ISTANBUL TECHNICAL UNIVERSITY

Digital System Design Applications

Experiment VII
SEQUENTIAL CIRCUITS

Preliminary

Students should cover sequential circuits and state reduction methods.

Objectives

- Learn design and implementation of sequential circuits.
- Learn difference between Mealy and Moore machines.
- Learn state reduction and encoding

Requirements

Students are expected to be able to

- have knowledge about what sequential circuits are
- know reduction methods

References:

1. Brown&Vrasenic, "Fundementals of Digital Logic with Verilog Design", McGraw-Hill, 1st edition. pp445-521
2. Spartan-3E Libraries Guide for HDL Designs
3. Constraints Guide

SEQUENTIAL CIRCUITS

Create a new ISE project with the settings below:

- Family: **Spartan 3E**
- Device: **XC3S500E**
- Package: **FG320**
- Speed: **-4**
- Synthesis Tool: **XST (VHDL\Verilog)**
- Simulator: **ISim (VHDL\Verilog)**
- Preferred Language: **Verilog**

CIRCUIT THAT DETECTS FOUR CONSECUTIVE 1 OR 0

Description of problem

You will design a circuit that has 1-bit x input and 1-bit z output. D-type flip-flops on the FPGA will be used and all flipflops will be triggered on the rising edge of the clock signal (clk). The circuit will sample the x input at each active clock edge. If four consecutive 1 or 0 come from x input, the output z will be 1. In all other cases, z output will be 0. The state machine to be designed should be a Mealy Machine. Therefore, the output of the circuit will be 1 simultaneously when four consecutive 1 or 0 come from the input. Remember that if the machine was Moore, the output will be 1 after one clock cycle. In Fig. 1, state diagram of this problem is given.

State Reduction

1. Explain in your report what the state reduction is in sequential circuits and how it is done.
2. Is state reduction possible in Fig. 1? Explain with your comments in your report. If it is possible, make reduction and show details.
3. If state reduction is not possible in Fig. 1, use the state table without reduction given in Fig. 2.

State Encoding

1. Explain in your report how to perform efficient state encoding.
2. Make state encoding for the experiment. Show it in your report.
3. If you do not suggest any encoding, use the table given in Fig. 3. The state table to be obtained when the coding in Fig. 3 is used is shown in Fig. 4.

Reduction of combinatorial Parts

1. Make reduction for input functions of each D flipflops (Q2, Q1, Q0) and output function of z by using Quine-McCluskety or Karnaugh Map.
2. Show your reduction results in the report. Do not forget arbitrary states.
3. Think that you will implement combinatorial parts with LUT4. In this case, can you make a different reduction to use the least number of LUT4? Explain in detail in your report.

Verilog Encoding

1. Create new source as **FSM1.v** in your project. Give a same name (FSM1) to your module.
2. Combinatorial parts of the circuit will be implemented with assign.
3. always@(posedge clk) will be used for flipflops.
4. Make synthesize, translate, map, P&R. Show your results in your report. Add schematics and layout to the report.
5. x input comes from switch, clk input comes from button. Connect z output to the LED.

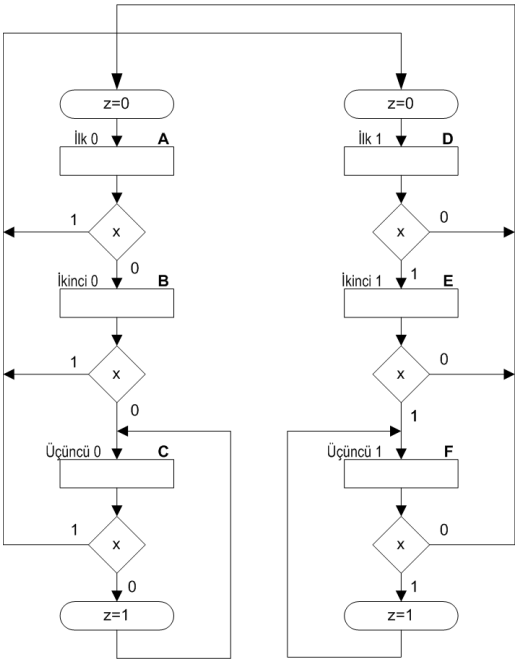

Figure 1: Algorithmic State Diagram

şimdiki durum	sonraki durum, çıkış	
	x = 0	x = 1
A	B,0	D,0
B	C,0	D,0
C	C,1	D,0
D	A,0	E,0
E	A,0	F,0
F	A,0	F,1

Figure 2: State Table without any reduction

durum	ikili kod
A	000
B	001
C	010
D	011
E	100
F	101

Figure 3: State Encoding Example

x	q2	q1	q0	Q2	Q1	Q0	z
0	0	0	0	0	0	1	0
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	0	k	k	k	k
0	1	1	1	k	k	k	k
1	0	0	0	0	1	1	0
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	k	k	k	k
1	1	1	1	k	k	k	k

Figure 4: State Table after Encoding

Simulation of the Circuit

Write a testbench with given information below:

1. Use as an input: 0100110001110000111100000111110000000111111
2. This input will be used from left to right and sampled at the each clock cycle.
3. Make behavioral and post place and route simulation. Give results and waveform in your report.

Analysis of Faulty Outputs

1. Determine if your circuit is producing faulty 0 or faulty 1.
2. Show that what faulty outputs are if any. Under which circumstances the circuit produces these outputs? Explain in the report.

3. For preventing these faulty outputs, what are the methods? Explain in your report.

Machine Type Changing

1. Add D flipflop to the output z of the circuit. Is the circuit still Mealy Machine? Explain in your report.
2. Test your circuit with the test code you write before.
3. Are faulty outputs still there? Explain in your report.

Stucking in Undesirable States

1. Delete D flipflop that you added to the z output in the previous section. Start your circuit from arbitrary states (i.e you do not use this state).
2. Is the circuit switch its state to used states or is it stuck in arbitrary states? Explain in your report.

DESIGN WITH DIVIDED STATE DIAGRAMS

1. Show that, state diagram in Fig. 5 has same functionality with Fig. 1.
2. Create new source, **FSM2.v**. Give same name to your module and design the circuit again by applying previous commands for FSM1. Add all results to the your report.

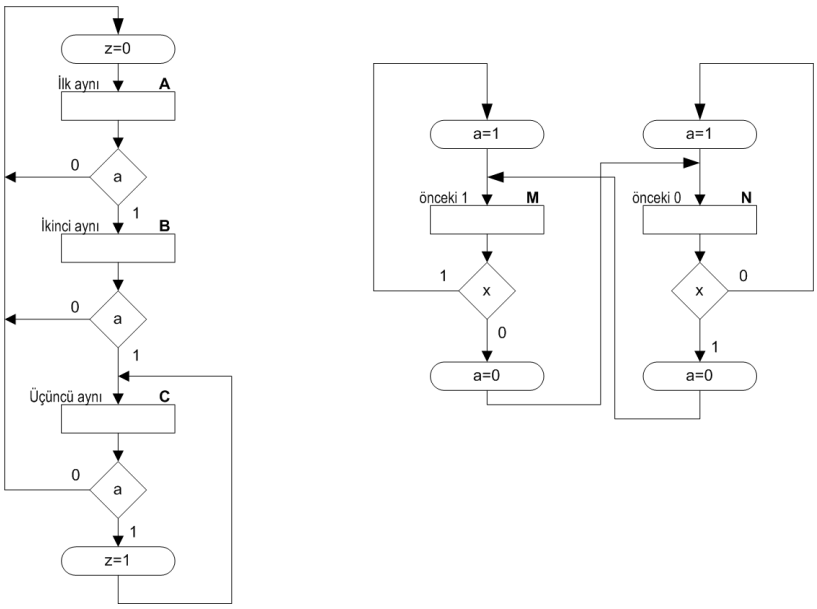

Figure 5: Divided Algorithmic State Table

Experiment Report

Each group member is going to prepare his/her own report. Reports should include:

- Screenshots and results asked above

- Your comments on results
- Design a circuit with 1-bit x input and 1-bit z output. This circuit detects '101' and '100' sequences. If one of these sequences comes twice in succession, the output will be 1 and the circuit stays in a state. Design this circuit with always and case. Examine report in ISE and find that which FSM coding technique is used for the design. Try to design with least number of LUT and D-FF. Try to minimize delays. The best design will be determined and awarded with extra points.

Reports are to be submitted before next experiment. Submission includes reports and project files.