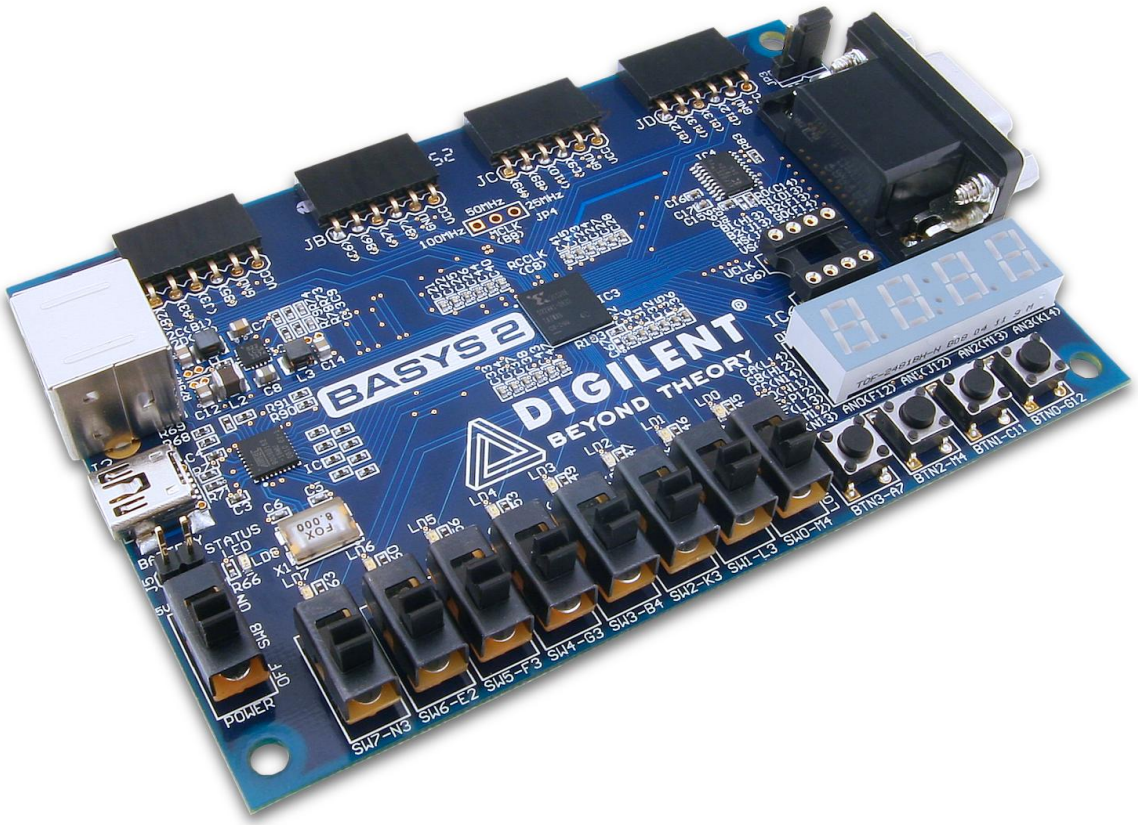
Washing Machine



Author

Diogo Vilar Sardinha
Electrical and Computer Engineering MSc student

Contents

1 Introduction

In this lab work I study the design of digital systems using one of the most common hardware description languages (HDL), Verilog (the other is VHL).

Using the Verilog language it is intended to implement in FPGA a state machine to control a washing machine. The board I use is the Basys 2 Spartan-3E FPGA Trainer Board.

I started by drawing the states and transitions diagram of Figure 1, detailing the intended operation for the state machine, and it was through this diagram that I wrote the code that was implemented on the board.

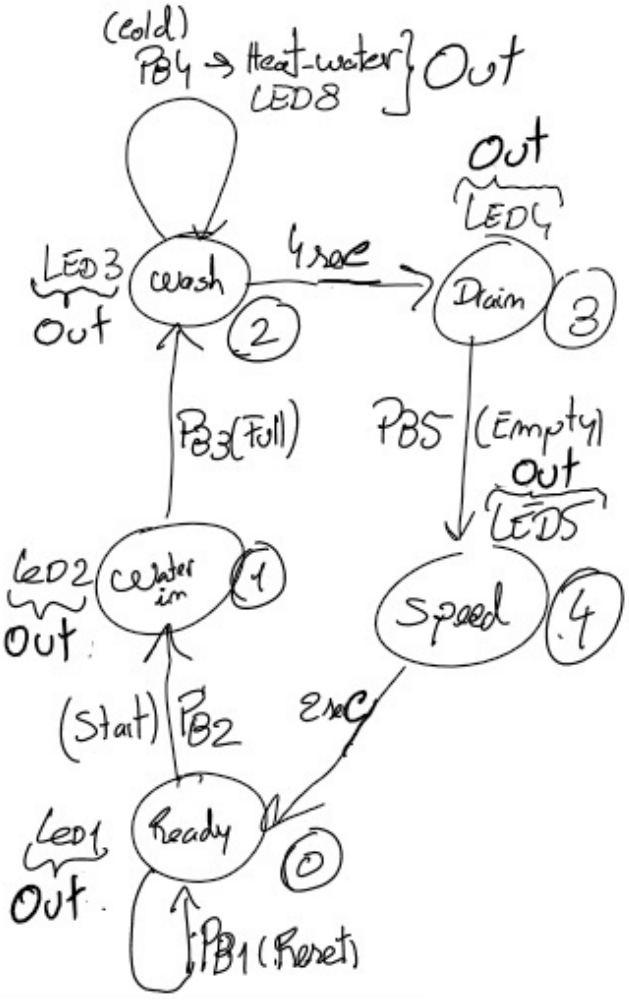


Figure 1: Finite state machine to be implemented

Once the desired implementation is defined, I developed the verilog code, as presented in the following section

2 Simulations

Once the code is created, I proceeded to the simulations. In Xilinx Software I created a *test bench* in verilog to test my developed model. The bench test is a very useful tool that allows you to test the digital circuits developed without the need for hardware.
In the test bench file I introduced a clock signal and several sequences of input signals.

Listing 1: wash_fsm_tb.v

```
////////////////////////////////////////////////////////////////////////////////
// Author: Diogo Vilar Sardinha
// Module Name:   wash_fsm_tb.v
// Project Name:  TB
// Target Device: XC2S300E
// Tool versions: ISE 10.1 + ModelSIM or ISIM
// Description:   Washing Machine Controller TestBench
////////////////////////////////////////////////////////////////////////////////
/*
 *    Notes:
 *       1) Verilog code
 *       2) Simulation images of the circuit
 *       3) Place and route report for 50MHz
 *       4) Place and route report for the maximum frequency to which the
  ↪  circuit can operate
 *
 *       Clocked at 50 MHz (T = 20 ns) (50M ticks per second)
 *       (ucf diz: T=10ns 50%High)
 *       To reduce simulation time, use smaller time delays in the
 *       state transitions Wash->Drain and Speed->Ready.
 *       Example:
 *                Wash->Drain 160 ns
 *                Speed->Ready 80 ns
 */

`timescale 1 ns / 1 ns
/* timescale 1ns/1ns means the time unit and time precision of a module that
  ↪  follow it.
 * The simulation time and delay values are measured using time unit.
 * The precision factor is needed to measure the degree of accuracy of the time
  ↪  unit,
 * in other words how delay values are rounded before being used in simulation.
 * Whatever times you mensioned in verilog code will be taken in ns.eg.
 * Resolution of 1ns means you can have 1ns as smallest representation of the
  ↪  time.
 */

module testbench;

 // Inputs
 reg reset;
 reg clk;
 reg start;
 reg full;
 reg empty;
 reg cold;

 // Outputs
 wire ready;
 wire water_in;
 wire heat_r;
 wire wash;
 wire drain;
 wire speed;

 // Instantiate the Unit Under Test (UUT) or (DUT) Device Under Test
 wash_fsm uut (
  .reset(reset),
  .clk(clk),
  .start(start),
  .full(full),
```

```
  .empty(empty),
  .cold(cold),
  .ready(ready),
  .water_in(water_in),
  .heat_r(heat_r),
  .wash(wash),
  .drain(drain),
  .speed(speed)
);

   // Clock generator
   always
   #10.0 clk = ~clk; // Toggle clock every 10 ticks (T=20ns)

   initial begin
      // Initialize Inputs
  reset = 1;
  clk = 1;
  start = 0;
  full = 0;
  empty = 0;
  cold = 0;

      // Display initial message
      $display("Washing Machine FSM");

  // Add stimulus here
      @(negedge clk)
         reset = 0;
  $display("Time = %t ns, reset assigned '0'", $time*10);
      @(negedge clk)
         start = 1;
      $display("Time = %t ns, start assigned '1'", $time*10);
      @(negedge clk)
         full = 1;
      $display("Time = %t ns, full assigned '1'", $time*10);
      @(negedge clk)
         cold = 1;
      $display("Time = %t ns, cold assigned '1'", $time*10);
      @(negedge clk)
         cold = 0;
      $display("Time = %t ns, cold assigned '0'", $time*10);
  while(drain != 1) begin #1; end
      @(negedge clk)
         empty = 1;
      while(ready != 1) begin #1; end
      $stop;


      @(negedge clk)
         start = 1;
      $display("Time = %t ns, start assigned '1'", $time*10);
      @(negedge clk)
         full = 1;
      $display("Time = %t ns, full assigned '1'", $time*10);
      @(negedge clk)
         cold = 1;
      $display("Time = %t ns, cold assigned '1'", $time*10);
      @(negedge clk)
         cold = 0;
      $display("Time = %t ns, cold assigned '0'", $time*10);
```

```
119   while(drain != 1) begin #1; end
120       @(negedge clk)
121          empty = 1;
122       while(ready != 1) begin #1; end
123       $stop;
124   end
125
126       always @(posedge ready)
127          $display("Time = %t ns, entering Ready state", $time*10);
128       always @(posedge water_in)
129          $display("Time = %t ns, entering Water In state", $time*10);
130       always @(posedge wash)
131          $display("Time = %t ns, entering Wash state", $time*10);
132       always @(posedge drain)
133          $display("Time = %t ns, entering Drain state", $time*10);
134       always @(posedge speed)
135          $display("Time = %t ns, entering Speed state", $time*10);
136       always @(posedge heat_r)
137          $display("Time = %t ns, heat_r asserted", $time*10);
138       always @(negedge heat_r)
139          $display("Time = %t ns, heat_r de-asserted", $time*10);
140
141 endmodule
```

In the Figure 2 we have an overview of the simulation going through all 5 states in the state machine and their associated sinais.
Representation and colors of the signals in the simulation:

- Clock - Yellow
- Reset - Red
- Inputs(Buttons) - Blue
- Outputs (5 States + heat r) - Green
- Current state (binary) - Green

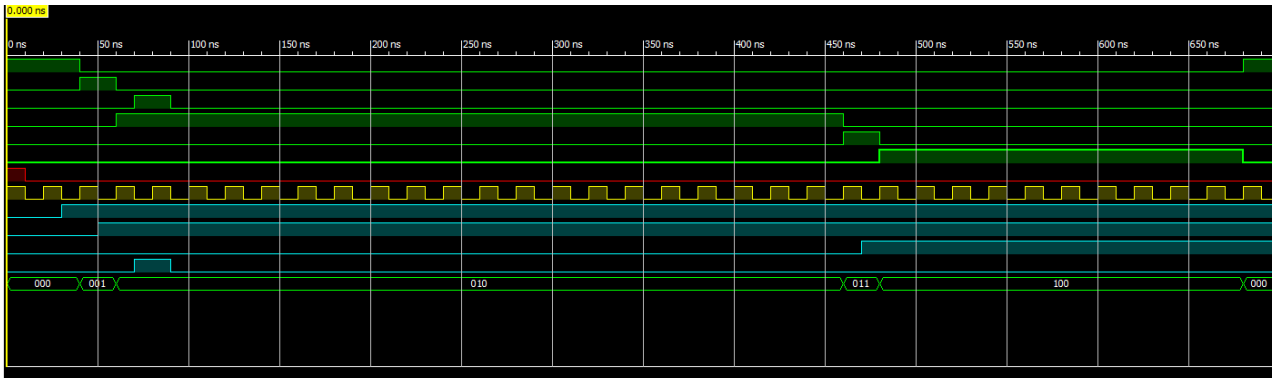


Figure 2: Overview of simulation with all existing state and signals

The transition between the first three states is in Figure 3.

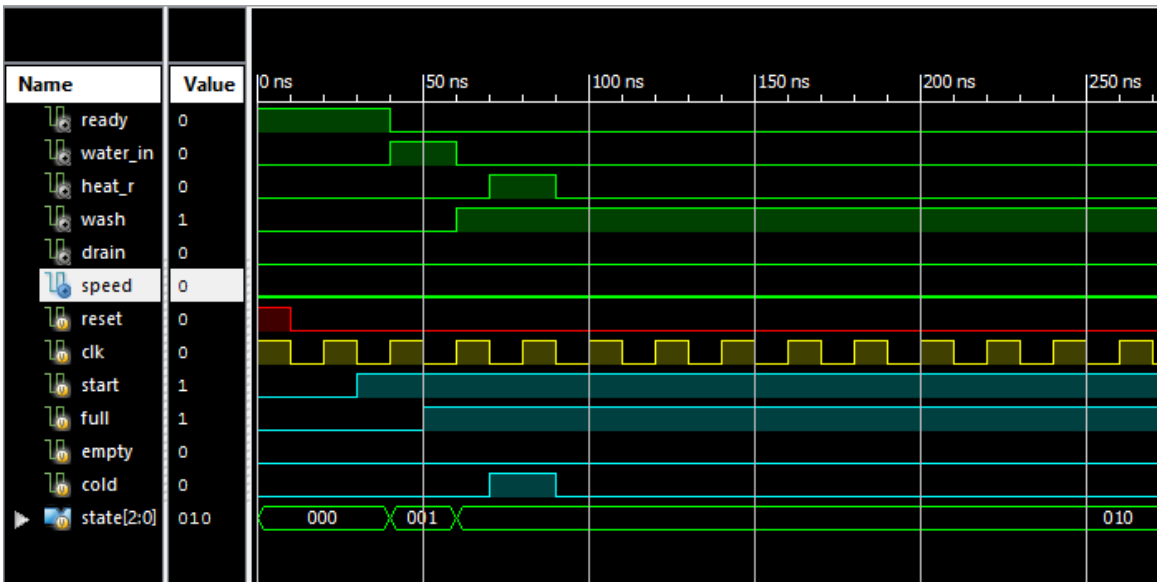


Figure 3: View of the simulation with the first 3 states

The dwell time in the state *wash* was changed from 4s to 400ns to be faster the simulation analysis as it happens in the Figure 4.

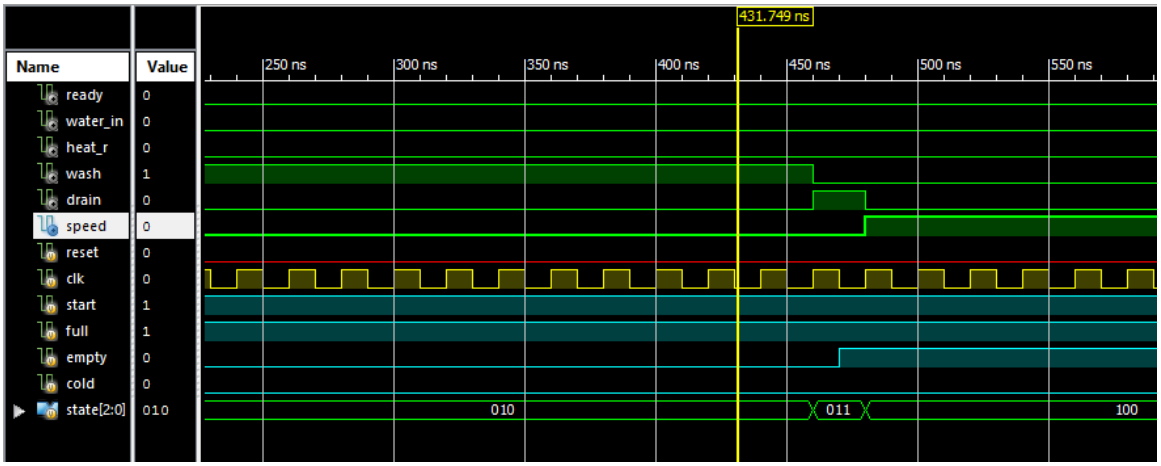


Figure 4: View of the simulation with states 2, 3 and 4

The dwell time in the *speed* state has been changed from 2s to 200ns to be faster to analyze the simulation Figure 5.

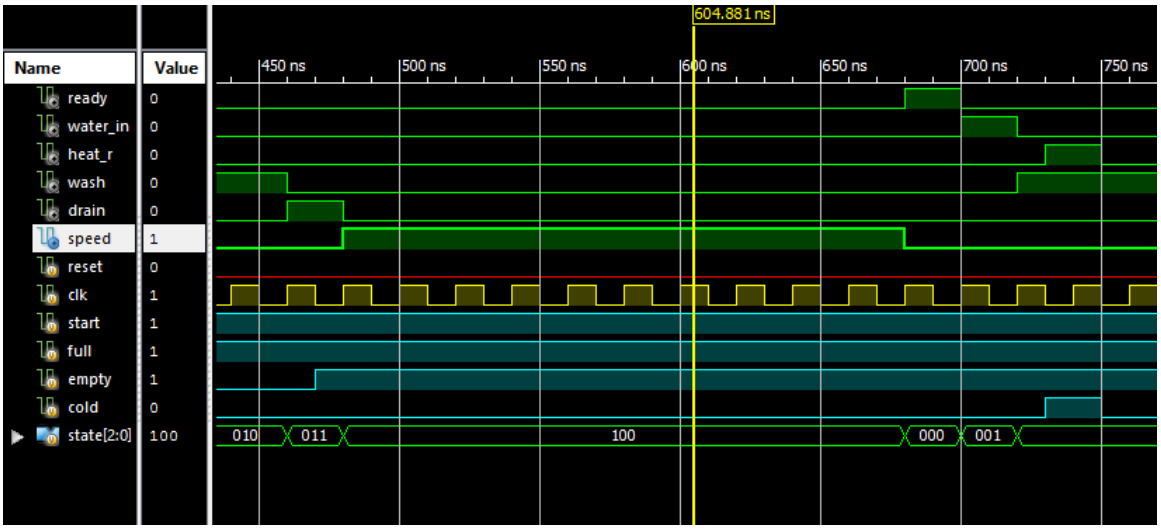


Figure 5: View of the simulation with states 2, 3, 4, 0, 1 and 2

3 Verilog code

Listing 2: wash_fsm.v

```
////////////////////////////////////////////////////////////////////////////////
// Author: Diogo Vilar Sardinha
// Module Name:    wash_fsm.v
// Project Name:  Laboratory - Digital Circuits - State machine
// Target Devices: Basys 2 Spartan-3E FPGA
// Description:
////////////////////////////////////////////////////////////////////////////////

`timescale 1ns / 1ns

module wash_fsm
  ↪  (clk,ready,water_in,wash,drain,speed,heat_r,reset,start,full,cold,empty);

 input clk, reset, start, full, cold, empty;
 output ready, water_in, wash,drain, speed, heat_r;
 // output is declared reg so that it can be assigned in an always block:
 reg ready, water_in, wash, drain, speed, heat_r;

 parameter state0=0, state1=1, state2=2, state3=3, state4=4;
 reg[2:0] nxt_st = state0, state = state0;

 /* ----------------------------------------------------------
 Clock -> f = 50MHz ->  T = 1/50M = 20ns
 ----------------------------------------------------------
 4 seconds -> 4 * 50M clock periods -> 4s = (200 000 000)*T
 2 seconds -> 2 * 50M clock periods -> 2s = (100 000 000)*T
 ----------------------------------------------------------
 200 000 000 = 101 11110 10111 10000 10000 00000
 Nr of bits needed on register = 28bits
 ----------------------------------------------------------*/

 // Clock counter for countdown :
 reg[27:0] t = 0 ;
 // State countdown start values :
 parameter st2_counter = 200000000 , st4_counter = 100000000 ;
```

```
// State countdown enabler :
reg countdown_on = 0;

/*-------------------------------------------------------------------------/
  Always Block -> Time countdowns for states transitions
/------------------------------------------------------------------------ */
always @(posedge clk)
 begin: state_change_countdown
   if (countdown_on) t = t + 1;
   else t = 0;
 end

/*-------------------------------------------------------------------------/
    Always Block -> Device Inputs detection
/------------------------------------------------------------------------ */
always @(*)
 begin : next_state_logic
  heat_r = 0;
  countdown_on = 0;
  nxt_st= state;
   case (state)
   state0: begin
       countdown_on = 0;
       if (start) nxt_st = state1;
       end
   state1: begin
       countdown_on = 0;
       if (full) nxt_st = state2;
       end
   state2: begin
       if (cold) heat_r = 1;
       else heat_r = 0;
       countdown_on = 1;
       if (t == st2_counter-1)
        begin
         nxt_st = state3;
         heat_r = 0;
        end
       end
   state3: begin
       countdown_on = 0;
       if (empty) nxt_st = state4;
       end
   state4: begin
       countdown_on = 1;
       if (t == st4_counter-1) nxt_st = state0;
       end

   default: nxt_st = state0; // default avoids latches*/
  endcase
 end

/*-----------------------------------------------------------------------------------/
      Always Block -> Reset input detection
/-------------------------------------------------------------------------------
↪  */
always @(posedge reset, posedge clk)
 begin : reset_detector
  if (reset) state <= state0;
  else state <= nxt_st;
 end
```

```
95
96  /*-------------------------------------------------------------------------------------*/
97        Always Block -> Output actions on each state
98  /*-------------------------------------------------------------------------------------*/
99  always @(state)
100   begin : output_logic
101     case (state)
102      state0: begin ready=1;water_in=0;wash=0;drain=0;speed=0;end
103      state1: begin ready=0;water_in=1;wash=0;drain=0;speed=0;end
104      state2: begin ready=0;water_in=0;wash=1;drain=0;speed=0;end
105      state3: begin ready=0;water_in=0;wash=0;drain=1;speed=0;end
106      state4: begin ready=0;water_in=0;wash=0;drain=0;speed=1;end
107      default begin ready=1;water_in=0;wash=0;drain=0;speed=0;end
108     endcase
109   end
110
111 endmodule
```

4 Place and Route

4.1 Report place and route (f = 50MHz)

I tested the behavior of the board for the frequency of 50Mhz (T = 10ns) by changing the clock settings in the ucf file that allows the mapping of the board outputs.

Listing 3: wash_fsm.par 50Hz

```
1  Release 14.7 par P.20131013 (nt)
2  Copyright (c) 1995-2013 Xilinx, Inc.  All rights reserved.
3
4  DIOGOLAPTOP::
5
6  par -filter C:/.Xilinx/wash_fsm/iseconfig/filter.filter -w -intstyle ise -ol
7  high -t 1 wash_fsm_map.ncd wash_fsm.ncd wash_fsm.pcf
8
9
10 Constraints file: wash_fsm.pcf.
11 Loading device for application Rf_Device from file '3s100e.nph' in environment
   ↪  C:\Xilinx\14.7\ISE_DS\ISE\.
12   "wash_fsm" is an NCD, version 3.2, device xc3s100e, package cp132, speed -4
13
14 Initializing temperature to 85.000 Celsius. (default - Range: -40.000 to
   ↪  100.000 Celsius)
15 Initializing voltage to 1.140 Volts. (default - Range: 1.140 to 1.320 Volts)
16
17
18 Device speed data version:  "PRODUCTION 1.27 2013-10-13".
19
20
21 Design Summary Report:
22
23  Number of External IOBs                          12 out of 83     14%
24
25    Number of External Input IOBs                  6
26
27       Number of External Input IBUFs              6
28         Number of LOCed External Input IBUFs      6 out of 6     100%
29
```

```
30
31    Number of External Output IOBs                 6
32
33       Number of External Output IOBs              6
34         Number of LOCed External Output IOBs      6 out of 6     100%
35
36
37    Number of External Bidir IOBs                  0
38
39
40    Number of BUFGMUXs                        1 out of 24      4%
41    Number of Slices                         30 out of 960     3%
42       Number of SLICEMs                      0 out of 480     0%
43
44
45
46 Overall effort level (-ol):   High
47 Placer effort level (-pl):    High
48 Placer cost table entry (-t): 1
49 Router effort level (-rl):    High
50
51 Starting initial Timing Analysis.  REAL time: 1 secs
52 Finished initial Timing Analysis.  REAL time: 1 secs
53
54
55 Starting Placer
56 Total REAL time at the beginning of Placer: 1 secs
57 Total CPU  time at the beginning of Placer: 1 secs
58
59 Phase 1.1  Initial Placement Analysis
60 Phase 1.1  Initial Placement Analysis (Checksum:10a301f5) REAL time: 1 secs
61
62 Phase 2.7  Design Feasibility Check
63 Phase 2.7  Design Feasibility Check (Checksum:10a301f5) REAL time: 1 secs
64
65 Phase 3.31  Local Placement Optimization
66 Phase 3.31  Local Placement Optimization (Checksum:10a301f5) REAL time: 1 secs
67
68 Phase 4.2  Initial Clock and IO Placement
69
70 Phase 4.2  Initial Clock and IO Placement (Checksum:11efa21d) REAL time: 1 secs
71
72 Phase 5.30  Global Clock Region Assignment
73 Phase 5.30  Global Clock Region Assignment (Checksum:11efa21d) REAL time: 1
   ↪  secs
74
75 Phase 6.36  Local Placement Optimization
76 Phase 6.36  Local Placement Optimization (Checksum:11efa21d) REAL time: 1 secs
77
78 Phase 7.8  Global Placement
79 ..
80 ..
81 Phase 7.8  Global Placement (Checksum:347ae458) REAL time: 2 secs
82
83 Phase 8.5  Local Placement Optimization
84 Phase 8.5  Local Placement Optimization (Checksum:347ae458) REAL time: 2 secs
85
86 Phase 9.18  Placement Optimization
87 Phase 9.18  Placement Optimization (Checksum:36060b8a) REAL time: 2 secs
88
89 Phase 10.5  Local Placement Optimization
```

```
Phase 10.5  Local Placement Optimization (Checksum:36060b8a) REAL time: 2 secs

Total REAL time to Placer completion: 2 secs
Total CPU  time to Placer completion: 2 secs
Writing design to file wash_fsm.ncd



Starting Router


Phase  1  : 187 unrouted;      REAL time: 3 secs

Phase  2  : 169 unrouted;      REAL time: 3 secs

Phase  3  : 14 unrouted;      REAL time: 3 secs

Phase  4  : 14 unrouted; (Setup:0, Hold:0, Component Switching Limit:0)
 ↪  REAL time: 3 secs

Phase  5  : 0 unrouted; (Setup:0, Hold:0, Component Switching Limit:0)     REAL
 ↪  time: 3 secs

Updating file: wash_fsm.ncd with current fully routed design.

Phase  6  : 0 unrouted; (Setup:0, Hold:0, Component Switching Limit:0)     REAL
 ↪  time: 3 secs

Phase  7  : 0 unrouted; (Setup:0, Hold:0, Component Switching Limit:0)     REAL
 ↪  time: 3 secs

Phase  8  : 0 unrouted; (Setup:0, Hold:0, Component Switching Limit:0)     REAL
 ↪  time: 3 secs

Total REAL time to Router completion: 3 secs
Total CPU time to Router completion: 3 secs

Partition Implementation Status
-------------------------------

  No Partitions were found in this design.

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

Generating "PAR" statistics.

**************************
Generating Clock Report
**************************

+---------------------+--------------+------+------+------------+-------------+
|        Clock Net    |   Resource   |Locked|Fanout|Net Skew(ns)|Max Delay(ns)|
+---------------------+--------------+------+------+------------+-------------+
|           clk_BUFGP | BUFGMUX_X2Y11| No   |   17 |  0.004     |  0.060      |
+---------------------+--------------+------+------+------------+-------------+

* Net Skew is the difference between the minimum and maximum routing
only delays for the net. Note this is different from Clock Skew which
is reported in TRCE timing report. Clock Skew is the difference between
the minimum and maximum path delays which includes logic delays.
```

```
* The fanout is the number of component pins not the individual BEL loads,
for example SLICE loads not FF loads.

Timing Score: 0 (Setup: 0, Hold: 0, Component Switching Limit: 0)

Asterisk (*) preceding a constraint indicates it was not met.
   This may be due to a setup or hold violation.

--------------------------------------------------------------------------------------------------------
  Constraint                                |    Check    | Worst Case |  Best
   ↪  Case | Timing |   Timing
                                            |             |    Slack   |
                                            ↪  Achievable | Errors |    Score
--------------------------------------------------------------------------------------------------------
  TS_clkin = PERIOD TIMEGRP "clk" 10 ns HIG | SETUP       |     4.468ns|
   ↪  5.532ns|       0|           0
  H 50%                                     | HOLD        |     1.478ns|
   ↪  |       0|           0
--------------------------------------------------------------------------------------------------------


All constraints were met.


Generating Pad Report.

All signals are completely routed.

Total REAL time to PAR completion: 3 secs
Total CPU time to PAR completion: 3 secs

Peak Memory Usage:  274 MB

Placement: Completed - No errors found.
Routing: Completed - No errors found.
Timing: Completed - No errors found.

Number of error messages: 0
Number of warning messages: 0
Number of info messages: 0

Writing design to file wash_fsm.ncd



PAR done!
```

4.2 Report place and route for the maximum operating frequency (f = 214,2MHz)

By changing the clock settings in the ucf file that allows the mapping of the board outputs, I tested the behavior of the board for the frequency of 214.2MHz. This was the maximum frequency for which the circuit still worked correctly and it was possible to correctly implement the place and route of the circuit.

Listing 4: wash_fsm.par 214.2MHz

```
Release 14.7 par P.20131013 (nt)
Copyright (c) 1995-2013 Xilinx, Inc.  All rights reserved.
```

```
3
4  DIOGOLAPTOP::
5
6  par -filter C:/.Xilinx/wash_fsm/iseconfig/filter.filter -w -intstyle ise -ol
7  high -t 1 wash_fsm_map.ncd wash_fsm.ncd wash_fsm.pcf
8
9
10 Constraints file: wash_fsm.pcf.
11 Loading device for application Rf_Device from file '3s100e.nph' in environment
   ↪  C:\Xilinx\14.7\ISE_DS\ISE\.
12    "wash_fsm" is an NCD, version 3.2, device xc3s100e, package cp132, speed -4
13
14 Initializing temperature to 85.000 Celsius. (default - Range: -40.000 to
   ↪  100.000 Celsius)
15 Initializing voltage to 1.140 Volts. (default - Range: 1.140 to 1.320 Volts)
16
17
18 Device speed data version:  "PRODUCTION 1.27 2013-10-13".
19
20
21 Design Summary Report:
22
23  Number of External IOBs                          12 out of 83     14%
24
25    Number of External Input IOBs                  6
26
27       Number of External Input IBUFs              6
28         Number of LOCed External Input IBUFs      6 out of 6      100%
29
30
31    Number of External Output IOBs                 6
32
33       Number of External Output IOBs              6
34         Number of LOCed External Output IOBs      6 out of 6      100%
35
36
37    Number of External Bidir IOBs                  0
38
39
40    Number of BUFGMUXs                        1 out of 24      4%
41    Number of Slices                         30 out of 960     3%
42       Number of SLICEMs                      0 out of 480     0%
43
44
45
46 Overall effort level (-ol):   High
47 Placer effort level (-pl):    High
48 Placer cost table entry (-t): 1
49 Router effort level (-rl):    High
50
51 Starting initial Timing Analysis.  REAL time: 1 secs
52 Finished initial Timing Analysis.  REAL time: 1 secs
53
54
55 Starting Placer
56 Total REAL time at the beginning of Placer: 1 secs
57 Total CPU  time at the beginning of Placer: 1 secs
58
59 Phase 1.1  Initial Placement Analysis
60 Phase 1.1  Initial Placement Analysis (Checksum:10a301f5) REAL time: 1 secs
61
```

```
Phase 2.7  Design Feasibility Check
Phase 2.7  Design Feasibility Check (Checksum:10a301f5) REAL time: 1 secs

Phase 3.31  Local Placement Optimization
Phase 3.31  Local Placement Optimization (Checksum:10a301f5) REAL time: 1 secs

Phase 4.2  Initial Clock and IO Placement

Phase 4.2  Initial Clock and IO Placement (Checksum:11efa21d) REAL time: 1 secs

Phase 5.30  Global Clock Region Assignment
Phase 5.30  Global Clock Region Assignment (Checksum:11efa21d) REAL time: 1
  ↪  secs

Phase 6.36  Local Placement Optimization
Phase 6.36  Local Placement Optimization (Checksum:11efa21d) REAL time: 1 secs

Phase 7.8  Global Placement
..
..
..
..
..
..
Phase 7.8  Global Placement (Checksum:341b8d44) REAL time: 2 secs

Phase 8.5  Local Placement Optimization
Phase 8.5  Local Placement Optimization (Checksum:341b8d44) REAL time: 2 secs

Phase 9.18  Placement Optimization
Phase 9.18  Placement Optimization (Checksum:365dd31f) REAL time: 2 secs

Phase 10.5  Local Placement Optimization
Phase 10.5  Local Placement Optimization (Checksum:365dd31f) REAL time: 2 secs

Total REAL time to Placer completion: 2 secs
Total CPU  time to Placer completion: 2 secs
Writing design to file wash_fsm.ncd



Starting Router


Phase  1  : 187 unrouted;      REAL time: 3 secs

Phase  2  : 169 unrouted;      REAL time: 3 secs

Phase  3  : 17 unrouted;      REAL time: 3 secs

Phase  4  : 22 unrouted; (Setup:0, Hold:0, Component Switching Limit:0)
  ↪  REAL time: 3 secs

Phase  5  : 0 unrouted; (Setup:199, Hold:0, Component Switching Limit:0)
  ↪  REAL time: 3 secs

Updating file: wash_fsm.ncd with current fully routed design.

Phase  6  : 0 unrouted; (Setup:199, Hold:0, Component Switching Limit:0)
  ↪  REAL time: 3 secs

```

```
Phase  7  : 0 unrouted; (Setup:0, Hold:0, Component Switching Limit:0)     REAL time: 3 secs

Phase  8  : 0 unrouted; (Setup:0, Hold:0, Component Switching Limit:0)     REAL time: 3 secs

Phase  9  : 0 unrouted; (Setup:0, Hold:0, Component Switching Limit:0)     REAL time: 3 secs

Total REAL time to Router completion: 3 secs
Total CPU time to Router completion: 3 secs

Partition Implementation Status
-------------------------------

  No Partitions were found in this design.

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

Generating "PAR" statistics.

**************************
Generating Clock Report
**************************

+----------------------+--------------+------+------+------------+-------------+
|        Clock Net     |   Resource   |Locked|Fanout|Net Skew(ns)|Max Delay(ns)|
+----------------------+--------------+------+------+------------+-------------+
|           clk_BUFGP | BUFGMUX_X2Y11| No   |   17 |  0.006     |  0.059      |
+----------------------+--------------+------+------+------------+-------------+

* Net Skew is the difference between the minimum and maximum routing
only delays for the net. Note this is different from Clock Skew which
is reported in TRCE timing report. Clock Skew is the difference between
the minimum and maximum path delays which includes logic delays.

* The fanout is the number of component pins not the individual BEL loads,
for example SLICE loads not FF loads.

Timing Score: 0 (Setup: 0, Hold: 0, Component Switching Limit: 0)

Asterisk (*) preceding a constraint indicates it was not met.
   This may be due to a setup or hold violation.

----------------------------------------------------------------------------------------------------------
  Constraint                                |    Check    | Worst Case |  Best Case | Timing |   Timing
                                            |             |    Slack   |            | Score
----------------------------------------------------------------------------------------------------------
  TS_clkin = PERIOD TIMEGRP "clk" 4.7 ns HI | SETUP       |     0.032ns|     4.668ns|       0|           0
  GH 50%                                    | HOLD        |     1.411ns|            |       0|           0
----------------------------------------------------------------------------------------------------------


All constraints were met.


Generating Pad Report.
```

```
173
174 All signals are completely routed.
175
176 Total REAL time to PAR completion: 4 secs
177 Total CPU time to PAR completion: 4 secs
178
179 Peak Memory Usage:  273 MB
180
181 Placement: Completed - No errors found.
182 Routing: Completed - No errors found.
183 Timing: Completed - No errors found.
184
185 Number of error messages: 0
186 Number of warning messages: 0
187 Number of info messages: 0
188
189 Writing design to file wash_fsm.ncd
190
191
192
193 PAR done!
```

5 Final considerations

The purpose of this lab was to increase my understanding of digital systems design using one of the most common hardware description languages (HDL) .

Using the Verilog language, this project was developed to implement a state machine to control a washing machine in FPGA.

I started by drawing the states and transitions diagram, detailing the intended operation for the state machine, and through it I wrote the code that was implemented on the board, after a brief review of the code writing process in Verilog.
Then, using xilinx software and using a Verilog test bench that allowed me to test several signals on the model without being necessary hardware. A test bench is a very useful working tool in the development of digital circuits.

After running a series of simulations I verified in Xilinx the correct functioning of the Verilog code developed to implement my model.

To obtain the maximum frequency at which the circuit can operate, I performed a set of tests: by changing the operating time to successively smaller values in the cfu file and I ran the "Place and Route" until it reported that it was impossible to implement the design correctly.

I ended up getting the incredible result of T = 4.688ns of period, which corresponds to an operating frequency of 214.2MHz! This may be due to the fact that in the code separate processes exist for the outputs and states, thus allowing the machine to operate with these two processes in parallel.
I also tested the operation of the circuit on the Basys2 board, using the diagram in Figure 1 to validate the simulations and verify the correct functioning of the state machine, which has been validated.

List of Figures

List of source codes