Laboratorio di Sistemi per la Progettazione Automatica
a.a. 2008/09

Giuseppe Di Guglielmo
Università degli Studi Di Verona
Dipartimento di Informatica

Revisione: mercoledì 18 marzo 2009 - 14.22

Lezione 3: HDL Designer

In questa lezione vengono introdotti gli aspetti basilari della modellazione di un dispositivo digitale mediante HDL Designer a partire dalla specifiche ad alto livello. HDL Designer è un prodotto di Mentor Graphics.

Questo tutorial è stato realizzato utilizzando la versione **2008.1 HDL Designer** su piattaforma Linux. Di seguito i comandi riportati con sfondo grigio (`$`) sono da intendersi come comandi della console Linux, mentre i comandi riportati con sfondo **rosa e grassetto** indicano i percorsi da seguire o i pulsanti dell'interfaccia grafica. Infine, i comandi riportati con sfondo rosa (`>`) sono da intedersi come comandi di **ModelSim**.

Introduzione

La lezione mostra come creare un dispositivo digitale mediante la sua modellazione come *Diagrammi a Blocchi* e come *Diagrammi di Stato*. Questa attività è una fase fondamentale del flusso di progettazione di un sistema *embedded*, tale flusso è schematizzato in Figura 1.

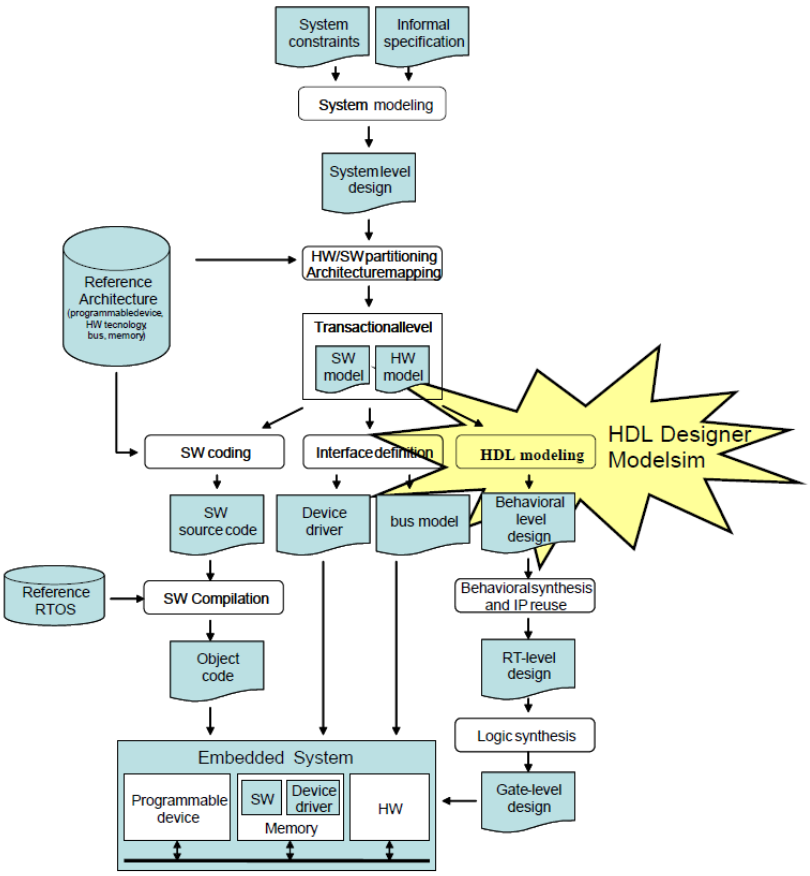

Figura 1: Flusso di progettazione di un sistema *embedded*.

Molti prodotti commerciali aiutano il progettista in questa fase di progettazione. HDL Designer è uno fra i più diffusi e apprezzati.

HDL Designer

Una guida completa all'utilizzo di HDL Designer è disponibile in
`/opt/EDA_Software/mentor/hdl_designer/2008.1a/docs/pdfdocs`
Solo i concetti chiave del suo utilizzo vengono presentati in questa lezione.

HDL Designer permette di progettare un sistema utilizzando differenti modelli descrittivi, come i Diagrammi a Blocchi, i Diagrammi di Stato, le Tabelle di Verità, e i Diagrammi di Flusso.

Diagrammi a blocchi

Il modello dei Diagrammi a Blocchi viene utilizzato per rappresentare il dispositivo mediante un insieme di componenti comunicanti. Figura 2 mostra un esempio. Il sistema è rappresentato come un modulo ad alto livello, per il quale vengono definiti porte in ingresso e uscita, segnali e ogni sotto-componente del sistema.

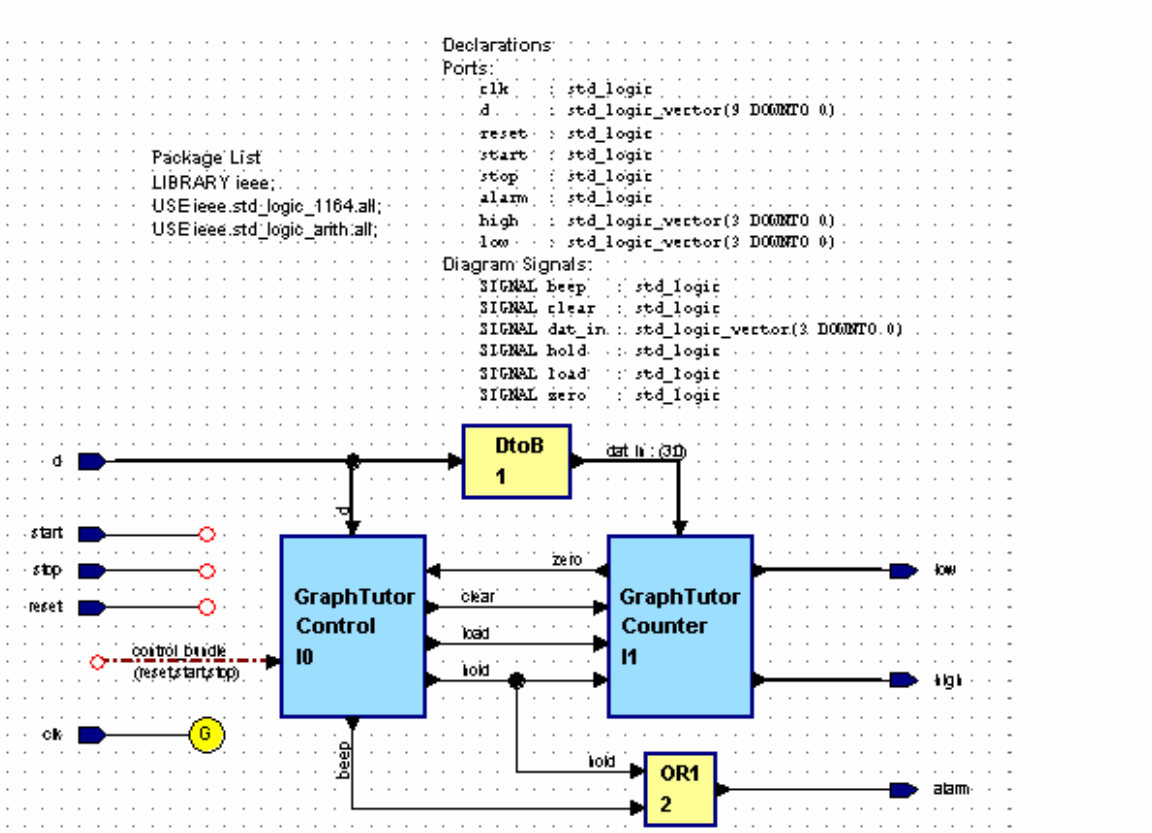

Figura 2: Esempio di diagramma a blocchi.

Tutte le componenti appaiono nel diagramma come delle *scatole nere*. A questo livello, infatti, il progettista si preoccupa esclusivamente della connessione dei componenti.

Diagrammi di stato

I Diagrammi di Stato permettono di rappresentare graficamente il comportamento del dispositivo mediante il modello delle Macchine a Stati Finiti. Il diagramma viene creato aggiungendo nuovi stati (specificando lo stato iniziale), nuove transizioni tra gli stati e definendo sia le condizioni che

le azioni per ciascuna transizione (nel caso di una macchina a stati finiti di Mealy). Un esempio di diagramma di stato creato con HDL Designer vien mostrato in Figura 3.

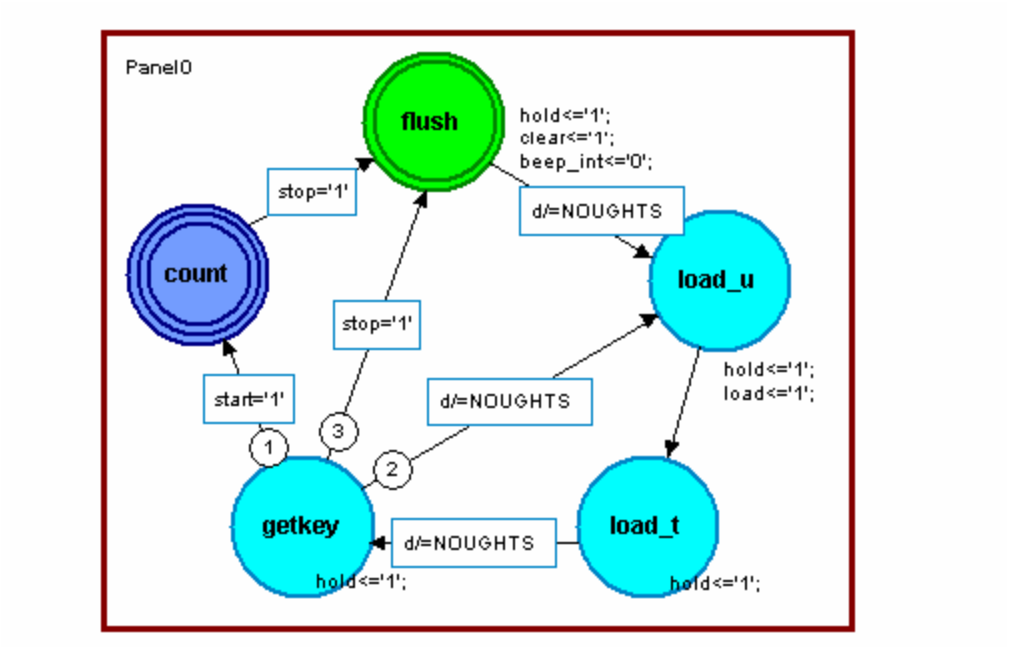

Figura 3: Esempio di diagramma di stato.

In particolare HDL Designer, mediante il pannello *State Machine Properties,* permette al progettista di selezionare le caratteristiche del codice HDL generato e di specificare la codifica della macchina a stati. Per esempio, quando l'FSM sarà completata, il progettista potrà scegliere di implementare un comportamento sincrono o asincrono, il tipo dei segnali di reset e clock, lo stato di *recovery*. Infine la rappresentazione della FSM può essere tradotta in codice HDL scegliendone lo stile, ad esempio se rappresentare a 2 o 3 processi, se utilizzare costrutti IF o CASE, etc.

Progettazione di un dispositivo digitale

Di seguito vengono brevemente riassunti alcuni passaggi per la progettazione di un semplice circuito sequenziale, mettendo in evidenza alcuni modelli/concetti/strumenti chiave.

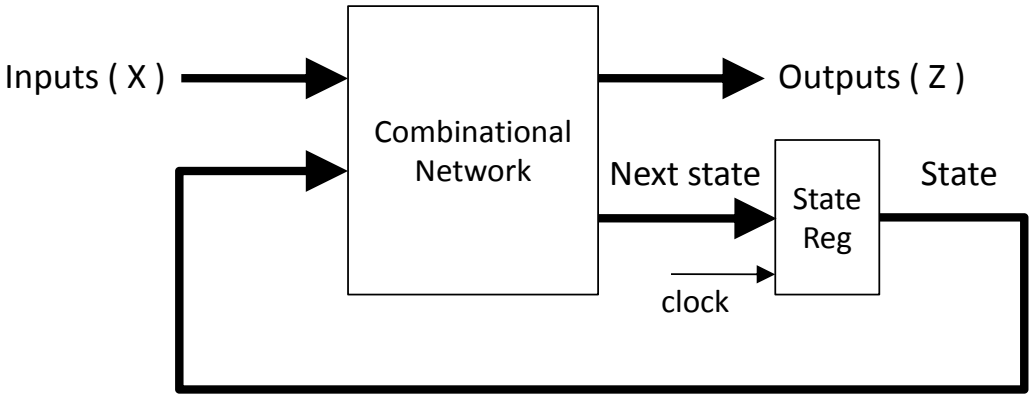

Figura 4: Modello generale di una Mealy Sequential Machine.

Un circuito sequenziale può essere rappresentato fondamentalmente mediante due modelli di macchine a stati finiti (FSM), modello di Mealy e modello di Moore. Nel modello di Mealy, i valori in uscita dipendono sia dallo stato presente sia dai valori in ingresso. Nel caso del modello di

Moore, i valori in uscita dipendono dal solo stato presente. Un modello generale per un circuito sequenziale di Mealy è rappresentato in Figura 4 e consiste in

- un circuito combinatorio, che genera i valori in uscita e lo stato prossimo;
- un registro di stato, che memorizza lo stato presente.

A partire dalle specifiche di un progetto è possibile definire una macchina a stati finiti come quella rappresentata in Figura 5 e in Figura 6, rispettivamente lo State Graph (Diagramma di Stati) e la State Table per la stessa FSM.

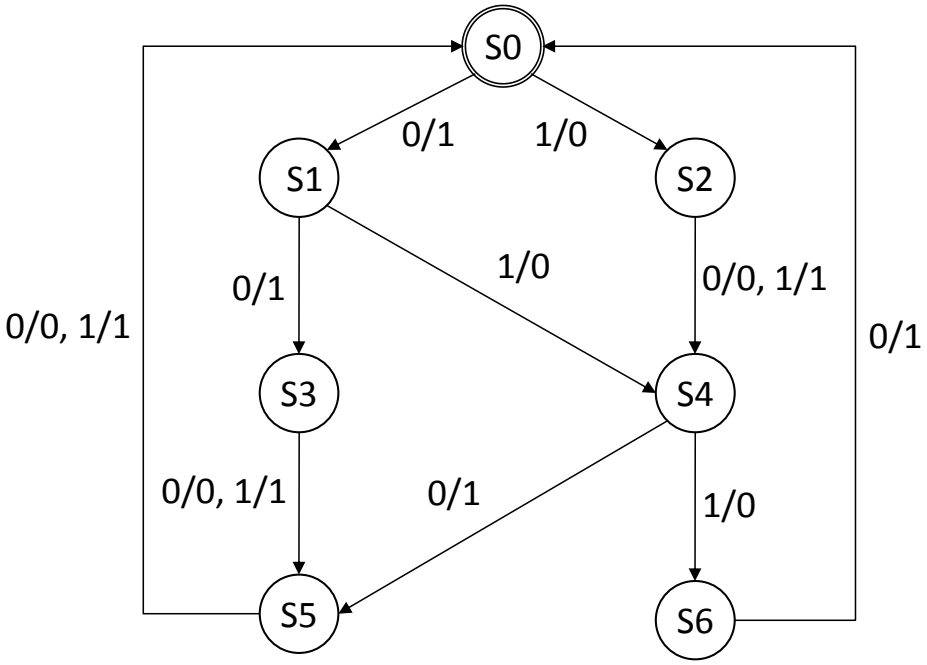

Figura 5: Mealy State Graph.

	NS		Z	
PS	X = 0	X = 1	X = 0	X = 1
S0	S1	S2	1	0
S1	S3	S4	1	0
S2	S4	S4	0	1
S3	S5	S5	0	1
S4	S5	S6	1	0
S5	S0	S0	0	1
S6	S0	-	1	-

Figura 6: State table.

Data una codifica per gli stati rappresentata in Figura 7 è possibile definire la Tabella delle Transizioni di Figura 8.

Q2Q3 \ Q1	0	1
00	S0	S1
01		S2
11	S5	S3
10	S6	S4

Figura 7: Assignment map.

$Q_1Q_2Q_3$	$Q_1^+Q_2^+Q_3^+$ X = 0	$Q_1^+Q_2^+Q_3^+$ X = 1	Z X = 0	Z X = 1
000	100	101	1	0
100	111	110	1	0
101	110	110	0	1
111	011	011	0	1
110	011	010	1	0
011	000	000	0	1
010	000	xxx	1	x
001	xxx	xxx	x	x

Figura 8: Transition table.

Infine è possibile definire mediante Mappe di Karnaugh (Figura 9) le funzioni per i segnali in uscita e per i segnali stato prossimo.

Il passaggio alla rappresentazione del circuito in Figura 10 è immediato.

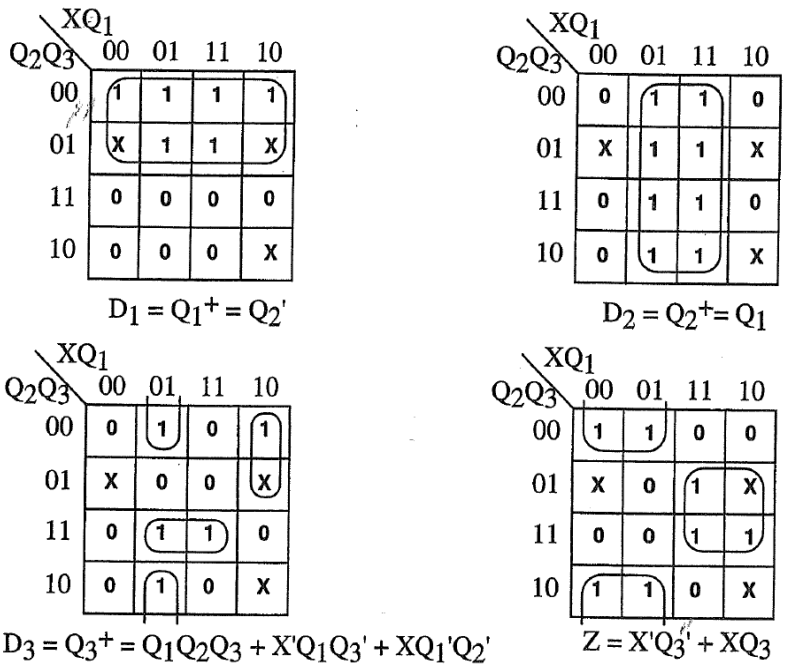

Figura 9: Mappe di Karnaugh.

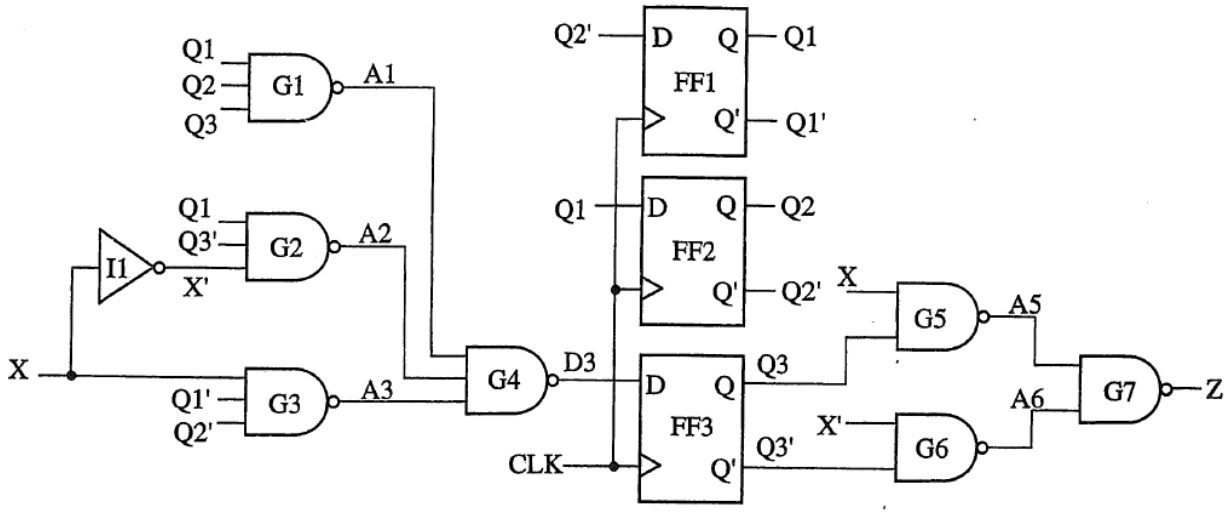

Figura 10: Realizzazione.

Realizzazione di un diagramma a blocchi mediante HDL Designer

In questa sezione viene descritto come creare il diagramma a blocchi di Figura 10. Per prima cosa è necessario autenticarsi su uno degli *host* del laboratorio ESD e quindi configurare l'ambiente.

Remote login

È necessario autenticarsi su una delle macchine del laboratorio ESD. Per far questo si può utilizzare una connessione *ssh* esportando il server grafico (*-X*) e richiedendo la compressione dei dati trasmessi (*-C*):

```
$ ssh -XC <_login_>@edalab-srv01.sci.univr.it
```

In alternativa, per esportare una sessione grafica, provare il comando:

```
$ X -query edalab-srv01.sci.univr.it
```

Impostazione variabili d'ambiente

È necessario esportare alcune variabili d'ambiente per poter utilizzare HDL Designer, in particolare sul terminale aperto remotamente, eseguire il comando:

```
$ source /opt/EDA_Software/start_eda.bash
```

Scegliendo in sequenza:

```
Mentor Graphics(1) -> Latest Version(1)
```

Questo script deve essere eseguito ogni qualvolta si apre un terminale remoto prima della sessione di progettazione.

Creazione directory di lavoro

Per prima cosa, al fine di mantenere organizzati e ordinati i vari file delle lezioni, è necessario creare una directory di lavoro.

```
$ mkdir lesson_3
$ cd lesson_3
$ mkdir ex_hds
$ cd ex_hds
```

Avvio di HDL Designer

Per avviare HDL Designer è sufficiente fornire il seguente comando sulla console linux:

```
$ hdl_designer
```

Avvertenza: attualmente il Laboratorio ESD dispone di 10 licenze di HDL Designer. Si chiede pertanto di eseguire questo tutorial in gruppi.

Creazione di un progetto

Dopo l'avvio di HDL Designer, è possibile creare un nuovo progetto.

- Selezionare
  File >> New >> Project…
- Nella finestra **Creating a New Project** inserire il nome del progetto, converter, e il percorso in cui posizionare i file di progetto, <…>/lesson_3/ex_hds.
- Selezionare **Next >**.
- È necessario importare una libreria di componenti che verranno instanziati in seguito, pertanto selezionare **Add existing designs files** e quindi **Finish**.
- Si aprirà la finestra **Add Existing Design**. Selezionare la voce **Copy Specified Files** e quindi **Please specify the required designs files >> Look in: >> Browse...**
- Selezionare come directory contenente i sorgenti:
  `/home/gdg/teaching/spa_lab/aa_2008_09/lesson_3/lib`
  Nella finestra **Content** selezionare il sorgente `bit_pack.vhd` e premere in sequenza **OK** e **Finish**.

Creazione del diagramma a blocchi

Ora è possibile definire le componenti del progetto. Si cominci aggiungendo un diagramma a blocchi come segue.

- Avviare il **Design Content Creation Wizard** con **File >> New >> Design Content…**
- Selezionare **Catergories >> Graphical View** e **File Types >> Block Diagram**. Assicurarsi che sia selezionato il campo **VHDL** e quindi premere **Next >**.
- Inserire il nome dell'`ENTITY` VHDL nel campo **Design Unit name:** `converter`.
- Premere **Finish**.
- Il pannello **Block Diagram** si presenta come un foglio millimetrato in cui trascinare le componenti e le connessioni.
- Premere il pulsante **Add Component**, si aprirà il **Component Browser**, trascinare sullo spazio di lavoro
  - 4 componenti `Nand3`
  - 3 componenti `Nand2`
  - 3 componenti `DFF`
  - 1 componente `Inverter`
- Cliccare su ciascuna componente col pulsante destro e dal menù a tendina selezionare **Object Properties…**
- Aggiornare il campo **Instance Name** con il nome dell'istanza della componente (si veda Figura 10). Se si volesse cambiare colore e forma alle componenti per migliorarne la visibilità, selezionare **Appearance** e quindi agire su **Color** e **Change Shape**.
- Per aggiungere porte in ingresso e uscita utilizzare **Add Signal with Port**.
- Per aggiungere segnali interni utilizzare **Add Signal**.

- Rappresentare il circuito di Figura 10, gestendo anche la corrispondenza dei nomi dei segnali, porte in ingresso uscita e componenti.

Per generare il codice HDL accedere a **Task >> Generate >> Run Single**. Eventuali errori vengono riportati in rosso nella finestra **Log Window**.
Per visualizzare il codice HDL generato premere il pulsante **View Generated HDL**.

Dopo aver completato la descrizione del Diagramma a Blocchi è possibile simularne il comportamento.

- Cliccare sullo spazio di lavoro la voce **Package List**.
- Selezionare in **Library** la voce `converter_lib` e in **Package** la voce `bit_pack`.
- Verificare che **Preview** sia `converter_lib.bit_pack` e quindi selezionare **Add**.
- Selezionare **Ok**.
- Selezionare il pulsante **ModelSim Flow** e nella finestra successiva premere **Ok**.
- Si avvierà **ModelSim**.
- Simulare il circuito con i seguenti comandi da fornire nel pannello *Transcript* di ModelSim:

```
VSIM > view wave
VSIM > add wave *
VSIM > force clk 0 0, 1 100 -r 200
VSIM > force x 0 0, 1 350, 0 550, 1 750, 0 950, 1 1350
VSIM > run 1600
```

Creazione del diagramma di stato

Un progetto può essere descritto mediante componenti rappresentate secondo modelli differenti. In questo caso si aggiunga un Diagramma di Stato come segue.

- Avviare il **Design Content Creation Wizard** con **File >> New >> Design Content…**
- Selezionare **Categories >> Graphical View** e **File Types >> State Diagram**. Assicurarsi che sia selezionato il campo **VHDL** e quindi premere **Next >**.
- Inserire il nome dell'`ENTITY` VHDL nel campo **Design Unit name:** `converter`.
- Premere **Finish**.
- Il pannello **State Diagram** si presenta come un foglio millimetrato in cui trascinare gli stati e le transizioni.
- Premere il pulsante **Add State** e posizionare un nuovo stato sullo spazio di lavoro.
- Cliccare sullo stato col pulsante destro e dal menù a tendina selezionare **Object Properties…**
    - Il campo **Outgoing transition for this state** permette di specificare
        - lo stile di codifica delle transizioni uscenti (IF, CASE)
        - loopback impliciti (se nessuna delle transizioni uscenti può essere soddisfatta rimani nello stato corrente)
    - I campi Entry Action, State Action e Exit Action sono da riempire nel caso si stia rappresentando una FSM di Moore e corrispondono al comportamento della FSM per lo stato corrente.
- Selezionare lo stato corrente e tramite copia-incolla creare in tutto 7 stati, corrispondenti agli stati della FSM rappresentata in Figura 5.
- Il pulsante Add Transition permette di aggiungere le transizioni fra stati.
- Dopo aver aggiunto una transizione, cliccare su di essa col pulsante destro e dal menù a tendina selezionare **Object Properties…**

- È possibile specificare per ciascuna transizione
  - Il comportamento rispetto al segnale di clock (alto, basso, etc)
  - Il comportamento rispetto al segnale di reset (reset sincrono, asincrono, etc)
  - La condizione che dovrà verificarsi per attivare la transizione. In questo caso il campo **IF Condition** permette di specificare tale condizione mediante sintassi VHDL, ed eventuali errori di sintassi vengono notificati.
  - Le azioni che verranno eseguite all'attivarsi della transizione. In questo caso il campo **Actions** permette di specificare tale condizione mediante sintassi VHDL (si è scelto VHDL come HDL per la descrizione della FSM corrente), ed eventuali errori di sintassi vengono notificati.
  - L'ordinamento secondo il quale valutare le transizioni (**Use Priority**).
- Completare la rappresentazione della FSM come in Figura 5.
- Cliccare sullo spazio di lavoro la voce **Package List**.
- Selezionare in **Library** la voce `converter_lib` e in **Package** la voce `bit_pack`.
- Verificare che **Preview** sia `converter_lib.bit_pack` e quindi selezionare **Add**.
- Selezionare **Ok**.

Attenzione: in VHDL le costanti numeriche di tipo bit e logic vengono rappresentate utilizzando gli apici, e.g. '1' e '0'.

Dopo aver rappresentato graficamente l'FSM è possibile specificare come codificarla in HDL, per fare questo accedere a **Diagram >> State Machine Properties…**
È possibile scegliere una serie di parametri di configurazione, in particolare

- Il campo **Machine** permette di decidere se la FSM è sincrona o asincrona;
- Il campo Generation Style offre le seguenti opzioni
  - 1 / 2 / 3 Process
  - IF / CASE

Si possono sperimentare le varie combinazioni e confrontare il codice HDL prodotto. Per generare il codice HDL accedere a **Task >> Generate >> Run Single**. Eventuali errori vengono riportati in rosso nella finestra **Log Window**. Per visualizzare il codice HDL generato premere il pulsante **View Generated HDL**.

Attenzione: se viene riportato come errore il fatto che manca la definizione del segnale di reset

- si acceda alla finestra principale **Design Manager** (Figura 11)
- si clicchi su `converter` per aprire la finestra **Interface**
- si aggiunga una nuova porta in ingresso, con nome `rst` e tipo `bit`.

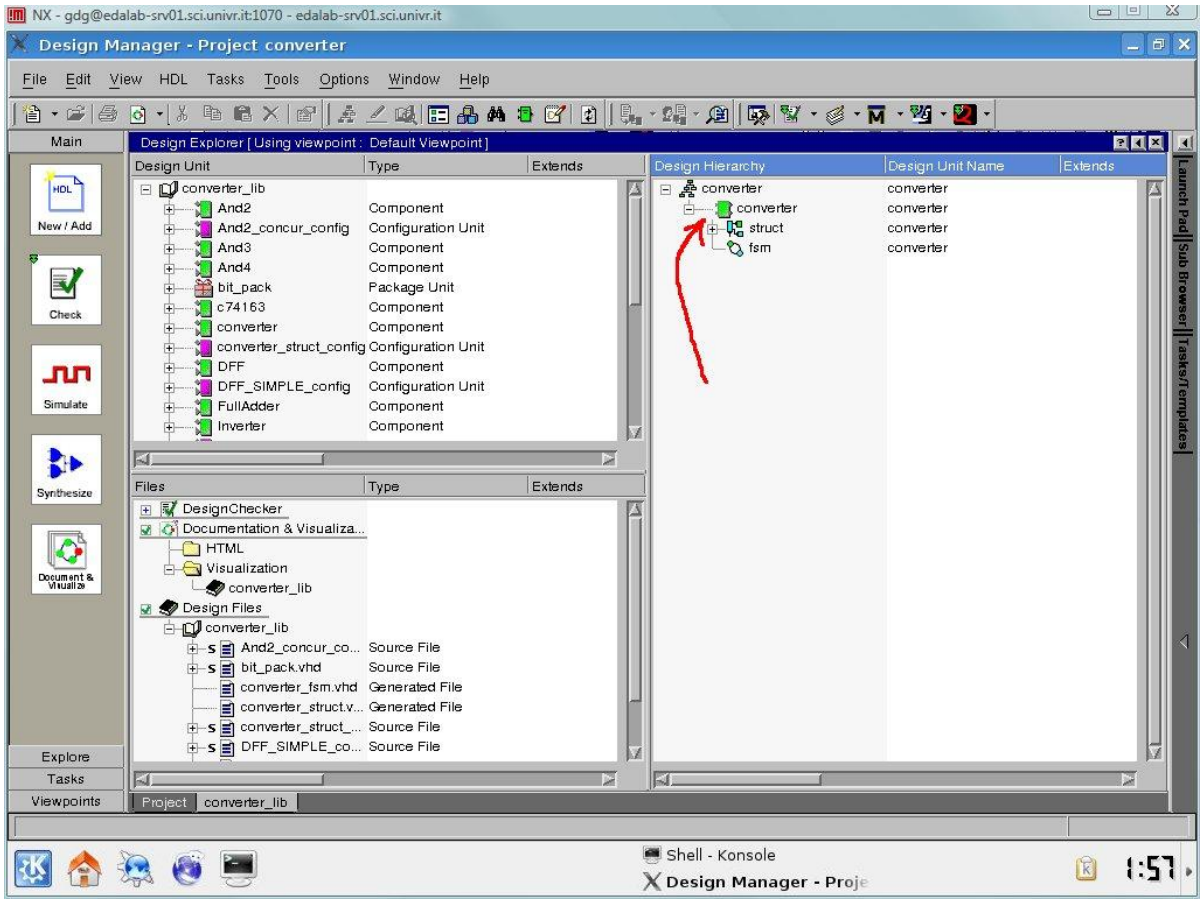

Figura 11: Cliccare sulla freccia rossa per accedere al pannello Interface

Dopo aver completato la descrizione del Diagramma di Stato è possibile simularne il comportamento.

- Selezionare il pulsante **ModelSim Flow** e nella finestra successiva premere **Ok**.
- Si avvierà **ModelSim**.
- Simulare il circuito con i seguenti comandi da fornire nel pannello *Transcript* di ModelSim:

```
VSIM > view wave
VSIM > add wave *
VSIM > force clk 0 0, 1 100 -r 200
VSIM > force x 0 0, 1 350, 0 550, 1 750, 0 950, 1 1350
VSIM > run 1600
```

Note: