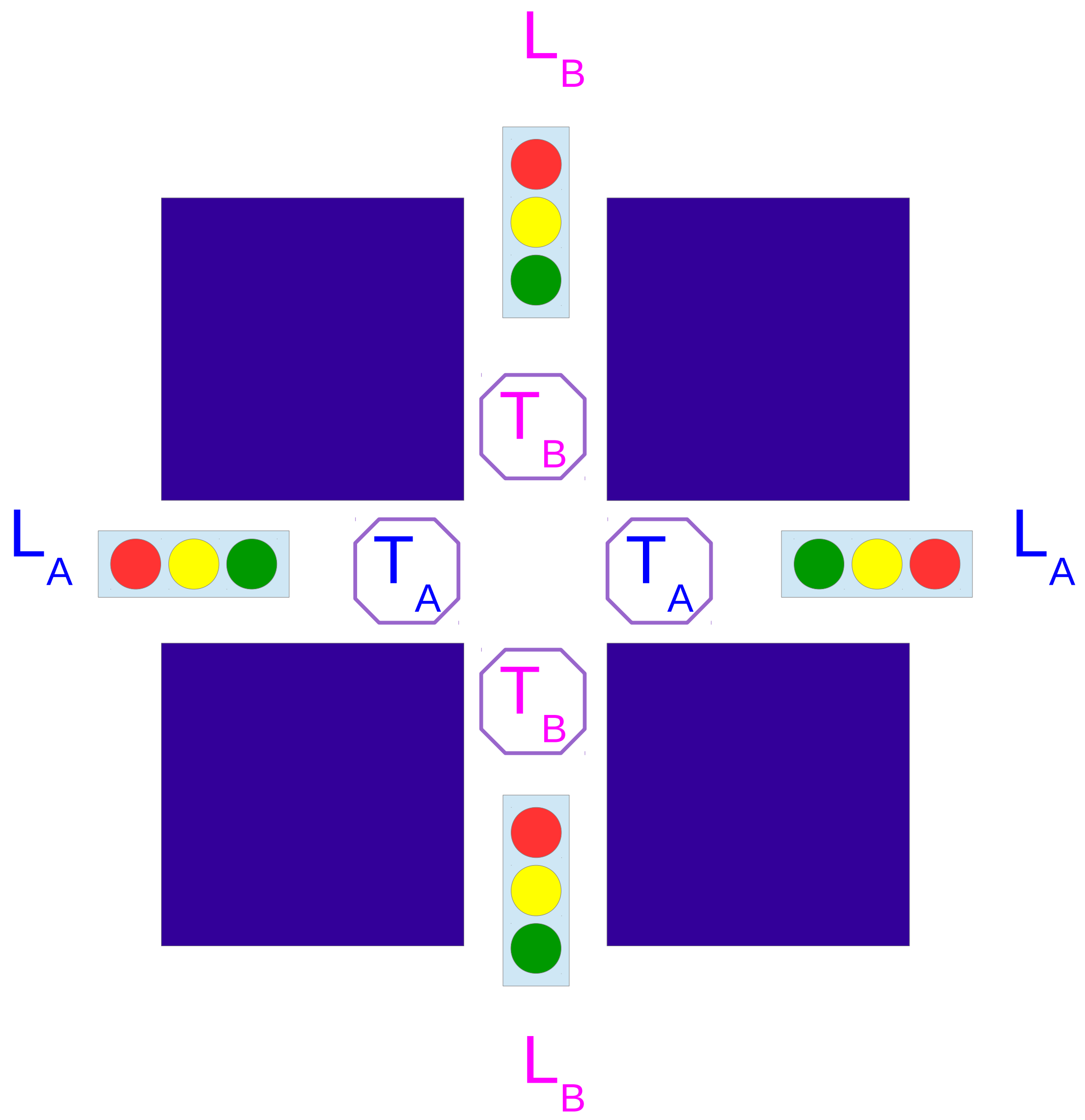
RTL Design Example (1A)

Young Won Lim
7/16/16

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

FSM Inputs and Outputs



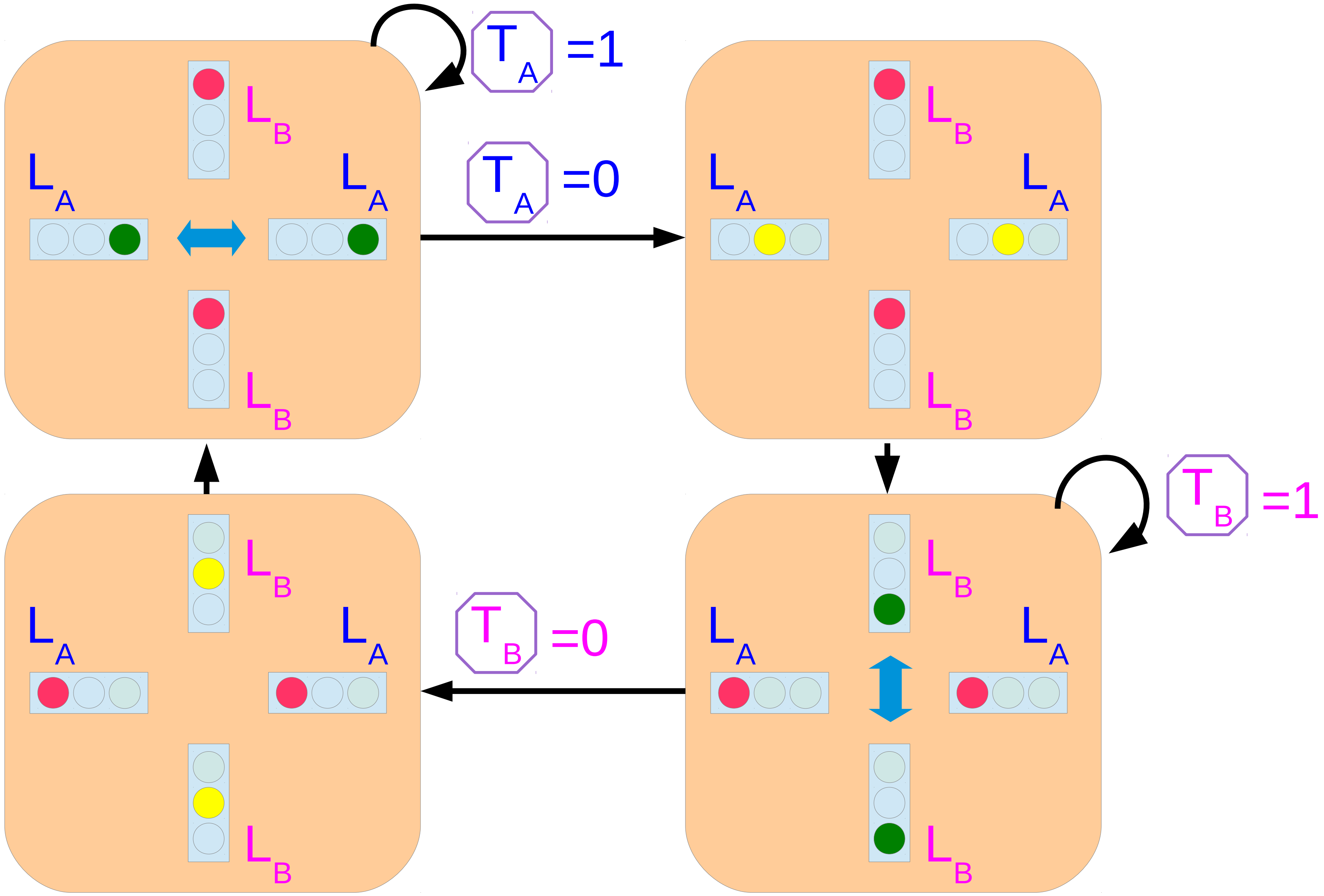
Traffic Lights - Outputs

L_A L_B

Sensor - Inputs

T_A T_B

States



traffic_controller – behavioral level (1)

```
module traffic_controller
                  (clock, reset, TA, TB, LA, LB);
  input clock, reset;
  input TA, TB;
  output [1:0] LA, LB;
  reg    [1:0] LA, LB;

  parameter [1:0] S0 = 0, S1 = 1, S2 = 2, S3 = 3;
  parameter [1:0] GREEN     = 2'b00,
                  YELLOW    = 2'b01,
                  RED       = 2'b10;



  reg    [1:0] S, NextS;

  always @(posedge clock)
    begin: SEQ
      if (reset)
        S = S0;
      else
         S = NextS;
    end
```

traffic_controller – behavioral level (2)

```
always @(S or TA or TB)
  begin: COMB
    case (S)
      S0:
        begin     🟦     end

      S1:
        begin     🟨     end

      S2:
        begin     🟧     end

      S3:
        begin     🟧     end

      default:
        begin
          NextS = S0;
          LA = GREEN;
          LB = RED;
        end
    endcase
  end

endmodule
```

```
if (TA)
  NextS = S0;
else
  NextS = S1;
LA = GREEN;
LB = RED;
```

```
NextS = S2;
LA = YELLOW;
LB = RED;
```

```
if (TB)
  NextS = S2;
else
  NextS = S3;
LA = RED;
LB = GREEN;
```

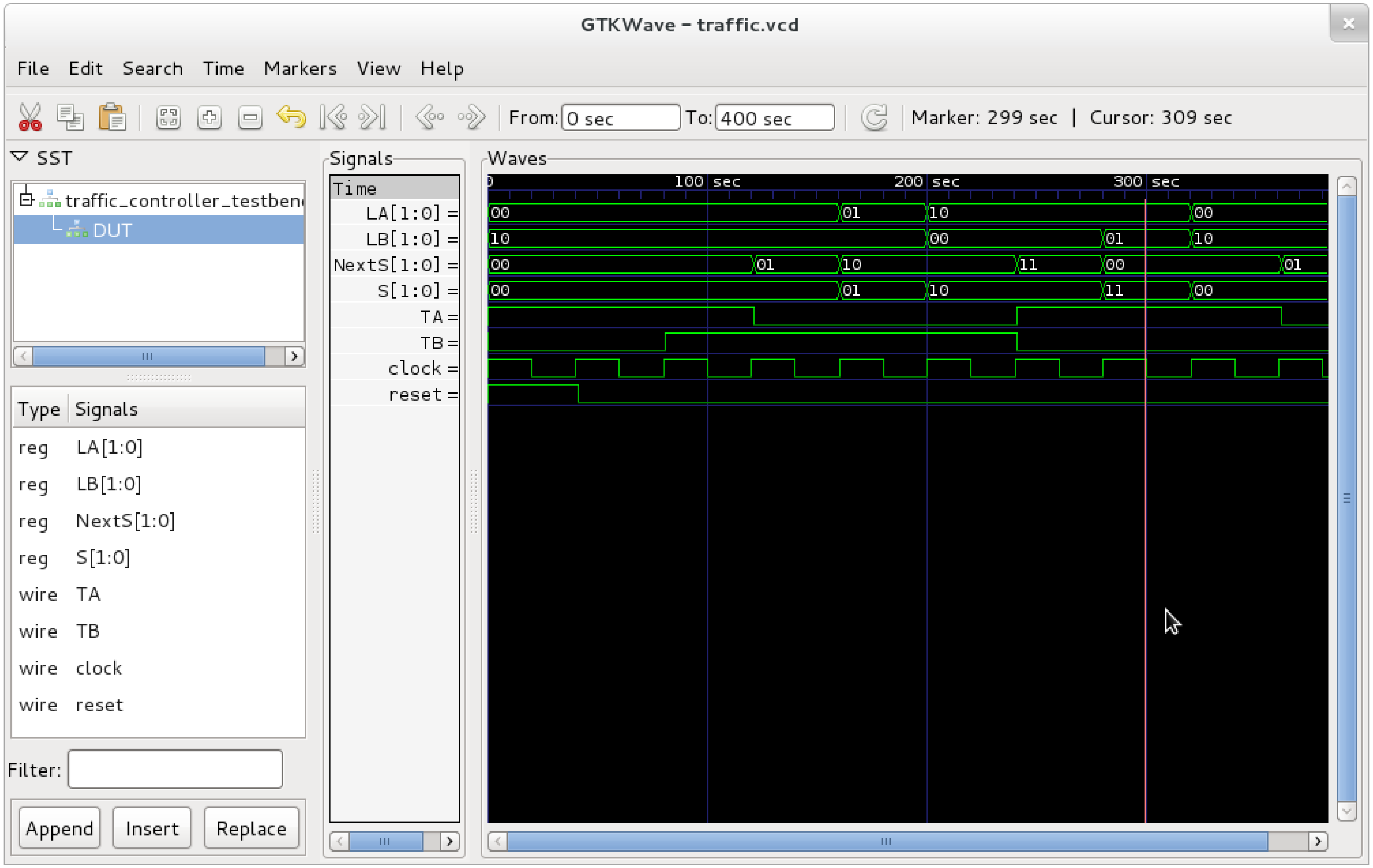
```
NextS = S0;
LA = RED;
LB = YELLOW;
```

Testbench

```
module traffic_controller_testbench;

  parameter cycle = 40;

  reg clock;

  always
    begin
      #(cycle/2) clock=~clock;
    end


  traffic_controller DUT (.clock(clock),
                          .reset(reset),
                          .TA(TA),
                          .TB(TB),
                          .LA(LA),
                          .LB(LB) );

  reg         reset;
  reg         TA, TB;
  wire [1:0] LA, LB;
```

```
  initial
    begin
      clock = 1;
      reset = 1;
      TA = 1;
      TB = 0;

      #1;
      #(cycle)      reset = 0;
      #(cycle)      TB = 1;
      #(cycle)      TA = 0;
      #(cycle*3)      TA = 1;      TB = 0;
      #(cycle*3)       TA = 0;
    end

  initial
    begin
      $dumpfile("traffic.vcd");
      $dumpvars(0, DUT);
      #(cycle * 10);
      $finish;
    end


endmodule
```

Gtkwave Results



Type	Signals
reg	LA[1:0]
reg	LB[1:0]
reg	NextS[1:0]
reg	S[1:0]
wire	TA
wire	TB
wire	clock
wire	reset

```
LA[1:0]    00  01  10  00
LB[1:0]    10  00  01  10
NextS[1:0] 00  01  10  11  00  01
S[1:0]     00  01  10  11  00
```

traffic_controller – synthesized gate level (1)

```
module traffic_controller (
  clock, reset, TA, TB, LA, LB
);
  input clock;
  input reset;
  input TA;
  input TB;
  output [1 : 0] LA;
  output [1 : 0] LB;
  wire clock_BUFGP_0;
  wire reset_IBUF_1;
  wire TA_IBUF_2;
  wire TB_IBUF_3;
  wire LB_1_OBUF_4;
  wire LB_0_OBUF_5;
  wire S_FSM_FFd1_6;
  wire LA_0_OBUF_7;
  wire \S_FSM_FFd2-In ;
  wire S_FSM_FFd2_9;

endmodule
```

```
  FDR   S_FSM_FFd2 (
    .C(clock_BUFGP_0),
    .D(\S_FSM_FFd2-In ),
    .R(reset_IBUF_1),
    .Q(S_FSM_FFd2_9)
  );

  FDR   S_FSM_FFd1 (
    .C(clock_BUFGP_0),
    .D(S_FSM_FFd2_9),
    .R(reset_IBUF_1),
    .Q(S_FSM_FFd1_6)
  );
```

Using Xilinx ISE

traffic_controller – synthesized gate level (2)

```
LUT2 #(
  .INIT ( 4'h2 ))
\S__n0018<0>1  (
  .I0(S_FSM_FFd2_9),
  .I1(S_FSM_FFd1_6),
  .O(LA_0_OBUF_7)
);

LUT2 #(
  .INIT ( 4'h2 ))
\S__n0018<2>1  (
  .I0(S_FSM_FFd1_6),
  .I1(S_FSM_FFd2_9),
  .O(LB_0_OBUF_5)
);

LUT4 #(
  .INIT ( 16'hC0F5 ))
\S_FSM_FFd2-In1  (
  .I0(TA_IBUF_2),
  .I1(TB_IBUF_3),
  .I2(S_FSM_FFd2_9),
  .I3(S_FSM_FFd1_6),
  .O(\S_FSM_FFd2-In )
);
```

```
IBUF   reset_IBUF (
  .I(reset),
  .O(reset_IBUF_1)
);

IBUF   TA_IBUF (
  .I(TA),
  .O(TA_IBUF_2)
);

IBUF   TB_IBUF (
  .I(TB),
  .O(TB_IBUF_3)
);
```

Using Xilinx ISE

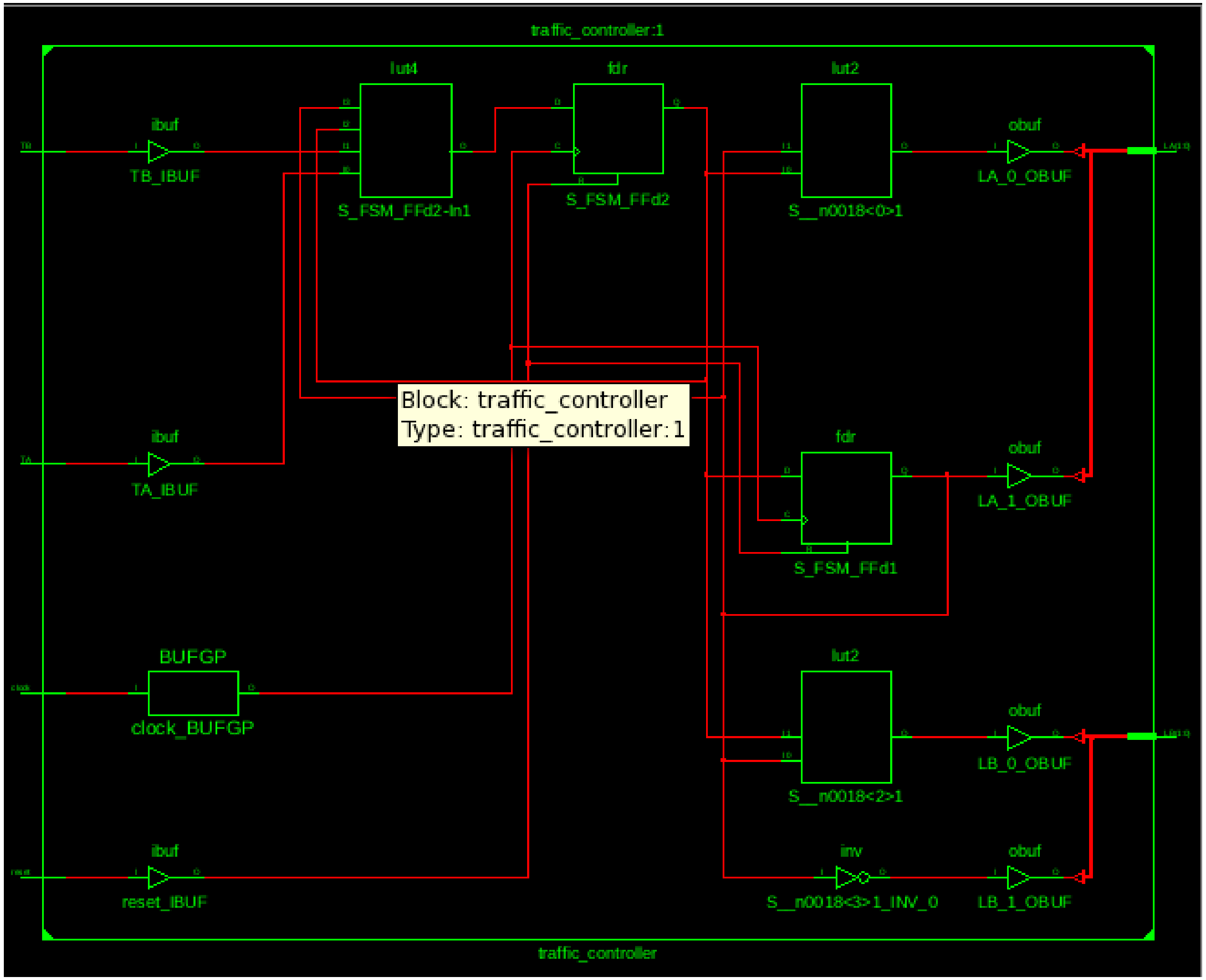
traffic_controller – synthesized gate level (3)

```
OBUF   LA_1_OBUF (
  .I(S_FSM_FFd1_6),
  .O(LA[1])
);

OBUF   LA_0_OBUF (
  .I(LA_0_OBUF_7),
  .O(LA[0])
);

OBUF   LB_1_OBUF (
  .I(LB_1_OBUF_4),
  .O(LB[1])
);

OBUF   LB_0_OBUF (
  .I(LB_0_OBUF_5),
  .O(LB[0])
);
```

```
BUFGP   clock_BUFGP (
  .I(clock),
  .O(clock_BUFGP_0)
);

INV   \S__n0018<3>1_INV_0  (
  .I(S_FSM_FFd1_6),
  .O(LB_1_OBUF_4)
);
```

Using Xilinx ISE

traffic_controller – synthesized gate level (5)



Using Xilinx ISE

References

[1] http://en.wikipedia.org/
[2] http://www.allaboutcircuits.com/
[3] W. Wolf, “Modern VLSI Design : Systems on Silicon
[4] N. Weste, D. Harris, “CMOS VLSI Design: A Circuits and Systems Perspective”
[5] J. P. Uyemura, “Introduction to VLSI Circuits and Systems”
[6] https://en.wikiversity.org/wiki/The_necessities_in_SOC_Design
[7] https://en.wikiversity.org/wiki/The_necessities_in_Digital_Design
[8] https://en.wikiversity.org/wiki/The_necessities_in_Computer_Design
[9] https://en.wikiversity.org/wiki/The_necessities_in_Computer_Architecture
[10] https://en.wikiversity.org/wiki/The_necessities_in_Computer_Organization