Ojas Anand, David Luria, Maulik Patel

EECE 2060C – 002

November 6, 2017

Pre-Lab Report

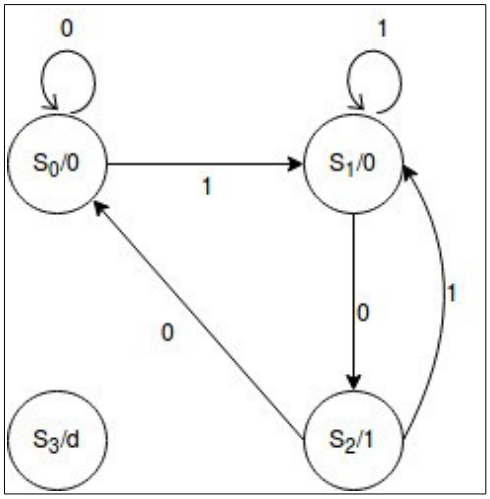
Lab 7: Sequence Detector Implementation using a Breadboard

Requirements:

- Design a synchronous sequential circuit to recognize a sequence of at least a single logic 1 followed by a logic 0
- Sequential Circuit must be a Moore machine and can use three states: S0, S1, and S2
- Two D-FFs will be utilized and an S3 state is optional for utilizing "don't cares"
- Input I can be implement via a switch whilst output Z can be implement by an LED
- It is recommend to utilize a push-button rather than a clock generator for a 1 Hz clock signal
- Implement an active-low preset option and an active-low clear option. Make sure to set them both to logic 1 after setting clear to 0 for state initialization

Design

- State Diagram



- State Table

Present State	Input (I)	Next State	Output (Z)
S_0	0 1	S_0 S_1	0 0
S_1	0 1	S_2 S_1	0 0
S_2	0 1	S_0 S_1	1 1
S_3	0 1	D D	D D

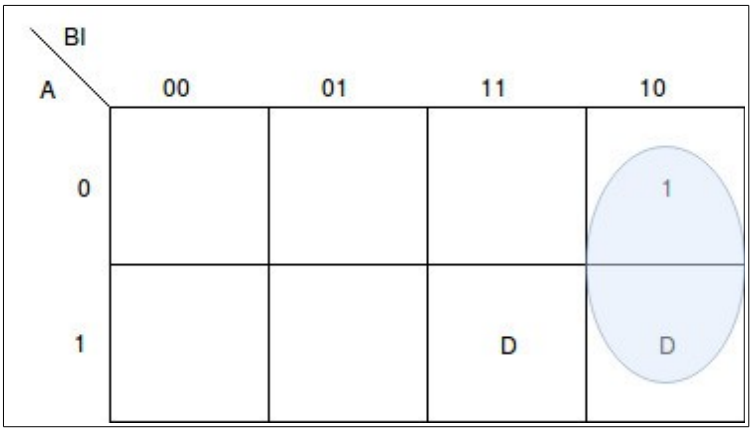
- State Assignment

Present State A B	Input (I)	Next State A B	Output (Z)
0 0	0 1	0 0 0 1	0 0
0 1	0 1	1 0 0 1	0 0
1 0	0 1	0 0 0 1	1 1
1 1	0 1	D D D D	D D

- FF-type selection
  - Two D flip-flops to represent three states with outputs A and B. There is a single input, "I", and a single output, "Z":
    - $A(t + 1) = D_A(A, B, I) = \Sigma(2)$
    - $B(t + 1) = D_B(A, B, I) = \Sigma(1, 3, 5)$
    - $Z(A, B, I) = \Sigma(4, 5)$

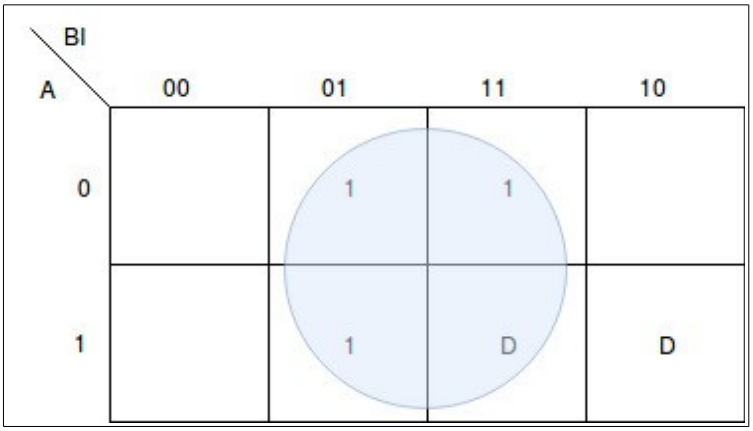
- Simplified FF input equations and output equations

- D_A:



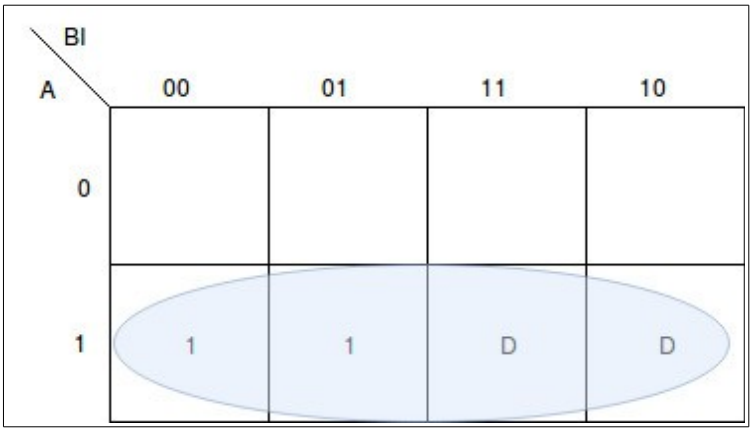
$D_A = BI'$

- D_B:



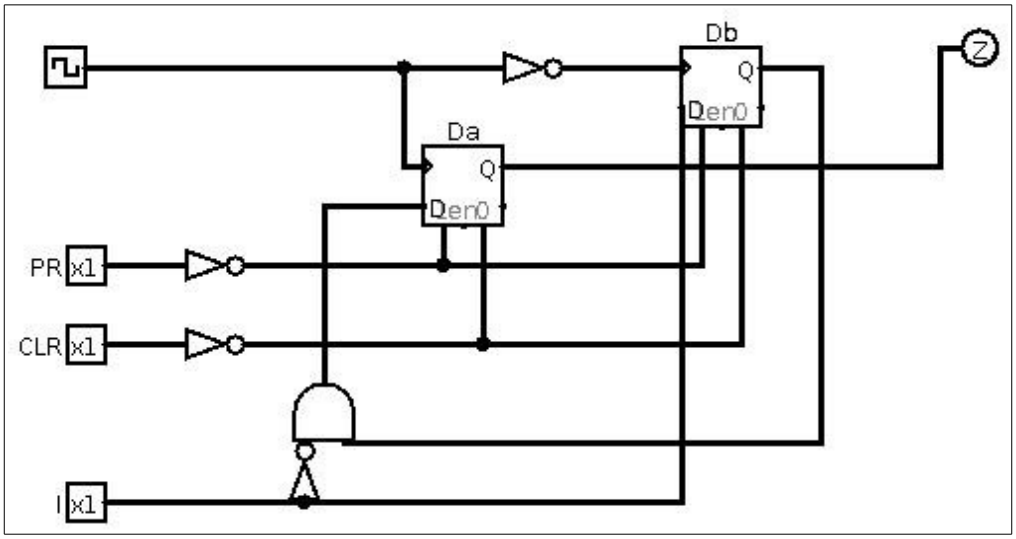
$D_B = I$

- Z:



Z = A

- Logic Diagram



Testing

- Test Patterns:

Input I	CLR	PR	Output Z
00000	1	1	00000
11111	1	1	00000
01100101001	1	1	00010010100
11101	0	1	00000

01001	1	0	11111
10110	0	0	00000

- Simulation:
  - Verilog D_FF:

```
module D_FF(Q, D, CLK, Clr);
        output Q;
        input D, CLK, Clr;
        reg Q;

   always @(posedge CLK, negedge Clr)
      if (~Clr) Q <= 1'b0; else Q <= D;
endmodule
```

  - Verilog Sequential Circuit:

```
module Moore_sequential(output Z, A, B,
        input I, clock, reset
);
        wire Da, Db;

        assign Da = B&&(~I);
        assign Db = I;
        assign Z = A;

        D_FF m1(A, Da, clock, reset);
        D_FF m2(B, Db, clock, reset);
endmodule
```

  - Verilog Testbench:

```
module Sim;
	// Inputs
	reg I;
	reg clock;
	reg reset;

	// Outputs
	wire Z;
	wire A;
	wire B;

	// Instantiate the Unit Under Test (UUT)
	Moore_sequential uut (
	.Z(Z),
	.A(A),
	.B(B),
	.I(I),
	.clock(clock),
	.reset(reset)
	);

	// Initialize Inputs
	initial begin
	  I = 0;
	  clock = 0;
	  reset = 1;
	  #2 reset = 0;
		#2 reset = 1;

	repeat (22)
		#5 clock = ~ clock; //used to generate 15 clock cycles
	end

	// Add stimulus here
	initial begin
		#10 I = 1; //Change the input patterns to fully simulate your circuit
		#20 I = 0;
		#20 I = 1;
		#10 I = 0;
		#10 I = 1;
		#10 I = 0;
		#20 I = 1;
		#10 I = 0;
	end
endmodule
```

Design Time: 5 hr