

## Esercitazione n° 2

Lo scopo dell'esercitazione è di imparare a creare un semplice disegno con lo Schematic Editor riuscendo a simulare il comportamento con il tool ModelSim.

### Problema

Realizzare mediante un file sorgente di tipo schematico un semplice full adder, cioè un dispositivo che preveda in input tre ingressi indipendenti da sommare fra loro e in output due uscite, una per il bit di somma e l'altra per il bit del carry. Ciascun ingresso sia controllato da uno switch (ad es. dipsw1 e dipsw2 e dipsw3) e il risultato della somma venga visualizzato come segue sul display a 7 segmenti:



Vista la non complessità della visualizzazione sul display, non è richiesta la creazione di alcun elemento che realizzi in modo specifico la codifica.

Ad esempio la somma:  $1 + 1 + 1$  si realizzerà ponendo tutti gli switch in posizione ON.

### Suggerimenti

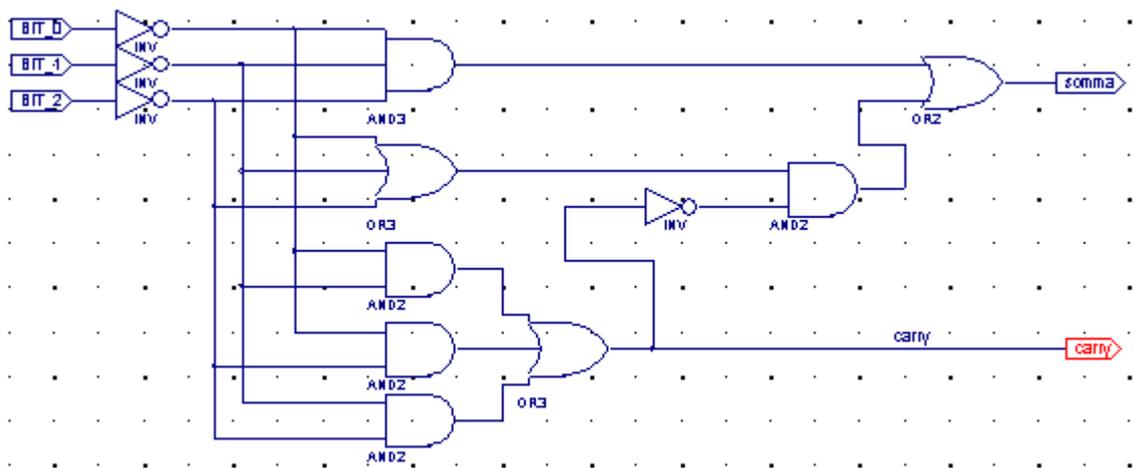
Si tenga conto del fatto che, allo scopo di rispettare le specifiche della board, i dipsw allo stato ON creano un collegamento a terra del circuito (consultare XSA User Manual), pertanto sarà necessario porre un invertitore per ogni ingresso al fine di rispettare la convenzione secondo cui ON  $\rightarrow$  1 e OFF  $\rightarrow$  0.

## Soluzione

Come per l'esercitazione n°1 è previsto l'uso di un file sorgente di tipo schematico.

- Una volta avviato il Project Navigator, creare un nuovo progetto (File → New Project). Seguendo la medesima procedura vista nella prima esercitazione, si assegni un nome al progetto, 'somma3bit' per esempio.
- Bisogna agganciare al progetto un file sorgente di tipo schematico (Project → New Source, Schematic); lo si chiami, ad esempio, 'somma3bit\_schem'. Si aprirà quindi la finestra dell'ECS.

Si costruisca il seguente diagramma:



L'invertitore 'inv', 'or2', 'and2', 'or3' e 'and3' si trovano in 'Symbols' nel gruppo 'Logic'. Si tratta dello schema di un full adder. Come già detto precedentemente, gli invertitori posti in corrispondenza degli ingressi hanno lo scopo di garantire la corrispondenza ON → 1 e OFF → 0 dei dipsw.

Si noti che in questo schematico non compaiono né buffer d'ingresso né buffer d'uscita. È una strategia possibile in virtù del fatto che nella fase di sintesi, come più in là si avrà modo di vedere, si può sfruttare l'opzione che aggiunge automaticamente i buffer di I/O dove necessario.

Salviamo il file e torniamo al Project Navigator.

- Con la medesima procedura vista nella prima esercitazione, si aggiunga il file sorgente per i vincoli (Project → New Source, Implementation Constraints File) e lo si chiami, ad esempio, 'somma3bit\_ucf'. Questo file è associato al file sorgente 'somma3bit\_schem', pertanto premeremo 'Avanti' e poi 'Fine'. Ora è necessario assegnare le location a ciascun ingresso e a ciascuna uscita. Consultando lo 'XSA Board User Manual' a pag. 24 si trovano sia le 'location' dei dipsw che quelle dei led del display a 7 segmenti. Le posizioni sono le seguenti:  
dipsw1 → p54  
dipsw2 → p64  
dipsw3 → p63

led\_s1 → p39  
led\_s2 → p62

Dalla finestra del Project Navigator, si proceda facendo doppio-click sul file 'somma3bit\_ucf.ucf' nel menu a tendina Module View posto in alto a sinistra; si aprirà la finestra del Constraints Editor.

Come visto nella prima esercitazione, si selezioni il menu a tendina 'Ports' del Constraints Editor.

Per assegnare le location è sufficiente fare doppio-click sulla casella 'Location' relativa alla porta desiderata.

La corrispondenza sarà la seguente:

Port Name	Port Direction	Location
bit_0	INPUT	p54
bit_1	INPUT	p64
bit_2	INPUT	p63
carry	OUTPUT	p62
somma	OUTPUT	p39

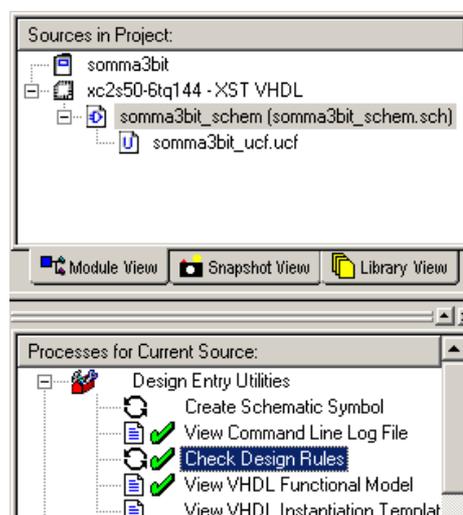
Verrà quindi generato il seguente file di testo 'somma3bit\_ucf.ucf':

```
NET "bit_0" LOC = "p54";  
NET "bit_1" LOC = "p64";  
NET "bit_2" LOC = "p63";  
NET "somma" LOC = "p39";  
NET "carry" LOC = "p62";
```

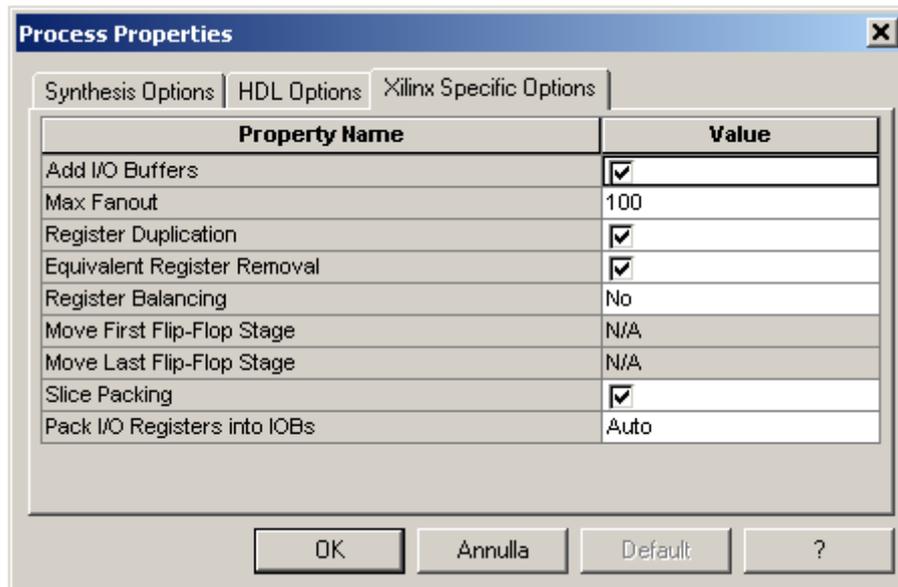
A questo punto il progetto è pronto per essere sottoposto ai seguenti processi.

*Attenzione: nell'eseguire le seguenti operazioni, si consiglia di osservare di volta in volta la finestra 'Console' posta nella parte bassa del Project Navigator. Vi si trovano indicazioni utili sul modo di procedere del tool nel corso delle operazioni. Qualora si verifici un errore, esso sarà notificato nella finestra 'Console'.*

- verifica: doppio click su 'Check Design Rules' dal 'Process View'



- sintesi: per ciò che concerne questa procedura si vada a verificare quanto osservato durante la creazione dello schematico 'somma3bit\_schem', ovvero l'esistenza di un'opzione che prevede l'inserimento automatico dei buffer di I/O. L'opzione è accessibile nel seguente modo: dal 'Project Navigator', selezionare 'Properties' premendo con il tasto destro del mouse su 'Synthesize' nel menu 'Project View'; accertarsi che vi sia un tick sull'opzione 'Add I/O Buffers' del menu a tendina 'Xilinx Specific Options':



Si avvii quindi il processo di sintesi con un doppio click su 'Synthesize' dal 'Process View'.

- Implementazione: doppio click su 'Implementation Design' dal 'Process View'.
- Creazione del file per la programmazione della board: doppio click su 'Generate Programming File' dal 'Process View'. Verrà creato il file 'somma3bit\_schem.bit' da caricare sulla XSA Board con la procedura standard.

## Simulazione con ModelSim

Si suggerisce di effettuare la simulazione sia prima che dopo la sintesi e di comparare tra loro i risultati ottenuti

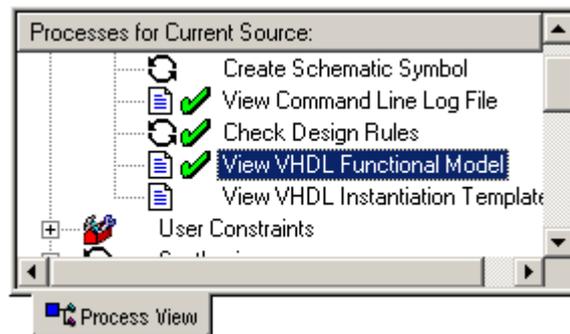
Per la simulazione del dispositivo appena creato sarà necessario utilizzare ModelSim.

La procedura sotto riportata si può applicare quando non si voglia fare uso del tool Foundation ISE5.2i, altrimenti può risultare più agevole generare un "testbench" per il circuito in esame ed impiegarlo quale generatore di segnali per il circuito da analizzare.

- Avviare ModelSim. Qualora compaia la finestra con le 'New ModelSim Features' premere 'Jumpstart', poi click sull'ipertesto 'Create a Project'. Tutta questa procedura può essere fatta ugualmente dalla finestra principale, ove compare il prompt di ModelSim:
- Si assegni al progetto un nome, ad esempio 'full\_adder', e la directory di lavoro desiderata nella 'Project Location'.

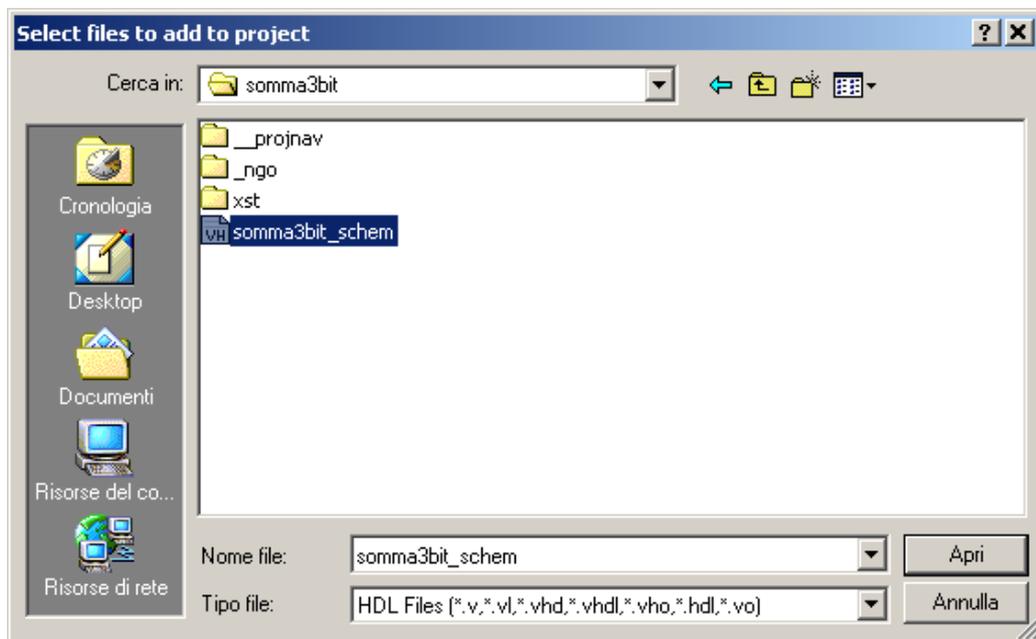
- Ora bisogna aggiungere i file che costituiscono il progetto da simulare, nel nostro caso tutti i file sorgente del progetto 'somma3bit'.

Attenzione: ModelSim accetta solo i file in formato HDL pertanto sarà necessario trasformare tutti i file sorgente schematici in file HDL. A questo scopo, si osservi la finestra 'Process View' dal Project Navigator, relativa al file sorgente di tipo schematico; si può notare che nel momento in cui si è fatto il 'Check Design Rules' è comparso un tick verde anche alla voce 'View VHDL Functional Model':



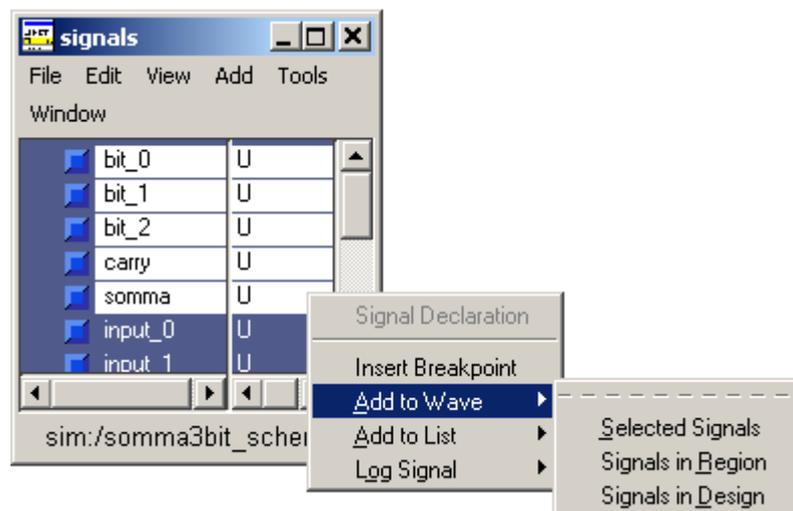
Con un doppio click su 'View VHDL Functional Model' si apre nella finestra a lato il listato del codice VHDL che descrive lo schematico. Il file generato è del tipo \*.vhf pertanto l'estensione va modificata in \*.vhd semplicemente risolvando il file come 'NomeFile.vhd' (File → Save As → NomeFile.vhd).

A questo punto ritorniamo al ModelSim e con un doppio click su 'Add Existing Files' aggiungiamo tutti i file sorgente (in questo caso solo il file 'somma3bit\_schem') al progetto ModelSim.



Nella finestra successiva 'Add file to Project' esiste la possibilità di copiare i file selezionati nella directory di lavoro del progetto ModelSim selezionando 'Copy to project directory' oppure di prelevare i file dalla loro posizione corrente selezionando 'Reference from current location'.

- Il passo successivo consiste nel compilare i file aggiunti, nel nostro caso solo uno. A tale scopo premere il tasto destro del mouse all'interno del menu a tendina 'Project' a sinistra, poi selezionare 'Compile' → 'Compile All'. A questo punto il file compilato è stato copiato nella cartella 'work' del menu a tendina 'Library'. Passiamo alla fase di simulazione vera e propria.
- Per avviare la simulazione di un determinato file, è sufficiente fare doppio click sul file che si desidera simulare nella cartella 'work' del menu 'Library'. Nel nostro caso 'somma3bit\_schem'. Vengono quindi caricati tutti i dati relativi al file in esame e si apre il menu 'sim'.  
Per avere un quadro completo della situazione è ora consigliabile aprire tutte le finestre relative alla simulazione, anche se poi ne utilizzeremo solo alcune: dalla finestra principale di ModelSim 'View' → 'All Windows'.  
Ci serviranno, oltre a quella principale dalla quale scriveremo i comandi per generare i segnali, saranno le finestre 'signals' e 'wave', pertanto per ora chiuderemo le altre.  
Si tratta ora di scegliere i segnali che si vogliono analizzare, cioè la totalità dei segnali che si vogliono controllare e osservare. Ad esempio si selezionino (ctrl + tasto sinistro del mouse): 'bit\_0', 'bit\_1', 'bit\_2', 'somma' 'carry'. Poi col tasto destro del mouse di selezioni 'Add to Wave' → 'Selected Signals':



I segnali selezionati verranno copiati nella finestra 'wave'. Dalla finestra wave è possibile comandare la simulazione mediante i tasti posti in alto.

Prima di far partire la simulazione è necessario generare gli stimoli per i segnali.

Il metodo più opportuno per creare dei segnali di test non è esattamente quelli di "forzarli" uno alla volta ad assumere determinati valori, ma piuttosto quello di realizzare un "testbench", ad esempio tramite una descrizione comportamentale in VHDL e poi usare quest'ultimo per simulare il funzionamento del circuito in esame a determinati stimoli

- Per generare gli stimoli la cosa più semplice è ritornare al prompt dei comandi. Il comando da utilizzare è 'force'. Digitando 'help force' dal prompt compariranno istruzioni relative alla sintassi:  
force [-freeze | -drive | -deposit] [-cancel <time>] [-repeat <time>] <item\_name> <value> [<time>] [, <value> <time> ...]

Utilizzeremo il comando force nei seguenti modi:

force <segnale> <valore> <fino a che tempo><unità di misura>.

Ad es. force dipsw1 1 10ns

Genera uno stimolo per il segnale dipsw1 portandolo al valore 1 fino a 10 ns dall'inizio della simulazione.

force <segnale> <valore> <fino a che tempo><unità di misura>, <valore> <fino a che tempo><unità di misura>, <valore> <fino a che tempo><unità di misura>, ... [..e così via]

Ad es. force dipsw1 1 10ns, 0 25ns, 1 27us

Genera uno stimolo per il segnale dipsw1 il quale assumerà il valore 1 fino a 10 ns dall'inizio della simulazione, poi assumerà il valore 0 fino a 25ns dall'inizio della simulazione e infine si nuovo il valore 1 fino a 27  $\mu$ s dall'inizio della simulazione.

Per generare un segnale periodico di un determinato periodo (utile ad esempio per simulare un segnale di clock) sarà sufficiente aggiungere alla fine della stringa -repeat <valore del periodo>.

Ad es. force dipsw1 1 10ns, 0 25ns, 1 27us -repeat 27us

Genera uno stimolo periodico di periodo 27  $\mu$ s, ripetendo ogni 27  $\mu$ s lo stimolo indicato (1 10ns, 0 25ns, 1 27us).

Per ulteriori informazioni relative al comando 'force' consultare lo 'User's Manual' ('Help' → 'PDF Documentation' → 'User's Manual').

Per quanto ci riguarda, poniamo di voler simulare la somma 1+0+1.

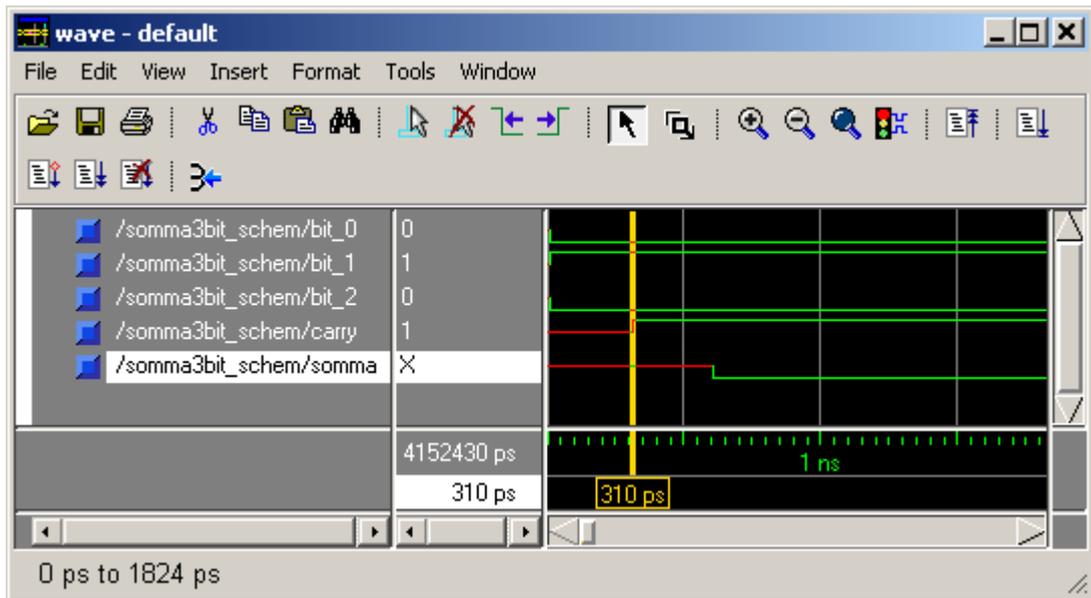
Attenzione: nel disegno si sono posti gli invertitori subito dopo i dipsw quindi bisogna creare gli stimoli al contrario; pertanto poniamo bit\_0 e bit\_2 al valore 0, bit\_1 al valore 1. Al prompt digitiamo:

```
force bit_0 0 10 -repeat 10
```

```
force bit_1 1 10 -repeat 10
```

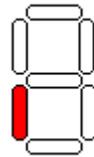
```
force bit_2 0 10 -repeat 10
```

- A questo punto torniamo alla finestra 'wave' e premiamo per esempio il pulsante  che corrisponde al comando 'run -all'. La simulazione parte ed è possibile controllarne l'avanzamento seguendo il contatore 'Now: ...'. Per bloccare la simulazione è sufficiente premere il tasto . A questo punto si può osservare il risultato nella finestra 'wave'.



Dalla simulazione è possibile vedere i tempi di propagazione dei segnali.

La somma  $1+0+1$  dà come risultato 10 quindi sarà solo il carry, collegato al led s2, ad essere a 1; somma sarà a 0 quindi il led s1 sarà spento. Sul display a 7 segmenti verrà visualizzato:



La simulazione ha confermato quanto ci aspettavamo.