INSTITUTO SUPERIOR TÉCNICO

Design Test and Reliability of Electronic Systems

Project 1 - Digital Controller

Group 2

n° 77973 - Pedro Pina

n° 84779 - Afonso Muralha

Professor Fernando Gonçalves

April 9, 2020

Contents

I. Introduction and Objectives

The goal of this project is to develop a controller that fulfills certain specifications.

The controller must be described in synthesizable Verilog and it must synthesize without any errors or warnings. (.....To check for these errors and warnings but also validate its behaviour the options on the Xilinx Vivado HL WebPACK Edition are used.....)

To validate the quality of the testbench, the code coverage must be evaluated using the Cadence simulator (Xcelium).

II. Specifications

The controller must synthesize without any errors or warnings.

As for the testbench, the goal should be a code coverage of 100% for the Block and FSM categories.

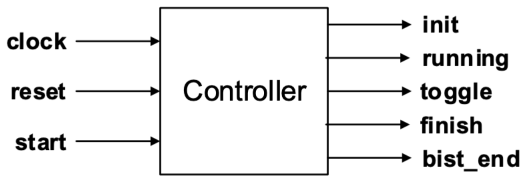


Figure 1: Controller interface

Figure 1 illustrates the controller interface, with the input and output signals. Every signal has certain sensitivity to the edges and logic levels which must be preserved when building the state machine.

Controller inputs

clock: The controller must be sensitive to the rising edges of the clock.

start: After a 0→1 transition in this signal, a new sequence must be initiated (see Figure 2). Its value is captured on the rising edge of the clock. While the full sequence is not completed (bist_end signal set to HIGH), further transitions in the start signal must be ignored.

reset: The reset must be synchronous and active at the logic level ‘1’. This signal is used to restart the state machine and the counters, preparing the controller to start a new sequence. When the reset signal goes LOW and the start signal is HIGH, a new sequence must not be restarted. The controller must wait for a 0→1 transition in the start signal to start a new sequence.

Controller outputs

init: This signal is a pulse with a duration of one clock cycle, indicating the start of a new sequence.

running: This signal must be HIGH for N clock cycles. If the reset signal goes HIGH, then this signal must go LOW.

toggle: This signal must toggle, while the running signal is HIGH. *finish*: This signal is a pulse with a duration of one clock cycle, indicating the end of a sequence.

bist_end: This signal must go HIGH when the sequence is completed. The bist_end signal must go LOW when the reset signal is activated or the start signal has a 0→1 transition.

III. Digital Controller Design

Finite State Machine

The designed finite state machine features 5 states in order to cop with the specifications mentioned above.

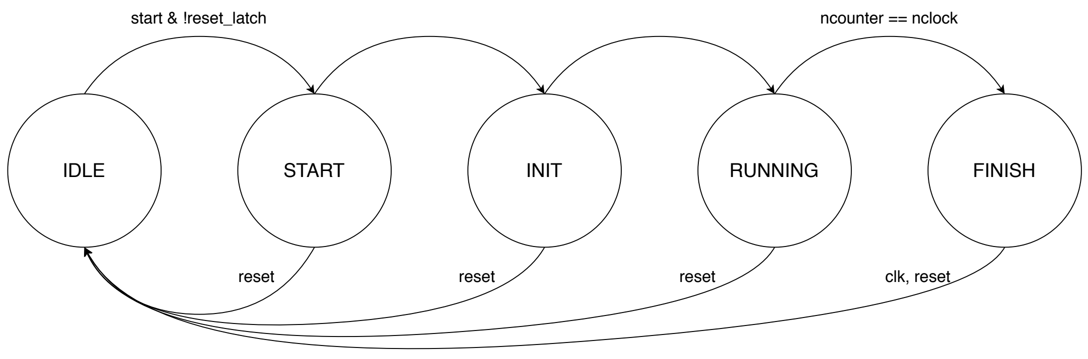


Figure 2: Finite state machine diagram.

Table 1: Output conditions. More detailed information on state description.

Outputs	Condition
init	state == INIT
running	(state == RUNNING) & ncounter <nclock + 1
toggle	(state == RUNNING) & toggle_r
finish	state == FINISH
bist_end	(complete) & !(reset | start);

IDLE state:

This state is activated while waiting for a new sequence to be started. When the start conditions are met, the controller changes into the START state.

These start conditions are defined by the following logic expression:

```
start & (state == IDLE) & !reset_latch
```

The main drive for the state change is the start pin becoming enabled but in order to meet requirements, a simple reset latch was implemented, so that if the start pin is already enabled on the falling edge of the reset signal, the controller waits for a full transition of the start signal before changing state. This latch was implemented as follows:

```
always @ (posedge start) begin
       if(start & !(reset))
              reset_latch <= 0;
       else
              reset_latch <= 1;
end
```

Code excerpt 1: Reset latch implementation.

START state:

On the start stage, the sequence is starting. The controller changes to the INIT state on the following positive clock transition.

INIT state:

Whilst on the INIT state, the controller enables the init output and is set to change into RUNNING state on the next positive clock transition.

RUNNING state:

On the RUNNING state, the controller generates the toggle and running outputs for the required number of clocks impulses. The number of clock impulses is set by a parameter and, in order to have a suitable sized register to hold it, the following primitive was used:

```
parameter NCLOCK = 650; //number of clocks for group 2
reg [$clog2(NCLOCK):0] ncounter; //uses the log2 primitive to calculate a suitable
    size for the register, this is done during compilation.
```

Code excerpt 2: Creation of the register for the NCLOCK parameter.

In this state, the running output is enabled, due to the condition:

```
    assign running = (state == RUNNING) & (ncounter < nclock+1);
```

Code excerpt 3: running output condition.

The toggle output is driven by the state and an auxiliary register that carries the output of a toggle process. This process is also responsible for counting the number of elapsed clocks and iterating them.

```
    assign toggle = (state == RUNNING) & toggle_r;
```

Code excerpt 4: toggle output condition.

```
always @ (posedge clock) begin
        if(reset | (state == FINISH)) begin
                toggle_r <= 0;
                ncounter <= 0;
        end
        else if(state == RUNNING) begin
                if(ncounter < NCLOCK) begin
                        toggle_r = !toggle_r;
                end
                else begin
                        toggle_r <= 0;
                end
                ncounter <= ncounter + 1;
        end
end
```

Code excerpt 5: Toggle generator and counter process.

When the iterating register reaches the determined parameter, the state changes to FINISH.

FINISH state:

Whilst on the FINISH state, the controller enables the finish output and is set to change into IDLE state on the next positive clock transition.

The bist_end signal is triggered by the following condition:

```
assign bist_end = (complete) & !(reset | start);
```

Code excerpt 6: bist_end register drive.

The complete register is used on order to assure a simultaneous transition of the bist_end signal.

```
always @ (negedge finish, posedge start, posedge reset) begin
       if(reset | start)
              complete = 0;
       else
              complete = 1;

end
```

Code excerpt 7: Complete register drive.

III.I Testbench and simulation

In order to test the controller, a test bench was created. This test bench instances the controller modules and features multiple test scenarios that can be set by commenting the non-relevant tests:

```
//===========SET TEST==========
`define normal
`define consequent_test
`define mid_start
`define start_reset
`define mid_reset
```

Code excerpt 8: Testbench test set.

For the following tests, an NCLOCK of 10 was used for easier visualization.

Normal test

Runs one sequence.

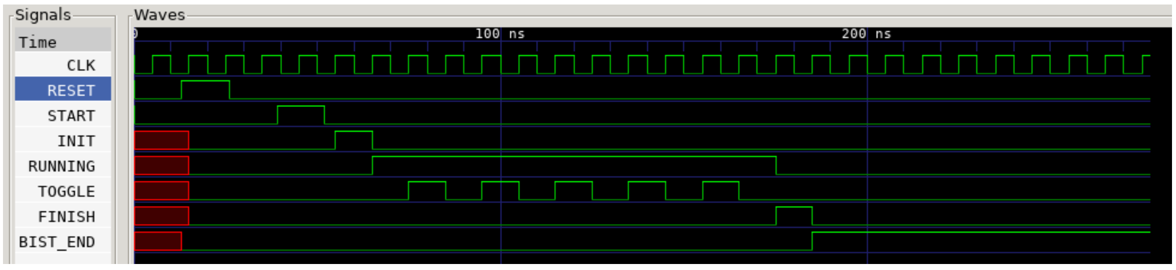


Figure 3: Normal test waveforms.

Consequent test

Runs two sequences in a row:

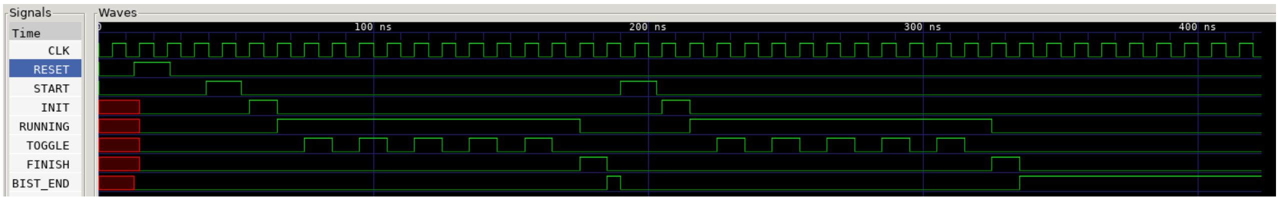


Figure 4: Consequent test waveforms.

Mid-sequence start test

Runs one sequence and toggles the start during the running sequence. The mid-sequence start should not affect the sequence.

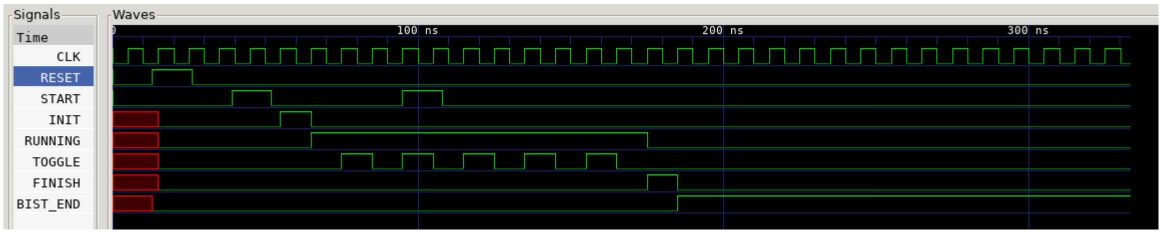


Figure 5: Mid-sequence start test waveforms.

Mid-sequence reset test

Runs one sequence and toggles the reset during the running sequence and starts a new one after that. The mid-sequence reset should abort the sequence and the following sequence should complete without any issues.

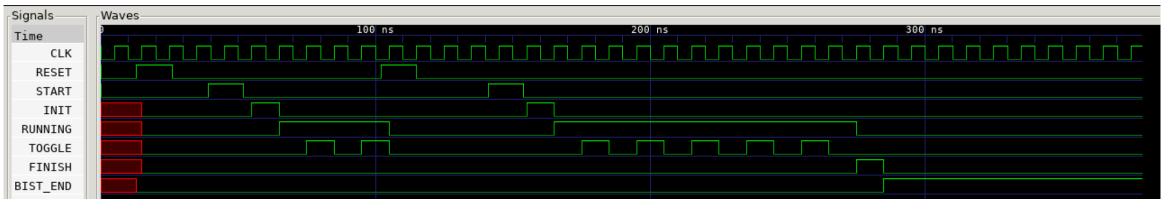


Figure 6: Mid-sequence reset test waveforms.

Start/reset latch test

Runs one sequence and after its completion, enables the start and reset signals and disables the reset signal. In this case the sequence should not start. After that a toggle of the start signal should start a new sequence and it should complete without issues.

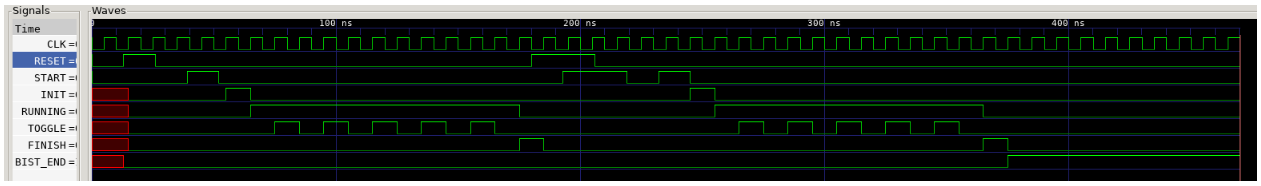


Figure 7: Start/reset latch test waveforms.

In order to validate the running and toggle signals, two counter variables were created on the test bench that are used for measuring then length of the running signal and the number of pulses of the toggle signal.

```
integer pulsecount = 0;
integer runningcount = 0;

always @ (posedge TOGGLE) begin
    pulsecount = pulsecount +1;
end

always @ (posedge CLK) begin
    if(RUNNING)
        runningcount = runningcount + 1;
end
```

Code excerpt 9: Counter variables.

These variables are printed and reseted after a sequence:

```
$display("Number of pulses: %d pulses.",pulsecount);
$display("Time of running: %d clock pulses.",runningcount-1); //-1 because the first
     rising edge of the clock doesent count
pulsecount = 0;
runningcount = 0;
```

Code excerpt 10: Conter variable prints.

Simulating one normal sequence with the NCLOCK set to the group's number (650) we can validate the result with the output of the program:

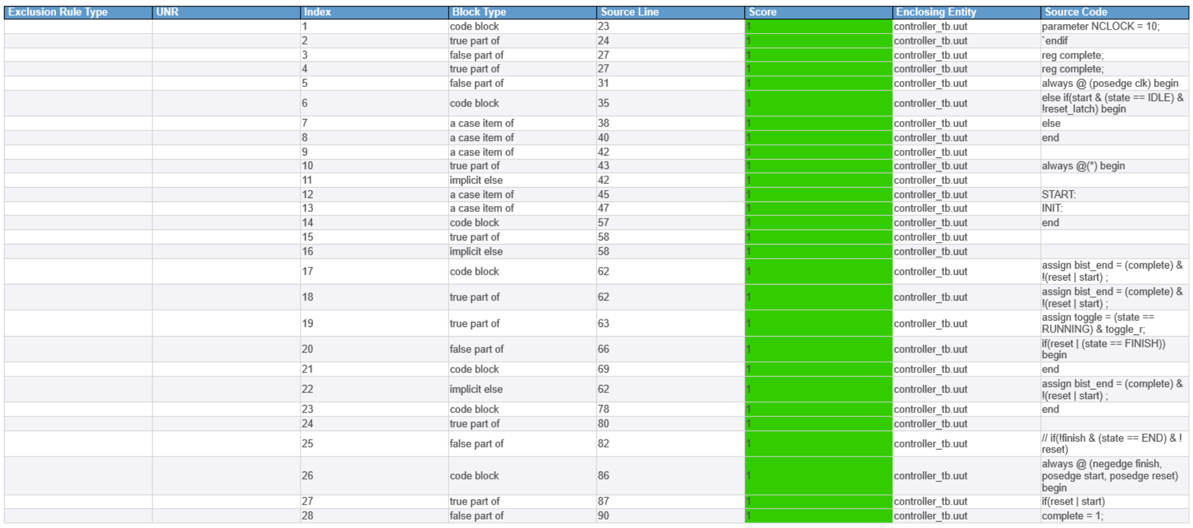
```
Number of pulses:        325 pulses.
Time of running:         650 clock pulses.
```

Code excerpt 11: Toggle and running validation.

Code Coverage and synthesis validation

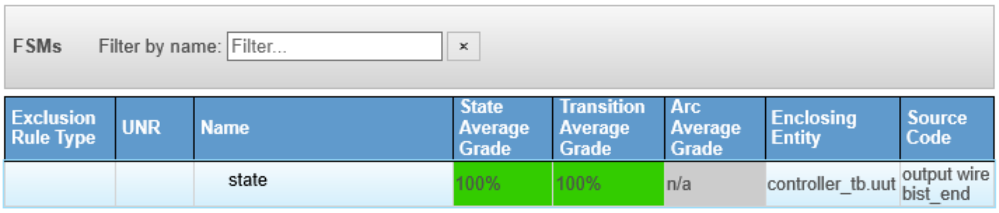
In order to validate the quality of the produced code, the controller module was synthesized on the Vivado IDE and the report can be found in annex X.

A code coverage and FSM reports were also generated using cadence tools for the controller using NCLOCK = 650 and NCLOCK = 10.



Exclusion Rule Type	UNR	Index	Block Type	Source Line	Score	Enclosing Entity	Source Code
		1	code block	23	1	controller_tb.uut	parameter NCLOCK = 10;
		2	true part of	24	1	controller_tb.uut	`endif
		3	false part of	27	1	controller_tb.uut	reg complete;
		4	true part of	27	1	controller_tb.uut	reg complete;
		5	false part of	31	1	controller_tb.uut	always @ (posedge clk) begin
		6	code block	35	1	controller_tb.uut	else if(start & (state == IDLE) & !reset_latch) begin
		7	a case item of	38	1	controller_tb.uut	else
		8	a case item of	40	1	controller_tb.uut	end
		9	a case item of	42	1	controller_tb.uut	
		10	true part of	43	1	controller_tb.uut	always @(*) begin
		11	implicit else	42	1	controller_tb.uut	
		12	a case item of	45	1	controller_tb.uut	START:
		13	a case item of	47	1	controller_tb.uut	INIT:
		14	code block	57	1	controller_tb.uut	end
		15	true part of	58	1	controller_tb.uut	
		16	implicit else	58	1	controller_tb.uut	
		17	code block	62	1	controller_tb.uut	assign bist_end = (complete) & !(reset | start) ;
		18	true part of	62	1	controller_tb.uut	assign bist_end = (complete) & !(reset | start) ;
		19	true part of	63	1	controller_tb.uut	assign toggle = (state == RUNNING) & toggle_r;
		20	false part of	66	1	controller_tb.uut	if(reset | (state == FINISH)) begin
		21	code block	69	1	controller_tb.uut	end
		22	implicit else	62	1	controller_tb.uut	assign bist_end = (complete) & !(reset | start) ;
		23	code block	78	1	controller_tb.uut	end
		24	true part of	80	1	controller_tb.uut	
		25	false part of	82	1	controller_tb.uut	// if(!finish & (state == END) & !reset)
		26	code block	86	1	controller_tb.uut	always @ (negedge finish, posedge start, posedge reset) begin
		27	true part of	87	1	controller_tb.uut	if(reset | start)
		28	false part of	90	1	controller_tb.uut	complete = 1;

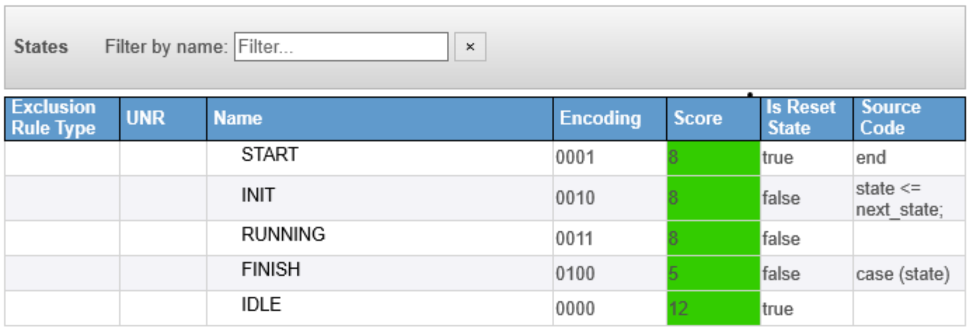
Figure 8: Code coverage report.



FSMs Filter by name: Filter...

Exclusion Rule Type	UNR	Name	State Average Grade	Transition Average Grade	Arc Average Grade	Enclosing Entity	Source Code
		state	100%	100%	n/a	controller_tb.uut	output wire bist_end

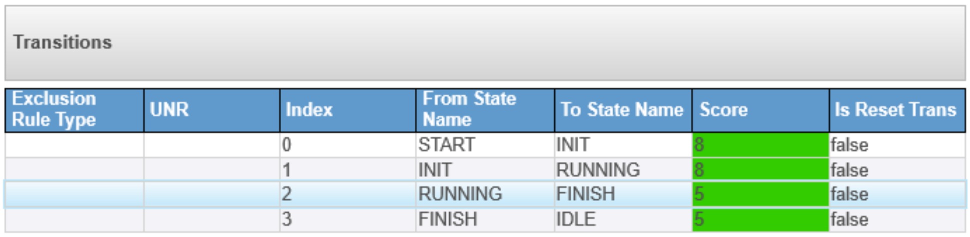
Figure 9: FSM report 1.



States Filter by name: Filter... ×

Exclusion Rule Type	UNR	Name	Encoding	Score	Is Reset State	Source Code
		START	0001	8	true	end
		INIT	0010	8	false	state <= next_state;
		RUNNING	0011	8	false	
		FINISH	0100	5	false	case (state)
		IDLE	0000	12	true	

Figure 10: FSM report 2.



Transitions

Exclusion Rule Type	UNR	Index	From State Name	To State Name	Score	Is Reset Trans
		0	START	INIT	8	false
		1	INIT	RUNNING	8	false
		2	RUNNING	FINISH	5	false
		3	FINISH	IDLE	5	false

Figure 11: FSM report 3.

IV. Conclusion

This project objective was to learn how to program and develop a controller with certain specifications. This meant that, for it to work properly, the ones developing it, needed to have a clear understanding of every input and ouput signals functionality and behaviour - how does a signal influences other signals and how does it develop throw a specific time period.

Therefore, having tools such as the Gtkwave and the Cadence simulator (Xcelium) are important to evaluate the code, which errors and warnings it may have, during the developing stage. In addition to that, having a graphic representation of what has been done can guarantee that the final product is working as intended, with every specification running correctly.

References

[1] Verilog Tutorial (Course slides)
https://fenix.tecnico.ulisboa.pt/downloadFile/1126518382240212/My%20Verilog%20Tutorial.pdf

Vivado synthesizing output

```
*** Running vivado
    with args -log controller.vds -m64 -product Vivado -mode batch -messageDb
        vivado.pb -notrace -source controller.tcl



****** Vivado v2019.2 (64-bit)
  **** SW Build 2708876 on Wed Nov 6 21:40:23 MST 2019
  **** IP Build 2700528 on Thu Nov 7 00:09:20 MST 2019
    ** Copyright 1986-2019 Xilinx, Inc. All Rights Reserved.

source controller.tcl -notrace
Command: synth_design -top controller -part xc7k70tfbv676-1
Starting synth_design
Attempting to get a license for feature 'Synthesis' and/or device 'xc7k70t'
INFO: [Common 17-349] Got license for feature 'Synthesis' and/or device '
    xc7k70t'
INFO: Launching helper process for spawning children vivado processes
```

```
INFO: Helper process launched with PID 25664
---------------------------------------------------------------------------------

Starting Synthesize : Time (s): cpu = 00:00:04 ; elapsed = 00:00:06 . Memory (
   MB): peak = 506.453 ; gain = 213.816
---------------------------------------------------------------------------------

INFO: [Synth 8-6157] synthesizing module 'controller' [F:/Documents/GitHub/
   PTFSE-Classes/Project1/controller.v:6]
      Parameter IDLE bound to: 0 - type: integer
      Parameter START bound to: 1 - type: integer
      Parameter INIT bound to: 32'sb00000000000000000000000000000010
      Parameter RUNNING bound to: 3 - type: integer
      Parameter FINISH bound to: 4 - type: integer
      Parameter NCLOCK bound to: 650 - type: integer
INFO: [Synth 8-6155] done synthesizing module 'controller' (1#1) [F:/Documents/
   GitHub/PTFSE-Classes/Project1/controller.v:6]
---------------------------------------------------------------------------------

Finished Synthesize : Time (s): cpu = 00:00:06 ; elapsed = 00:00:07 . Memory (
   MB): peak = 579.242 ; gain = 286.605
---------------------------------------------------------------------------------

---------------------------------------------------------------------------------

Finished Constraint Validation : Time (s): cpu = 00:00:06 ; elapsed = 00:00:08
   . Memory (MB): peak = 579.242 ; gain = 286.605
---------------------------------------------------------------------------------

---------------------------------------------------------------------------------

Start Loading Part and Timing Information
---------------------------------------------------------------------------------

Loading part: xc7k70tfbv676-1
---------------------------------------------------------------------------------
```

```
Finished Loading Part and Timing Information : Time (s): cpu = 00:00:06 ;
   elapsed = 00:00:08 . Memory (MB): peak = 579.242 ; gain = 286.605
---------------------------------------------------------------------------------

INFO: [Device 21-403] Loading part xc7k70tfbv676-1
INFO: [Synth 8-5544] ROM "next_state" won't be mapped to Block RAM because
   address size (4) smaller than threshold (5)
WARNING: [Synth 8-327] inferring latch for variable 'next_state_reg' [F:/
   Documents/GitHub/PTFSE-Classes/Project1/controller.v:46]
---------------------------------------------------------------------------------

Finished RTL Optimization Phase 2 : Time (s): cpu = 00:00:06 ; elapsed =
   00:00:08 . Memory (MB): peak = 579.242 ; gain = 286.605
---------------------------------------------------------------------------------


Report RTL Partitions:
+-+--------------+------------+----------+
| |RTL Partition |Replication |Instances |
+-+--------------+------------+----------+
+-+--------------+------------+----------+
No constraint files found.
---------------------------------------------------------------------------------

Start RTL Component Statistics
---------------------------------------------------------------------------------

Detailed RTL Component Info :
+---Adders :
         2 Input   11 Bit      Adders := 1
+---Registers :
                   11 Bit   Registers := 1
                    4 Bit   Registers := 1
                    1 Bit   Registers := 3
+---Muxes :
```

```
         3 Input    4 Bit      Muxes := 1
         2 Input    3 Bit      Muxes := 1
         5 Input    3 Bit      Muxes := 1
         2 Input    1 Bit      Muxes := 4
         6 Input    1 Bit      Muxes := 1
---------------------------------------------------------------------------------

Finished RTL Component Statistics
---------------------------------------------------------------------------------

---------------------------------------------------------------------------------

Start RTL Hierarchical Component Statistics
---------------------------------------------------------------------------------

Hierarchical RTL Component report
Module controller
Detailed RTL Component Info :
+---Adders :
         2 Input   11 Bit     Adders := 1
+---Registers :
                  11 Bit   Registers := 1
                   4 Bit   Registers := 1
                   1 Bit   Registers := 3
+---Muxes :
         3 Input    4 Bit      Muxes := 1
         2 Input    3 Bit      Muxes := 1
         5 Input    3 Bit      Muxes := 1
         2 Input    1 Bit      Muxes := 4
         6 Input    1 Bit      Muxes := 1
---------------------------------------------------------------------------------

Finished RTL Hierarchical Component Statistics
---------------------------------------------------------------------------------

---------------------------------------------------------------------------------
```

```
Start Part Resource Summary
---------------------------------------------------------------------------------

Part Resources:
DSPs: 240 (col length:80)
BRAMs: 270 (col length: RAMB18 80 RAMB36 40)
---------------------------------------------------------------------------------

Finished Part Resource Summary
---------------------------------------------------------------------------------

No constraint files found.
---------------------------------------------------------------------------------

Start Cross Boundary and Area Optimization
---------------------------------------------------------------------------------

Warning: Parallel synthesis criteria is not met
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\
   next_state_reg[3] )
WARNING: [Synth 8-3332] Sequential element (next_state_reg[3]) is unused and
   will be removed from module controller.
---------------------------------------------------------------------------------

Finished Cross Boundary and Area Optimization : Time (s): cpu = 00:00:11 ;
   elapsed = 00:00:15 . Memory (MB): peak = 726.262 ; gain = 433.625
---------------------------------------------------------------------------------


Report RTL Partitions:
+-+--------------+------------+----------+
| |RTL Partition |Replication |Instances |
+-+--------------+------------+----------+
+-+--------------+------------+----------+
No constraint files found.
```

```
---------------------------------------------------------------------------------

Start Timing Optimization
---------------------------------------------------------------------------------

---------------------------------------------------------------------------------

Finished Timing Optimization : Time (s): cpu = 00:00:12 ; elapsed = 00:00:15 .
   Memory (MB): peak = 728.090 ; gain = 435.453
---------------------------------------------------------------------------------


Report RTL Partitions:
+-+--------------+------------+----------+
| |RTL Partition |Replication |Instances |
+-+--------------+------------+----------+
+-+--------------+------------+----------+
---------------------------------------------------------------------------------

Start Technology Mapping
---------------------------------------------------------------------------------

---------------------------------------------------------------------------------

Finished Technology Mapping : Time (s): cpu = 00:00:12 ; elapsed = 00:00:15 .
   Memory (MB): peak = 728.266 ; gain = 435.629
---------------------------------------------------------------------------------


Report RTL Partitions:
+-+--------------+------------+----------+
| |RTL Partition |Replication |Instances |
+-+--------------+------------+----------+
+-+--------------+------------+----------+
---------------------------------------------------------------------------------

```

```
Start IO Insertion
---------------------------------------------------------------------------------


---------------------------------------------------------------------------------

Start Flattening Before IO Insertion
---------------------------------------------------------------------------------


---------------------------------------------------------------------------------

Finished Flattening Before IO Insertion
---------------------------------------------------------------------------------


---------------------------------------------------------------------------------

Start Final Netlist Cleanup
---------------------------------------------------------------------------------


---------------------------------------------------------------------------------

Finished Final Netlist Cleanup
---------------------------------------------------------------------------------


---------------------------------------------------------------------------------

Finished IO Insertion : Time (s): cpu = 00:00:17 ; elapsed = 00:00:20 . Memory
    (MB): peak = 729.082 ; gain = 436.445
---------------------------------------------------------------------------------


---------------------------------------------------------------------------------

Start Renaming Generated Instances
---------------------------------------------------------------------------------


---------------------------------------------------------------------------------
```

```
Finished Renaming Generated Instances : Time (s): cpu = 00:00:17 ; elapsed =
    00:00:21 . Memory (MB): peak = 729.082 ; gain = 436.445
---------------------------------------------------------------------------------



Report RTL Partitions:
+-+--------------+------------+----------+
| |RTL Partition |Replication |Instances |
+-+--------------+------------+----------+
+-+--------------+------------+----------+

Report Check Netlist:
+------+------------------+-------+---------+-------+------------------+
|      |Item              |Errors |Warnings |Status |Description       |
+------+------------------+-------+---------+-------+------------------+
|1     |multi_driven_nets |      0|        0|Passed |Multi driven nets |
+------+------------------+-------+---------+-------+------------------+
---------------------------------------------------------------------------------

Start Rebuilding User Hierarchy
---------------------------------------------------------------------------------

---------------------------------------------------------------------------------

Finished Rebuilding User Hierarchy : Time (s): cpu = 00:00:17 ; elapsed =
    00:00:21 . Memory (MB): peak = 729.082 ; gain = 436.445
---------------------------------------------------------------------------------

---------------------------------------------------------------------------------

Start Renaming Generated Ports
---------------------------------------------------------------------------------

---------------------------------------------------------------------------------

Finished Renaming Generated Ports : Time (s): cpu = 00:00:17 ; elapsed =
   00:00:21 . Memory (MB): peak = 729.082 ; gain = 436.445
---------------------------------------------------------------------------------

---------------------------------------------------------------------------------
Start Handling Custom Attributes
---------------------------------------------------------------------------------

---------------------------------------------------------------------------------
Finished Handling Custom Attributes : Time (s): cpu = 00:00:17 ; elapsed =
   00:00:21 . Memory (MB): peak = 729.082 ; gain = 436.445
---------------------------------------------------------------------------------

---------------------------------------------------------------------------------
Start Renaming Generated Nets
---------------------------------------------------------------------------------

---------------------------------------------------------------------------------
Finished Renaming Generated Nets : Time (s): cpu = 00:00:17 ; elapsed =
   00:00:21 . Memory (MB): peak = 729.082 ; gain = 436.445
---------------------------------------------------------------------------------

---------------------------------------------------------------------------------
Start Writing Synthesis Report
---------------------------------------------------------------------------------


Report BlackBoxes:
+-+--------------+----------+
| |BlackBox name |Instances |
+-+--------------+----------+
+-+--------------+----------+
Report Cell Usage:
+------+-----+------+
|      |Cell |Count |
+------+-----+------+
|1     |BUFG |     2|
|2     |LUT1 |     1|
|3     |LUT2 |     5|
|4     |LUT3 |    10|
|5     |LUT4 |     5|
|6     |LUT5 |     8|
|7     |LUT6 |     9|
|8     |FDCE |     1|
|9     |FDRE |    16|
|10    |LD   |     3|
|11    |IBUF |     3|
|12    |OBUF |     5|
+------+-----+------+

Report Instance Areas:
+------+---------+-------+------+
|      |Instance |Module |Cells |
+------+---------+-------+------+
|1     |top      |       |    68|
+------+---------+-------+------+
---------------------------------------------------------------------------------

Finished Writing Synthesis Report : Time (s): cpu = 00:00:17 ; elapsed =
   00:00:21 . Memory (MB): peak = 729.082 ; gain = 436.445
---------------------------------------------------------------------------------

Synthesis finished with 0 errors, 0 critical warnings and 2 warnings.
Synthesis Optimization Runtime : Time (s): cpu = 00:00:17 ; elapsed = 00:00:21
   . Memory (MB): peak = 729.082 ; gain = 436.445
Synthesis Optimization Complete : Time (s): cpu = 00:00:17 ; elapsed = 00:00:21
    . Memory (MB): peak = 729.082 ; gain = 436.445
```

```
INFO: [Project 1-571] Translating synthesized netlist
Netlist sorting complete. Time (s): cpu = 00:00:00 ; elapsed = 00:00:00 .
   Memory (MB): peak = 742.207 ; gain = 0.000
INFO: [Netlist 29-17] Analyzing 3 Unisim elements for replacement
INFO: [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
INFO: [Project 1-570] Preparing netlist for logic optimization
INFO: [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
Netlist sorting complete. Time (s): cpu = 00:00:00 ; elapsed = 00:00:00 .
   Memory (MB): peak = 842.684 ; gain = 0.000
INFO: [Project 1-111] Unisim Transformation Summary:
  A total of 3 instances were transformed.
  LD => LDCE: 3 instances

INFO: [Common 17-83] Releasing license: Synthesis
13 Infos, 2 Warnings, 0 Critical Warnings and 0 Errors encountered.
synth_design completed successfully
synth_design: Time (s): cpu = 00:00:21 ; elapsed = 00:00:28 . Memory (MB): peak
    = 842.684 ; gain = 550.047
Netlist sorting complete. Time (s): cpu = 00:00:00 ; elapsed = 00:00:00 .
   Memory (MB): peak = 842.684 ; gain = 0.000
WARNING: [Constraints 18-5210] No constraints selected for write.
Resolution: This message can indicate that there are no constraints for the
   design, or it can indicate that the used_in flags are set such that the
   constraints are ignored. This later case is used when running synth_design
   to not write synthesis constraints to the resulting checkpoint. Instead,
   project constraints are read when the synthesized design is opened.
INFO: [Common 17-1381] The checkpoint 'F:/Documents/GitHub/PTFSE-Classes/
   Project1/Vivado/Vivado.runs/synth_1/controller.dcp' has been generated.
INFO: [runtcl-4] Executing : report_utilization -file
   controller_utilization_synth.rpt -pb controller_utilization_synth.pb
INFO: [Common 17-206] Exiting Vivado at Thu Apr 9 18:13:39 2020...
```

Code excerpt 12: Complete register drive.