Experiment 2 - Sequential Synthesis and FPGA Device Programming

Amirhossein Tayebi 810198430
Armin Eslami 810198349

***Abstract*— This document is about designing and implementing an interaction mechanism using the Cyclone II board**

***Keywords*—state machine design**

I. Serial Transmitter

A. Onepulser

The Moore state machine design of the onepulser, the Verilog code and the testbench results are shown in fig. 1, fig. 2 and fig. 3.

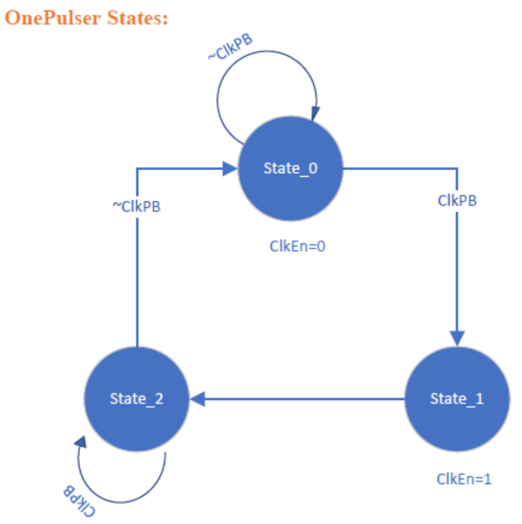
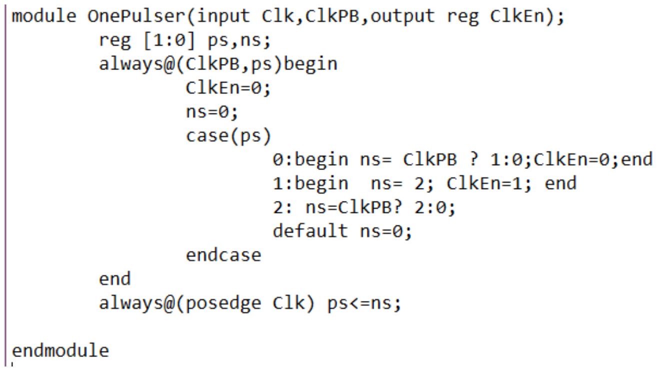


Fig. 1: The moore state machine design of Onepulser



```
module OnePulser(input Clk,ClkPB,output reg ClkEn);
        reg [1:0] ps,ns;
        always@(ClkPB,ps)begin
                ClkEn=0;
                ns=0;
                case(ps)
                        0:begin ns= ClkPB ? 1:0;ClkEn=0;end
                        1:begin  ns= 2; ClkEn=1; end
                        2: ns=ClkPB? 2:0;
                        default ns=0;
                endcase
        end
        always@(posedge Clk) ps<=ns;

endmodule
```

Fig. 2: The Verilog code of Onepulser

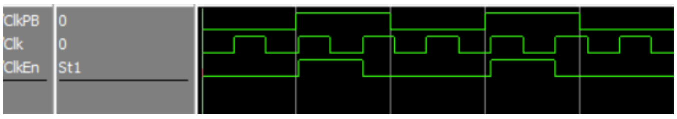


Fig. 3: Testbench waveform of Onepulser module

B. Orthogonal Finite State Machine

The moore state machine design of sequence detector and counter , the Verilog code and the testbench results are shown in fig. 4, fig. 5, fig. 6 and fig. 7.

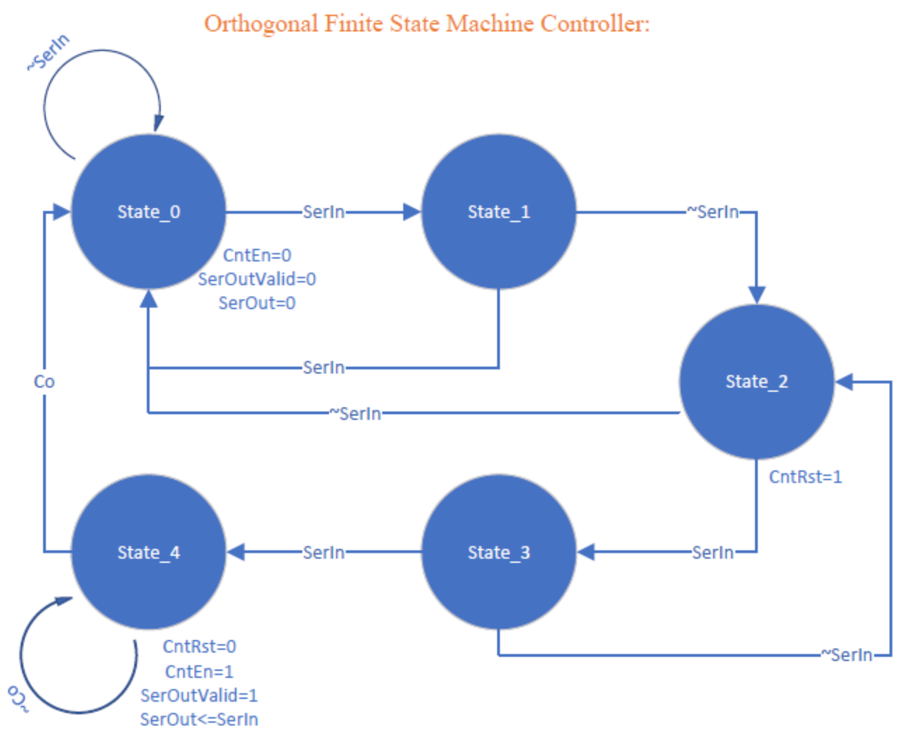
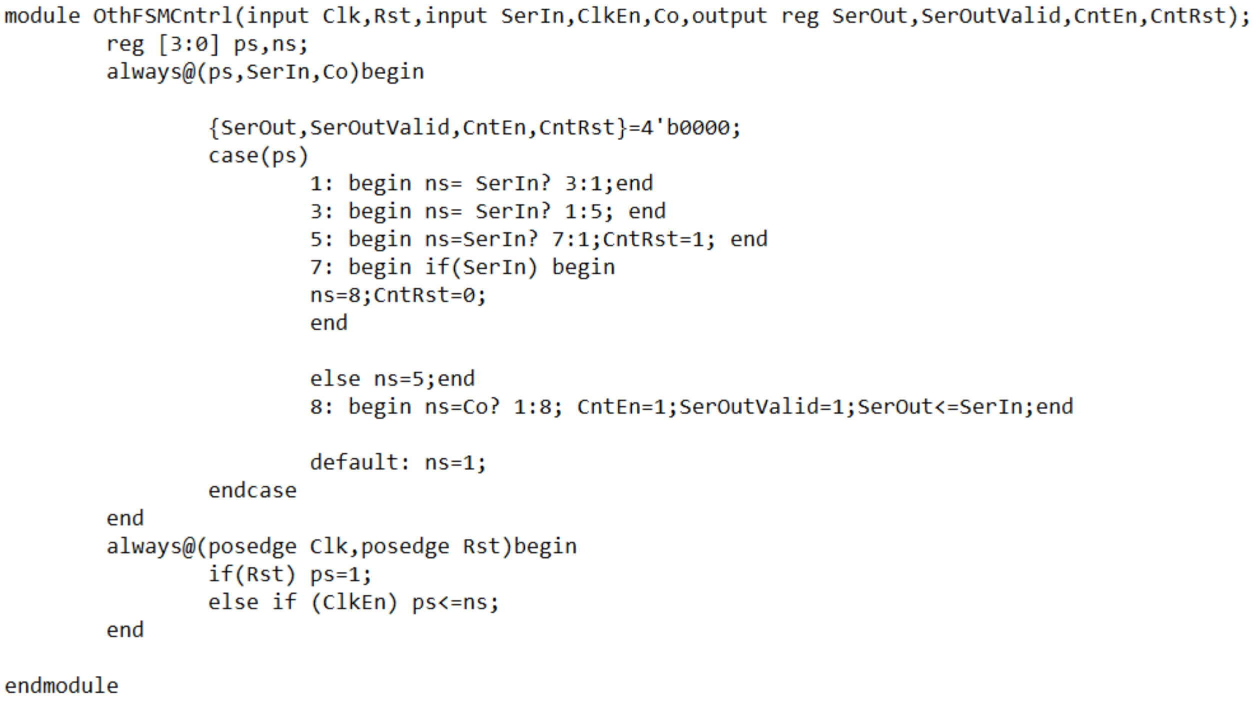
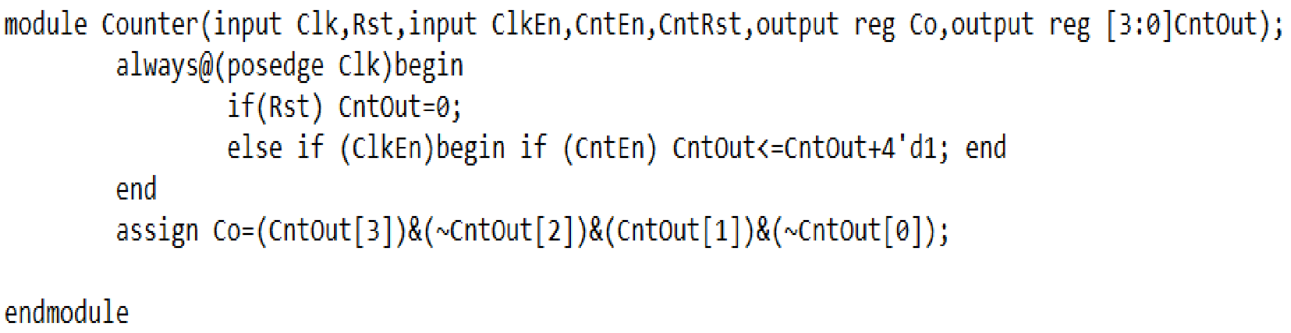


Fig. 4: The moore state machine design of OthFSM



```
module OthFSMCntrl(input Clk,Rst,input SerIn,ClkEn,Co,output reg SerOut,SerOutValid,CntEn,CntRst);
        reg [3:0] ps,ns;
        always@(ps,SerIn,Co)begin

                {SerOut,SerOutValid,CntEn,CntRst}=4'b0000;
                case(ps)
                        1: begin ns= SerIn? 3:1;end
                        3: begin ns= SerIn? 1:5; end
                        5: begin ns=SerIn? 7:1;CntRst=1; end
                        7: begin if(SerIn) begin
                        ns=8;CntRst=0;
                        end

                        else ns=5;end
                        8: begin ns=Co? 1:8; CntEn=1;SerOutValid=1;SerOut<=SerIn;end

                        default: ns=1;
                endcase
        end
        always@(posedge Clk,posedge Rst)begin
                if(Rst) ps=1;
                else if (ClkEn) ps<=ns;
        end

endmodule
```

Fig. 5: The Verilog code of OthFSM controller



```
module Counter(input Clk,Rst,input ClkEn,CntEn,CntRst,output reg Co,output reg [3:0]CntOut);
        always@(posedge Clk)begin
                if(Rst) CntOut=0;
                else if (ClkEn)begin if (CntEn) CntOut<=CntOut+4'd1; end
        end
        assign Co=(CntOut[3])&(~CntOut[2])&(CntOut[1])&(~CntOut[0]);

endmodule
```

Fig. 6: The Verilog code of OthFSM counter

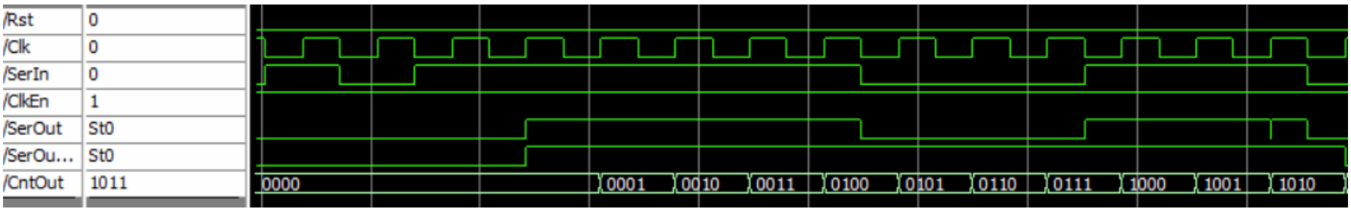


Fig. 7: The testbench waveform of OthFSM module

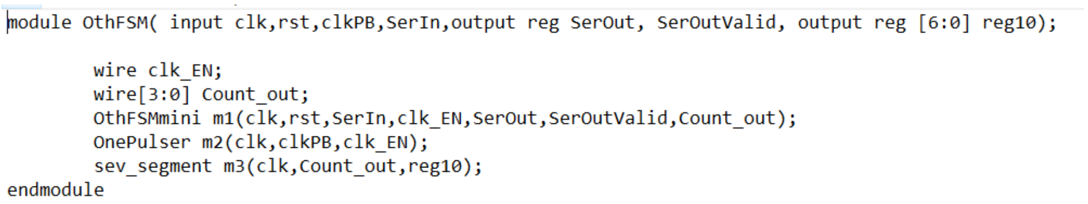
C. *Seven Segment Display*

The Verilog code of the seven segment display is shown in fig. 8.

```
module sev_segment(input clk, input [3:0]d , output reg [6:0] seg);
        always @(d,clk) begin
                case(d)
                        4'd0: seg <= 7'b1000000;
                        4'd1: seg <= 7'b1111001;
                        4'd2: seg <= 7'b0100100;
                        4'd3: seg <= 7'b0110000;
                        4'd4: seg <= 7'b0011001;
                        4'd5: seg <= 7'b0010010;
                        4'd6: seg <= 7'b0000010;
                        4'd7: seg <= 7'b1111000;
                        4'd8: seg <= 7'b0000000;
                        4'd9: seg <= 7'b0010000;
                        default: seg <= 7'b1111111;
                endcase
        end
endmodule
```

Fig. 8: The Verilog code of OthFSM counter

And the top layer wiring is shown in fig. 9.



```
module OthFSM( input clk,rst,clkPB,SerIn,output reg SerOut, SerOutValid, output reg [6:0] reg10);

        wire clk_EN;
        wire[3:0] Count_out;
        OthFSMmini m1(clk,rst,SerIn,clk_EN,SerOut,SerOutValid,Count_out);
        OnePulser m2(clk,clkPB,clk_EN);
        sev_segment m3(clk,Count_out,reg10);
endmodule
```

Fig. 9: The Verilog code of OthFSM module

II. Serial Transmitter Implementation

We are using DE1 development board for our design implementation. The RTL design, Board

image and wiring setup are shown in fig. 10, fig. 11, fig. 12.

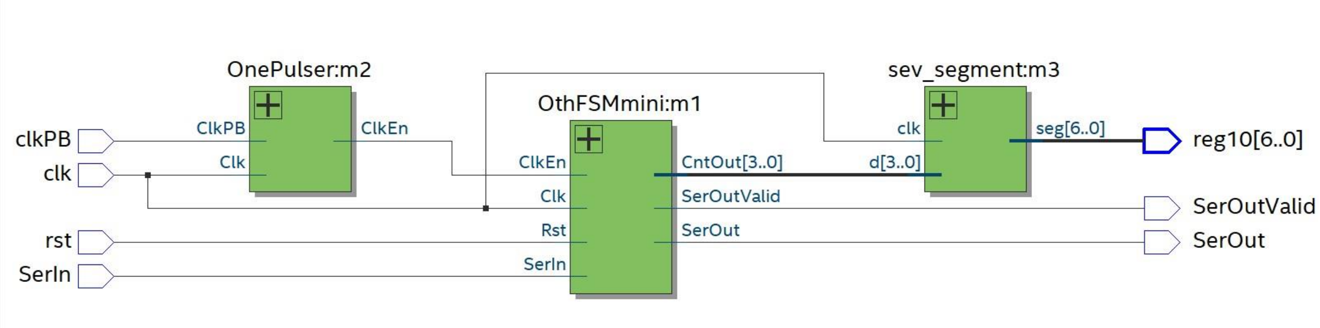
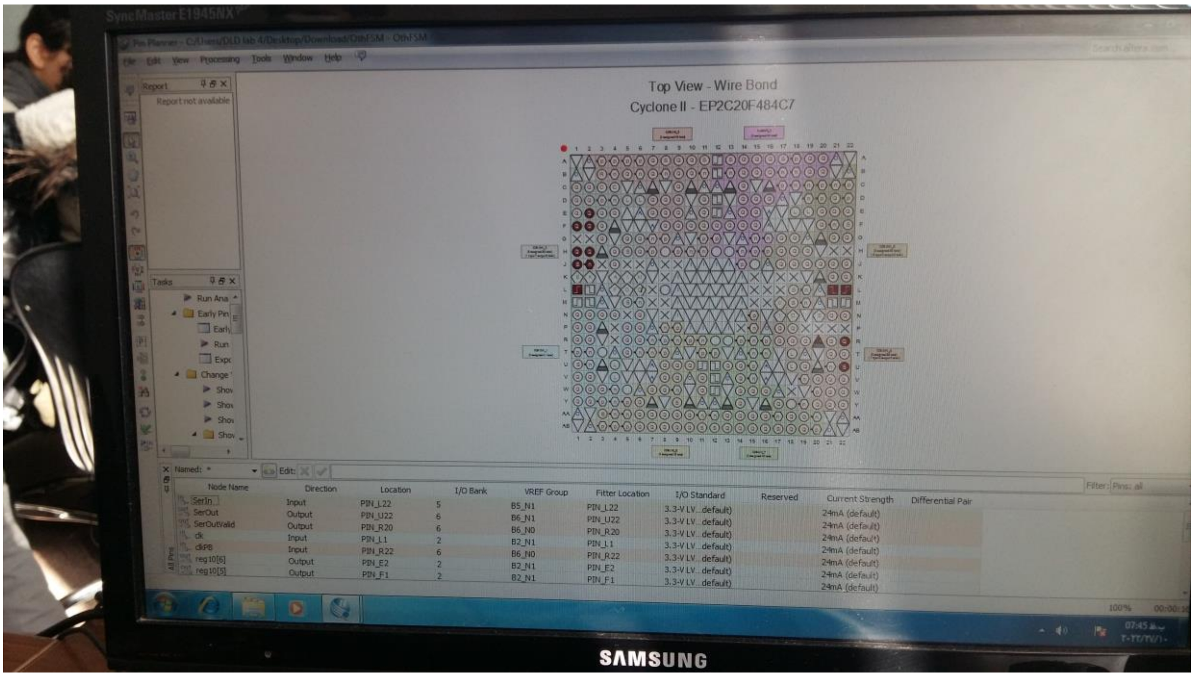


Fig. 10: The RTL design of OthFSM module



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
SerIn	Input	PIN_L22	5	B5_N1	PIN_L22	3.3-V LV...default)		24mA (default)	
SerOut	Output	PIN_U22	6	B6_N1	PIN_U22	3.3-V LV...default)		24mA (default)	
SerOutValid	Output	PIN_R20	6	B6_N0	PIN_R20	3.3-V LV...default)		24mA (default)	
clk	Input	PIN_L1	2	B2_N1	PIN_L1	3.3-V LV...default)		24mA (default)	
clkPB	Input	PIN_R22	6	B6_N0	PIN_R22	3.3-V LV...default)		24mA (default)	
reg10[6]	Output	PIN_E2	2	B2_N1	PIN_E2	3.3-V LV...default)		24mA (default)	
reg10[5]	Output	PIN_F1	2	B2_N1	PIN_F1	3.3-V LV...default)		24mA (default)	

Fig. 11: The wiring view of DE1 board

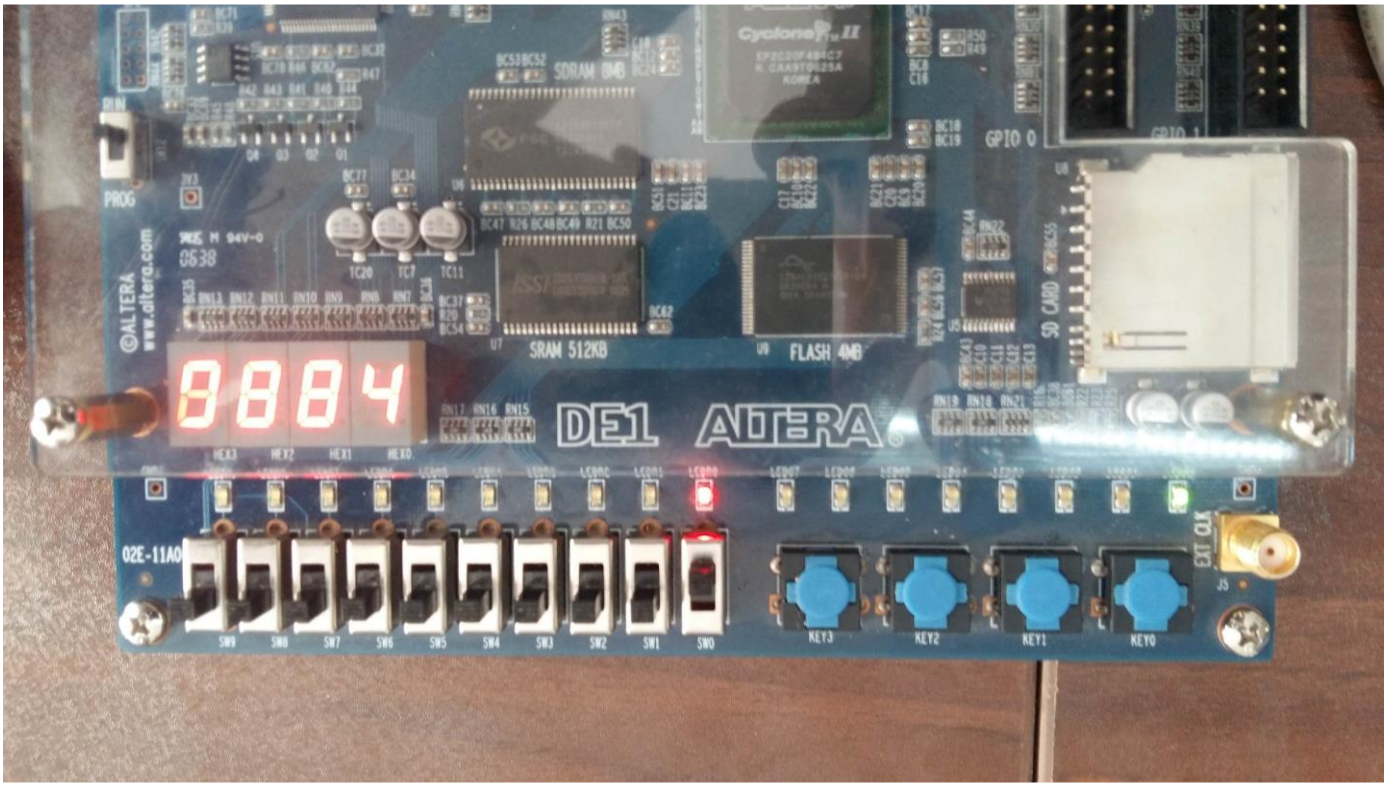


Fig. 12: A glance of DE1 board while executing the rogram