

Pràctica 2: Introducció al flux de disseny i síntesis en FPGA

Durada: 1 sessió

Introducció

La Figura 5 mostra el diagrama RTL simplificat del mestre I²C. Alguns blocs, com el de control de bit y byte a la figura estan representats de forma genèrica como una maquina d'estats amb les entrades y sortides necessàries. En altres, mes senzills, com el bit timer es dona tot el detall d'implementació. Un cop definida l'arquitectura del nostre sistema digital, els següents passos són codificar a nivell RTL i verificar-ne la funcionalitat, per finalment sintetitzar el sistema complet. En aquesta pràctica ens familiaritzarem amb els passos bàsics a seguir pel flux de disseny i síntesis en FPGA. Per fer-ho ens centrarem en la implementació dels registres de desplaçament que es mostren al diagrama RTL de la Figura 5.

Objectius

- Codificar RTL de l'estructura descrita en base a les especificacions donades.
- Familiaritzar-se en l'ús de l'entorn de simulació ModelSim.
- Introducció al **self-checking** mitjançant l'ús de tasques, funcions pròpies del Verilog (*\$display*, *\$monitor*, *\$wait*, *\$force*, *\$release*, ...) i la monitorització de senyals internes del dispositiu a verificar (DUT, de l'anglès Device Under Test).
- Familiaritzar-se en l'ús de l'entorn de síntesis Quartus Prime d'Altera i **simulació post-implementació**.

Material

El material de suport el podeu descarregar del campus virtual:

- **tb_shiftreg.v** : testbench a completar.
- Plantilla informe de la pràctica.



Figura 5 Arquitectura del mestre I2C a implementar.

Tasques a realitzar

1. Dissenyeu un registre de desplaçament genèric no-cíclic parametrizable de N-bits, que anomenarem *shiftreg*. El registre de desplaçament ha de permetre la càrrega en paral·lel de dades. Ha de tenir una entrada i sortida en sèrie i la sortida en paral·lel del contingut del registre. El registre ha de tenir una senyal d'habilitació i un reset asíncron actiu per nivell baix. Els ports d'entrada i sortida seran doncs:

Clk	Relotge del sistema.
Rst_n	Reset asíncron actiu per nivell baix.
Load	Habilita la carrega de dades en paral·lel. Actiu per nivell alt.
En	Habilitació del desplaçament. Actiu per nivell alt.
DataIn	Entada de dades en paral·lel de N-bits.
DataOut	Sortida de dades en paral·lel de N-bits.
SerIn	Entrada de dades sèrie, que s'insereixen per el bit menys significatiu (LSB).
SerOut	Sortida de dades sèrie que es correspon amb el bit més significatiu (MSB).

- Examineu el fitxer *tb_shiftreg.v* que trobareu al campus virtual i instancieu el vostre registre de desplaçament com a DUT i verifiqueu-ne el funcionament.
- Completeu el test del registre de desplaçament creant la següent tasca:
test_serin, que carregui pel port d'entrada sèrie i comprovar que després de N cicles el valor de dins el registres es correspon al valor esperat. El valor esperat s'ha de calcular automàticament dins la mateixa tasca. Per tant, la tasca ha de tenir com a inputs: la dada a rebre pel port *SerIn*, i el nombre de desplaçaments que volem que fer.
- Utilitzeu la tasca del sistema *\$monitor* per veure l'estat de les entrades i sortides del registre de desplaçament en tot moment (excloent els senyals de *Clk* i *Rst_n*).
- Simuleu i verifiqueu el correcte funcionament per un registre de 8-bits per diferents valors d'entrada i nombre de desplaçaments.
- Verifiqueu visualment mitjançant el diagrama d'ones cada un dels test. Examineu els estímuls externs, les senyals internes i les sortides del vostre registre de desplaçament. També podeu examinar les variable de les tasques.
- Sintetitzeu el registre de desplaçament de 8 bits amb el Quartus per una Cyclone IVE model EP4CE22F17C6.
 - Creeu un projecte nou. Consulteu la guia de Quartus que teniu al campus virtual.
 - Elaboreu el disseny i comproveu quins missatge d'*error*, *info*, i *alertes* ús dona.
 - Examineu el esquema RTL generat per el Quartus. És l'esperat?
 - Creeu un fitxer de restriccions temporals (*shiftreg.sdc*) dins la carpeta misc, que contingui la següent comanda.

```
create_clock -name clk 100MHz -period 10.0 [get_ports Clk]
```

Aquesta comanda defineix un rellotge de 100 MHz al port d'entrada *Clk*. Aquest rellotge serà utilitzat per l'eina de síntesis per realitzar l'anàlisi temporal estàtic i comprova si el disseny implementat pot funcionar aquesta freqüència.

- Afegiu el fitxer SDC al projecte de Quartus.
 - Assigneu les entrades i sortides als pins de la FPGA indicats a la **Taula 2** de l'Annex *Pràctica 2 – Simulació netlist*.
8. Sintetitzeu de nou el disseny i examineu el *Compilation Report*.
- Compleix els requisits temporals? Quina és la freqüència màxima del disseny?
 - Hi ha alguna E/S del vostre mòdul sense definir (*unconstrained*)?
 - Quants i quins recursos s'utilitzen?
9. Configureu el projecte de Quartus per poder simular la *netlist* generada. Consulteu la guia *guia_SimulacióNetlisQuartus* que trobareu al campus virtual.
- Quina diferències veieu respecte al simulació RTL? Examineu el retard de les senyals respecte el flanc de rellotge.
10. Aquest apartat es optatiu. Per últim comproveu que efectivament funciona el registre de desplaçament amb la placa DE0-CV. Per fer-ho heu de seguir els següents passos:
- A. Canvieu el model de FPGA per el de la DE0-CV una **Cyclone V 5CEBA4F23C7N**. Per fer-ho d'anar al menú *Assignments > Device*.
 - B. Actualitzeu l'assignació de pins d'acord amb la **Taula 3** de l'Annex *Pràctica 2 – Demostració*.
 - C. Sintetitzeu de nou el disseny.
 - D. Carregeu-lo a la placa DE0-CV per comprovar que funciona. Heu de comprovar totes les seves funcions.

Entrega

Fer demostració al professor un cop s'ha implementat a la FPGA. En cas de no poder fer la demostració a l'aula, heu de gravar un vídeo que adjuntareu dins la carpeta doc. Recordeu que la demostració es optativa.

Un fitxer ZIP amb el directori de treball:

1. A la carpeta **rtl**: el codi RTL sintetitzable.
2. A la carpeta **tb**: el codi del testbench complet.

3. A la carpeta **misc**: el fitxer de restriccions temporals (.sdc).
4. Dins la carpeta **doc** hi heu de posar l'informe complimentat que trobareu al campus virtual, que constarà de:
 - Enumerar les tasques del banc de proves i explicar breument què fan.
 - Captures de les simulacions, amb una explicació breu i ressaltant les zones d'interès.
 - Captura del terminal de simulació amb els missatges de l'auto verificació.
 - Captura del esquema RTL resultant de la síntesi amb el Quartus (expandiu les caixetes!).
 - Taula on consti freqüència màxima d'operació, nombre de portes utilitzades de cada tipus i percentatge d'ocupació de la Cyclone V.

Annex Pràctica 2

Simulació netlist

Per la simulació de la netlist amb els retards de la lògica de la pràctica 2 farem servir els següents pins de la Cyclone IV-E muntada en una placa DE0-Nano (no és la mateixa que per la demostració):

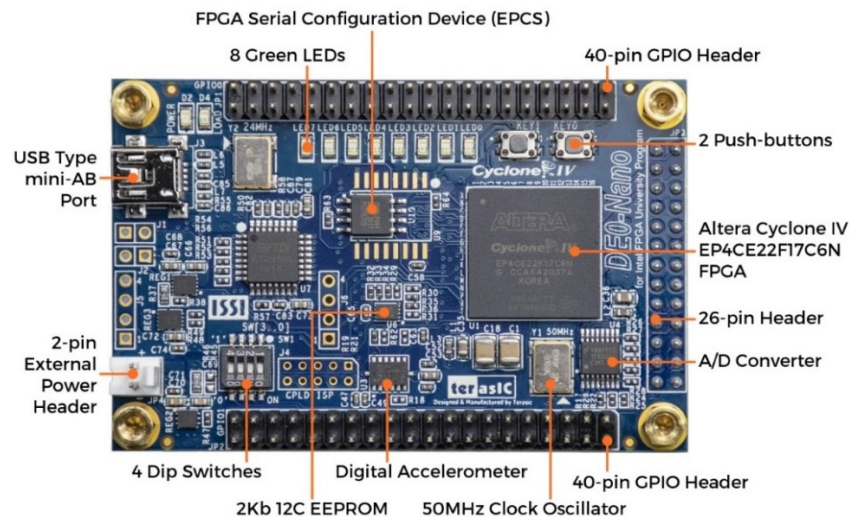


Figura 6 Vista superior de la placa de desenvolupament DE0-Nano amb els diferents elements identificats.

Taula 2 Assignació de pins de la Cyclone IV-E de la DE0-Nano.

Senyal	Element	Pin	Senyal	Element	Pin
En	SW[0]	M1	DataOut[7]	LED7	L3
SerIn	GPIO_0[1]	C3	DataOut[6]	LED6	B1
DataIn[7]	GPIO_0[10]	A6	DataOut[5]	LED5	F3
DataIn[6]	GPIO_0[9]	B6	DataOut[4]	LED4	D1
DataIn[5]	GPIO_0[8]	D5	DataOut[3]	LED3	A11
DataIn[4]	GPIO_0[7]	B5	DataOut[2]	LED2	B13
DataIn[3]	GPIO_0[6]	A4	DataOut[1]	LED1	A13
DataIn[2]	GPIO_0[5]	B4	DataOut[0]	LED0	A15
DataIn[1]	GPIO_0[4]	B3	Clk	CLOCK_50	R8
DataIn[0]	GPIO_0[3]	A3	Rst_n	KEY[0]	J15
SerOut	GPIO_0[2]	A2	Load	GPIO_0[0]	D3

Demostració

Per la demostració de la pràctica 2 farem servir els següents elements de la placa DE0-CV:

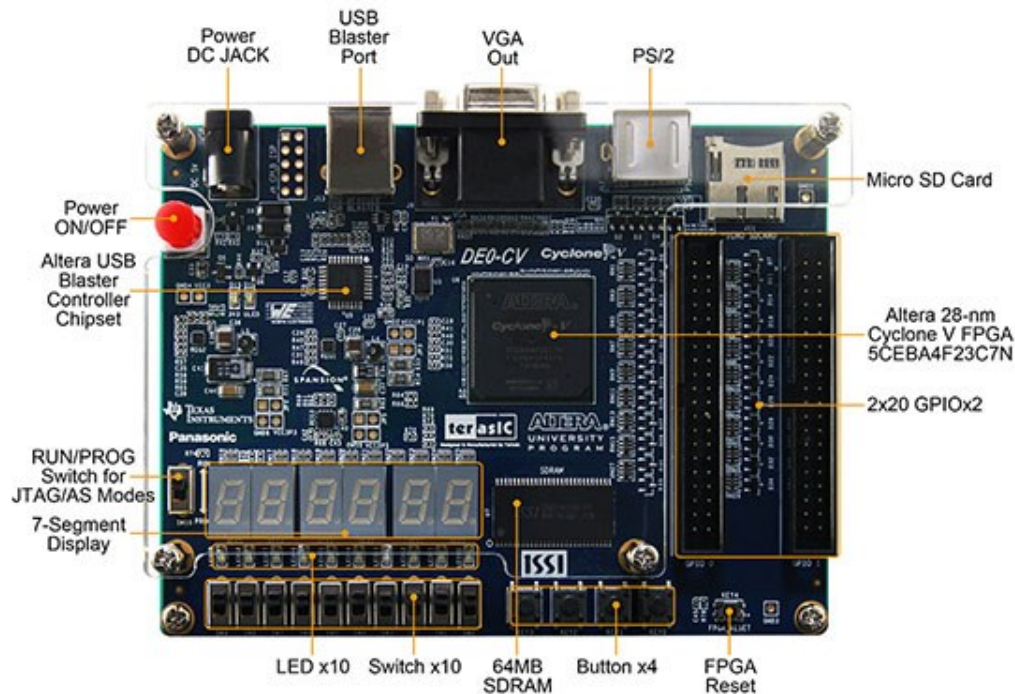


Figura 7 Vista superior de la placa de desenvolupament DE0-CV amb els diferents elements identificats.

Taula 3 Assignació de pins de la Cyclone V de la DE0-CV.

Senyal	Element	Pin	Senyal	Element	Pin
En	SW9	PIN_AB12	DataOut[7]	LED7	PIN_U1
SerIn	SW8	PIN_AB13	DataOut[6]	LED6	PIN_U2
DataIn[7]	SW7	PIN_AA13	DataOut[5]	LED5	PIN_N1
DataIn[6]	SW6	PIN_AA14	DataOut[4]	LED4	PIN_N2
DataIn[5]	SW5	PIN_AB15	DataOut[3]	LED3	PIN_Y3
DataIn[4]	SW4	PIN_AA15	DataOut[2]	LED2	PIN_W2
DataIn[3]	SW3	PIN_T12	DataOut[1]	LED1	PIN_AA1
DataIn[2]	SW2	PIN_T13	DataOut[0]	LED0	PIN_AA2
DataIn[1]	SW1	PIN_V13	Clk	KEY0	PIN_U7
DataIn[0]	SW0	PIN_U13	Rst_n	KEY1	PIN_W9
SerOut	LED9	PIN_L1	Load	KEY2	PIN_M7