

Esercitazione n° 5

Nella seguente esercitazione, che si propone di mettere in pratica quanto finora imparato, la soluzione non sarà curata nei dettagli come le precedenti, dal momento che, a questo punto, si ritiene lo studente in grado di effettuare autonomamente le operazioni fondamentali.

Problema

Creare un dispositivo che, premendo con un click corto il PUSHBUTTON sulla XSA Board avvii un contatore (a 3 bit) sul display a 7 segmenti. Il contatore visualizzerà in sequenza, a partire da 0, i numeri 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3... finchè non verrà premuto il PUSHBUTTON con un click breve: quindi il contatore si arresterà. A quel punto, se si premerà il PUSHBUTTON con un click lungo il contatore si riavvierà, attendendo un click breve per ripartire; se si premerà con un click breve, il contatore riprenderà a contare da dove si era fermato.

- Valgono le seguenti specifiche:

$\sim 0.35s < \text{'click breve'} < \sim 1s$

$\text{'click lungo'} > \sim 1s$

Suggerimenti

Si sfrutti quanto fatto nelle precedenti esercitazioni, eventualmente con qualche piccola modifica.

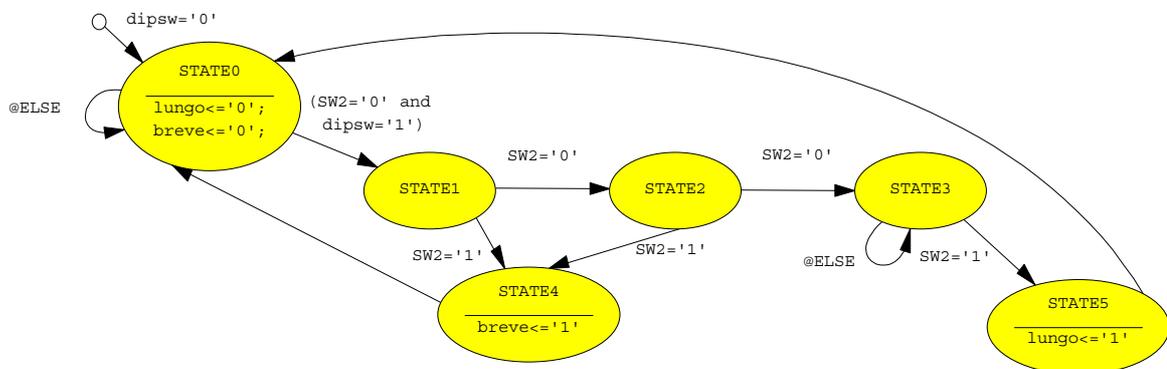
Soluzione

- Si inizi come sempre creando un nuovo progetto dal 'Project Navigator' dal nome 'contatore' ad esempio.
- Sarà quindi necessario aggiungere i seguenti file sorgenti creati nelle precedenti esercitazioni (Project → Add Source):

div_clk.vhd derivato dalla FSM con cui si è creato il divisore di clock
 dec7segm3bit.vhd relativo al decoder per il display a 7 segmenti

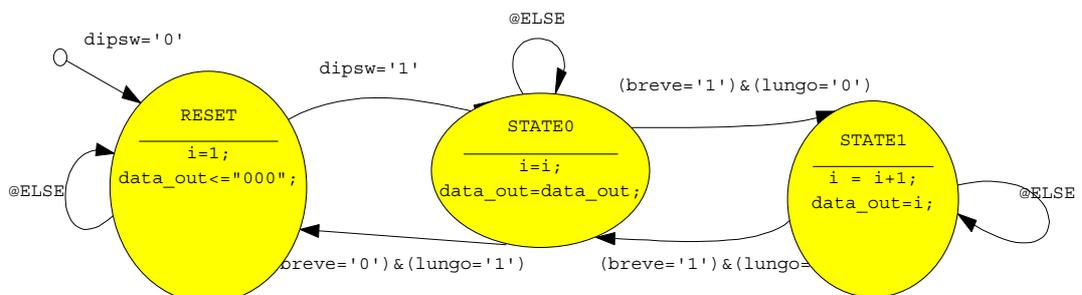
Si creino i simboli schematici ('Create Schematic Symbol' dalla finestra Process View) per ciascuno dei due file sorgenti.

- Ora riutilizzeremo il file 'click.dia' creato con lo StateCAD, ma sarà necessario apportare qualche piccola modifica:



Si noti che si sono levate le condizioni nelle transizioni che portano dallo STATE4 allo STATE0 e dallo STATE5 allo STATE0; questo perché è necessario che il dispositivo che riconosce il tipo di click si riporti nella condizione iniziale immediatamente. Si salvi il file come 'click_2.dia', ad esempio; si generi quindi il codice VHDL (premendo su tasto 'Generate HDL') e si aggiunga il file 'click_2.vhd' come file sorgente al progetto. Si crei quindi il simbolo ('Create Schematic Symbol') per lo schematico. Ora è necessario creare la macchina che, in base alla sequenza di click ricevuta, conti, si arresti o si azzeri.

- Torniamo allo StateCAD e realizziamo la seguente FSM:



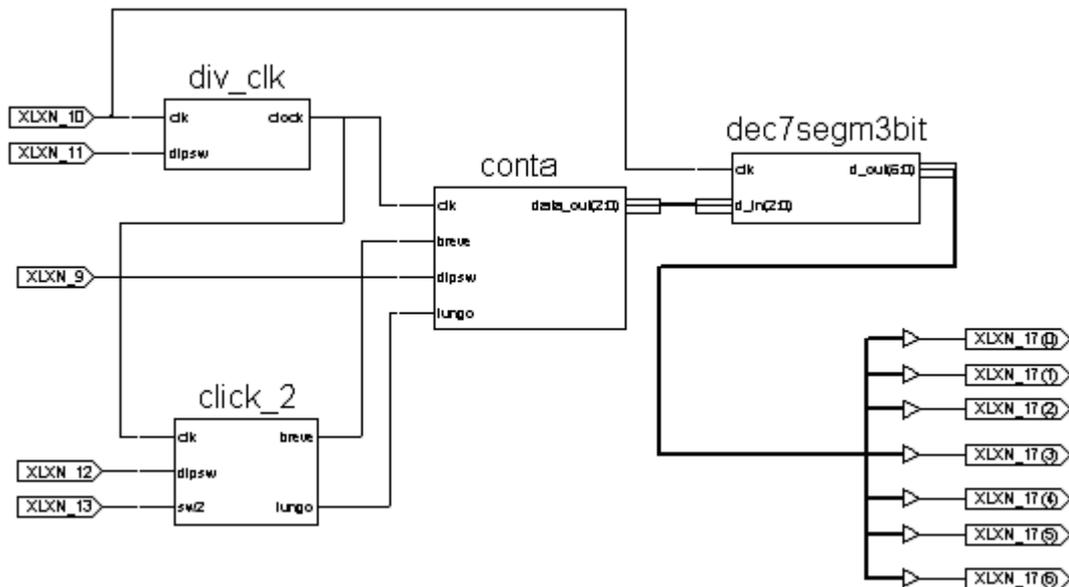
Salviamo il file come 'conta.dia'. Tutte le variabili (Options → Variable) dovranno essere Active:HIGH. Il vettore data_out non è del tipo NODE ma del tipo PIN. Attenzione alla fase

di ottimizzazione durante la generazione del codice VHDL. Generiamo il codice VHDL. Aggiungiamo il file 'conta.vhd' come file sorgente al progetto. Generiamo il simbolo per lo schematico finale.

Se lo si desidera si utilizzi StateBench per simulare il funzionamento.

A questo punto resta solo da realizzare lo schematico finale.

- Si crei un nuovo file sorgente di tipo schematico (Project→New Source→Schematic) dal nome 'contatore_fin', ad esempio.
Dall'ECS creare il seguente schema:



Ora manca solo constraints file.

- Si crei un nuovo file sorgente per i constraints (Project → New Source → Implementation Constraints File) e lo si chiami 'contatore_ucf' ad esempio.
Dalla finestra del Constraints Editor (per aprirla basta fare doppio click sul file 'contatore_ucf.ucf' nella finestra Module View) si compili il menu Ports al solito modo. Il file generato avrà il seguente listato:

```
NET "xlxn_9" LOC = "p64";
NET "xlxn_10" LOC = "p88";
NET "xlxn_11" LOC = "p54";
NET "xlxn_12" LOC = "p63";
NET "xlxn_13" LOC = "p93";
NET "xlxn_17<0>" LOC = "p67";
NET "xlxn_17<1>" LOC = "p39";
NET "xlxn_17<2>" LOC = "p62";
NET "xlxn_17<3>" LOC = "p60";
NET "xlxn_17<4>" LOC = "p46";
NET "xlxn_17<5>" LOC = "p57";
NET "xlxn_17<6>" LOC = "p49";
```

- E ora possibile procedere con le successive fasi di sintesi, implementazione e generazione del file di programmazione della scheda.