' nella quale bisogna scegliere se ottimizzare le variabili elencate. Si scelga 'Optimize'.
Comparirà di seguito la finestra 'Optimize Port I/O' per scegliere le variabili che restano interne al progetto, nel nostro caso il registro 'i'. Si scelga 'Optimize'
Quindi il tool verificherà il disegno, lo analizzerà, lo ottimizzerà e genererà il codice ad esso relativo:



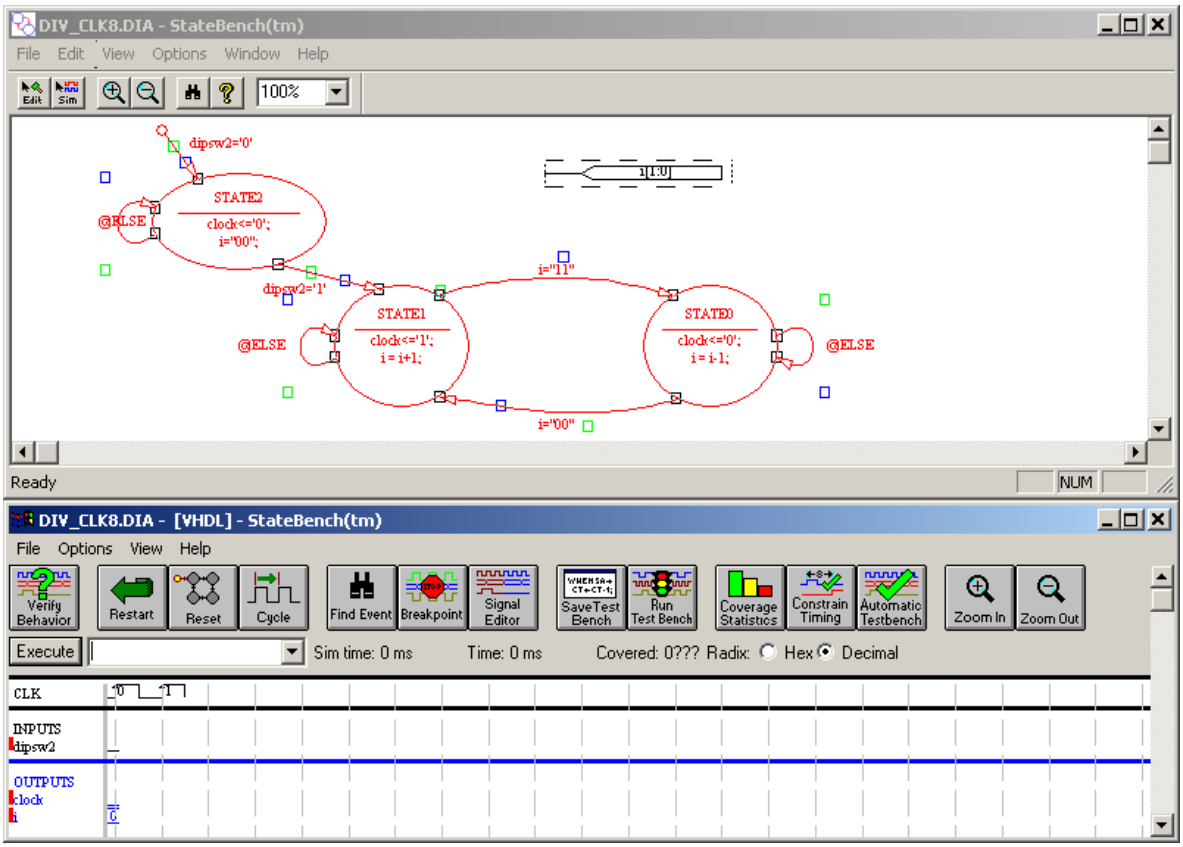
Premendo su ‘Close’ comparirà la finestra con il codice VHDL generato. Il file contenente il codice avrà nome ‘DIV_CLK.vhd’.

- A questo punto si potrebbe, ad esempio, utilizzare lo ‘State Bench’ 

per controllare il funzionamento di quanto creato. Se però dovessimo simulare le transizioni almeno fino al primo passaggio di stato, avremmo bisogno di 16384 transizioni. Quindi non fattibile in tempi ragionevoli. Se si vuole quindi provare l’effettivo funzionamento di questo tipo di diagramma è consigliabile di ridurre la capacità del registro ‘i’, per esempio creando un diagramma come quello seguente:



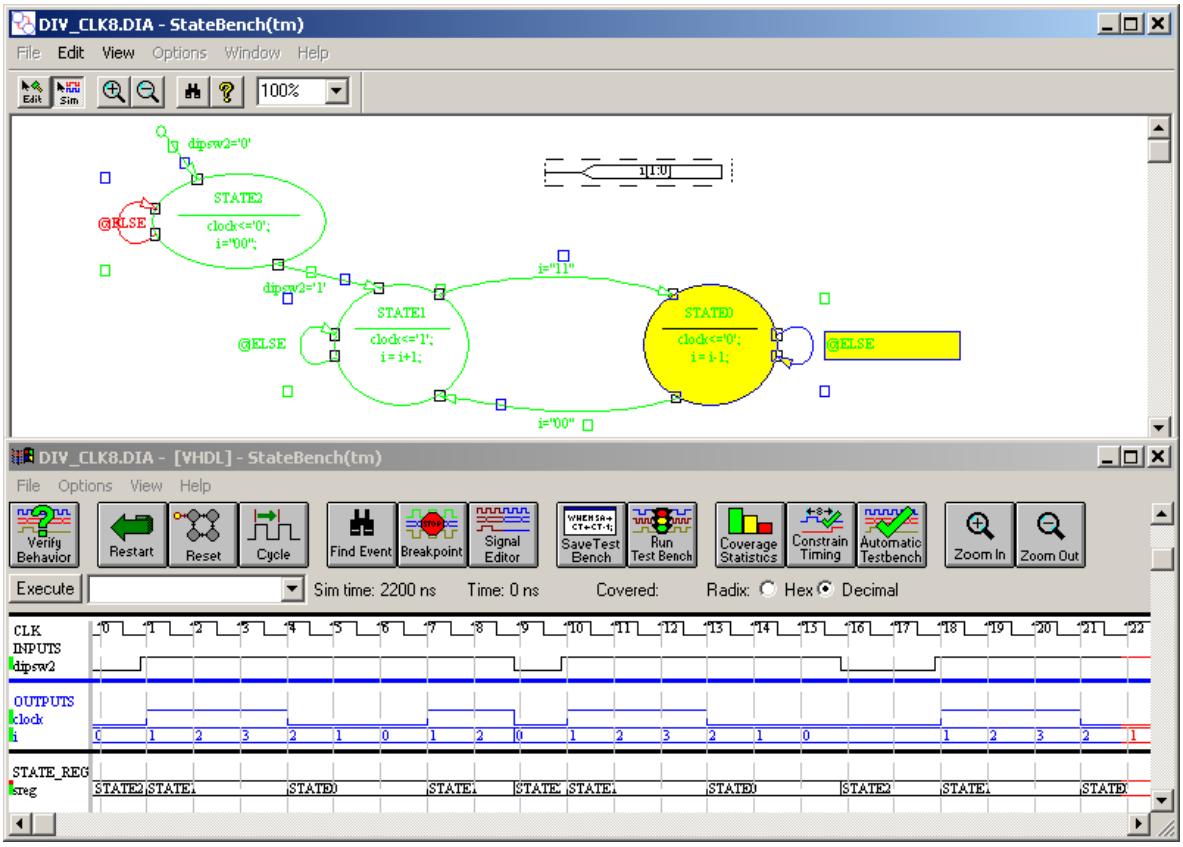
È in tutto e per tutto uguale al diagramma precedente, se non per il fatto che la transizione avviene ogni 3 colpi di clock, quindi risulta facilmente controllabile.
Si provi ad avviare lo State Bench per operare una simulazione di tipo funzionale. Si aprono due finestre:



Nella finestra soprastante è possibile andare a selezionare i vari stati o le varie condizioni di passaggio per simulare il comportamento del diagramma. Clickando il pulsante 

è possibile effettuare una simulazione. È necessario partire dallo stato di Reset.
Nella finestra sottostante si possono leggere i valori relativi alle variabili coinvolte. Ad esempio si provi la seguente successione:

STATE2 → STATE1 → STATE1 → STATE1 → STATE0 → STATE0 → STATE0 → STATE1 → STATE2 → STATE1 → STATE1 → STATE1 → STATE0 → STATE0 → STATE0 → STATE2 → STATE1.



Osservando la finestra sottostante è possibile verificare l'effettivo funzionamento del diagramma. La simulazione ha confermato ciò che ci aspettavamo.
Il tool StateBench genera anche un file in codice VHDL dov'è descritto il Test Bench su cui si è provato il disegno creato. Il file ha nome 'DIV_CLK_TB.vhd

- Ora ritorniamo al Project Navigator e aggiungiamo il file sorgente 'div_clk.vhd': Project → Add Source, scegliere 'DIV_CLK.vhd' che è un 'VHDL Module'.

A questo punto dobbiamo procedere con la realizzazione del dispositivo vero e proprio.

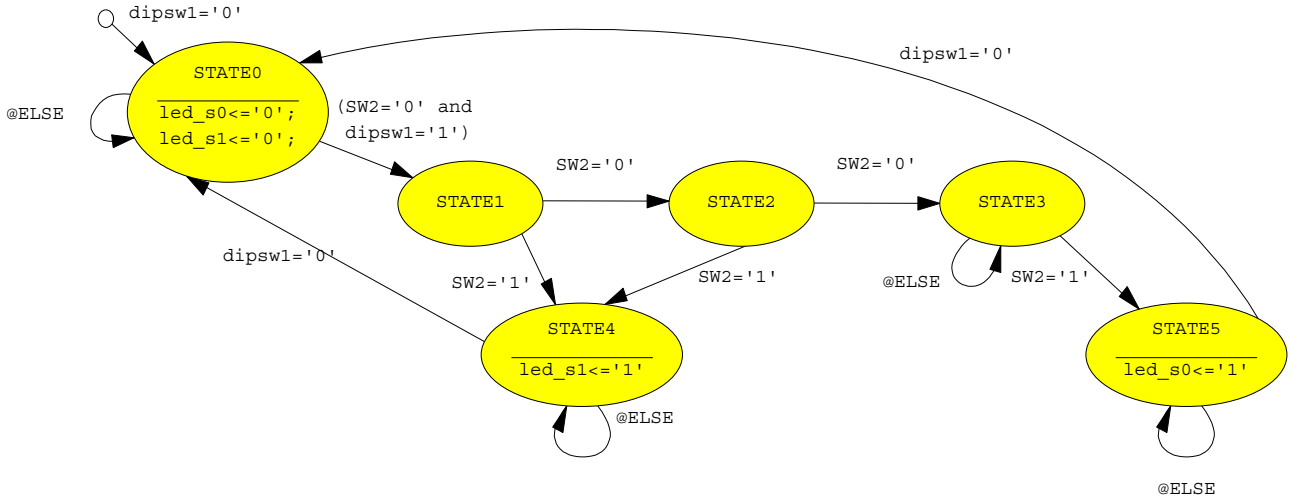
Realizzazione del circuito

Ritorniamo allo StateCad.
Il dispositivo deve riconoscere come 'click lungo', un click che duri più di 1 secondo. Gli altri saranno 'click breve'.
Il disegno che creeremo riceverà il clock dal dispositivo divisore di clock, quindi un clock a circa 3Hz. Bisogna fare in modo che ci siano 6 stati:

- uno stato di reset STATE0 in cui si attende che lo SW2 venga premuto e i led sono spenti
- uno stato STATE1 in cui si perviene nel momento in cui SW2 viene premuto (al primo colpo di clock non è ancora passato 1 secondo, quindi al momento del rilascio di SW2 si è in 'click breve')
- uno stato STATE2 in cui si perviene se SW2 è ancora premuto (SW2 è tenuto premuto per 2 colpi di clock, quindi non è ancora passato 1 secondo, pertanto al momento del rilascio di SW2 si è di nuovo in 'click breve')
- uno stato STATE3 in cui si perviene quando SW2 resta premuto per 3 colpi di clock, quindi per 1 secondo; al rilascio di SW2 si sarà in 'click lungo'
- lo stato STATE4 di 'click breve'

- lo stato STATE5 di ‘click lungo’

Il disegno è pertanto il seguente:



In questo caso per mantenere i LED illuminati si è optato per rimanere un tempo indeterminato nello stato 4 o nello stato 5, (ossia fino ad un eventuale segnale di reset). Si puo' pero' pensare ad una soluzione diversa che pur utilizzando gli stati 4 o 5 come stati di passaggio riescano a "congelare" l'ultimo valore di uscita sui LED.

Osservazione: dal momento che si richiede di effettuare un reset per riavviare il dispositivo, non è necessario alcun contatore. Volendo ritornare allo stato di attesa STATE0 automaticamente, senza dover azionare alcun tasto, sarebbe sufficiente, ad esempio, introdurre un contatore negli stati STATE4 (‘click breve’) e STATE5 (‘click lungo’) che, giunto a un determinato numero piloti il ritorno allo stato di attesa STATE0 nel quale si andrebbero azzerate non solo le uscite, ma anche il contatore.

Non occorre per forza impiegare un contatore per usire dagli stati 4 o 5.

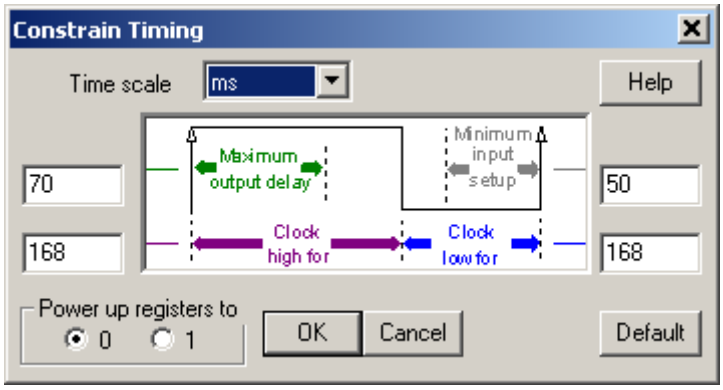
- Controlliamo le variabili (Options → Variable). Facciamo in modo che siano tutte ‘Active HIGH’ per verificare la corrispondenza 1→HIGH, 0 → LOW.

- Possiamo nuovamente procedere con la simulazione sullo StateBench per verificare se il disegno funziona.
  Aggiungiamo un particolare rispetto alla precedente simulazione. Aggiungiamo dei vincoli temporali, premendo il pulsante 

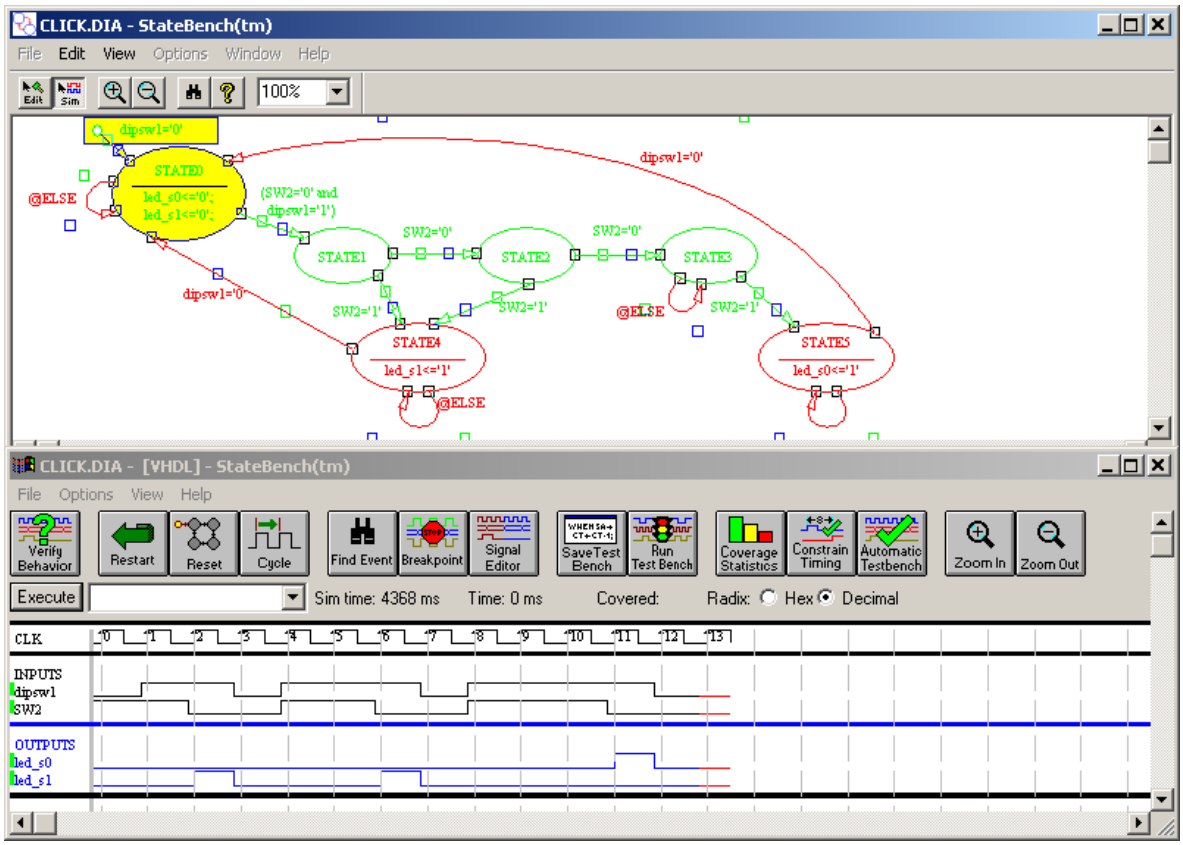
.

  A livello di State CAD la simulazione avviene a livello comportamentale, pertanto la definizione di una frequenza che si avvicini a quella reale di funzionamento risulta abbastanza inutile

  Partendo da un clock di macchina di 48.7 kHz, un ciclo di clock all’uscita del divisore di clock dura 1:48700*16383 = 0,33640657 secondi. Quindi il clock vale 1 per 0,33640657/2=0,168203285 secondi e zero per altrettanti secondi. Approssimiamo in millisecondi, cioè un semiciclo dura 168ms. Pertanto compiliamo come segue:



Confermiamo con 'OK' e proviamo la seguente successione:
STATE0 → STATE1 → STATE4 → STATE0 → STATE1 → STATE2 → STATE4 → STATE0 → STATE1 → STATE2 → STATE3 → STATE5 → STATE0



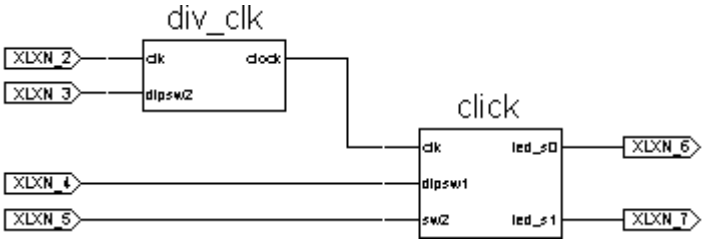
La simulazione con lo StateBench ha confermato il corretto funzionamento del disegno.
Salviamo il file 'CLICK.dia'.

- Procediamo ora con la generazione del codice VHDL nel modo visto per il divisore di clock: premere il pulsante 'Generate HDL'.

- Ritorniamo al Project Navigator e aggiungiamo al progetto il file sorgente 'CLICK.vhd' (Project → Add Source) che è un VHDL Module.

- Il disegno finale che unisce i due dispositivi darà uno schematico, pertanto, per ciascuno dei due disegni creiamo il simbolo ('Create Schematic Symbol') che utilizzeremo nello schematico finale.

- Aggiungiamo al progetto un nuovo file sorgente di tipo schematico (Project → New Source → Schematic) che chiameremo 'click2_finale'.
  Lo schema è il seguente:



- Occupiamoci dei vincoli per i collegamenti. A tale scopo aggiungiamo un nuovo file sorgente che sarà un 'Implementation Constraints File' dal nome 'click2_ucf', per esempio. Il file andrà associato allo schematico finale 'click2_finale'. Procediamo facendo doppio click sul file 'click2_ucf.ucf' nel menu a tendina 'Model View' per aprire il Constraints Editor. Le location sono le seguenti:

Port Name	Port Direction	Location	Pad to Setup	Clock to Pad
xlxn_2	INPUT	p88	N/A	N/A
xlxn_3	INPUT	p64		N/A
xlxn_4	INPUT	p54		N/A
xlxn_5	INPUT	p93		N/A
xlxn_6	OUTPUT	p67	N/A	
xlxn_7	OUTPUT	p39	N/A	

- Non resta che procedere con la verifica di quanto progettato, poi la sintesi, l'implementazione e la generazione dei file di programmazione della Board.