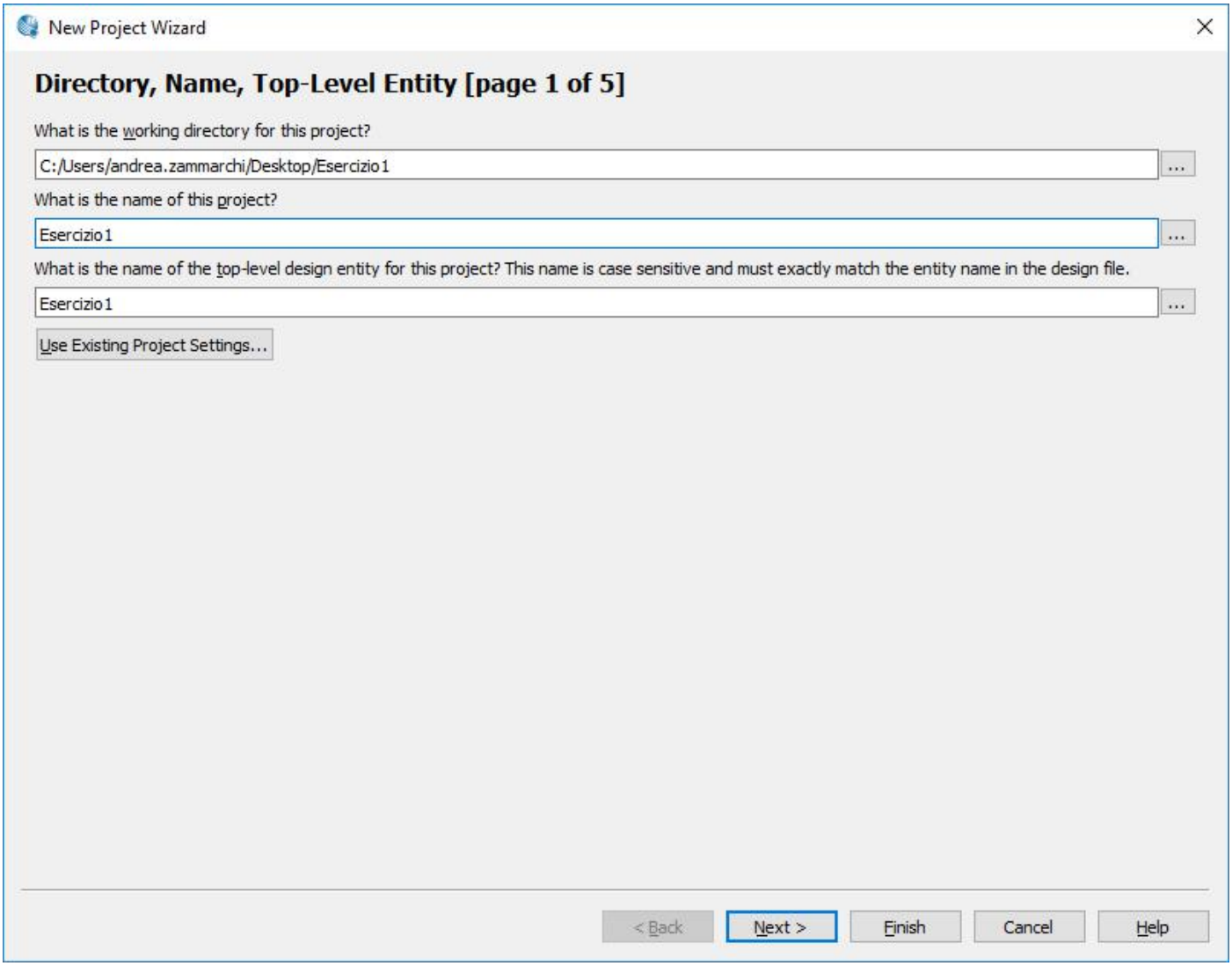
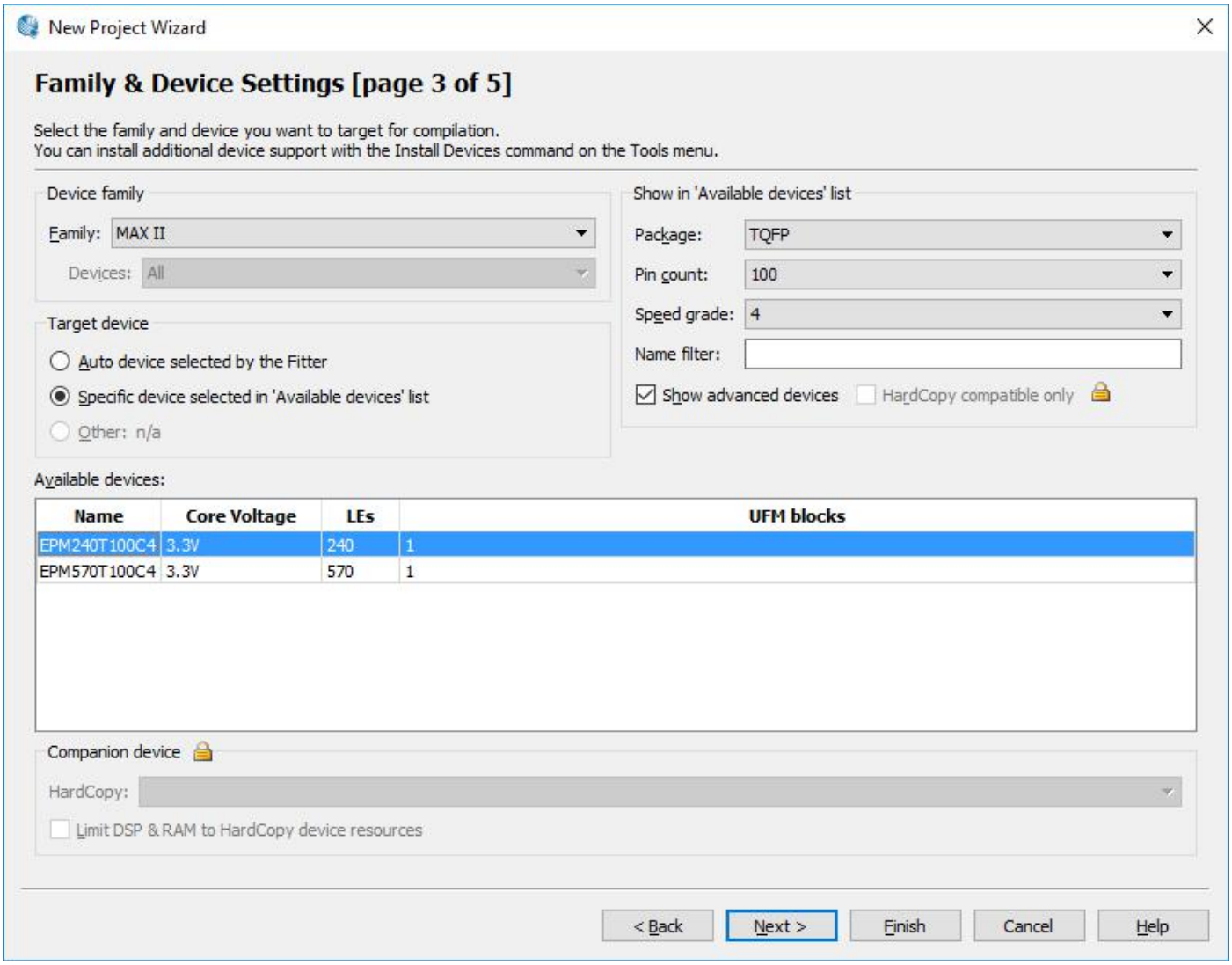
Ottimizzare un programma in Verilog

Ottimizzare un programma in verilog generato automaticamente da Quartus II.
Creare sul desktop due directory: “Esercizio1” e “Esercizio1sim”, dopodichè apriamo Quartus II e creaiamo un nuovo progetto cliccando in >> File > New Project Wizard...

Ci comparirà la seguente finestra dove nella prima TextBox (cliccando sui tre puntini a fianco) selezioneremo la cartella di progetto ovvero “Esercizio1” creata precedentemente sul desktop; nella seconda TextBox digitiamo il nome del nostro progetto che sarà sempre “Esercizio1”; la terza TextBox verrà compilata automaticamente.



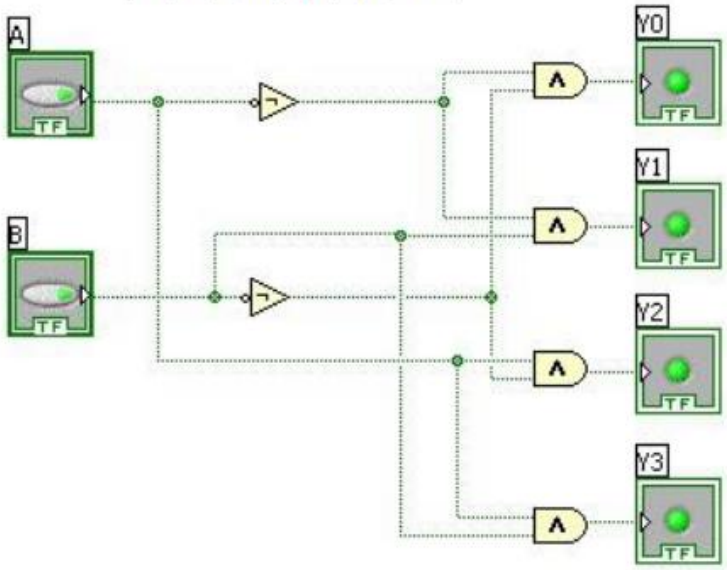
Clicchiamo quindi su “Next”.
Andremo alla seconda pagina che per il momento non ci interessa e clicchiamo nuovamente su “Next”.
Nella terza pagina dovremo selezionare il tipo di dispositivo sul quale verranno applicati i nostri circuiti:
-Nella TextBox “Family” selezioniamo **“MAX II”**;
-Nella TextBox “Package” selezioniamo **“TQFP”**;
-Nella TextBox “Pin count” selezioniamo **“100”**;
-Nella TextBox “Speed grade” selezioniamo **“4”**;
-Nel riquadro bianco sottostante selezioniamo il primo dispositivo, come nella figura.



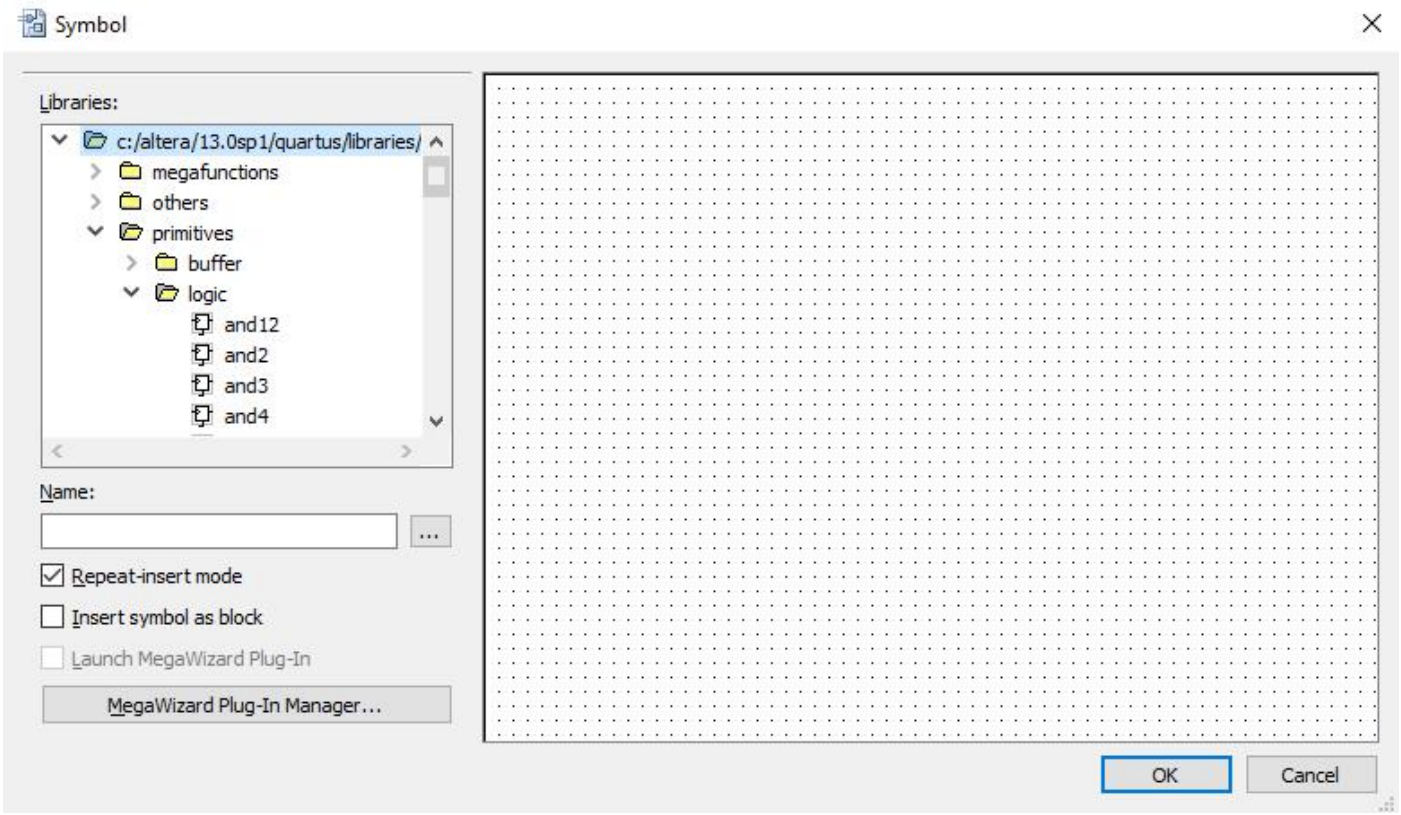
Clicchiamo quindi su “finish” perchè le pagine successive non ci interessano.
A questo punto clicchiamo su File > New... e selezioniamo Block diagram/Schematic file e pigiamo OK.
Adesso dovremmo creare un circuito, ad esempio ne creiamo uno con 6 porte logiche:

A	B	Y0	Y1	Y2	Y3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Y0 = /A /B
Y1 = /A B
Y2 = A /B
Y3 = A B

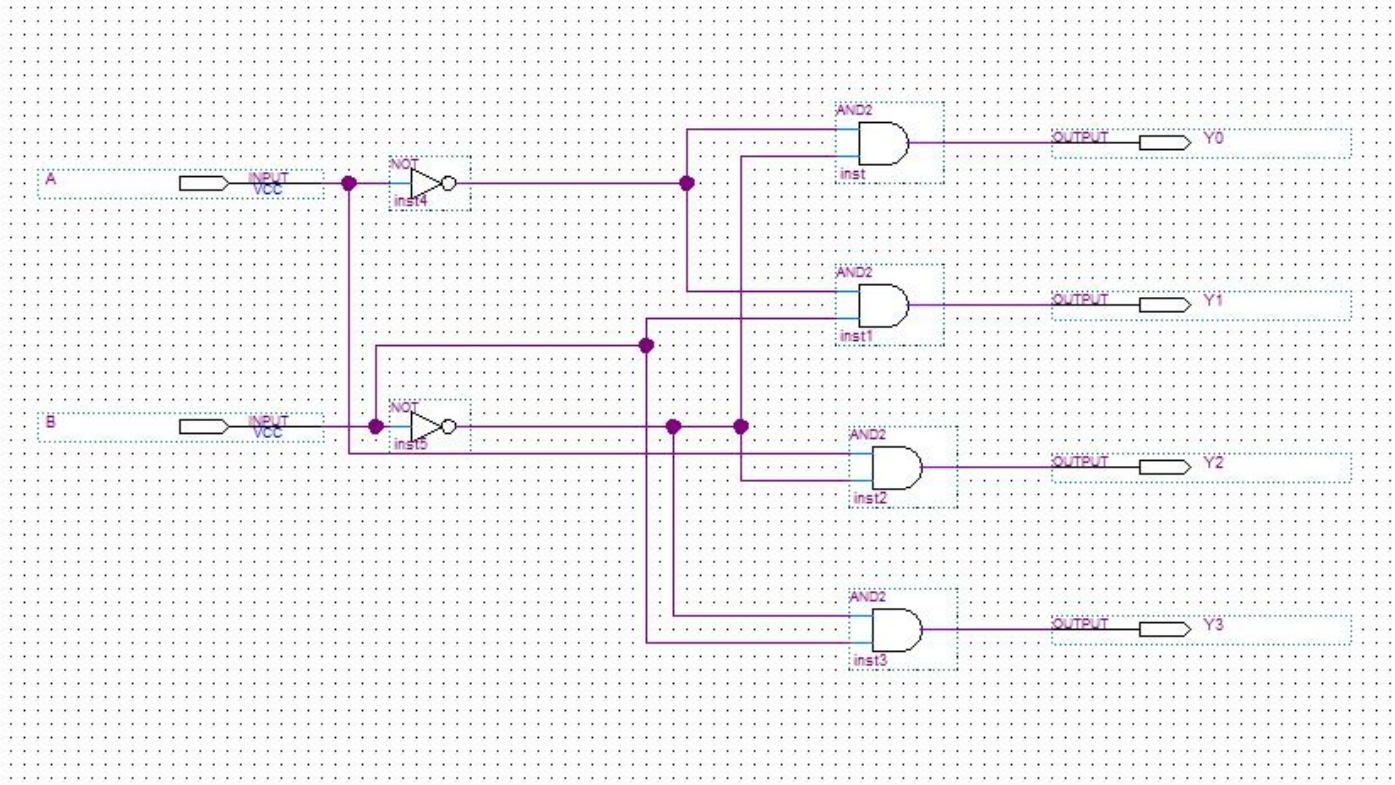


Per aggiungere delle porte logiche al nostro file clicchiamo su  e nella finestra che ci compare troveremo le porte logiche nel seguente percorso:



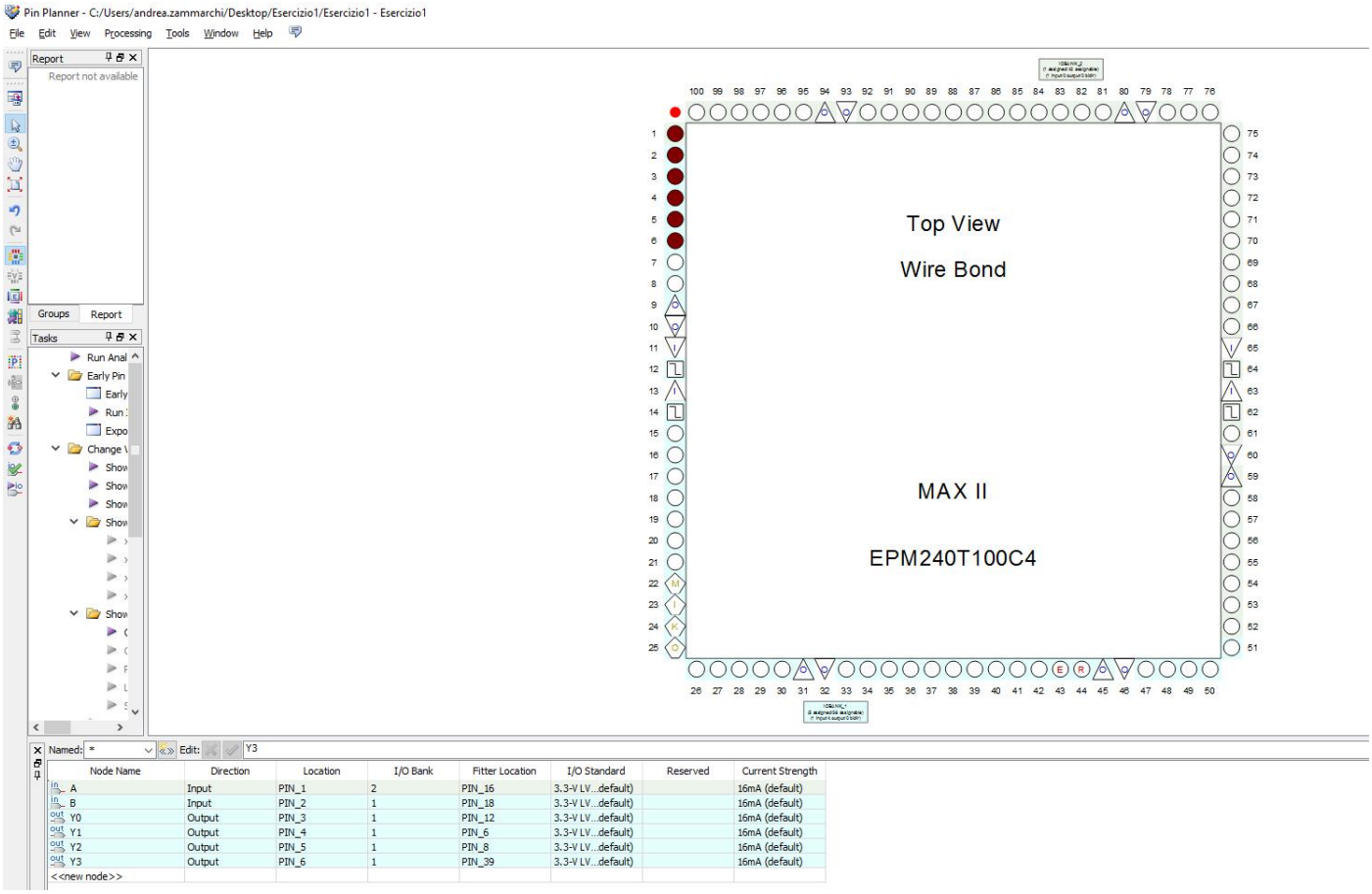
Prendiamo quindi quattro “and2” e due “not”.

Poi dobbiamo aggiungere due Ingressi e quattro Uscite, per farlo clicchiamo su:  .
Quindi creiamo i collegamenti come nell’immagine sopra riportata.
Per rinominare gli ingressi e le uscite basta fare doppio click nel nome violetto solitamente chiamato “pin_name1”, chiamiamo i due ingressi A e B e le quattro uscite Y0,Y1,Y2,Y3.
Alla fine il circuito sara’ questo:



Compilare il file cliccando su , ci chiederà di salvare e ovviamente lo salviamo nella cartella sul desktop “Esercizio1”, una volta compilato non ci dovranno essere errori, al massimo dei warnings che sono trascurabili.

Clicchiamo poi su e nella finestra che ci compare trasciniamo i pin dalla schermata in basso sui pin i primi 6 pin del MAX II (mi raccomando scegliere soltanto i pin tondi della nostra scheda!).



Ora possiamo chiudere la finestra del “Pin planner” e RiCompilare.

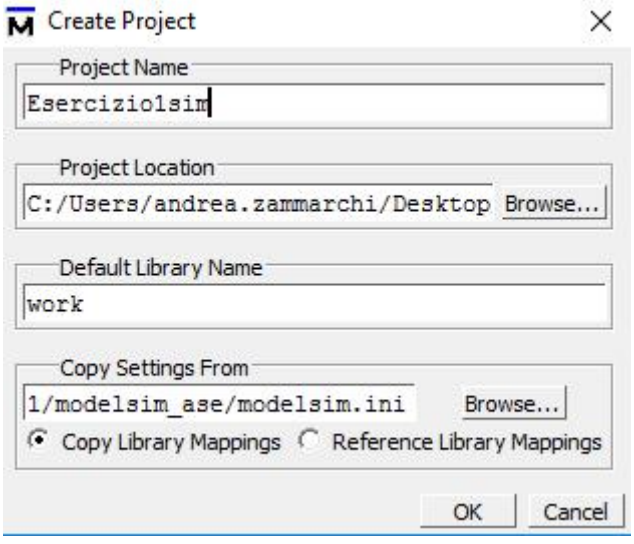
Creiamo il file verilog del nostro circuito cliccando su

File > Create/Update > Create HDL design file from current file...

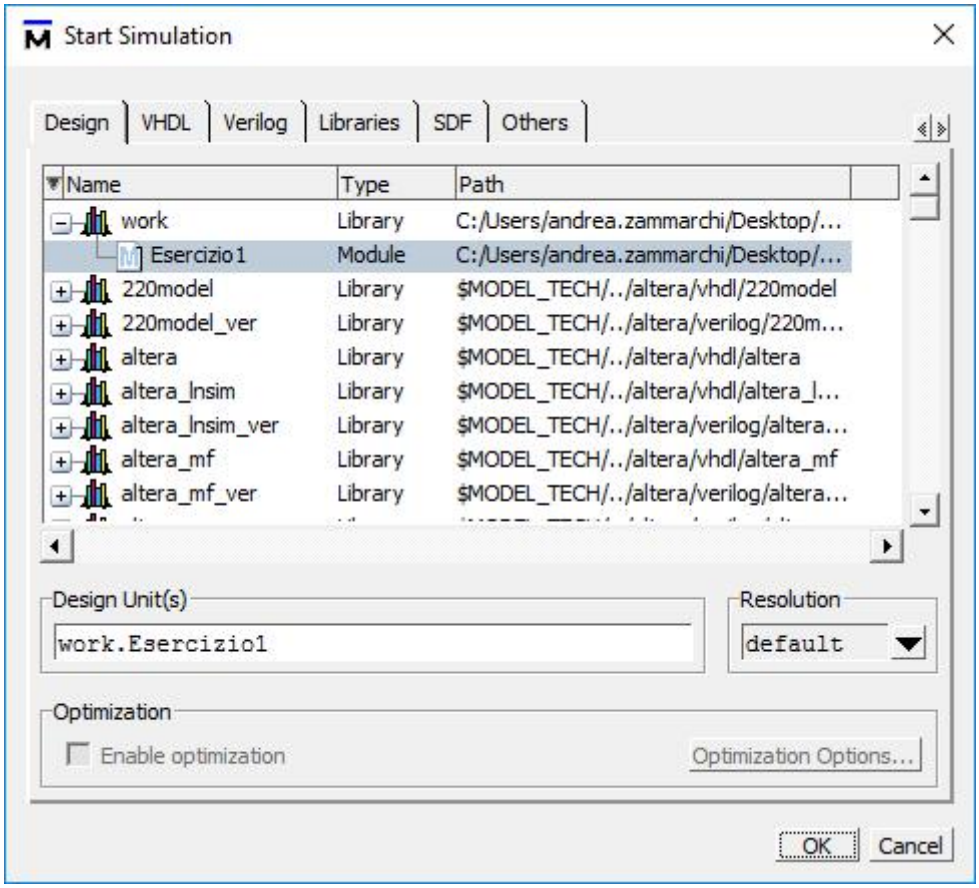
nella finestrina che ci compare clicchiamo “Verilog HDL” poi su OK.

Chiudere Quartus II e sul nostro desktop copiamo il file “Esercizio1.v” che è nella cartella “Esercizio1” e lo incolliamo nella cartella “Esercizio1sim”.

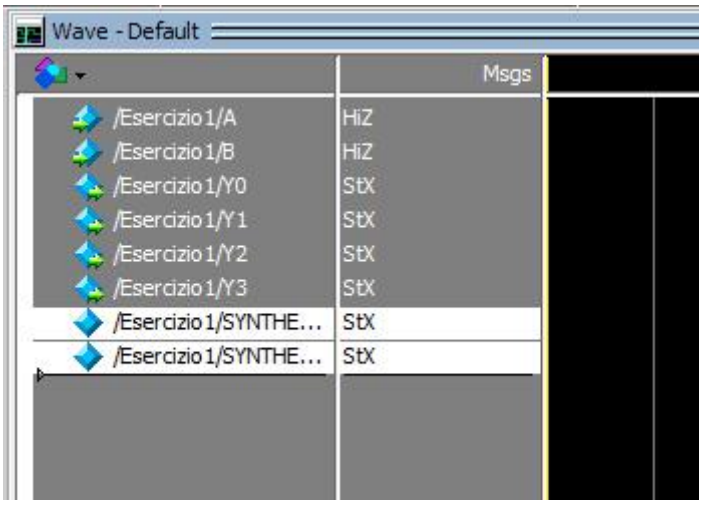
Apriamo ModelSim-Altera e clicchiamo su File > New > Project... e nella finestra che ci compare completiamo le TextBox nel seguente modo:



Nella seconda TextBox selezioniamo la nostra cartella “Esercizio1sim” sul desktop.
Clicchiamo su OK e nella finestrina che ci compare clicchiamo su “Create Simulation”,
scrivere “Esercizio1sim” nella TextBox “Design Units” e cliccare su Save.
Clicchiamo poi su “Add existing file” e nella prima TextBox selezioniamo il nostro file “Esercizio1.v” nella cartella “Esercizio1sim” poi spuntiamo il RadioButton “Copy to project directory”, clicchiamo su OK poi su Close.
Nella schermata principale clicchiamo col tasto destro sul file “Esercizio1.v” e clicchiamo su
Compile > Compile Selected.
Poi Clicchiamo su Simulate > Start simulation e selezioniamo il file nel seguente percorso:

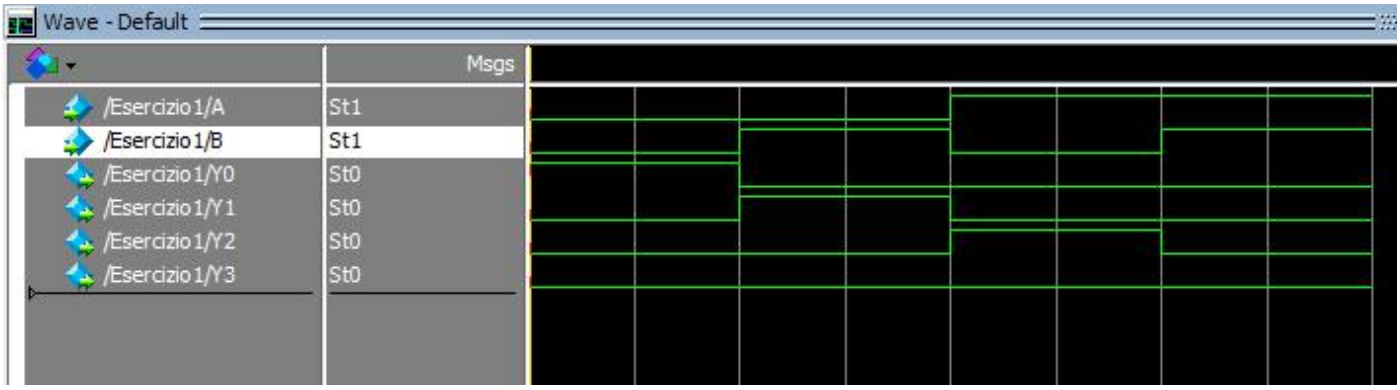


Clicchiamo su OK e selezioniamo la finestra Objects poi clicchiamo su Add > To Wave... > Signal in region.
Nella finestra Wave selezioniamo col tasto destro i seguenti elementi e cliccare su Edit > Delete.



Forzare i valori di A e B cliccando col tasto destro su A e poi su Force, ci apparirà una finestra e nella TextBox Value scriviamo 0.

Ripetiamo il procedimento per B, una volta fatto pigiamo il tasto F9 per fare la simulazione.
Poi forziamo:
-A=0 e B=1 e pigiamo F9;
-A=1 e B=0 e pigiamo F9;
-A=1 e B=1 e pigiamo F9;
Otterremo il seguente grafico:



Se il vostro grafico è uguale a questo allora la simulzaione è andata a buon fine.

Ora, apriamo il file Esercizio1.v sul NotePad++:

```
module Esercizio1(
	A,
	B,
	Y0,
	Y1,
	Y2,
	Y3
);

input wire	A;
input wire	B;
output wire	Y0;
output wire	Y1;
output wire	Y2;
output wire	Y3;

wire	SYNTHESIZED_WIRE_4;
wire	SYNTHESIZED_WIRE_5;

assign	SYNTHESIZED_WIRE_4 =	~A;
assign	SYNTHESIZED_WIRE_5 =	~B;
assign	Y0 = SYNTHESIZED_WIRE_4 & SYNTHESIZED_WIRE_5;
assign	Y1 = SYNTHESIZED_WIRE_4 & B;
assign	Y2 = A & SYNTHESIZED_WIRE_5;
assign	Y3 = A & B;

endmodule
```

Questo è il file sorgente in verilog che Quartus II ha generato automaticamente tramite i collegamenti realizzati graficamente dall'utente.

Nonostante sia generato automaticamente non vuol dire che sia perfetto.
Per perfetto si intende ottimizzato e per ottimizzato si intende più breve possibile ma funzionante e allo stesso tempo facilmente comprensibile.
Per ottimizzare il file bisogna però conoscere il linguaggio verilog.
Brevemente alla programmazione in questo linguaggio.

VERILOG

In Verilog si possono definire componenti attraverso il costrutto **module**.
Un modulo è molto simile ad una dichiarazione di procedura:
- ha un nome;
- una lista di parametri formali (di ingresso o di uscita);
- un corpo, che definisce come i parametri di uscita vengono calcolati a partire dai parametri in ingresso.

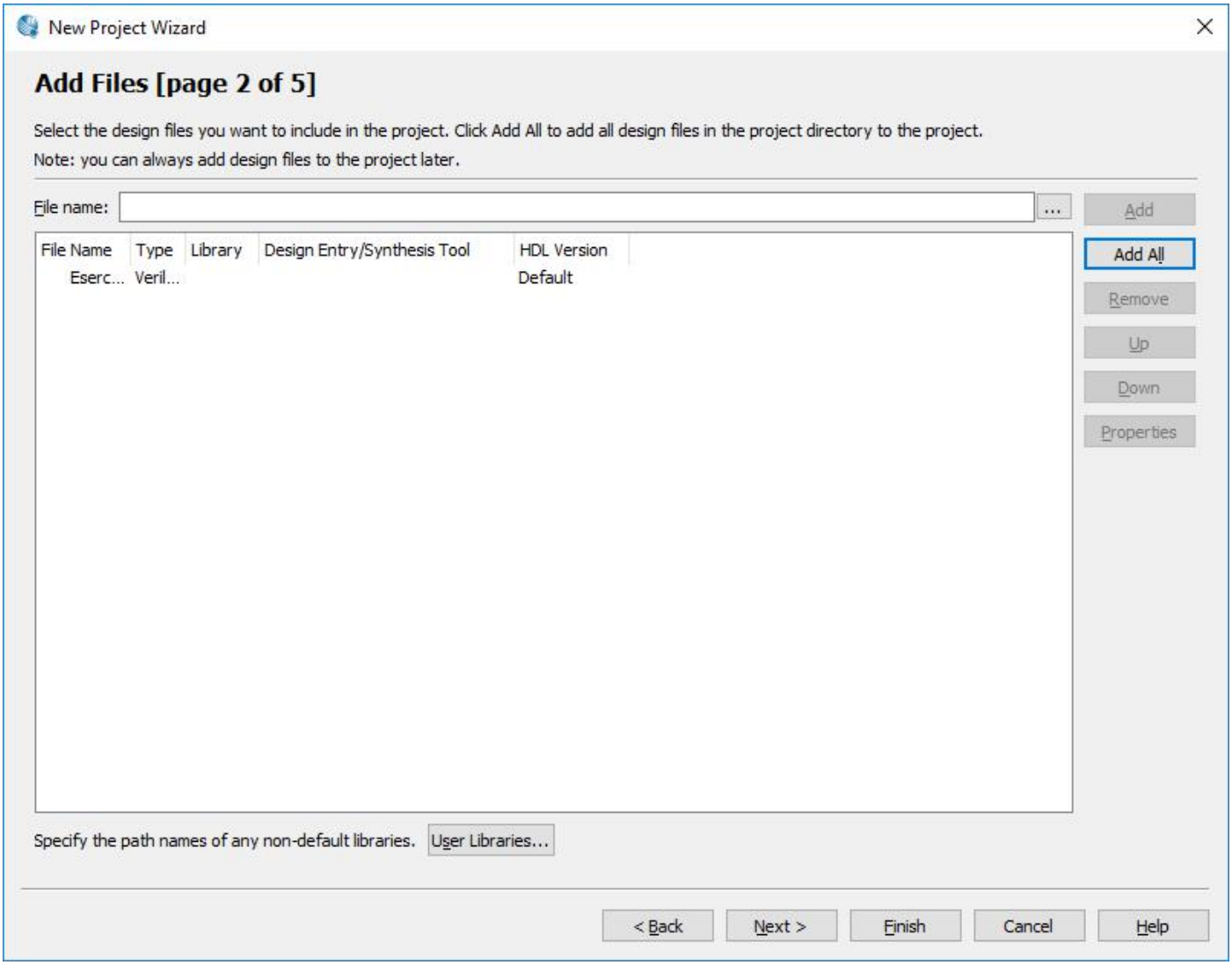
I **wire** sono fili ovvero collegamenti utilizzati per connettere componenti.
Un wire si puo' dichiarare mediante la parola chiave wire.
Le prime due dichiarazioni introducono 2 collegamenti di nome A B come entrate, in realta' sono dichiarazioni di ingressi e uscite. Le altre quattro dichiarazioni introducono 4 fili di nome Y0 Y1 Y2 Y3 come uscite. Ognuno realizza un collegamento da 1 bit.
Input wire = dichiarazione unilinea degli ingressi, **output wire** = dichiarazione unilinea delle uscite.
Assegnamento continuo (**assign**) la cui semantica e': assegna in continuazione il risultato della parte destra alla parte sinistra. Se varia un valore utilizzato nella parte destra, rivalutala e riassegnala alla parte sinistra.
La tilde prima di A e B indica che gli ingressi sono negati.
Viene assegnato ad A negato SYNTHESIZED_WIRE_4 e a B negato SYNTHESIZED_WIRE_5.
L'uscita Y0 è data dall'And logico di A negato e B negato.
L'uscita Y1 è data dall'And logico di A negato e B.
L'uscita Y2 è data dall'And logico di A e B negato.
L'uscita Y3 è data dall'And logico di A e B.
Con **endmodule** si chiude il modulo precedentemente iniziato (module Esercizio1).
A questo punto ottimizziamo il file:

```
module Esercizio1(A,B,Y0,Y1,Y2,Y3);

input wire    A,B;
output wire  Y0,Y1,Y2,Y3;

wire      SYNTHESIZED_WIRE_4,SYNTHESIZED_WIRE_5;

assign    SYNTHESIZED_WIRE_4=~A;
assign    SYNTHESIZED_WIRE_5=~B;
assign    Y0=SYNTHESIZED_WIRE_4 & SYNTHESIZED_WIRE_5;
assign    Y1=SYNTHESIZED_WIRE_4 & B;
assign    Y2=A & SYNTHESIZED_WIRE_5;
assign    Y3=A & B;

endmodule
```

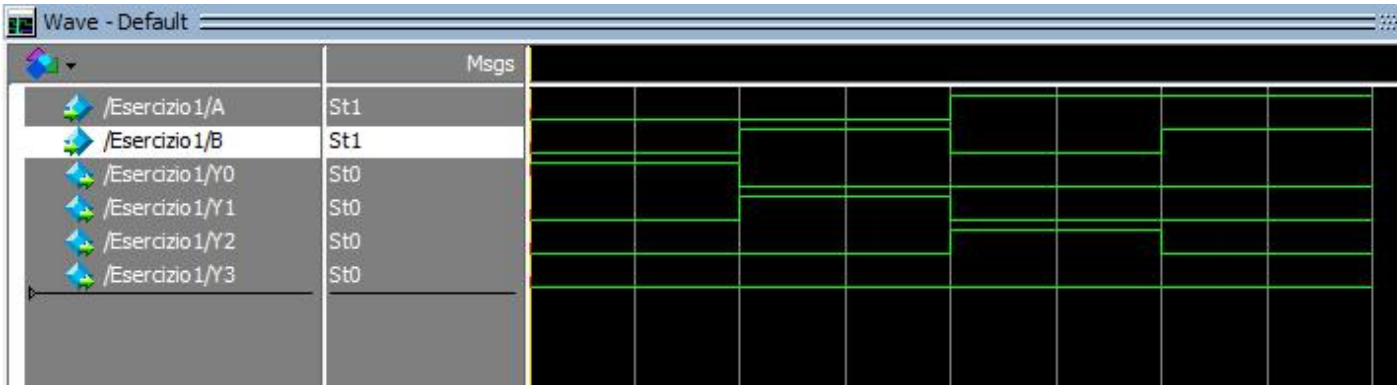
Perchè non abbiamo ottimizzato anche la funzione assign?
Questo perchè scriverlo in un'unica riga sarebbe diventato difficilmente comprensibile.

Adesso dobbiamo verificare di non aver commesso errori durante l'ottimizzazione del file.
Perciò creiamo altre due cartelle sul desktop e chiamiamole rispettivamente "Esercizio1V" e "Esercizio1Vsim".

In queste due cartelle copiamo il file che abbiamo appena ottimizzato.
Apriamo poi Quartus II e creiamo un nuovo progetto.
Ripetiamo i passaggi all'inizio ,ma questa volta nella seconda pagina dobbiamo aggiungere al progetto il nostro file sorgente.
Clicchiamo quindi sui tre puntini a fianco alla TextBox principale e selezioniamo il file sorgente nella cartella Esercizio1V e clicchiamo poi su Add.

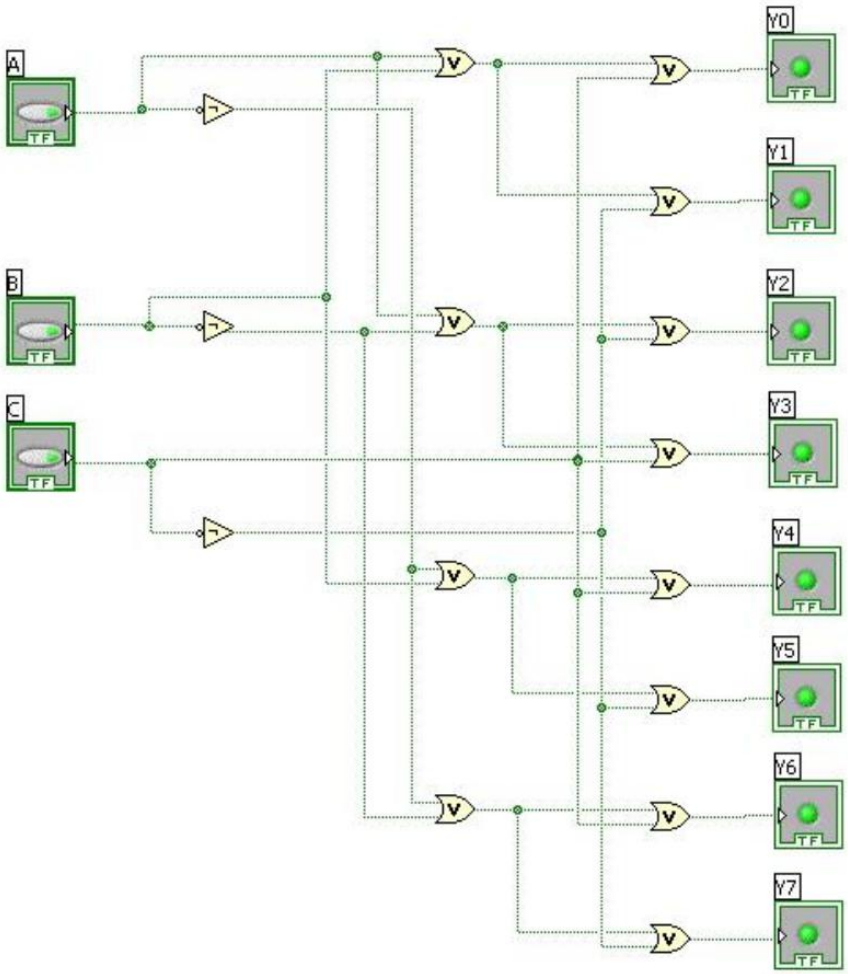


Completiamo le seguenti pagine come inizialmente riportato.
Nella finestrina "Project Navigator" selezioniamo la scheda "Files" e clicchiamo col tasto destro sul file sorgente e clicchiamo su "Set as top level entity" poi facciamo doppio click sul file stesso che si aprirà e compiliamolo, se ci darà tre errori, per risolverli, facciamo doppio click su "edit settings" nella finestra in basso a sinistra.
Nella scheda "General" e nella ComboBox "recently selected top level entity" selezioniamo "Esercizio1". Compiliamo nuovament.
Eseguiamo i passaggi del Pin Planner già precedentemente spiegati poi compiliamo nuovamente e chiudiamo Quarus II.
Apriamo ModelSim e ripetiamo gli stessi passi spiegati precedentemente selezionando la cartella "Esercizio1Vsim" anzichè "Esercizio1sim".
Se il grafico finale è uguale a questo allora siete diventati dei programmatori on verilog!!! ;D



Guida: Come creare un programma in Verilog:

Partiamo da uno schema gia fatto:



Partendo da questo schema creiamo il file verilog e scriviamo il programma

```
module Esercizio1(A,B,C,Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);

input wire    A,B,C;
output wire  Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7;

wire      SYNTHESIZED_WIRE_4,SYNTHESIZED_WIRE_5,SYNTHESIZED_WIRE_6;

assign    SYNTHESIZED_WIRE_4=~A;
```

```
assign   SYNTHESIZED_WIRE_5=~B;
assign   SYNTHESIZED_WIRE_6=~C;
assign   Y0=A | B | C;
assign   Y1=A | B | SYNTHESIZED_WIRE_6;
assign   Y2=A | SYNTHESIZED_WIRE_5 | C;
assign   Y3=A | SYNTHESIZED_WIRE_5 | SYNTHESIZED_WIRE_6;
assign   Y4=SYNTHESIZED_WIRE_4 | B | C;
assign   Y5=SYNTHESIZED_WIRE_4 | B | SYNTHESIZED_WIRE_6;
assign   Y6=SYNTHESIZED_WIRE_4 | SYNTHESIZED_WIRE_5 | C;
assign   Y7=SYNTHESIZED_WIRE_4 | SYNTHESIZED_WIRE_5 | SYNTHESIZED_WIRE_6;

Endmodule
```

Assegnamo quindi ad A negato SYNTHESIZED_WIRE_4, a B negato SYNTHESIZED_WIRE_5, a C negato SYNTHESIZED_WIRE_4.
Vediamo quindi dallo schema che le uscite si ottengono nella seguente maniera:

Y0 = A + B + C
Y1 = A + B + /C
Y2 = A + /B + C
Y3 = A + /B + /C
Y4 = /A + B + C
Y5 = /A + B + /C
Y6 = /A + /B + C
Y7 = /A + /B + /C

Una volta creato il file verilog su Quartus II compreso dentro un nuovo progetto, andiamo ad, effettuare la simulazione su ModelSim

Zammarchi Andrea 3C 2017