

Esercitazione di Elettronica Digitale

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

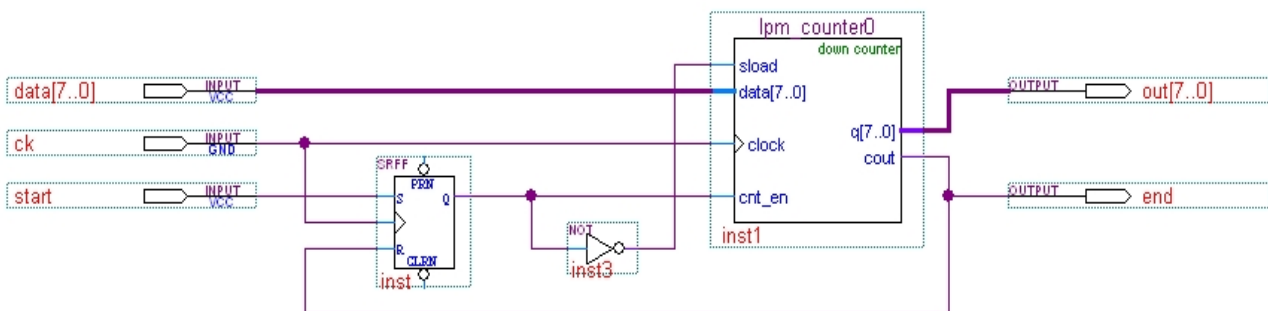
Timer Programmabile con contatore.

Introduzione.

Nella presente esercitazione si realizzerà un timer programmabile facendo uso di due dispositivi digitali di tipo sequenziale e precisamente un Flip-flop RS e un contatore. Si vedrà anche come realizzare un divisore di frequenza con un contatore asincrono e si verificherà, simulando e sperimentando con il CPLD MAX3000A il funzionamento del timer.

Il timer dovrà fornire un impulso di fine conteggio sul pin end dopo un ritardo da noi preimpostato. Il funzionamento è il seguente: inizialmente l'uscita Q del Flip-flop è bassa, il pin load è alto e in uscita al contatore verrà caricato il dato presente sugli ingressi data[7..0], cioè il ritardo da noi programmato. Portando alto il pin start, l'uscita Q del Flip-flop RS verrà settata alta, in questo modo si abilita il contatore al conteggio all'ingù che, conterà gli impulsi di clock fino ad azzerarsi e quindi a produrre sul pin cout l'impulso di fine conteggio.

Lo schema da realizzare sarà il seguente :



Lo schema presenta i seguenti pin:

data[7..0]: rappresenta un bus a 8 bit, da qui si imposta il ritardo come dato binario.

ck: a questo pin deve essere applicato il segnale di clock.

start: questo pin dà lo start al conteggio.

out[7..0]: rappresenta un bus a 8 bit che identifica il dato in uscita, ovvero il conteggio.

end: su questo pin è fornito l'impulso di fine conteggio.


Progetto del Timer programmabile.

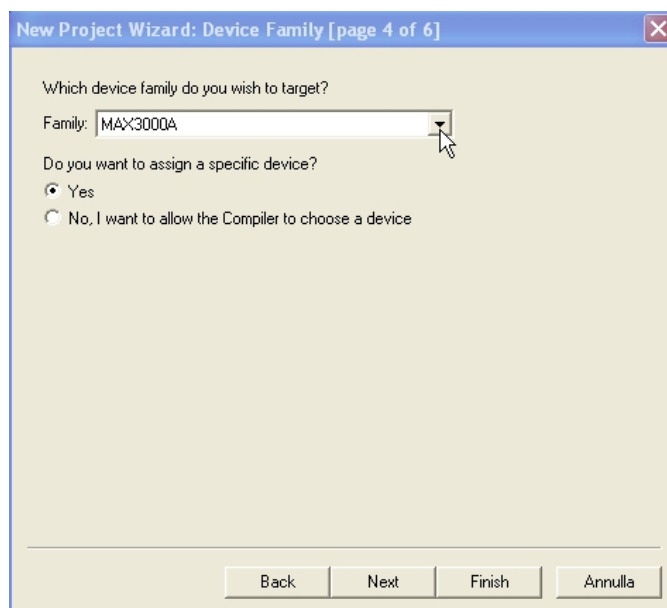
Per prima cosa si crei sul desktop una cartella chiamandola **Eledig**, e dentro se ne crei un'altra chiamata **Timer**. In questa cartella si inseriranno tutti i file del progetto.



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 i dati salvati in precedenza.

Si apra Quartus II

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 **Timer** precedentemente creata e si dia il nome "**Timer**" (senza apici) 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**, si prema su **Next**.

Si aprirà una finestra, 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**.


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

Si clicchi su **OK**

Si aprirà una finestra con una griglia, è qui che si dovranno inserire i componenti come mostrato dalla schema iniziale.

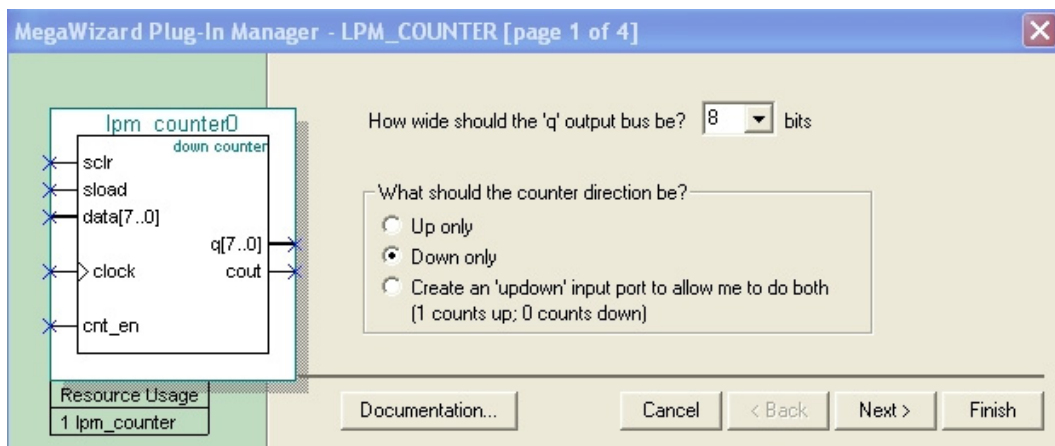
Si clicchi sull'icona , si aprirà la finestra denominata "**Symbol**".

Libraries:

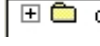
Si apra la libreria cliccando sull'icona , si scelga Primitives -> Storage e si inserisca srff (flip-flop di tipo RS) disabilitando la funzione Repeat-insert mode.

Ora si scelga, sempre dalla libreria, Megafunction e la sottocartella Arithmetic, si scelga "lpm_counter". Si disabiliti la funzione Repeat-insert mode, altrimenti si inseriranno più componenti e si controlli che sia abilitata la funzione LaunchMegawizard Plug-In.


Si faccia click su **Ok**, si aprirà il wizard, si imposti AHDL come tipo di output file e si dia **Next**. Si scelga il numero di bit, 8 bit, si selezioni Down only e si dia **Next**. Si scelga Plain binary, Count Enable e Carry-out e si dia **Next**. Ora si scelga di inserire anche l'ingresso Load dalla sezione Asynchronous inputs e si clicchi su **Finish**.



Libraries:

Si inserisca il blocco creato. Ora si apra la libreria cliccando sull'icona , Si disabiliti la funzione Repeat-insert mode e si scelga dal gruppo Primitives, la sottocartella Logic e si inserisca la porta Not come mostrato nello schema.

Libraries:


Si apra la libreria cliccando sull'icona , Si disabiliti la funzione Repeat-insert mode e si scelga dal gruppo Primitives, la sottocartella Pin, qui si scelga "Input". Si inseriscano nello schema i pin di input e si colleghino come mostrato dallo schema.

Si rinominino i pin di input nel modo seguente: start, data[7..0] e Ck.

Si inseriscano i pin "Output" e si colleghino al bus q[7..0] e al cout rinominandoli rispettivamente q[7..0] e al cout. Per effettuare il collegamento tra i componenti si avvicini il puntatore del mouse all'ingresso o all'uscita degli stessi. Si noti il cambiamento del cursore, a questo punto tenendo premuto il pulsante sinistro del mouse si tracci il collegamento tra i componenti.



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

Si imposti il valore di default di tutti gli ingressi a VCC. 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.

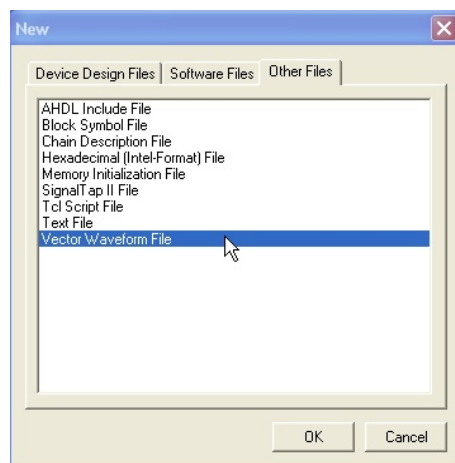
Timing Analyzer: in questa fase il compilatore analizza i tempi di propagazione nel CPLD, calcola il tempo peggiore e il tempo migliore specificando il percorso per cui si hanno.

Simulazione del Progetto.

Si impostino nel menù Tool -> Options e nella categoria Waveform Editor i seguenti valori:


- 1) Default file end time: 15 s
- 2) Default grid period : 100 ms

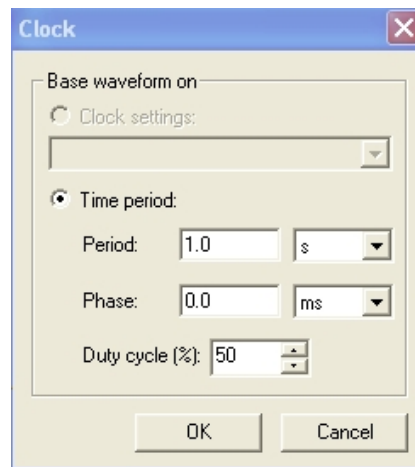
Si continui con l'esperienza cliccando su . Si scelga Vector Waveform File dal sottomenu Other Files e si clicchi su **OK**. Si aprirà un ulteriore finestra





Si guardi la scala dei tempi che si sta utilizzando, l'ordine di tempo deve essere dei secondi. Per ingrandire selezionate la lente presente al lato sinistro della finestra e cliccate con il tasto destro sulle forme d'onda.

Si faccia doppio clic nella parte sinistra (sotto la dicitura Name) e si imposti, nell'opzione Radix il tipo di dato Binary, 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 selezioni ck e start, e si clicchi su **OK**. Si ripeta il procedimento ma si imposti il tipo di dato in Radix al valore Unsigned


Decimal e si selezionino i pin data e out. Ora si selezionino ck e si clicchi sull'icona  apparirà la finestra seguente. Si imposti il periodo del clock a 1 secondo, come nella figura sotto e si dia **OK**



Si clicchi sull'icona  e si selezionino, per l'ingresso start un intervallo di tempo di circa 1 secondo all'inizio della time bar, (prima di 3 secondi ad esempio). Si è impostato l'ingresso per quell'intervallo ad un valore alto. Si continui selezionando l'ingresso data e cliccando sull'icona  si imposti il valore 5.

Che cosa stiamo facendo?

Stiamo indicando al contatore di caricare il dato presente in ingresso (5 secondi) e stiamo impostando il timer a 5 secondi.

Si salvi il file delle forme d'onda e si clicchi sull'icona  per la compilazione; se non ci sono errori passate alla simulazione cliccando su Tools -> Simulator Tool.

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

Functional: simulazione ideale, non tiene conto dei tempi di propagazione.

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 selezionino il file delle forme d'onda (.vwf), e si dia **Start** per iniziare la simulazione. Se non si sono avuto errori, si clicchi su **Report**.

Nella finestra che si aprirà si visualizzeranno gli ingressi da noi impostati e le uscite calcolate dalla simulazione. Si verifichi se il risultato è quello voluto.

Si ripeta la simulazione scegliendo questa volta il tipo 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.

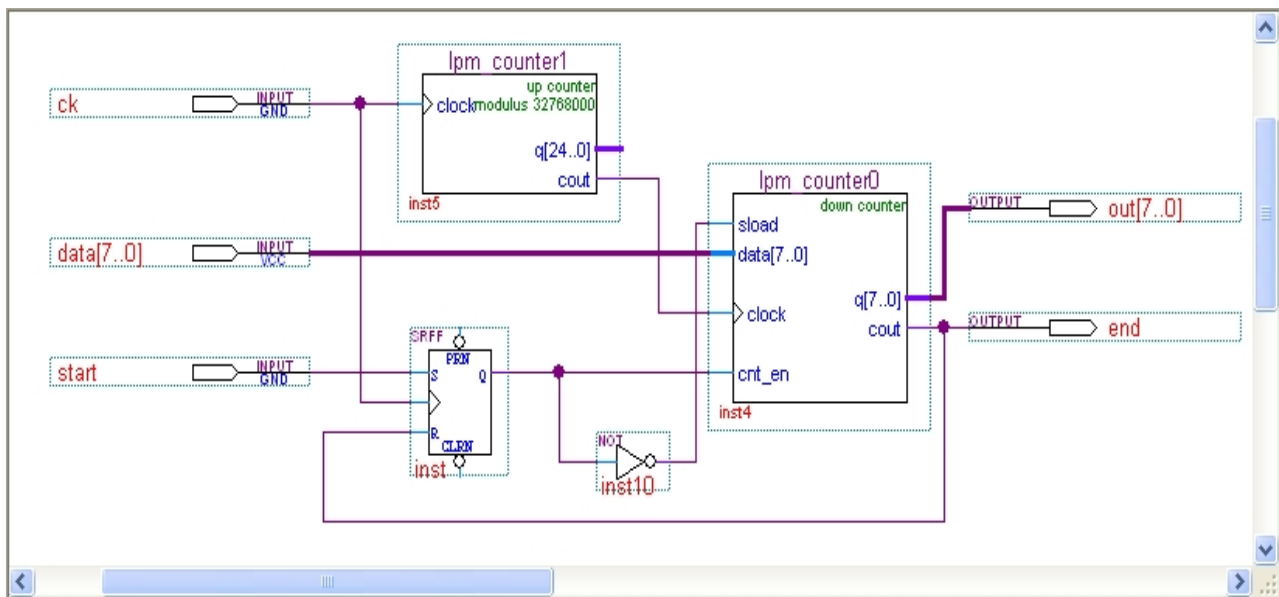
Tra poco andremo a programmare la basetta per sperimentare il funzionamento del nostro timer, dobbiamo però modificare il nostro progetto per adattarlo al clock della nostra basetta e poi dobbiamo assegnare i pin.

Cosa significa assegnare i pin?

Bisogna dire al nostro hardware programmabile quali sono gli input e gli output corrispondenti sulla basetta.

Si può pensare di visualizzare il conteggio, cioè gli 8 bit che scandiscono il conteggio sugli otto led (i 7 segmenti e il DP) mentre l'uscita end si potrà visualizzare con lo spegnimento di un intero display ad esempio DIGIT4.

Prima di assegnare i pin però si modifichi il circuito come riporta lo schema qui sotto.

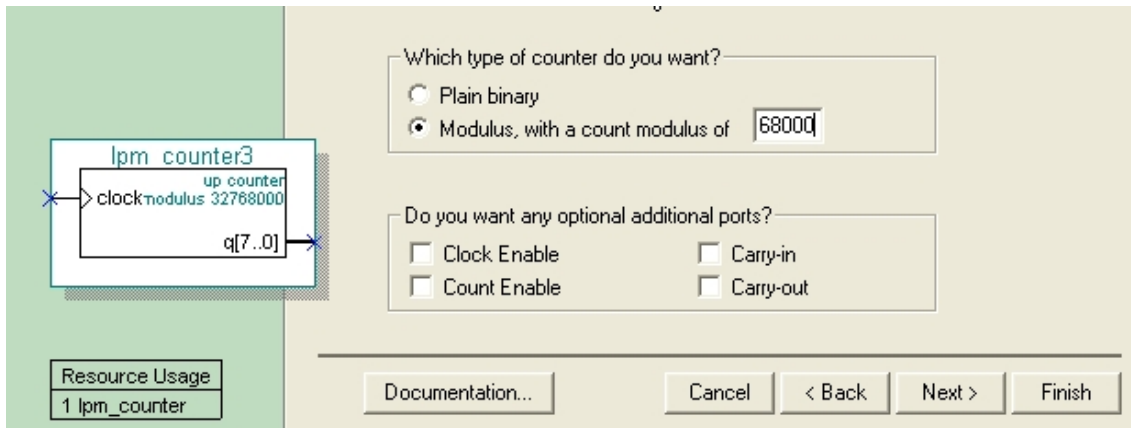


La modifica è richiesta in quanto il clock presente sulla board è di 32,768 MHz, quindi è troppo alto per la nostra applicazione, allora si utilizza un altro contatore che funge da divisore di frequenza.

Si crei un altro progetto e si inseriscano gli stessi componenti del progetto precedente. Il divisore di frequenza si scelga nel modo seguente.

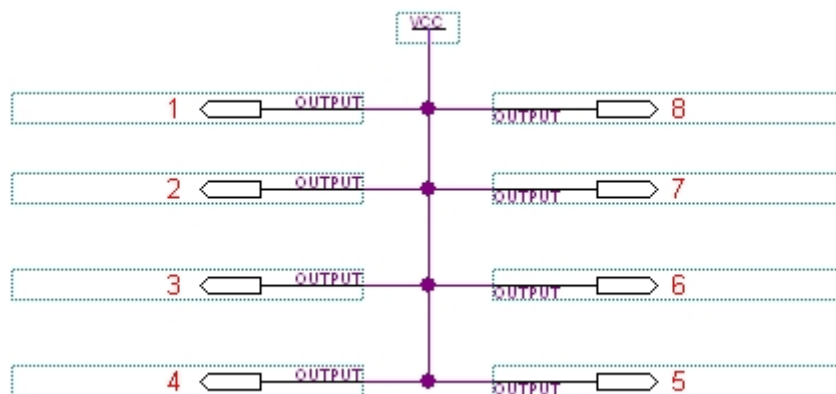
Si scelga dalla libreria il contatore LPM_COUNTER e si personalizzi impostando le seguenti opzioni nella creazione guidata.

- Up only
- Number of bit: 25
- Modulus, with a count modulus of : 32768000, come mostra la figura sotto



- Cout: SI
- Load – Clear – Clock enable: NO

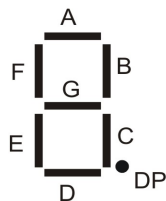
Si è scelto un contatore modulo 32768000 che divide la frequenza del clock iniziale e la porta ad 1 Hz. Il contatore usato da divisore di frequenza si è dimensionato a 25 bit in quanto $2^{25} = 33554432$ è sufficiente a fornire il modulo del contatore. Come si nota dallo schema circuitale, il clock al contatore che conta i secondi proviene dal divisore di frequenza. In questo modo il timer contando i colpi di clock conterà effettivamente un ritardo in secondi. Questa modifica però rende il circuito asincrono, in quanto non c'è un solo clock a sincronizzare tutti i componenti. Si inserisca anche nel progetto la rete di pull-up mostrata in figura.



Modificato il progetto lo si compili con l'icona . Successivamente si entri nel menù Assignments -> Pins, si aprirà una finestra, in questa si assegnino i pin come nelle tabelle qui sotto.

<u>Corrispondenza Pin progetto - Pin basetta</u>	
1	PIN_1
2	PIN_5
3	PIN_7
4	PIN_9
5	PIN_12
6	PIN_14
7	PIN_17
8	PIN_20


<u>Corrispondenza Pin progetto - Pin basetta</u>	
DATA[0]	PIN_2
DATA[1]	PIN_6
DATA[2]	PIN_8
DATA[3]	PIN_10
DATA[4]	PIN_13
DATA[5]	PIN_16
DATA[6]	PIN_19
DATA[7]	PIN_21
CK	PIN_87
START	PIN_27
OUT[0]	PIN_70
OUT[1]	PIN_71
OUT[2]	PIN_72
OUT[3]	PIN_75
OUT[4]	PIN_76
OUT[5]	PIN_77
OUT[6]	PIN_80
OUT[7]	PIN_79
END	PIN_100



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

Fatte le assegnazioni si compili il progetto, si salvi il file .pof, generato dalla compilazione, su un floppy e si passi alla programmazione del CPLD.

Verifica del progetto.

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 impostino i jumper sui pin assegnando il ritardo per lo start. Si prema il tasto SW3 per dare lo start e si contino i secondi per verificare se effettivamente il timer funziona in modo corretto. Si facciano diverse prove.

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.