

FPGA Tutorial on DE1 Board

TUTORIAL 1

Using Quartus II V13.0

Marsi 2014 - University of TRIESTE

Tutorial FPGA n.1

Realizzazione di un semplice circuito logico su DE1

Descrizione: Si realizzi un semplice circuito logico su DE1 [1-2] con ingressi pilotati dagli switches ed uscita su LED.

Scopo: familiarizzare col tool di sviluppo, con la scheda di sviluppo DE1, apprendimento dei passi fondamentali da seguire per portare a termine un progetto completo.

Apprendimento previsto:

- Primi rudimenti per navigare all'interno del tool quartus
- Compilazione di un progetto descritto in forma schematica
- Compilazione di un progetto descritto tramite HDL (Verilog o VHDL)
- Imposizione di alcuni vincoli realizzativi (Pin position)
- Simulazione e debugging del circuito (a diversi livelli di astrazione)
- Visualizzazione e verifica delle caratteristiche temporali (ritardi di propagazione)
- Visualizzazione del layout del circuito
- Download e running del circuito su scheda DE1

Procedimento

1. Fase introduttiva

Si apra il tool quartus [3] tramite icona sul desktop oppure dall'elenco dei programmi installati

Si crei un nuovo progetto o cliccando sull'apposito tasto oppure tramite

File > New Project Wizard

Si definisca

- il direttorio nel quale salvare i files di sistema (non si usino spazi – per cui evitare ad esempio direttori quali “My Documents”)
- il nome del progetto, e definire (provvisoriamente)
- il nome del blocco più alto in graduatoria del progetto (top level design entity) – tipicamente questo prende il nome stesso del progetto e comunque può essere modificato in seguito.

(Next)

Ove si volessero importate files già scritti si può agire su questa pagina (in questo caso si può passare alla pagina successiva

(Next)

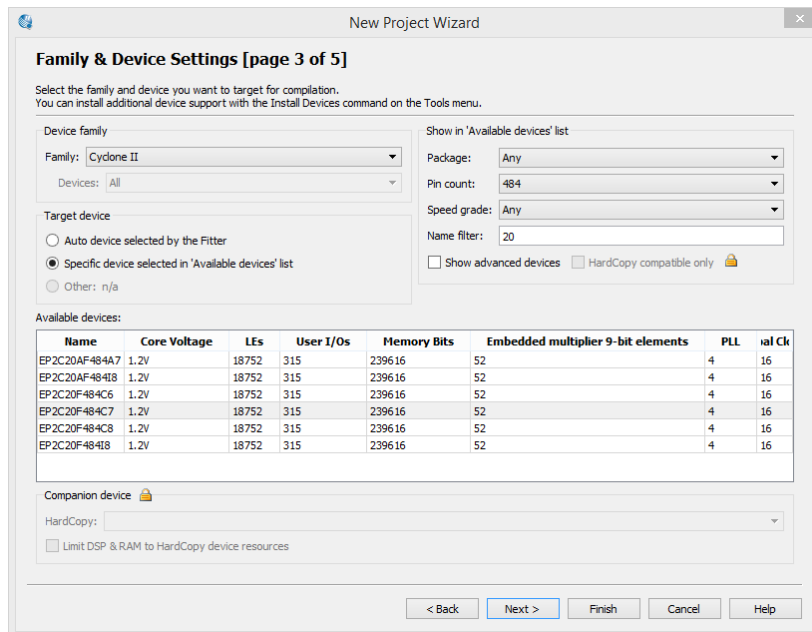
Si definisca il dispositivo (FPGA) per il quale sviluppare l'intero progetto. Ovviamente il progetto stesso, le risorse disponibili e la struttura stessa dipendono dal dispositivo utilizzato – esistono centinaia di FPGA diverse per famiglia, caratteristiche, prestazioni, piedinatura ecc. **E' FONDAMENTALE** scegliere l'FPGA

corretta [4-5] . Nel nostro caso sulla DE1 è montata una EP2C20F484C7N della famiglia Cyclone II pertanto sceglieremo la:

Altera - Cyclone II - EP2C20F484C7

(Si noti che il suffisso N indica solo che il dispositivo rispetta la normativa comunitaria 2002/95/CE la quale impone alcune restrizioni sull'uso di determinate sostanze pericolose nella costruzione di apparecchiature elettroniche).

In questa fase potrebbe essere comodo l'impiego di opportuni "filtri" per isolare velocemente il dispositivo di nostro interesse come ad esempio fissare il Pin Count a 484 e vincolare il fatto che la sigla del dispositivo debba contenere la stringa "20"



La tabella presente in questa pagina consente di paragonare le prestazioni dei vari dispositivi (celle logiche, memoria a disposizione, moltiplicatori integrati, PLL, risorse di I/O, linee di clock dedicato) ... questo potrebbe essere utile laddove non sia stato definito ancora il dispositivo sul quale realizzare il sistema.

(Next)

In questa pagina si possono fissare programmi di terze parti da usare per la sintesi, la simulazione e l'analisi temporale. In questo caso si definisca come tool per la simulazione ModelSim – Altera e come linguaggio da adottare Verilog HDL – disattivare "Run Gate-level simulation automatically after compilation" .

(Next)



Una pagina riassuntiva elenca le scelte fatte, verificarne la coerenza

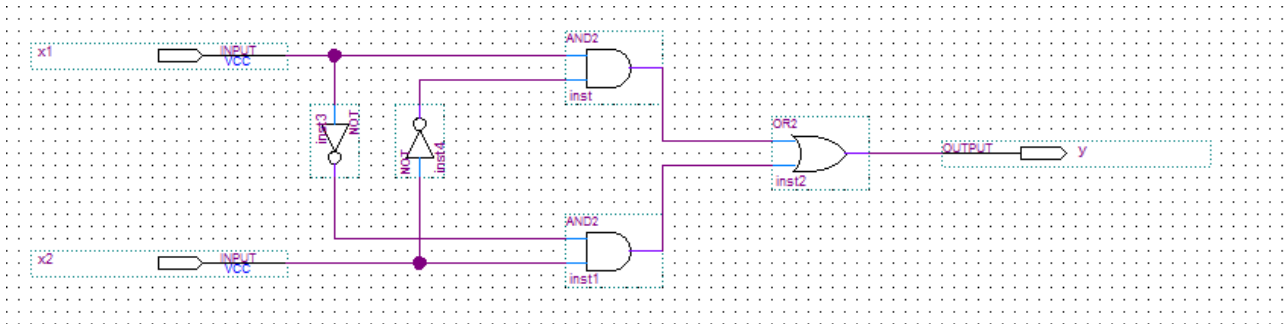
(Finish)

2. Definizione dell'architettura sistema (Tramite schematico) [5-6]

File > New
Block Diagram / Schematic file (OK)

Nella nuova pagina grafica che compare, tracciare lo schematico del sistema da realizzare simile a quanto riportato in figura.

Si impieghino i tasti  e  per accedere alla libreria delle porte logiche e per creare i vari collegamenti.



In particolare si suggerisce:

- di prendere visione delle diverse porte logiche e strutture hardware disponibili suddivise tra
Megafunctions: funzioni complesse ad alto livello di astrazione (verranno utilizzate in seguito)
Others: funzioni particolari legate ad una particolare famiglia di FPGA
Primitives: funzioni generiche
- di analizzare le varie opzioni grafiche disponibili (ad esempio quella di poter modificare la posizione dei gate mantenendo inalterati i collegamenti o viceversa spostare i gate senza i collegamenti)
- dare un nome mnemonico convenuto alle varie porte logiche e soprattutto ai pin di ingresso ed uscita.
NOTA BENE: utilizzare dei nomi abbastanza specifici ed evitare possibili parole "riservate" quali ad esempio: "in", "out", "inout", "and", "inv", ..., così come parole magari più improbabili come "for", "while", "goto" ecc. ecc.

Si salvi lo schematico e se ne verifichi la correttezza del circuito attraverso un processo di analisi.

Processing > Start > Analysis and Elaboration

Si noti che esistono eventualmente delle scorciatoie nella finestra dedicata ai Tasks.

Si correggano eventuali errori (un doppio click sul messaggio d'errore aiuta ad evidenziare la parte interessata)

3. Simulazione funzionale [7]

Creare un file di stimoli :

```
File > New  
University Program VWF (OK)
```

(si apre la finestra ove definire gli stimoli)

Definire quali segnali debbano essere monitorati:

```
Edit > Insert > Insert Node or Bus  
Clk on Node Finder  
Filters: Pin:all  
Cilk on List  
Cilk on >>  
OK  
OK
```

Definire la scala dei tempi

```
Edit > Set End Time
```

Configurare a 200 ns

OK

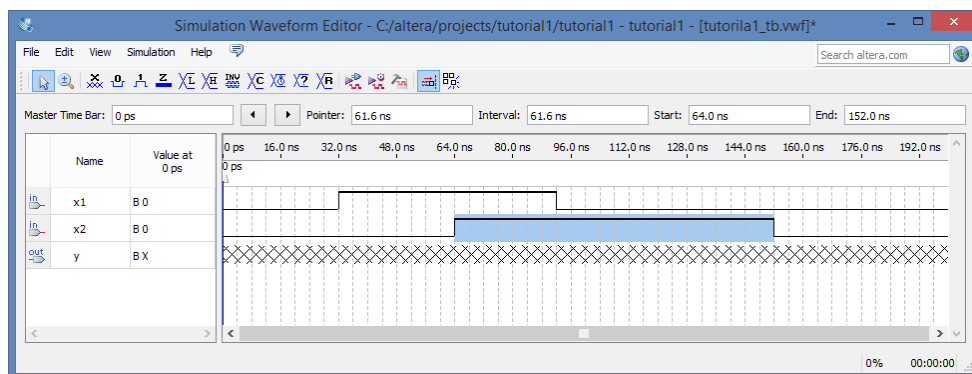
Definire lo la risoluzione adottata dalla scala temporale

```
Edit > Grid Size
```

Configurare a 1 ns

OK

Utilizzando il tool grafico a disposizione disegnare le f.d.o. degli stimoli desiderati



Salvare il file di stimoli

File > Save

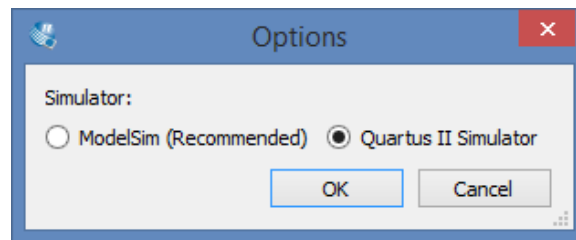
Dare un nome opportuno o lasciare il nome di default (di solito si usa per convenzione lo stesso nome del blocco da simulare seguito dal suffisso “_tb” onde ricordare a quale blocco si applica detto file di stimoli)

OK

Verificare nelle opzioni di utilizzare il simulatore integrato all'interno di Quartus.

Simulation > Option


E scegliere Il Quartus II Simulator



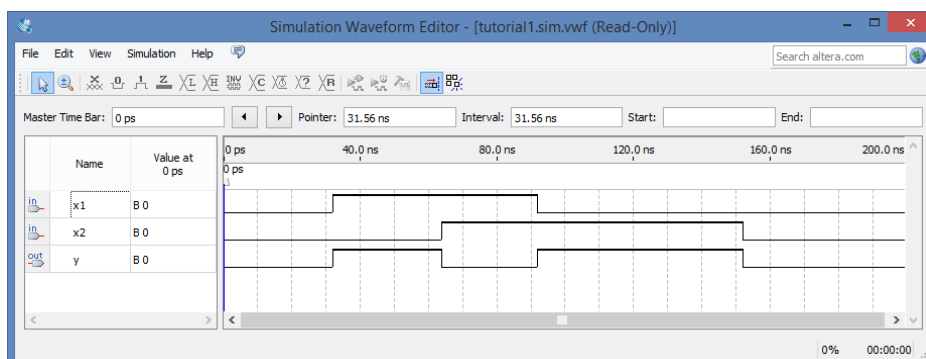
Si tenga presente che questo è un simulatore molto semplice da usarsi solamente in ambito didattico. Per simulazioni più approfondite si consiglia di impiegare ModelSim. Per il cui utilizzo si rimanda al tutorial specifico.

Far quindi partire la simulazione


Simulation > Run Functional Simulation

(esiste eventualmente lo shortcut  sul pannello comandi)

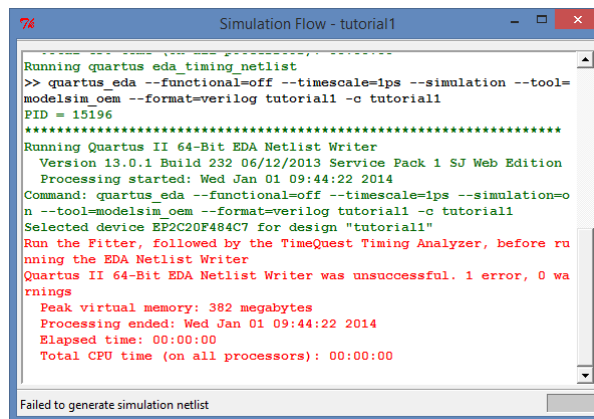
In questo modo si attiva una simulazione funzionale a livello RTL. Essa rispecchia il funzionamento ad alto livello di astrazione della funzione sviluppata ed essa non possiede ancora le informazioni su eventuali tempi di propagazione e di ritardo. Verificare le f.d.o in uscita e la loro congruenza con quanto ci si aspettava.



Ove si provasse ora a lanciare la simulazione temporale

Simulation > Run Timing Simulation 

Si otterrebbe probabilmente un messaggio di errore



```
Running quartus_eda_timing_netlist
>> quartus_eda --functional=off --timescale=1ps --simulation --tool=
modelsim_oem --format=verilog tutorial1 -c tutorial1
PID = 15196
*****
Running Quartus II 64-Bit EDA Netlist Writer
Version 13.0.1 Build 232 06/12/2013 Service Pack 1 SJ Web Edition
Processing started: Wed Jan 01 09:44:22 2014
Command: quartus_eda --functional=off --timescale=1ps --simulation=
n --tool=modelsim_oem --format=verilog tutorial1 -c tutorial1
Selected device EP2C20F484C7 for design "tutorial1"
Run the Fitter, followed by the TimeQuest Timing Analyzer, before ru
nning the EDA Netlist Writer
Quartus II 64-Bit EDA Netlist Writer was unsuccessful. 1 error, 0 wa
rnings
Peak virtual memory: 382 megabytes
Processing ended: Wed Jan 01 09:44:22 2014
Elapsed time: 00:00:00
Total CPU time (on all processors): 00:00:00
Failed to generate simulation netlist
```

Questa è infatti una simulazione a più basso livello che può essere eseguita solo dopo che il circuito è stato completamente definito in tutti i suoi collegamenti interni all’FPGA. Essa richiede pertanto di eseguire preventivamente un processo completo di sintesi e mapping che pertanto deve venir realizzato (vedi seguito) prima che tale simulazione possa venir attivata.

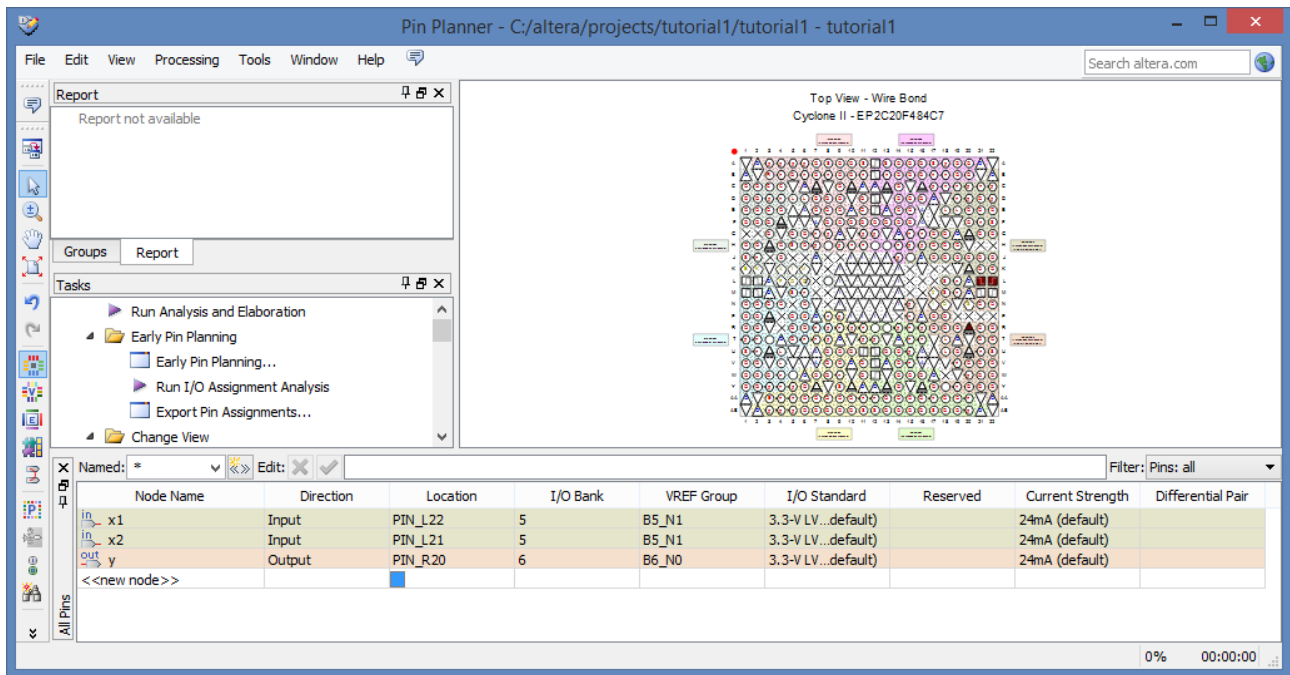
4. Definizione dei vincoli [8-9]

Al momento ci si limiterà a fissare dei semplici vincoli di posizionamento per i pin di ingresso e di uscita, onde fare in modo che gli ingressi siano associati rispettivamente agli switches 0 e 1 e che l’uscita sia collegata al led rosso numero 0.

Dal manuale utente della DE1 [1-2] si evince che SW[0] è collegato al piedino PIN_L22, mentre SW[1] è collegato al piedino PIN_L21. Inoltre il LEDRO è collegato al PIN_R20. Si tratta pertanto di associare queste posizioni ai piedini di ingresso e di uscita del nostro circuito (provare eventualmente altre associazioni, ma sempre col vincolo che ovviamente gli **switches possono essere impiegati esclusivamente come ingressi**. Sebbene sia la scheda che l’FPGA hanno delle protezioni, da un punto di vista generale, assegnare un valore logico ad un segnale che poi viene esternamente forzato ad un valore logico diverso crea un conflitto che potrebbe **danneggiare** il dispositivo stesso.

Assignments > Pin Planner (ctrl-shift-N)

Editare opportunamente la tabella come da figura:



Chiudere la finestra

Nota1: analogo risultato è ottenibile anche utilizzando il tool “assignment editor” che può essere impiegato anche per assegnare vincoli di altra natura..

Assignments > Assignments Editor (ctrl-Shift-A)

Nota2: Un altro modo più comodo per l’assegnazione dei vincoli di piedinatura (utile soprattutto quando il numero di piedini da assegnare è piuttosto elevato è quello di realizzare con un text editor un file testuale con estensione .csv (comma spaced values) ove i singoli valori sono separati da una virgola;

per esempio

```
To,Location
SW[0],PIN_L22
SW[1],PIN_L21
LEDR[0],PIN_R20
```

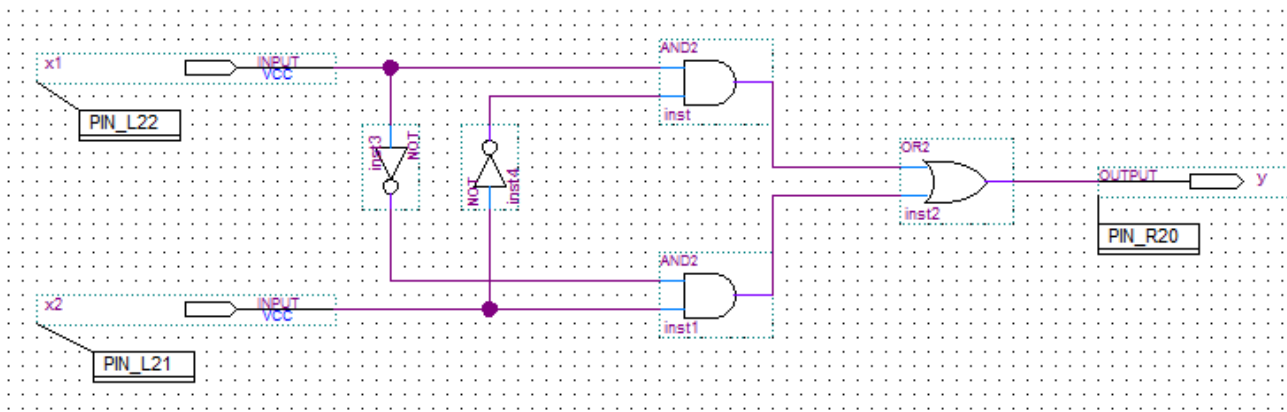
Ed importarlo nel progetto mediante

Assignments > Import Assignments

Se si riapre lo schematico, verificando che sia attiva l’opzione

View > Show Location Assignments

Si ha una rappresentazione grafica direttamente su schematico della locazione dei vari PINS



5. Compilazione completa del progetto


A questo punto si può procedere con la compilazione completa del progetto. Questa prevede diversi passi “Analisi”, “Sintesi”, Place & Route”, “Generate programming file”, “Time Quest Timing analysis”, ecc. ecc. che però vengono automaticamente svolti in sequenza:

Processing > Start Compilation (ctrl-L).

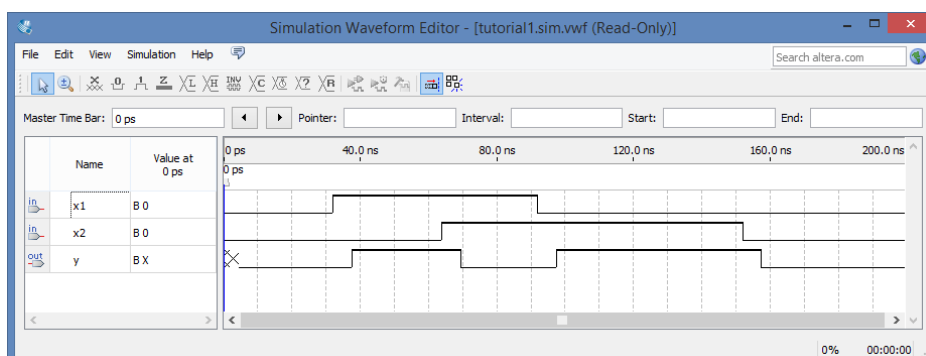
6. Simulazione temporale:

A questo punto il circuito è completamente realizzato e perfettamente definito in tutte le sue funzionalità ed in tutti i suoi ritardi strutturali, si può pertanto effettuare una simulazione a più basso livello di astrazione che tenga conto anche di queste peculiarità.

Si riapra il file (o la scheda) dove sono stati definiti gli stimoli, ovvero le forme d’onda e si attivi:

Simulation > Run Timing Simulation 

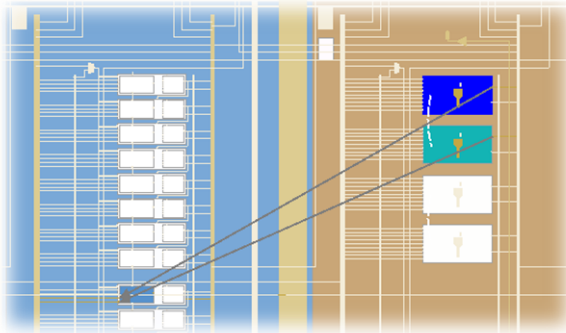
Verificare le f.d.o in uscita, la loro congruenza e soprattutto si notino i ritardi rispetto gli stimoli di ingresso.



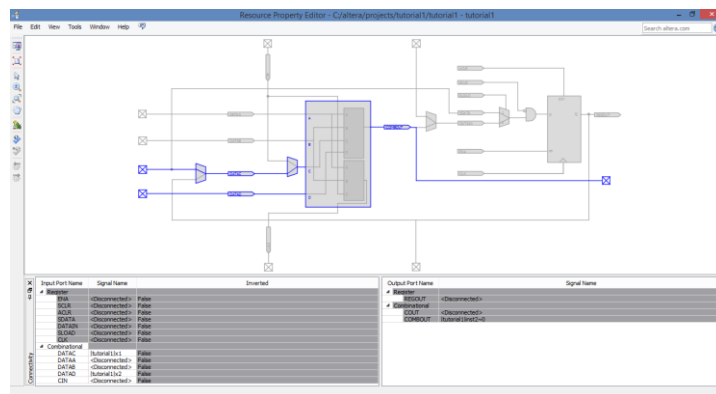
7. Visualizzazione del layout del circuito [10]:

Tools > Chip Planners

Utilizzare lo zoom per evidenziare piccole parti del circuito e tramite doppio click sulla singola cella evidenziarne la configurazione specifica.



Evidenziare in particolare i LE coinvolti dal progetto, i loro collegamenti, le porte di I/O interessate, i collegamenti e le loro configurazioni interne



8. Visualizzazione ed analisi dei vari report di processo:

Molto probabilmente la finestra contenente tutti i reports di processo dovrebbe essere già aperta, altrimenti si può richiamare con

Processing > Compilation Reports (ctrl-R)

I vari reports sono organizzati per sezioni così come lo sono le varie fasi del processo di compilazione. Analizzarli nel dettaglio, in particolare soffermarsi sulle risorse impiegate all'interno dell'FPGA

Analysis & Settings > Resources Usage Summary

sulla congruenza della pedinatura

Fitter > Pin-out File

e sui ritardi previsti secondo diversi modelli (ottimistico e pessimistico)

```
Time Quest Timing Analysis > Slow Model > Datasheet Report > ...  
... > Propagation Delay
```

```
Time Quest Timing Analysis > Fast Model > Datasheet Report > ...  
... > Propagation Delay
```

9. *Analisi dettagliata dei ritardi*

Per effettuare un'analisi maggiormente dettagliata dei ritardi di propagazione dei vari percorsi del segnale si può utilizzare uno strumento dedicato a tale scopo, denominato "Time Quest Timing Analyzer" [11-12-13]

```
Tools > Time Quest Timing Analyzer
```

Nella nuova finestra si creino le informazioni relative alle temporizzazioni

```
Netlist > Create Timing Netlist
```

E si scelga la modalità di analisi (**Post-fit**) ed il modello da adottare (**Slow**).

A questo punto bisognerebbe fissare il clock del sistema, la sua frequenza di lavoro ed i vincoli entro i quali il sistema dovrebbe sottostare, ma data la semplicità del progetto in questione si può passare alla fase successiva:

```
Netlist > Update Timing Netlist
```

Ora si visualizzi il report di interesse

```
Reports > Custom Reports > Reports Path
```

La finestra che si apre ha funzione di filtro onde visualizzare, all'interno di un progetto complesso solo i paths di nostro interesse, ma nel caso in esame si possono visualizzare tutti i paths essendo essi solamente due. Si porti pertanto la variabile "**Reports Number of Paths**" ad un valore superiore a 2 e si clicchi su "**Report Paths**"

Command Info				Summary of Paths			
Delay	From Node	To Node					
1	5.517	x1	y				
2	5.367	x2	y				

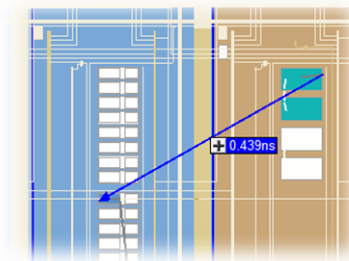
Path #1: Delay is 5.517								Path #1: Delay is 5.517					
Path Summary		Statistics		Data Path				Path Summary		Statistics		Data Path	
Total	Incr	RF	Type	Fanout	Location	Element	Property	Value					
1	5.517	5.517				data path	From Node	x1					
1	0.000	0.000		1	PIN_L22	x1	To Node	y					
2	1.026	1.026	RR	CELL	IOC_X50_Y14_N0	x1 combout	Delay	5.517					
3	1.465	0.439	RR	IC	LCCOMB_X49_Y14_N16	inst2~0 datac							
4	1.787	0.322	RR	CELL	LCCOMB_X49_Y14_N16	inst2~0 combout							
5	2.677	0.890	RR	IC	IOC_X50_Y10_N0	y datain							
6	5.517	2.840	RR	CELL	PIN_R20	y							

Nella finestra che così si apre si avrà un report dettagliato in tutti i passaggi dei due paths che compongono il nostro circuito dall'ingresso del segnale fino all'uscita di quest'ultimo. Inoltre cliccando col tasto destro sul path di nostro interesse e scegliendo "locate path" si può individuare il path in oggetto in alcune delle possibili visualizzazioni del circuito.

Ad esempio nel Chip Planner, sfruttando il comando

View > Show Delays

Si possono visualizzare il loco i ritardi dei vari step del percorso.



10. Test su Scheda

Giunti a questo punto non rimane che scaricare il bitstream (che contiene tutte le informazioni sulla configurazione dell'FPGA) sulla FPGA medesima

- Verificare che il microswitch SW12 sia nella posizione RUN
- Collegare il cavo di download USB
- Accendere la scheda

Dopo un istante la scheda si configurerà in base al file di configurazione residente nella flash [1] che di default prevede la realizzazione di pattern grafici sui led, la creazione di un disegno sulla porta VGA e la generazione di un segnale sinusoidale all'uscita della porta audio (disattivabile tramite lo switch SW[9]).

Tool > Programmer

Eventualmente - Click on "Hardware setup" e scegliere USB-Blaster
Eventualmente - evidenziare con un click il file ".sof" da utilizzare
Click on Start

Verificare il corretto funzionamento del sistema

11. Descrizione mediante HDL [11]

Un altro metodo molto più utilizzato per descrivere sistemi digitali è quello di utilizzare linguaggi di descrizione hardware (HDL). Fra questi i più comuni sono Verilog HDL [14-15-16] e VHDL [17].

Per descrivere un sistema in HDL:

```
File > New  
Verilog HDL File -oppure- VHDL File
```

A questo punto si apre una finestra di testo per descrivere il sistema in oggetto. Molto utile può essere in questo caso l'utilizzo dei "template"

```
Edit > Insert Template
```

Una descrizione del progetto più sopra riportato utilizzando il Verilog HDL potrebbe essere la seguente:

```
module tutorial1(x1,x2,y);  
  
    input x1,x2;  
    output y;  
    wire not1,not2,t1,t2;  
  
    assign not1 = ! x1;  
    assign not2 = ! x2;  
    assign t1 = x1 & not2;  
    assign t2 = x2 & not1;  
    assign y = t1 | t2;  
  
endmodule
```

Salvare il file con un nome appropriato (adottare lo stesso nome del "module")

A questo punto nel progetto vi sono due descrizioni dello stesso oggetto, si deve evidenziare a quale si vuole fare riferimento per procedere. Per selezionare la descrizione adatta

- Andare nella finestra Project navigator

Ove questa non fosse già aperta aprirla tramite

```
View > Utility Windows > project Navigator (Alt-0)
```

- Selezionare il tab "Files"
- Right-click sul file appena salvato
- Selezionare "Set as Top Level Entity"

Eventualmente per evitare confusione si può rimuovere la descrizione tramite schematico(utile specialmente se entrambe le descrizioni hanno il medesimo nome).

Click destro sul file e scegliere **“Remove File from the Project”**

Se si sono mantenuti I medesimi nomi per le porte di ingresso e di uscita si possono utilizzare le medesime procedure descritte sopra (senza dover re-impostare il file di stimoli ed il file di vincoli) per:

- Effettuare una simulazione funzionale
- Compilare il nuovo sistema
- Effettuare una simulazione temporale
- Visualizzare il floorplanning
- Visualizzare i vincoli
- Effettuare il download su scheda

Altrimenti bisogna creare nuovi files che rispecchino i nuovi nomi assunti dalle porte.

Bibliografia

- [1] Altera Corporation: **“DE1 Development and Education Board - User Manual”**
https://www.terasic.com.tw/attachment/archive/83/DE1_UserManual_v1018.pdf
- [2] Altera Corporation: **“Cyclone II FPGA Starter Development Board -Reference Manual”**
http://www.altera.com/literature/manual/mnl_cii_starter_board_rm.pdf
- [3] Altera Corporation: **“Quartus II Handbook Version 13.1 - Volume 1: Design and Synthesis”**
http://www.altera.com/literature/hb/qts/quartusii_handbook.pdf
- [4] Altera Corporation: **“Cyclone II Device Handbook”**
http://www.altera.com/literature/hb/cyc2/cyc2_cii5v1.pdf
- [5] Altera Corporation: **“Documentation on Cyclone II Devices”**
<http://www.altera.com/literature/lit-cyc2.jsp>
- [5] Altera Corporation: **“My First FPGA Design Tutorial”**
http://www.altera.com/literature/tt/tt_my_first_fpga.pdf
- [6] Altera Corporation: **“Quartus II Introduction Using Schematic Designs”**
ftp://ftp.altera.com/up/pub/Altera_Material/13.0/Tutorials/Schematic/Quartus_II_Introduction.pdf
- [7] Altera Corporation: **“Quartus II Handbook Version 13.1-Volume 3: Verification”**
http://www.altera.com/literature/hb/qts/qts_qii5v3.pdf
- [8] Altera Corporation: **“Using the Assignment Editor in the Quartus II Software”**
http://www.altera.co.jp/literature/wp/wp_assigned.pdf
- [9] Altera Corporation: **“Constraining Designs”**
http://www.altera.com/literature/hb/qts/qts_qii52001.pdf

- [10] Altera Corporation: **“Analyzing and Optimizing the Desig Floorplan with the Chip Planner”**
http://www.altera.com/literature/hb/qts/qts_qii52006.pdf
- [11] Altera Corporation: **“TimeQuest Timing Analyzer - Quick Start Tutorial”**
http://www.altera.com/literature/uq/uq_tq_tutorial.pdf
- [12] Altera Corporation: **“Quartus II TimeQuest Timing Analyzer Cookbook”**
http://www.altera.com/literature/manual/mnl_timequest_cookbook.pdf
- [13] Altera Corporation: **“The Quartus II TimeQuest Timing Analyzer”**
http://www.altera.com/literature/hb/qts/qts_qii53018.pdf
- [14] Altera Corporation: **“Quartus II Introduction Using Verilog Design”**
ftp://ftp.altera.com/up/pub/Tutorials/DE2/Digital_Logic/tut_quartus_intro_verilog.pdf
- [15] Stuart Sutherland: **“Verilog® HDL - Quick Reference Guide”**
http://www.sutherland-hdl.com/online_verilog_ref_guide/verilog_2001_ref_guide.pdf
- [16] Synopsys Corporation: **“FPGA Compiler II / FPGA Express Verilog HDL Reference Manual”**
<http://classes.soe.ucsc.edu/cmpe225/Fall01/synver.pdf>
- [16] **“IL LINGUAGGIO HDL VERILOG PER LA SINTESI E LA. SIMULAZIONE DI CIRCUITI DIGITALI”**
<http://unina.stidue.net/Sistemi%20Elettronici%20Programmabili/Materiale/Manuale%20Verilog.pdf>
- [17] Carlo Brandolese - Politecnico di Milano: **“Introduzione al linguaggio VHDL - Aspetti teorici ed esempi di progettazione”**,
[http://www.fabiopanozzo.it/download/Reti_logiche/VHDL%20\(Dispensa%20a%20cura%20di%20Carlo%20Brandolese\).pdf](http://www.fabiopanozzo.it/download/Reti_logiche/VHDL%20(Dispensa%20a%20cura%20di%20Carlo%20Brandolese).pdf)