

# TUTORIAL 3

Realizzazione di un contatore su scheda XSA50

# 1 Problema proposto

In questo tutorial ci si propone di realizzare un contatore che, associato ad un display a sette segmenti, visualizzi il numero di volte che si è premuto un tasto; ovvero ogni volta che il tasto sulla board viene premuto, il contatore incrementa di una unità e questo viene visualizzato sul display.

Si deve realizzare quindi uno schema che riunisca un contatore a 4 bit ed un decoder atto a convertire l'uscita binaria del contatore in un codice opportuno per accendere la corretta serie di LED in modo da ottenere una "visualizzazione grafica" del numero in esame. L'ingresso del contatore sarà pilotato direttamente dall'utente.

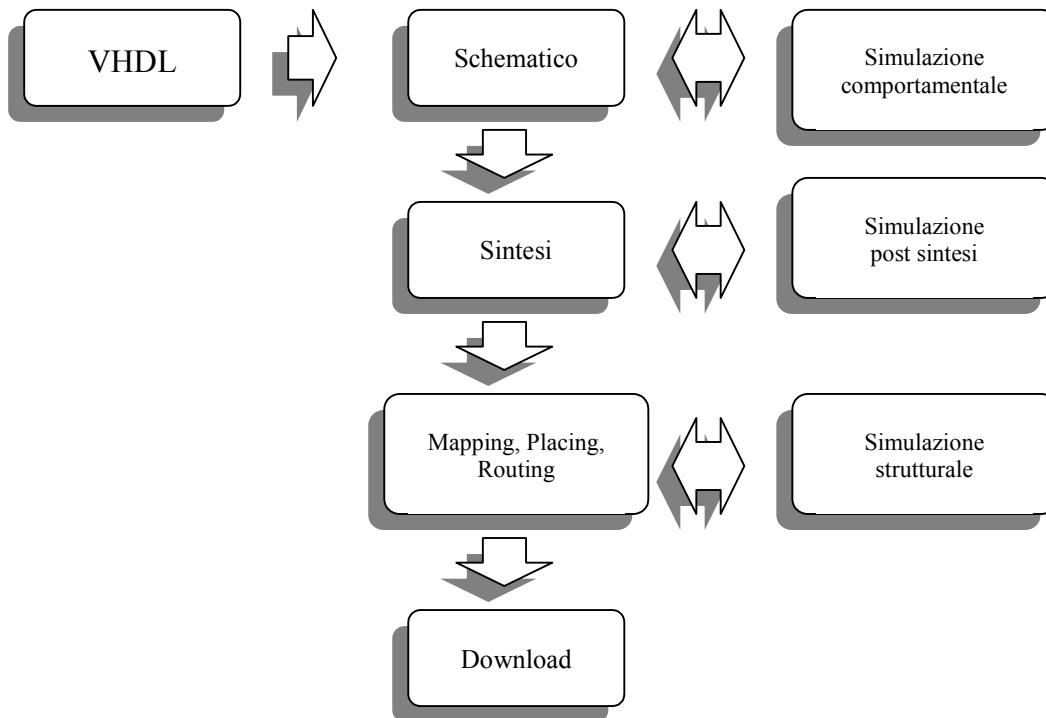
Il decoder può essere progettato utilizzando il linguaggio VHDL e, quindi è necessaria una piccola conoscenza di base, anche se il tool di sviluppo della Xilinx viene in aiuto con dei wizard che, data una descrizione di un blocco in termini di ingressi ed uscite, ricostruisce lo scheletro in VHDL, che dev'essere completato assegnandone i valori opportuni.

Si seguiranno quindi i seguenti passi:

1. Si realizzi il decoder in VHDL;
2. Si generi il "simbolo" associato;
3. Si realizzi lo schematico del progetto completo;
4. Si crei un segnale di test;
5. Si effettui una simulazione comportamentale;
6. Si proceda con l'operazione di sintesi e successiva analisi del report;
7. Si passi quindi alla traduzione ed alla simulazione post sintesi;
8. Si crei il file di vincoli con il Constraints Editor;
9. Si effettuino le operazioni di mappatura e sua simulazione;
10. Si faccia il placing&routing e relativa simulazione;

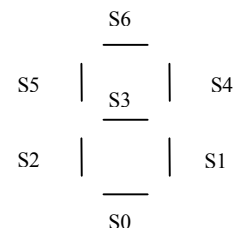
Naturalmente ad ogni simulazione, nel caso non si abbia il risultato aspettato si tornerà al passo precedente.

11. A completamento di tutto si esegua il download su scheda.



Per quanto riguarda i vincoli, si tengano presente che:

- il tasto sulla scheda è collegato al piedino 93 sull'FPGA;
- i sette segmenti del display S0,S1,S2,S3,S4,S5,S6 sono collegati rispettivamente ai piedini 67,39,62,60,46,57,49 sull'FPGA.



ed inoltre è utile sapere che il tasto è attivo nello stato basso, mentre i led sono attivi nello stato alto.

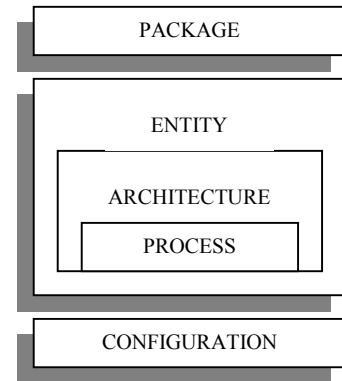
Per la realizzazione di questo progetto i requisiti richiesti sono

- il software ISE 5.x della Xilinx;
- il simulatore Modelsim;
- il software della XESS;
- la board XSA50;
- il cavo di download;
- l'alimentatore;

## 2 Cenni di VHDL

Il VHDL (VHSIC Hardware Description Language) è un linguaggio di descrizione hardware. Fra i componenti principali del VHDL ci sono le così dette library units o design units, esse sono costituite da uno o più elementi tra i quattro elencati di seguito:

- Package: insieme di definizioni che i vari elementi condividono;
- Entity: contiene la definizione degli ingressi e delle uscite;
- Architecture: è la descrizione interna di funzionamento del blocco;
- Configuration: serve ad associare una particolare architettura ad una entity;



in cui il primo e l'ultimo sono opzionali.

Uno schema tipico è presentato qui a fianco.

Una entity contiene appunto la dichiarazione delle porte di ingresso ed uscita e la sua sintassi è del tipo

**entity nome entity is**

Port (nome porta : in tipo di porta;  
nome di porta : out tipo di porta);

**end nome entity;**

La architecture è forse il blocco più importante: essa è composta da una sezione di dichiarazione e da un corpo che può contenere sia elementi concorrenti sia elementi sequenziali (quest'ultimi iniziano solitamente con la parola chiave *process*).

La sintassi di una architecture è del tipo

**architecture nome architecture of nome blocco is**

dichiarazioni

**begin**

corpo

**end nome architecture;**

La sintassi di un process è del tipo

**process begin**

corpo



Si confermi cliccando su *Avanti* > , si controlli il riepilogo e poi si clicchi su *Fine*. Viene quindi aggiunto al progetto un file col nome *Decoder4bit.vhd* che contiene lo scheletro in VHDL di quello che poi sarà il decoder; questo contiene una descrizione dell'entità ed una definizione di architettura a cui però manca il corpo centrale, ovvero il funzionamento del blocco. Si completi quindi il listato nel modo seguente:

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

-- Uncomment the following lines to use the declarations that are
-- provided for instantiating Xilinx primitive components.
-- library UNISIM;
-- use UNISIM.VComponents.all;

entity decoder4bit is
  Port ( DecoderIn : in std_logic_vector(3 downto 0);
        DecoderOut : out std_logic_vector(6 downto 0));
end decoder4bit;

architecture Behavioral of decoder4bit is

begin
  DecoderOut <= "1110111" when DecoderIn="0000" else
    "0010010" when DecoderIn="0001" else
    "1011101" when DecoderIn="0010" else
    "1011011" when DecoderIn="0011" else
    "0111010" when DecoderIn="0100" else
    "1101011" when DecoderIn="0101" else
    "1101111" when DecoderIn="0110" else
    "1010010" when DecoderIn="0111" else
    "1111111" when DecoderIn="1000" else
    "1111011" when DecoderIn="1001" else
    "1111110" when DecoderIn="1010" else
    "0101111" when DecoderIn="1011" else
    "1100101" when DecoderIn="1100" else
    "0011111" when DecoderIn="1101" else
    "1101101" when DecoderIn="1110" else
    "1101100" when DecoderIn="1111" else
    "0001000";

end Behavioral;
```

in cui si è associato a ciascun numero in ingresso la sua corrispondente visualizzazione sul display a sette segmenti utilizzando la notazione esadecimale. Si è impiegata anche una visualizzazione di default per i casi inattesi, data da un trattino orizzontale al centro del display.

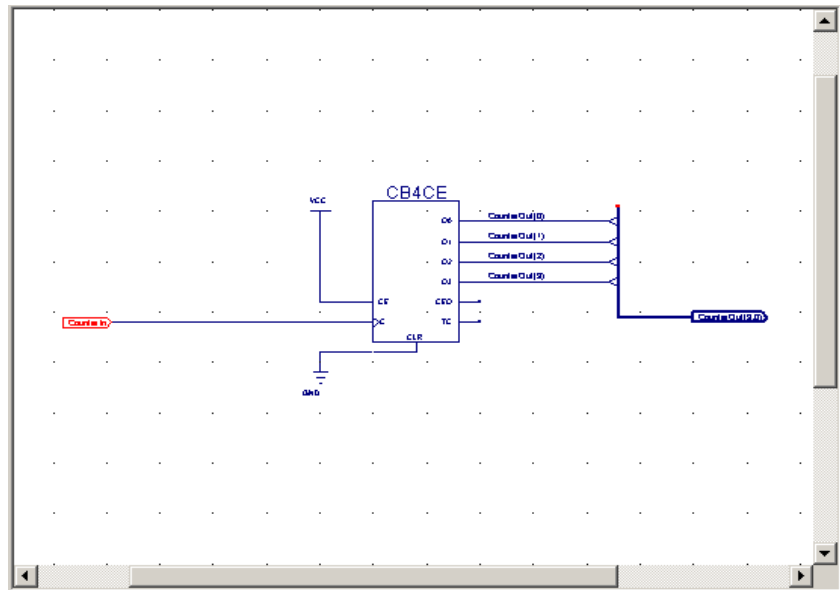
Una volta terminato si salvi dal Project Navigator.

Si crei inoltre lo schematic symbol, che verrà poi utilizzato nel progetto globale.

### 3.2 Realizzazione del progetto

Si aggiunga un nuovo schematico al progetto: esso dovrà contenere l'oggetto contatore.

Con la metodologia già vista, si realizzi il contatore, impiegando il simbolo *CB4CE*, collegando il clear a massa ed il chip enable all'alimentazione e riunendo i quattro fili in uscita sotto un unico bus:

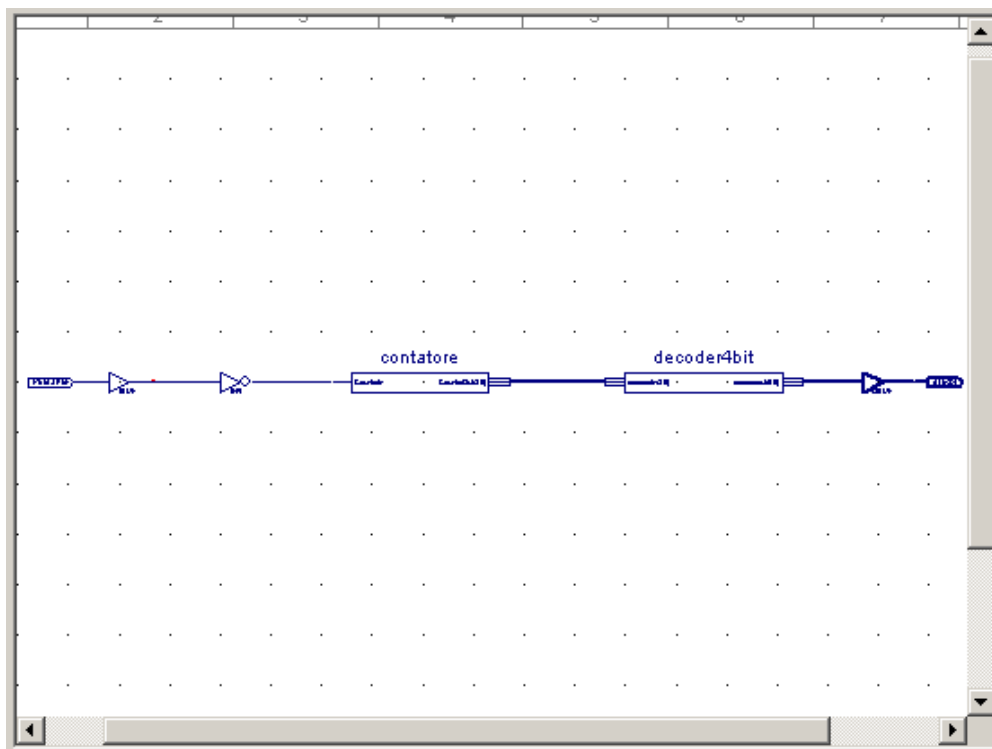


Si operi la verifica degli errori e si salvi il tutto.

Si ritorni al Project Navigator e si crei lo schematic symbol.

A questo punto si aggiunga un nuovo schematico al progetto; qui si collegherà il contatore al decoder, in modo che le uscite del contatore coincidano con gli ingressi del decoder.

Inoltre visto che si dovrà contare il numero di volte che viene premuto il tasto e che il contatore incrementa ad ogni ciclo del suo clock in ingresso, il tasto ed il clock coincideranno (Nello schematico bisognerà inoltre ricordarsi che il pulsante è premuto quando si trova nello stato basso).



Si lanci ora il processo di sintesi con le opzioni desiderate: quelle di default comunque vanno bene. Si valuti se la frequenza di lavoro è appropriata.

A questo punto si possono aggiungere i vincoli: si crei quindi un file di constraints nel modo già visto per quest'ultimo schematico, associando correttamente ciascun filo del bus in uscita dal decoder a ciascun segmento del display. Esso, nella forma testuale, simile al seguente

```
NET "pulsante" LOC = "P93";  
NET "s<0>" LOC = "P67";  
NET "s<1>" LOC = "P39";  
NET "s<2>" LOC = "P62";  
NET "s<3>" LOC = "P60";  
NET "s<4>" LOC = "P46";  
NET "s<5>" LOC = "P57";  
NET "s<6>" LOC = "P49";
```

Infine si lanci l'Implement design.

In particolare nella fase di Place&Route si analizzi il report relativo in cui viene riportato se i vincoli imposti sono stati raggiunti o meno.

Sotto la voce Pad Report è possibile inoltre vedere i pin impiegati e loro localizzazione nell'fpga fisica.

Si generi il bitstream e successivamente si esegua il download su scheda.

N.B. Si ricordi che ad ogni passo è meglio valutare la relativa simulazione per verificare il corretto funzionamento del progetto, avendo creato precedentemente un'onda di test opportuna.

Si noti che quando si va a provare praticamente su scheda il progetto appena creato, il comportamento non è proprio quello aspettato; in quanto a volte può capitare che il conteggio non sia proprio sequenziale, ma succeda che vengano saltati dei numeri: questo è dovuto ad un fenomeno, noto con il nome di "rebounding". Può accadere infatti che nel passaggio da uno stato all'altro del pulsante vi siano dei "rimbalzi" nei contatti che verrebbero così rilevati come una serie di commutazioni multiple.

Questo fenomeno può essere evitato sia fisicamente (introducendo un opportuno condensatore), ma anche può essere corretto tenendone opportunamente conto in fase di progetto: si potrebbe, cioè, campionare il tasto con frequenza sufficientemente bassa, in modo che questi rimbalzi non vengano rilevati e l'uscita risulti pulita (si potrebbe, ad esempio, impiegare una appropriata macchina a stati finiti tra tasto e contatore).

