

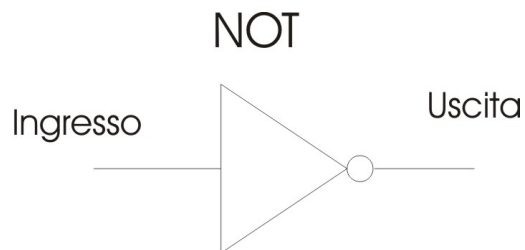
Esercitazione di Elettronica Digitale

Ing. F. Iannuzzo - A.A. 2004/2005

Progetto di una porta logica NOT con Hardware programmabile.

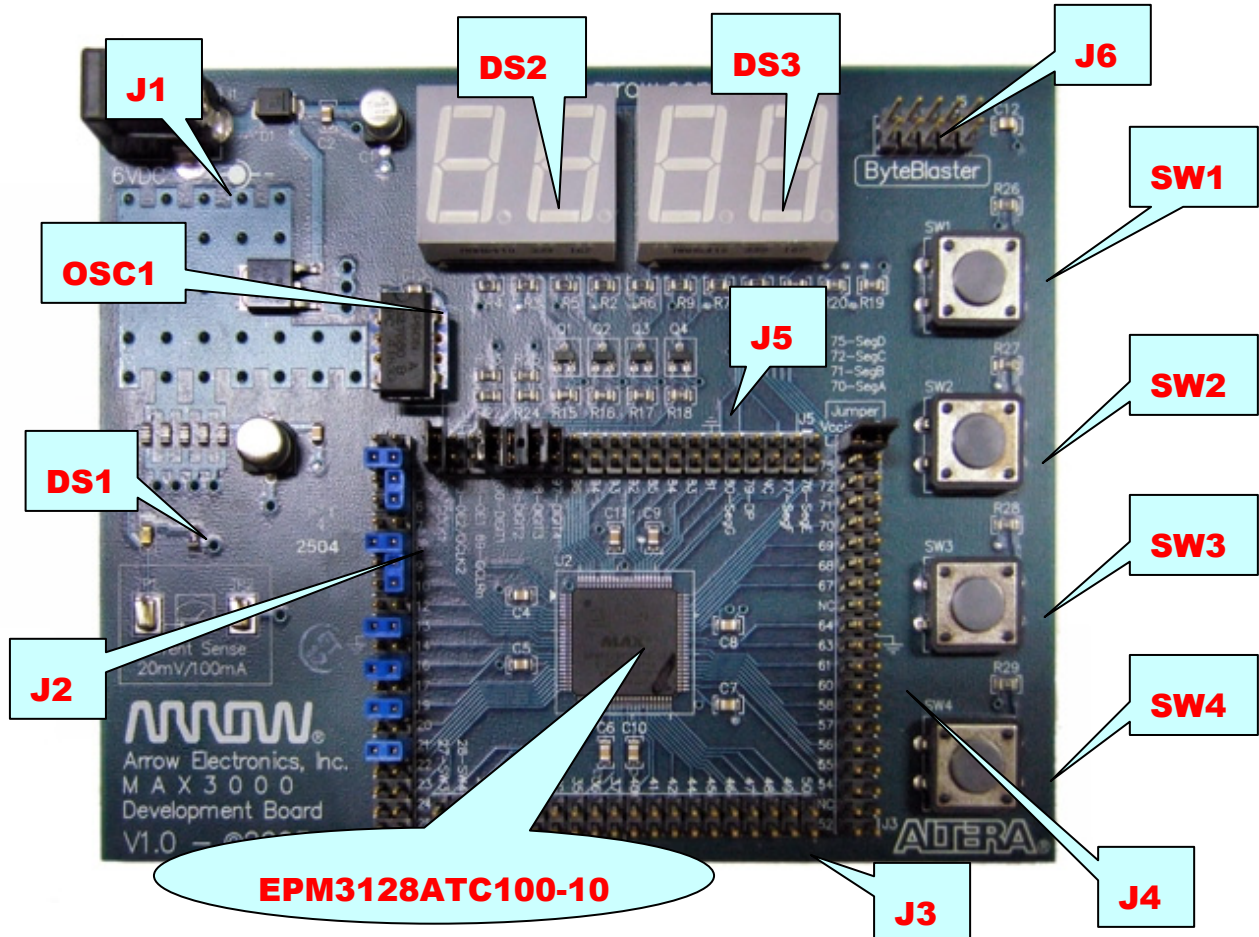
Introduzione.

In questa esercitazione impareremo l'uso di Quartus II per la simulazione e programmazione di hardware programmabile Altera, conosceremo la Development Board (basetta) con il MAX3000A ed infine realizzeremo un semplice invertitore NOT il cui schema è qui sotto riportato.



Il cuore centrale del dispositivo che useremo per le prossime esercitazioni è l'hardware programmabile MAX 3000A prodotto dall'Altera Corporation.

Questo chip, più precisamente è un CPLD (Complex – Programmable – Logic – Device) etichettato EPM3128ATC100-10 e per la sua utilizzazione è montato su una basetta rappresentata nella foto sotto.



Vediamo gli elementi che fanno parte della basetta:

- **SW1...SW4:** 4 pulsanti.

SW1 è connesso al pin 24 del CPLD, SW2 al pin 25, SW3 al 27 e SW4 al 28. Quando vengono premuti SW1 e SW2 portano a massa (livello logico basso "0") i rispettivi pin, mentre SW3 e SW4 portano a Vccio (livello logico alto "1"). Tutti e 4 sono inseriti in serie con resistori da 1 Kohm come protezione da corto circuiti.

- **J1...J6:** 6 connettori con svariate funzioni.

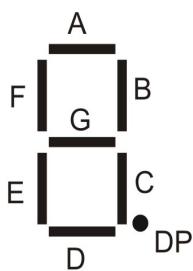
J1: connettore per l'alimentazione della basetta (6V DC)

J2...J5: connettori disposti ai 4 lati del PLD (2x20 pin). A fianco di ogni pin c'è un numero che indica il rispettivo pin del PLD. Oltre al numero in alcuni vi è anche una sigla che identifica la funzione del pin o il collegamento (es. "80 – SEG G" indica che il pin 80 del PLD è connesso al segmento G dei display)

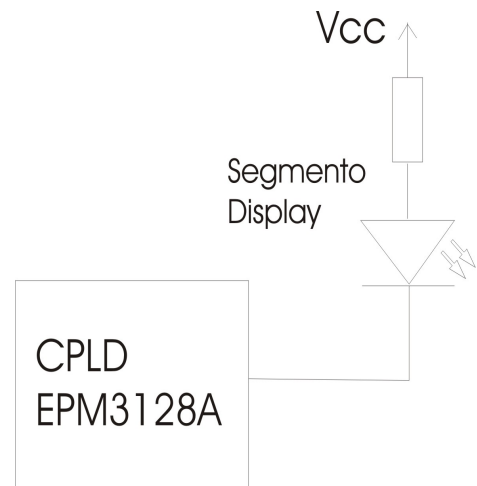
J6: connettore per interfacciare la basetta con il PC, attraverso il cavo ByteBlasterII siamo in grado di programmare il PLD.

- **OSC1:** è un oscillatore che genera un clock di 32,768 Mhz, utile per le applicazioni sequenziali, è connesso al pin 87.
- **DS1:** led di segnalazione alimentazione.
- **DS2, DS3:** sono 4 display a sette segmenti + punto (a, b, c, d, e, f, g, decimal point).

Questi display sono ad anodo comune, ciò implica che per l'accensione di un segmento bisogna portare il rispettivo pin a massa, quindi un livello logico basso corrisponde a segmento acceso, mentre livello alto, segmento spento. Di seguito sono specificate le connessioni di ogni display.



Segmento	PIN
A	70
B	71
C	72
D	75
E	76
F	77
G	80
DP	79




Progetto con il software Quartus II.

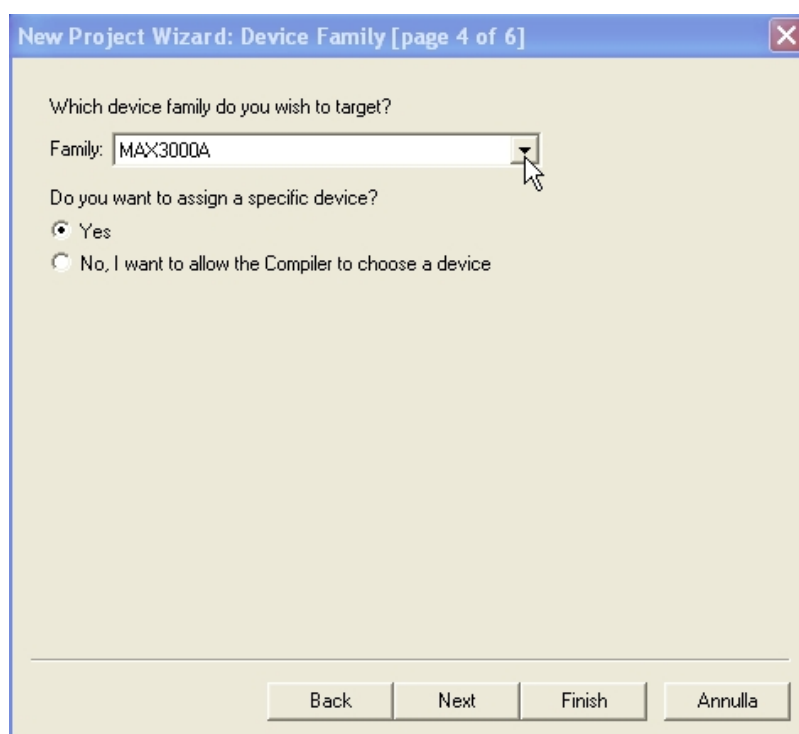
Prima di iniziare la realizzazione del progetto, si prenderà dimestichezza con il programma Quartus II.

Per prima cosa si crei la cartella **c:\Elettronica**, e dentro se ne crei un'altra chiamata **Esercitazione_1** (si scriva senza inserire spazi). In questa cartella si inseriranno tutti i file del progetto.



Alla fine dell'esperienza si copi questa cartella su un supporto di memorizzazione rimovibile (FLOPPY o PENDRIVE) in quanto allo spegnimento del PC si perderanno tutti dati salvati in precedenza.

- a) Si apra dal menù Avvio/Programmi/Altera il software Quartus II
- b) Si faccia click su: File -> New Project Wizard, all'apertura della finestra, si scelga **Next**, si scelga facendo click su  nel primo spazio, la directory **Esercitazione_1** precedentemente creata e si dia il nome **Invertitore** al progetto, specificandolo nel secondo e terzo spazio bianco. Si faccia click su **Next** per tre volte di seguito fino ad arrivare alla finestra seguente:



In questa finestra si specifichi la famiglia MAX3000A, si selezioni Yes e si prema su **Next**.

Nella finestra che si apre si scelgano i seguenti dati:

Si scelga dalla casella "PIN COUNT" il valore 100.

Si scelga dalla casella "SPEED GRADE" il valore 10.

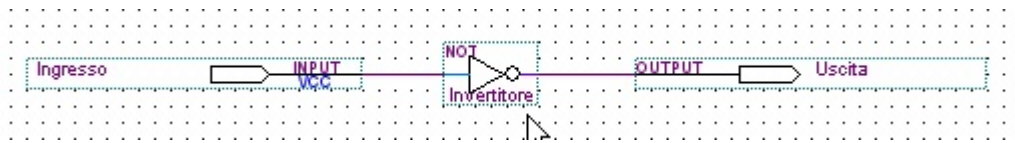
Si scelga il CPLD EPM3128ATC100-10 dalla lista.


Si clicchi su **Finish**.

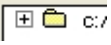
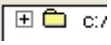
c) Si clicchi sull'icona  e si scelga Block Diagram / Schematics File.

d) Si clicchi su **OK**

Si aprirà una finestra con una griglia, è qui che si dovranno inserire i componenti del progetto da realizzare. Come mostrato dallo schema seguente:




Si clicchi sull'icona . Si aprirà la finestra denominata Symbol. Si disabiliti la funzione Repeat-insert mode, altrimenti si inseriranno più componenti. Si apra la libreria cliccando

sull'icona  Libraries:  c:/ e si scelga Primitives e dalla sottocartella Logic si scelga di inserire la porta Not. Si faccia click su **Ok** e si inserisca sul foglio Schematics la porta NOT. Dalla libreria Pin, si scelga uno alla volta Input ed Output, disabilitando di nuovo la funzione Repeat-insert mode, e per una maggior chiarezza si rinomino i pin chiamandoli Ingresso e Uscita (per rinominare il pin bisogna cliccare sul nome). Si effettui il collegamento tra i componenti avvicinando il puntatore del mouse all'ingresso e all'uscita degli stessi. Si noti il cambiamento del cursore, a questo punto tenendo premuto il pulsante sinistro del mouse si tracci il collegamento.



Si noti che il pin di ingresso presenta un valore di default (VCC o GND) modificabile con doppio click. Questo valore specifica al compilatore quale livello logico assegnare al pin come default.

Fatto ciò si salvi il file e si proceda alla compilazione tramite l'icona  posta in alto sulla barra dei comandi.


Finita la compilazione si visualizzi il Report della compilazione. Le informazioni che vi sono indicate riguardano le fasi della compilazione. Le fasi sono le seguenti:

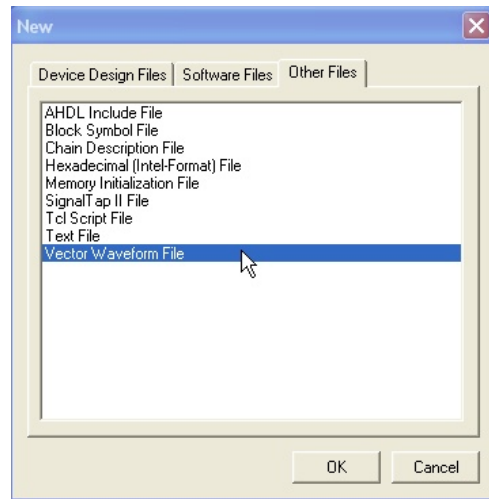
Analysis and Synthesis: in questa fase si analizza il progetto e viene generata la Netlist che specifica i collegamenti tra i vari componenti.

Fitter: in questa fase il compilatore distribuisce nello spazio del CPLD le macrocelle contenenti i componenti del progetto.

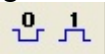
Assembler: in questa fase il compilatore assembla il file per la programmazione del CPLD.

Timing Analyzer: in questa fase il compilatore analizza i tempi di propagazione nel CPLD, calcola il tempo peggiore e il tempo migliore specificando il percorso per cui si hanno.

Si continui con l'esperienza cliccando su . Si scelga Vector waveform file dal sottomenu Other Files e si clicchi su **OK**.



Si aprirà un'ulteriore finestra, si faccia doppio clic nella parte sinistra (sotto la dicitura Name) e si scelga **Node Finder...**. Si utilizzi il menù filter e si scelga PINS: all e poi **List**. Apparirà la lista di tutti i pin del progetto. Si faccia doppio click sul pin Ingresso e si chiudano le finestre Node Finder e Insert Node or Bus cliccando su **OK**.


- e) Si specifichino le transizioni dell'ingresso. Con il mouse si selezionino di volta in volta diversi intervalli di tempo e con le icone  si imposti se nell'intervallo selezionato l'ingresso debba assumere valore alto (1) oppure basso (0). Si salvi il file.



Attenzione: Si guardi bene la scala dei tempi che si sta utilizzando quando si imposta la forma d'onda. Una transizione che duri meno del tempo di propagazione non sarà apprezzabile in uscita. Per ingrandire si usi la lente presente al lato sinistro della finestra e si clicchi con il tasto destro sulla finestra delle forme d'onda.

Simulazione del Progetto



Si clicchi sull'icona  per la compilazione e se non ci sono errori si passi alla simulazione cliccando sul menu Tools -> Simulator Tool.

Si imposti il tipo di simulazione tra le due a disposizione.

- 1) **Functional**: simulazione ideale, non tiene conto dei tempi di propagazione.
- 2) **Timing**: simulazione reale, tiene conto dei tempi di propagazione.

Si scelga per prima la simulazione Functional, e si clicchi su **Generate Functional Simulation Netlist**. Si selezioni il file delle forme d'onda (.vwf), e si dia **Start** alla simulazione. Se non si sono avuto errori, si clicchi su **Report**.

Nella finestra si visualizzino le transizioni dell'uscita e si verifichi se il risultato è quello voluto.

Si ripeta la simulazione scegliendo questa volta Timing, si dia **Start** e si visualizzi, cliccando su **Report**, il risultato ottenuto in modo da confrontare le due simulazioni e notare le differenze.

Programmazione della basetta.

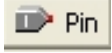
Tra poco andremo a programmare la basetta ma rimane da fare una cosa molto importante, dobbiamo assegnare i pin al nostro progetto.

Cosa significa assegnare i pin?

Bisogna dire al nostro hardware programmabile quali sono gli input e gli output corrispondenti sulla basetta. Per noi quale sarà il segnale d'ingresso? E cosa collegheremo al pin di uscita?

Possiamo, ad esempio, disporre che l'uscita sia visualizzata da un segmento di un display e l'ingresso sarà la pressione di un pulsante. Procediamo.

Si entri nel menù `Assignments` -> `Pins`, si aprirà una finestra, si faccia click sull'icona




posta in alto a destra.

- a) Si faccia doppio click su **new** sotto la colonna **To** ; si aprirà un menù a tendina; si scelga il pin Ingresso .
- b) Si faccia doppio click sotto la colonna **Location** in corrispondenza del pin Ingresso; si aprirà un menù a tendina, si scelga il **Pin_24** corrispondente al pulsante SW1.
- c) Si faccia di nuovo doppio click su **new** sotto la colonna **To** e si scelga ora il pin Uscita .
- d) Si faccia doppio click sotto la colonna **Location** e si scelga il **Pin_80** che corrisponde al segmento G dei display,

Fatta l'assegnazione dei pin, si salvi il file e si compili di nuovo il progetto. Se non si hanno errori si vada nella cartella che si trova sul desktop, **Eledig\Esercitazione_1** e si copi il file **".pof"** su un floppy. Questo file servirà alla programmazione del CPLD.

Verifica del risultato.

Si apra Quartus II e si faccia click sull'icona . Nella finestra che si aprirà, con il comando **Add file...** si specifichi il file `.pof` salvato poco fa sul floppy. Si scelga l'opzione `Program / Configure` e si click su **Start** per iniziare la programmazione.

Si verifichi che alla pressione di SW1, il segmento centrale del display si dovrà spegnere. Questo perché il pulsante porta all'ingresso del nostro invertitore un livello logico basso e di conseguenza l'uscita sarà un livello alto che porta allo spegnimento del segmento.

Ringraziamenti.

Desidero ringraziare il Dott. Paolo Mele, laureato di questa Facoltà, per l'insostituibile contributo dato alla preparazione di questa e delle altre esercitazioni sugli hardware programmabili, nell'ambito del suo lavoro di tesi di laurea.