Informe Pràctica 3 Sessió 1: Generador de polsos

Testbench

Taula 1: Llista dels diferents tasques del testbench i la seva funcionalitat.

Tasca	**Descripció**
Reset	Restableix tots els valors a un estat conegut
Wait_cycles	El sistema s'espera el nombre de cicles de rellotges escollit
Async_check	Comprova que no hi ha errors asíncronament
Check_errors	Comprova que la variable errors és 0
Test_and_result	Mostra els resultats del test
Load_ticks	Carrega el valor del comptador al timer
Test_hold	Comprova el correcte funcionament del timer permetent afegir el nombre de cicles que es vol que s'aturi. Mesura: - Que l'output sigui sempre l'esperat - El temps entre dos polsos consecutius - El nombre de cicles entre dos polsos consecutius
Test_spread (deprecated)	Permet distribuir uniformement els polsos d'espera. Desestimat degut a que no és el que demana l'enunciat però pot ser útil en futures ocasions.

Verificació Funcional

```
# [Info-   47.00 ns] Test counter
#
# ------------------------------------------------------------------------------------------------------------------
# [Info-   47.00 ns] Reset
# No ticks!
# ------------------------------------------------------------------------------------------------------------------
# ** TEST PASSED **
# [Results @ n =   0, StopCycles =   8] Errors =          0 | TimeGap =    0.00 ns | CyclesGap =          0
# ------------------------------------------------------------------------------------------------------------------
# ------------------------------------------------------------------------------------------------------------------
# [Info-   97.00 ns] Reset
# [Info-  167.00 ns] Successful check at time
# [Info-  167.00 ns] Successful check at time
# [Info-  187.00 ns] Successful check at time
# [Info-  187.00 ns] Successful check at time
# ------------------------------------------------------------------------------------------------------------------
# ** TEST PASSED **
# [Results @ n =   1, StopCycles =   8] Errors =          0 | TimeGap =   20.00 ns | CyclesGap =          1
# ------------------------------------------------------------------------------------------------------------------
```

Figura 1: Verificació funcional del comptador quan n = 0 i n = 1 (variable "ticks"). Es pot veure que en el primer cas s'emet un avís de que no hi ha un valor de "ticks" usable i que en el segon el comportament és el que s'espera. Encara que StopCycles = 8, com que no hi ha punt intermig entre 1 i 0, el comptador no s'atura → El "gap" entre polsos és d'un (1) cicle; 20ns.

```
# [Info-  187.00 ns] Reset
# [Info-  257.00 ns] Successful check at time
# [Info-  267.00 ns] Successful check at time
# [Info-  277.00 ns] Successful check at time
# [Info-  287.00 ns] Successful check at time
# [Info-  377.00 ns] Successful check at time
# [Info-  387.00 ns] Successful check at time
# [Info-  397.00 ns] Successful check at time
# [Info-  407.00 ns] Successful check at time
# [Info-  407.00 ns] Successful check at time
# [Info-  427.00 ns] Successful check at time
# [Info-  437.00 ns] Successful check at time
# [Info-  447.00 ns] Successful check at time
# [Info-  457.00 ns] Successful check at time
# [Info-  547.00 ns] Successful check at time
# [Info-  557.00 ns] Successful check at time
# [Info-  567.00 ns] Successful check at time
# [Info-  577.00 ns] Successful check at time
# [Info-  577.00 ns] Successful check at time
# ----------------------------------------------------------------------------------------------------------------------
# ** TEST PASSED **
# [Results @ n =   8, StopCycles =   8] Errors =           0 | TimeGap =  170.00 ns | CyclesGap =          16
# ----------------------------------------------------------------------------------------------------------------------
```

Figura 2: Verificació funcional del comptador quan n = 8. Es pot veure com amb n = 8 i StopCycles = 8, el "gap" entre polsos és de 16 cicles (8 de comptatge + 8 de stop) i que el temps entre aquests és de 170ns.

```
# [Info-  577.00 ns] Reset
# [Info-  647.00 ns] Successful check at time
# [Info-  657.00 ns] Successful check at time
# [Info-  667.00 ns] Successful check at time
# [Info-  677.00 ns] Successful check at time
# [Info-  687.00 ns] Successful check at time
# [Info-  697.00 ns] Successful check at time
# [Info-  707.00 ns] Successful check at time
# [Info-  717.00 ns] Successful check at time
# [Info-  807.00 ns] Successful check at time
# [Info-  817.00 ns] Successful check at time
# [Info-  827.00 ns] Successful check at time
# [Info-  837.00 ns] Successful check at time
# [Info-  847.00 ns] Successful check at time
# [Info-  857.00 ns] Successful check at time
# [Info-  867.00 ns] Successful check at time
# [Info-  867.00 ns] Successful check at time
# [Info-  887.00 ns] Successful check at time
# [Info-  897.00 ns] Successful check at time
# [Info-  907.00 ns] Successful check at time
# [Info-  917.00 ns] Successful check at time
# [Info-  927.00 ns] Successful check at time
# [Info-  937.00 ns] Successful check at time
# [Info-  947.00 ns] Successful check at time
# [Info-  957.00 ns] Successful check at time
# [Info- 1047.00 ns] Successful check at time
# [Info- 1057.00 ns] Successful check at time
# [Info- 1067.00 ns] Successful check at time
# [Info- 1077.00 ns] Successful check at time
# [Info- 1087.00 ns] Successful check at time
# [Info- 1097.00 ns] Successful check at time
# [Info- 1107.00 ns] Successful check at time
# [Info- 1107.00 ns] Successful check at time
# ----------------------------------------------------------------------------------------------------------------------
# ** TEST PASSED **
# [Results @ n =  15, StopCycles =   8] Errors =           0 | TimeGap =  240.00 ns | CyclesGap =          23
# ----------------------------------------------------------------------------------------------------------------------
```

Figura 3: Verificació funcional del comptador quan n = 15. Es pot veure com amb n = 8 i StopCycles = 8, el "gap" entre polsos és de 23 cicles (15 de comptatge + 8 de stop) i que el temps entre aquests és de 240ns.

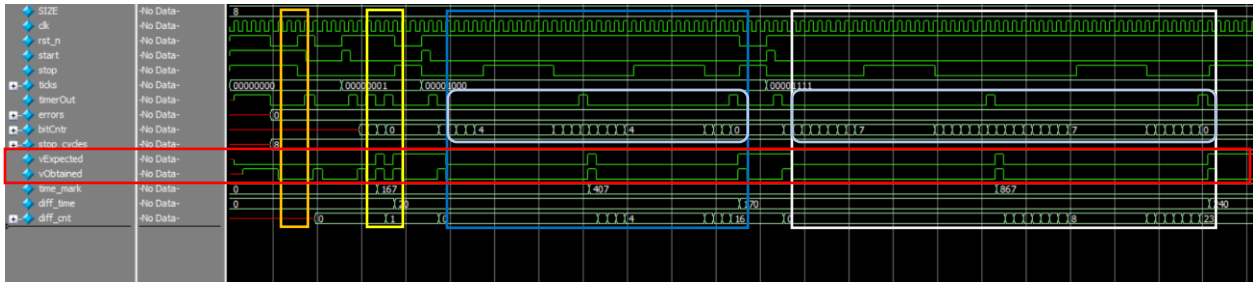

Figura 1: Diagrama d'ones de la simulació.

- Zona taronja (n=0): es veu que vExpected i vObtained són 0 tota l'estona.
- Zona groga (n=1): vExpected i vObtained són idèntics tot el període d'avaluació i no hi ha stop ja que no hi ha punt mig entre 0 i 1.
- Zona blava (n=8): vExpected i vObtained són idèntics tot el període d'avaluació i podem veure com el comptador es dilata 8 cicles quan arriba a 4 (la meitat del comptatge).
- Zona blanca (n=15): vExpected i vObtained són idèntics tot el període d'avaluació i podem veure com el comptador es dilata 8 cicles quan arriba a 7 (la meitat del comptatge).

Síntesis en FPGA

La **Figura 3** mostra el esquema RTL de la netlist generada amb el Quartus del timer.

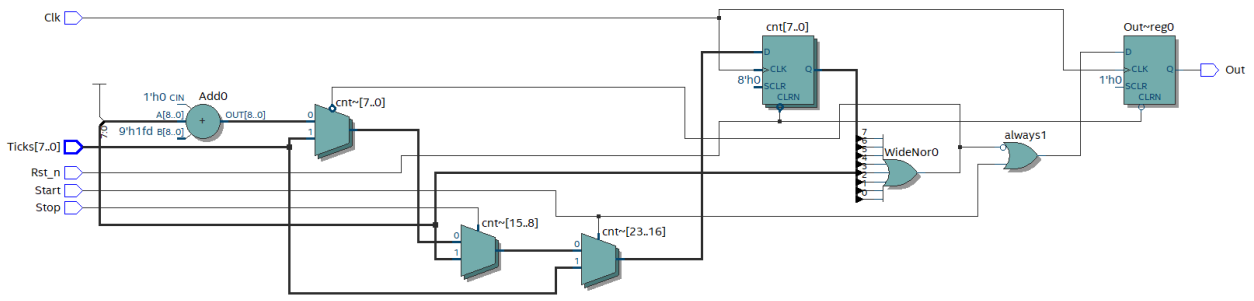

Figura 3: Esquema RTL de la netlist generada amb el Quartus.

Taula 2: Llista dels recursos utilitzats en la implementació de registre de desplaçament.

Recurs	Utilitzats	%
Pins E/S	13/224	6
Elements Lògics	7/18.480	<1
Registres	9	-
RAMs	0	0
DSPs	0	0
PLLs	0	0
FMAX	376,36MHz	-

Verificació Post-síntesis

A la **Figura 4** es mostra ...

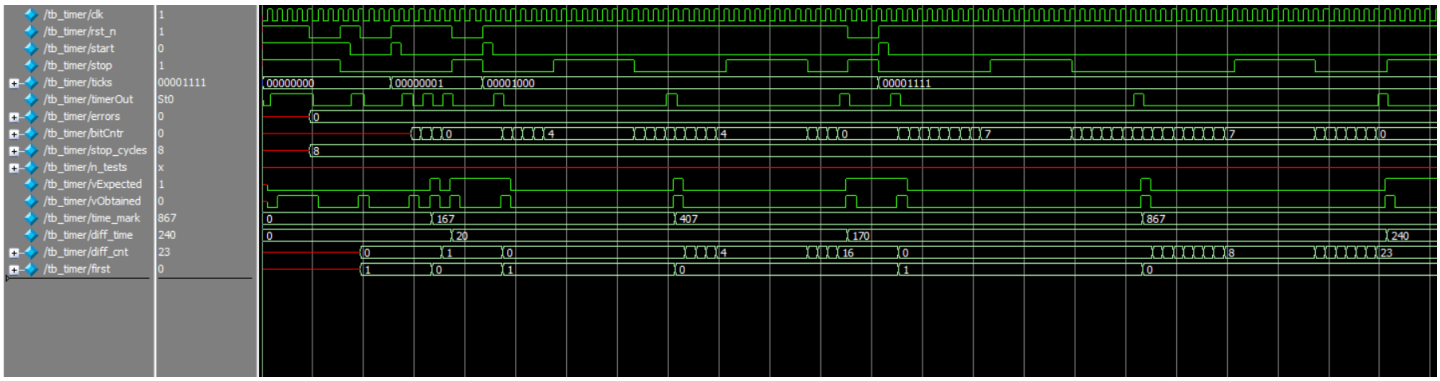

Figura 4: Diagrama d'ones de la verificació post síntesi. Es pot veure que no hi ha molts canvis excepte algun petit delay.