

TUTORIAL 4

Realizzazione di un cronometro su scheda
XSA50

1 Problema proposto

In questo tutorial si propone di realizzare un semplice cronometro, controllato da un tasto che funziona da "Start/Stop" ossia fa partire e ferma il conteggio.

Lo scopo primario è quello di familiarizzare con l'utilizzo di un tool, lo StateCad, che viene impiegato per la realizzazione di macchine a stati finiti.

Per la realizzazione del progetto è necessario un contatore, un display a sette segmenti ed un sistema di controllo, pilotato dal tasto presente sulla board, che avvii o fermi il conteggio sul contatore. Quest'ultimo è di facile implementazione tramite un'appropriata macchina a stati finiti e riceve in ingresso il segnale di clock, quello di reset e quello associato al tasto, mentre in uscita c'è un unico segnale che indica lo stato in cui dev'essere il conteggio.

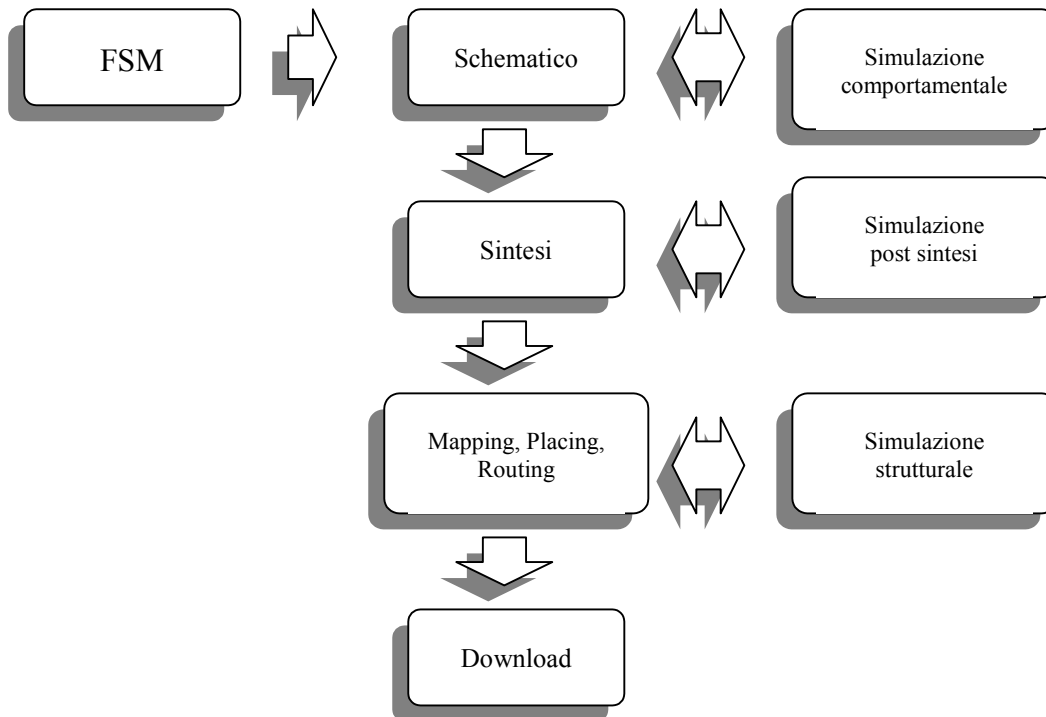
Si seguiranno quindi i seguenti passi:

1. Innanzitutto si utilizzi lo StateCAD per la creazione della macchina a stati finiti;
2. Si simuli con lo StateBench e se ne verifichi il corretto funzionamento;
3. Si generi il simbolo associato alla macchina a stati finiti così sviluppata;
4. (Opzionale) Si generi un segnale di test;
5. (Opzionale) Si simuli dal punto di vista comportamentale con il Modelsim;
6. Si crei il decoder per il display a sette segmenti;
7. Si realizzi il simbolo associato;
8. Si realizzi lo schematico del progetto completo, utilizzando i simboli creati ai passi 3. e 7.;
9. Si esegua una simulazione comportamentale e se ne verifichi il comportamento;
10. Si proceda con l'operazione di sintesi;
11. Si analizzi il report prodotto;
12. Si effettui la traduzione;
13. Si esegua una simulazione post sintesi;
14. Si crei un file di vincoli;
15. Si mappi e si simuli;
16. Si esegua il place&route e si verifichi che i vincoli siano stati raggiunti;
17. Si faccia una simulazione post place&route;

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

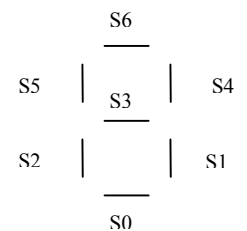
18. A completamento di tutto si generi il bitstream e si faccia il download su scheda;

Si valuti eventualmente l'impiego di un opportuno divisore di clock da inserire nello schematico del progetto completo.



Per quanto riguarda i vincoli, si tengano presente che:

- il tasto sulla scheda è collegato al piedino 93 sull’FPGA;
- l’interruttore sulla scheda è collegato al piedino 54 sull’FPGA (utile ad esempio , come tasto di reset);
- 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.



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 con prese a 25 poli;
- l’alimentatore;

2 Realizzazione proposta

Si lanci il software ISE e si crei un nuovo progetto.

Si aggiunga un nuovo sorgente e questa sia del tipo `State Diagram`: dopo aver dato il nome al nuovo file, si aprirà automaticamente lo StateCad.

Il tool di sviluppo StateCad è uno dei pacchetti compresi nell'installazione del software della Xilinx ISE Webpack: è di semplice utilizzo, in quanto molto spesso si fa uso di wizard che semplificano notevolmente il progetto.

Si vuole che la macchina a stati riceva in ingresso il tasto ed in uscita dia il comando al contatore di fermarsi o di ripartire.

Sebbene di stati "utili" ve ne siano all'atto pratico solamente due, ossia uno di Start ed uno di Stop, la metodologia di passaggio da uno stato all'altro ci impone una certa cautela nella realizzazione del diagramma rappresentativo della macchina.

Infatti, essendo la transizione tra i due stati pilotata dal medesimo segnale in ingresso, si correrebbe il rischio, qualora la pressione del tasto perdurasse per più di un ciclo di clock, di trovare una macchina che cambia stato ripetutamente ad ogni ciclo. Si deve pertanto introdurre degli stati d'attesa che consentono il passaggio dallo stato di Start a quello di Stop e viceversa, non semplicemente quando viene rilevato un segnale alto all'ingresso, bensì quando sull'ingresso viene rilevata una transizione dallo stato alto a quello basso.

Si proceda nel seguente modo:




- si clicchi allora su `Draw State Machine`;
- si scelga una macchina a 4 stati e forma geometrica;
- si scelga un reset asincrono, ma per questo progetto la sincronia sul reset è ininfluente;
- si spunti la voce `Loop Back`, che permette di associare ad ogni stato una transizione su se stesso e si spunti la voce `Next` e si scriva sulla rispettiva casella `TASTO`, questo permette di definire automaticamente una transizione al prossimo stato controllata dalla variabile `TASTO`.

Si supponga che lo stato iniziale sia quello di stop: si clicchi due volte su di esso e gli si dia un nome, poi si selezioni `Output Wizard` (questo serve per impostare l'uscita corrispondente al presente stato) e sulla casella contrassegnata come `DOUT` si scriva il nome della variabile d'uscita (qui si è scelto `ChipEnable`) e sotto `Constant` a sinistra si scelga il valore (per questo stato è pari a 0; si associa cioè al valore 1 l'inizio del conteggio e allo 0 il viceversa); si confermi il tutto e si torni all'area di lavoro.

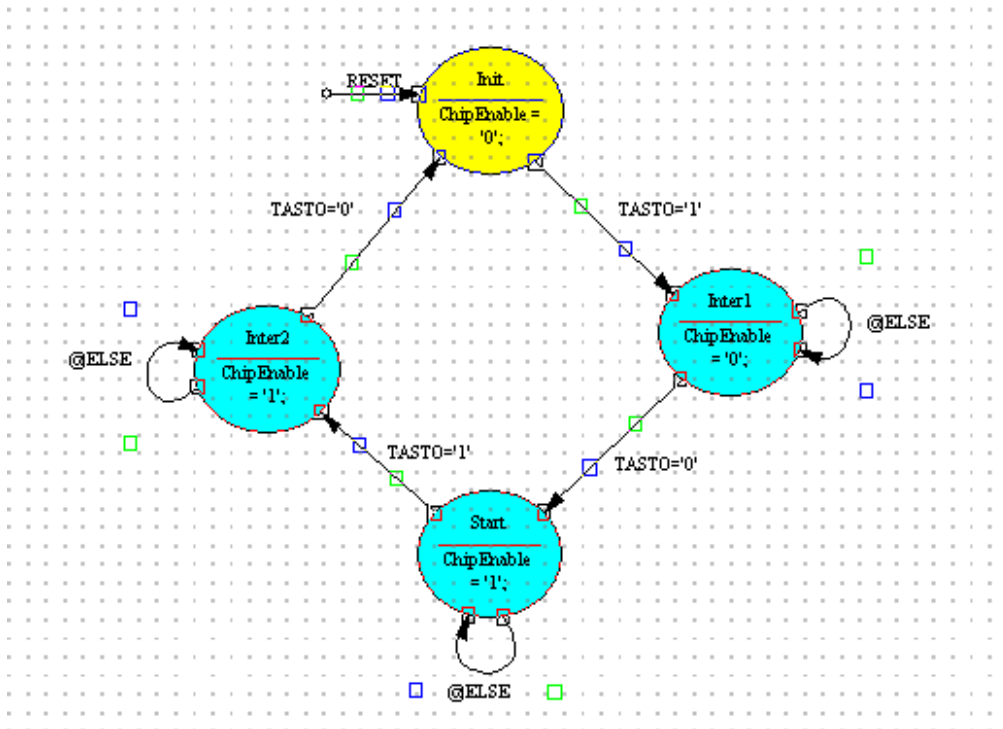
Si noti come ora al primo stato è stata associata anche l'uscita desiderata.

Allo stesso modo ora si clicchi sul secondo stato e se ne assegni un nome e l'uscita corrispondente (in questo caso `ChipEnable='0'`): si osservi che ora su `Output Wizard` sotto `DOUT` compare automaticamente la variabile `ChipEnable` e basta solo selezionarla.

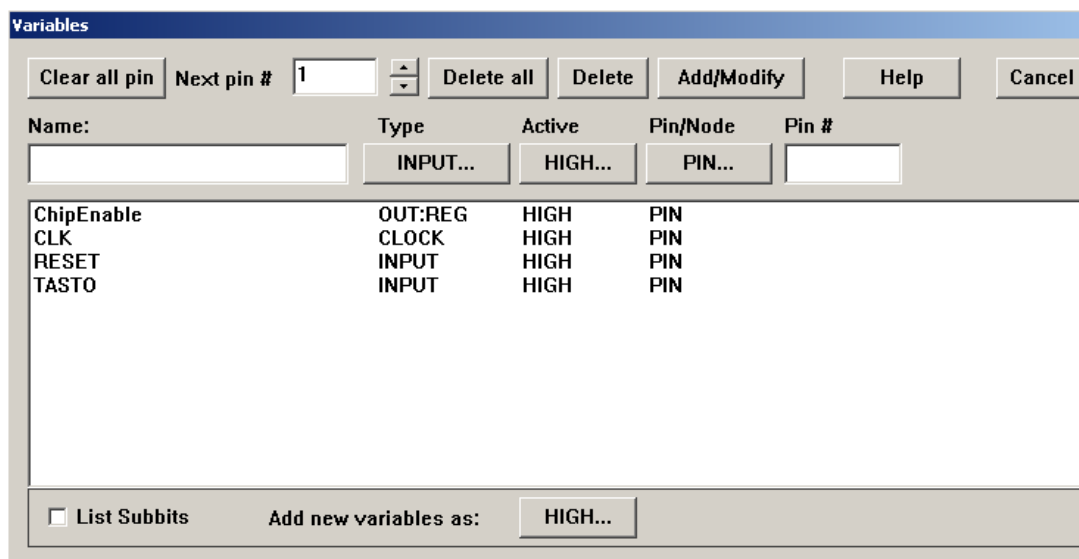
Analogamente si proceda col terzo stato, impostandone il nome (qui `Start`) e l'uscita (qui `ChipEnable='1'`), e successivamente con l'ultimo impostando la variabile d'uscita nuovamente a 1. Ora mancano le condizioni per effettuare il passaggio da uno stato all'altro: si clicchi due volte sulla prima transizione e sotto `Condition` impostare `TASTO='1'`; poi si vada alla seconda e si imposti `TASTO='0'` e analogamente per le altre due.

In caso di modifiche/cancellazioni, è possibile eventualmente aggiungere/creare stati, cliccando su , nuove transizioni con il tasto , oppure reset aggiuntivi col tasto . Si possono modificare i colori tramite i pulsanti appropriati a lato dell'area di lavoro.

Lo schema dovrebbe risultare simile alla figura qui sotto

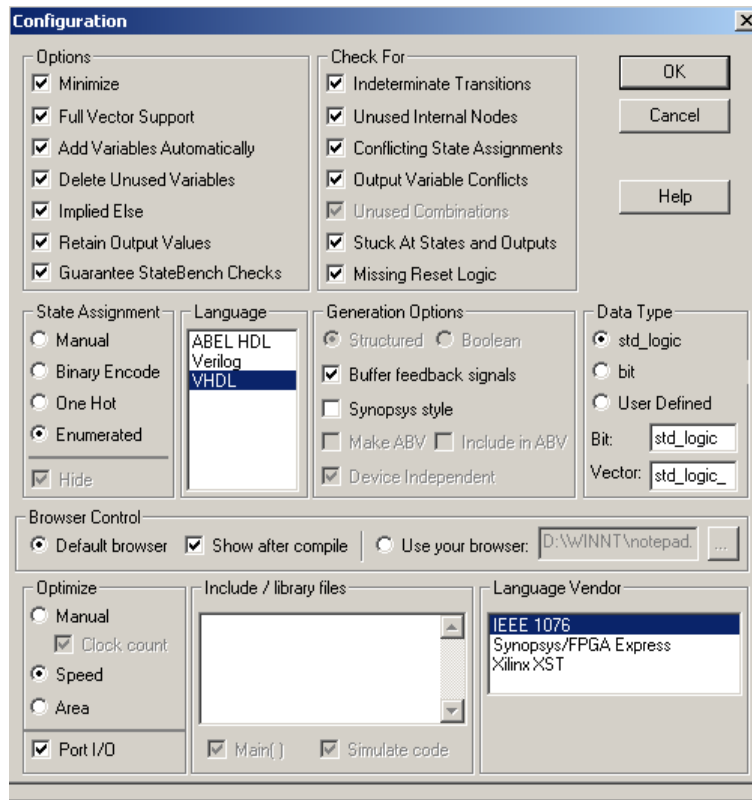


Se si clicca col tasto destro sull'area di lavoro e si sceglie Variables, si dovrebbe ottenere una lista di variabili simile alla seguente



in cui, cliccando sopra ad una determinata variabile, è possibile modificarne il tipo (Type) ed il fronte di attivazione (Active). Cliccando col tasto destro nell'area di lavoro si può inoltre accedere al menu di configurazione (Configuration), in cui in particolare interessa la scelta del linguaggio in cui dev'essere

tradotta la macchina a stati una volta finita: in questo caso è il VHDL secondo lo standard IEEE 1076 come mostrato di seguito



Si confermi, si salvi e si compili (`Generate HDL`). Se tutto è andato per il meglio, si avrà una schermata di conferma ed una successiva finestra in cui compare il sorgente VHDL in cui è stata tradotta la macchina a stati.

N.B. E' buona regola controllare sempre il listato prodotto per una ulteriore verifica di quello che si è progettato e della sua successiva corretta traduzione.

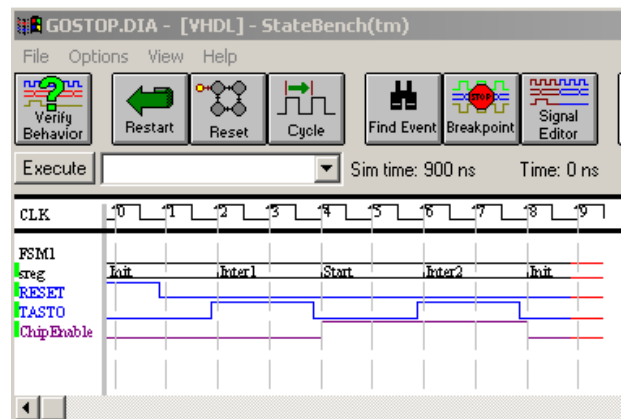
Si potrebbe, ora, simularne il funzionamento con il simulatore incluso nello StateCAD: si clicchi quindi sull'icona che porta il nome di `StateBench`.

Si hanno due scelte principali:

- effettuare una simulazione automatica, che prevede una transizione completa su tutti gli stati della macchina (`Automatic TestBench`);
- simulare passo passo.

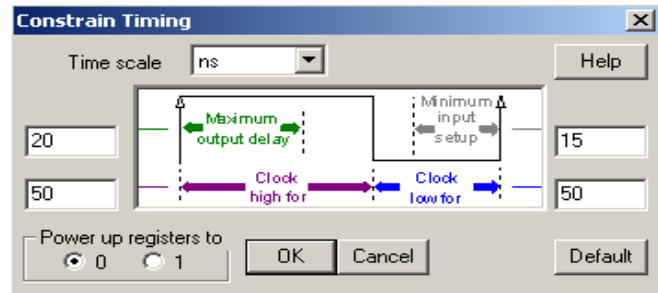
Nel primo caso al termine viene visualizzata anche una schermata di report che include eventuali errori riscontrati ed il numero di cicli di clock per eseguire l'intera simulazione, nonchè il suo tempo di esecuzione.

Per questo progetto si avrà un risultato simile a quello mostrato a fianco.



Nel questo secondo caso, invece, sotto Options Constrain Timing è possibile settare il tipo di clock, scegliendone la scala, i tempi in cui il fronte è alto/basso ed i fronti di salita/discesa.

Si confermi e si ritorni alla schermata principale.



E' possibile scegliere i segnali da valutare, cliccando su Signal Editor.

Per iniziare la simulazione bisogna innanzitutto portare la macchina a stati finiti nel suo stato di reset; si clicchi quindi sul pulsante denominato Reset (se quest'operazione non viene eseguita compare una finestra di Warning in cui si avverte che non si hanno inizializzazioni).

Cliccando poi di volta in volta su Cycle, è possibile procedere con la simulazione un ciclo di clock alla volta.

Inoltre, ad ogni passo, cliccando due volte su ciascun ingresso, si possono valutare istantaneamente le uscite prodotte dalla macchina a stati e gli stati attraversati.

Cliccando due volte su un'uscita, invece, è possibile settarne il valore atteso, che può essere eventualmente reimpostato nel caso differisca da quello prodotto.

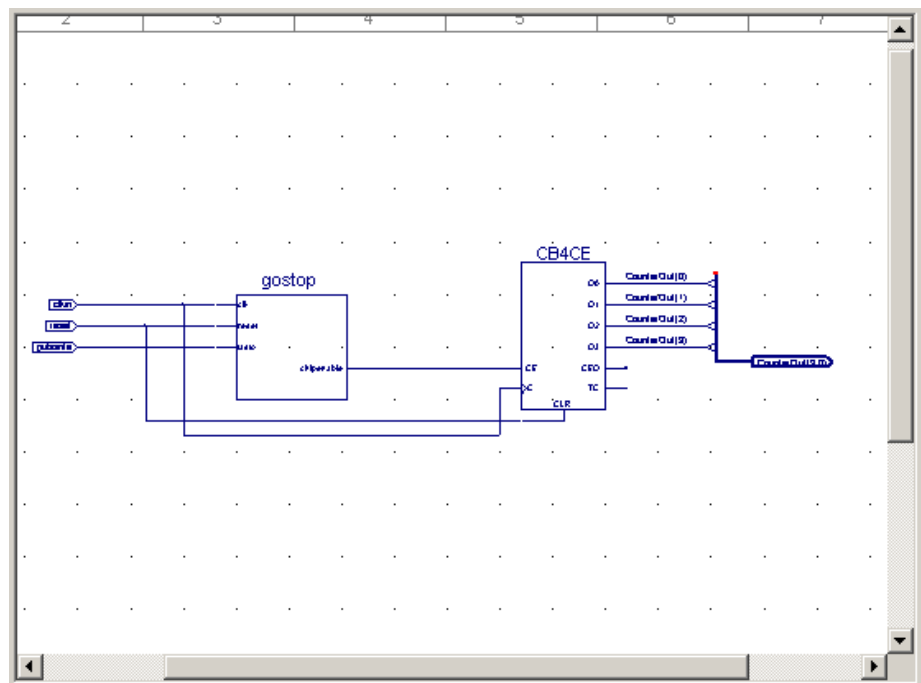
Si chiuda lo StateCad e si ritorni al Project Navigator.

Ora al progetto si aggiunga la sorgente esistente data dal codice VHDL della macchina a stati finiti appena creata: è il file che porta il nome del diagramma ed ha estensione vhd.

Si crei lo schematic symbol.

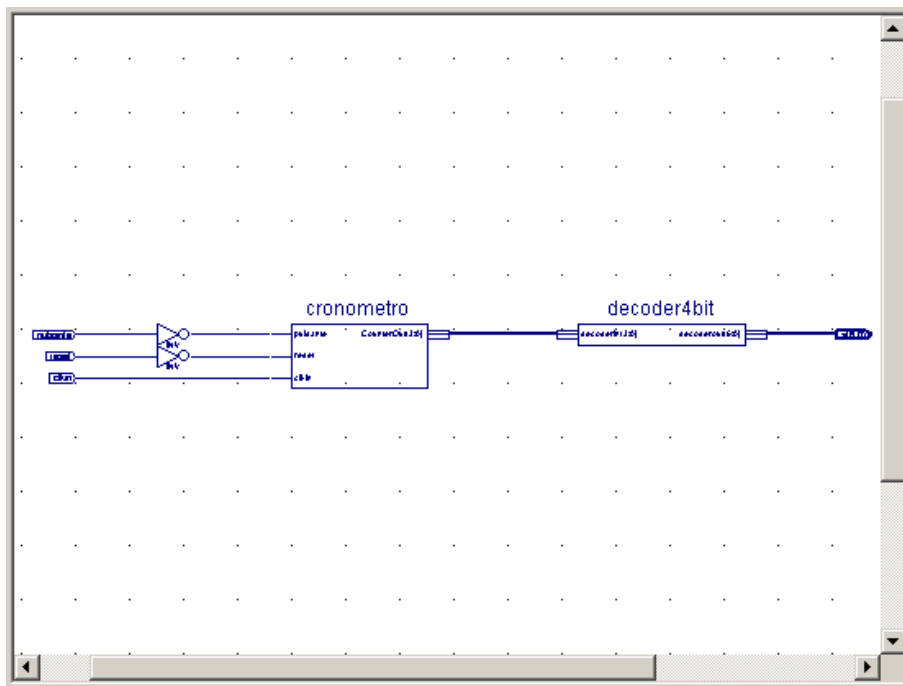
Si aggiunga al progetto un nuovo schematico ad esempio col nome *cronometro*, in cui si collega l'uscita del simbolo che rappresenta la macchina a stati (in questo caso la fsm è stata chiamata *GoStop*) con il *Chip Enable* del contatore *CB4CE*; si estraggano le uscite del contatore e si riuniscano in un unico bus; si estraggano infine anche il tasto, il reset ed il clock:

Si esegua quindi la verifica degli errori e si salvi.



Ora se ne crei lo schematic symbol e si aggiunga un nuovo schematico, in cui si collegherà il cronometro con un decoder per display a sette segmenti: ora, si può riciclare il decoder creato nel

tutorial precedente, oppure ricrearlo da capo (aggiungendo una nuova sorgente del tipo VHDL Module e completando il listato con i valori desiderati).
 Si estraggano ingressi ed uscite, ottenendo



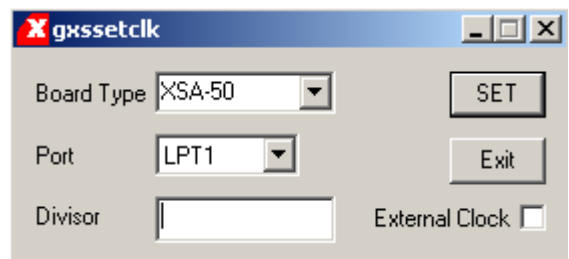
Ora si può effettuare una simulazione comportamentale, per valutarne il corretto funzionamento; nonchè si può sintetizzare e valutare quindi se la frequenza di lavoro della scheda è appropriata per il circuito ed anche se i ritardi visualizzati nel report sono accettabili.
 E' consigliato anche passare per i processi di traduzione, mappatura e piazzamento delle piste con relative simulazioni.

Si osservi che però vi è un problema che è stato sottovalutato finora: il clock con cui lavora la macchina a stati e, più in generale, tutto il cronometro è quello della board, cioè 50MHz, il che significa che si ha un campionamento ogni 20ns, ovvero il contatore incrementa ogni 20ns, che è un tempo troppo veloce per essere visualizzato sul display.

Bisogna allora impostare una frequenza ragionevole: questo può essere fatto o andando direttamente a modificare il clock sulla board con il software apposito della Xess, oppure aggiungendo allo schematico un opportuno divisore di clock.

Seguendo la prima via:

- si lanci il tool della Xess GXSSSETCLK;
- si imposti il tipo di board;
- si setti la porta a cui è collegata la scheda tramite cavo 25 poli;
- si scelga il divisore di clock voluto, tenendo conto che l'oscillatore della scheda produce un clock a 100MHz: si può scegliere quindi un fattore di divisione da 1 a 2052, che permette un range di frequenze che vanno da 100MHz a 48KHz rispettivamente;
- Si confermi cliccando su SET;

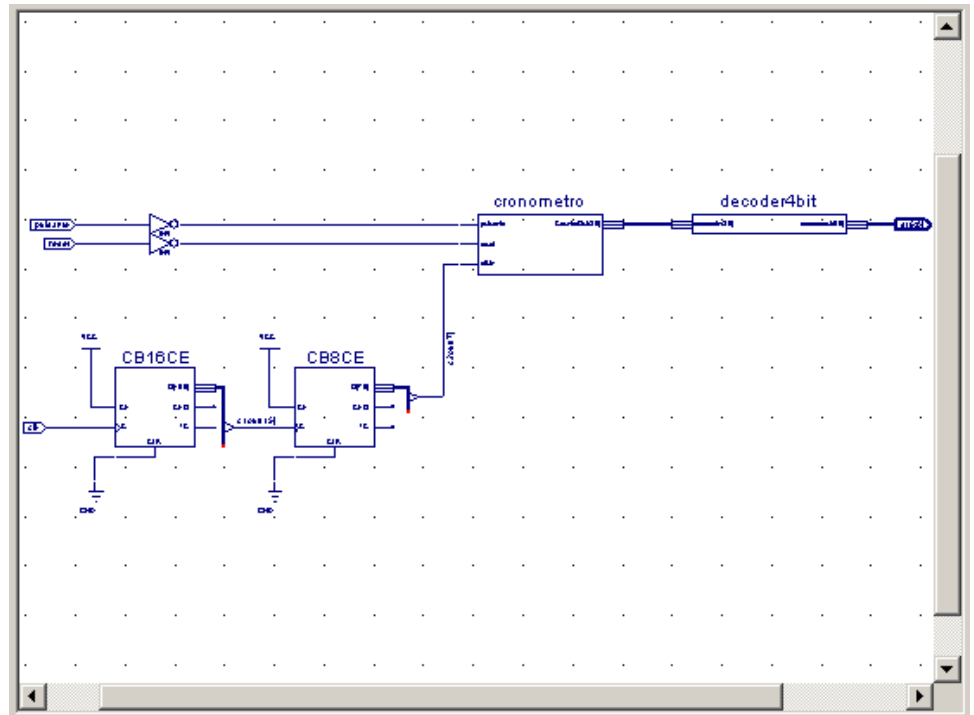


il tutto va eseguito con la board collegata al Pc ed opportunamente alimentata.

Per questo progetto anche la frequenza di 48KHz risulta però eccessiva, quindi è meglio scegliere la seconda soluzione.

Il divisore di clock non è altro che un contatore che ha in ingresso il clock di sistema; se si prende un CB16CE e si preleva il sedicesimo bit in uscita, questo cambierà ogni 2^{16} cicli di clock in ingresso e con un clock di 50MHz questo corrisponde ad un cambiamento ogni 1.3ms: è ancora troppo veloce!

Si metta allora in cascata un ulteriore contatore, ad esempio un CB8CE, in cui in ingresso si collega l'uscita del precedente ed in uscita si prelevi l'ottavo bit e lo si inietti nel clock del cronometro, in questo modo si ha un incremento ogni tre decimi di secondo circa.



Con questo procedimento, al check error si otterranno diversi "warning", legati al fatto che quasi la totalità delle uscite dei due contatori sono rimaste sconnesse. Si può quindi salvare il progetto e passare alla fase successiva.

Si risintetizzi e si valuti se la frequenza di lavoro è appropriata.

Si aggiunga ora un file di vincoli, in cui si associano le uscite al display a sette segmenti della scheda, il pulsante in ingresso al tasto sulla board ed il reset ad uno switch. Il clock viene impostato automaticamente se lo si è nominato con la parola riservata clk. Il listato del file di vincoli dovrebbe apparire simile al seguente

```
#PINLOCK_BEGIN
NET "pulsante" LOC = "P93";
NET "clk" LOC = "P88";
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";
NET "reset" LOC = "P54";
#PINLOCK_END
```

Si esegua quindi l' Implement Design e si esegua il download sulla scheda.