EEE 102: Introduction to Digital Circuit Design
Recitation 6 Solutions

Fall 2023

Problem 1: A sequential circuit with two D flip-flops A and B, two inputs X and Y, and one output Z is specified by the following input equations:

$$D_A = \bar{X}A + XY \qquad D_B = \bar{X}A + XB \qquad Z = XB$$

(a) Draw the logic diagram (truth table) of the circuit.

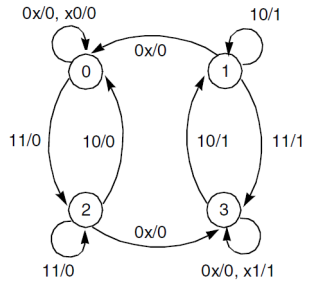
A	B	X	Y	D_A	D_B	Z
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	1	1
0	1	1	1	1	1	1
1	0	0	0	1	1	0
1	0	0	1	1	1	0
1	0	1	0	0	0	0
1	0	1	1	1	0	0
1	1	0	0	1	1	0
1	1	0	1	1	1	0
1	1	1	0	0	1	1
1	1	1	1	1	1	1

(b) Derive the state table.

State	A	B
S0	0	0
S1	0	1
S2	1	0
S3	1	1

Present State	Next State				Output Z			
	XY=00	XY=01	XY=10	XY=11	XY=00	XY=01	XY=10	XY=11
S0	S0	S0	S0	S2	0	0	0	0
S1	S0	S0	S1	S3	0	0	1	1
S2	S3	S3	S0	S2	0	0	0	0
S3	S3	S3	S1	S3	0	0	1	1

(c) Derive the state diagram.



Format: XY/Z (x = unspecified)

Problem 2: Design a sequential circuit with two D flip-flops A and B and one input X. When $X = 0$, the state of the circuit remains the same. When $X = 1$, the circuit goes through the state transitions from 00 to 10 to 11 to 01, back to 00, and then repeats.

Present State		Input	Next State	
A	B	X	A*	B*
0	0	0	0	0
0	0	1	1	0
0	1	0	0	1
0	1	1	0	0
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	1

D_A:

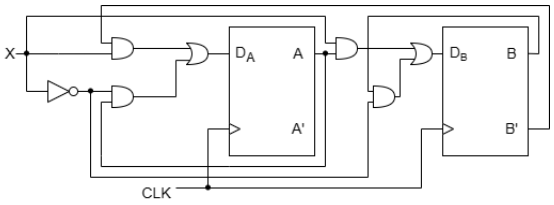
X\AB	00	01	11	10
0	0	0	1	1
1	1	0	0	1

$D_A = \overline{X}A + X\overline{B}$

D_B:

X\AB	00	01	11	10
0	0	1	1	0
1	0	0	1	1

$D_B = \overline{X}B + XA$



Problem 3: The state table for a twisted ring counter is given below. The circuit has no inputs, and its outputs are uncomplemented outputs of the flip-flops.

(a) Design the circuit using D flip-flops. Assume that the unspecified next states are don't care conditions.

(b) Add the necessary logic to the circuit to initialize it to state 000 on power-up master reset.

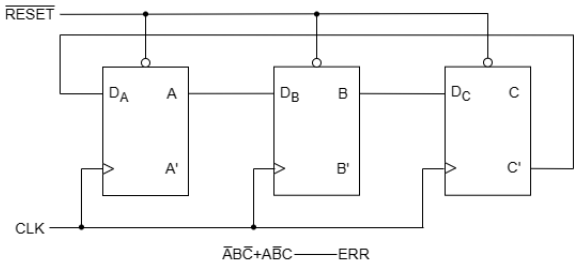
(c) For the circuit you designed, how would you deal with a situation in which it accidentally enters an unused state if it is implemented in (i) a child's toy, or (ii) a commercial airliner?

Present State	Next State
ABC	ABC
000	100
100	110
110	111
111	011
011	001
001	000

Present State			Next State		
A	B	C	A	B	C
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	x = 1	x = 0	x = 1
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	x = 0	x = 1	x = 0
1	1	0	1	1	1
1	1	1	0	1	1

(a)
$D_A = \overline{C}$
$D_B = A$
$D_C = B$

(b)
CLEAR A = $\overline{\text{RESET}}$
CLEAR B = $\overline{\text{RESET}}$
CLEAR C = $\overline{\text{RESET}}$

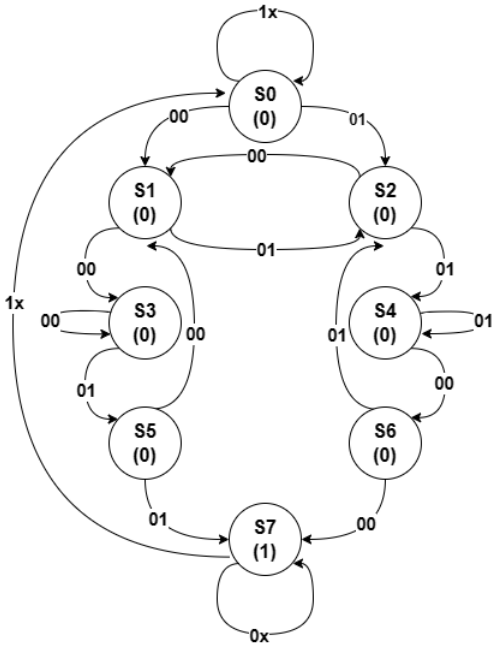


(c) It is possible that outside interference or malfunction will cause the circuit to enter one of the unused states, 010 or 101. Thus, sometimes it is desirable to specify, fully or at least partially, the next state values or output values for the unused states. Depending on the application of the circuit, a number of ideas may be applied. First, the outputs for the unused states may be specified so that any actions that result from entry and transaction in between the unused states are not harmful. Second, an additional output which indicates that the circuit has entered an incorrect state can be provided. Third, to ensure that a return to normal operation is possible without resetting the entire system, the next state behavior for the unused states may be specified [1].

In this problem, the circuit is suitable for a child's toy, where the implementation is the cheapest, but not for life critical applications. If an error occurs, the child just needs to reset it. In the life critical applications, it is critical to detect the errors immediately. Once the above circuit enters an unused state, 010 or 101, it will be stuck in those two states because of the way the corresponding next states are defined. For a life critical application, additional circuitry is needed for the immediate detection of error (ERR = $\overline{A}B\overline{C} + A\overline{B}C$).

[1] M. M. Mano and C. R. Kime, Logic and Computer Design Fundamentals, Third Edition, Pearson Prentice Hall, 2004, pp. 277–279

Problem 4: Draw a state diagram for a clocked synchronous state machine with two inputs, $INIT$ and X, and one Moore-type output Z. As long as $INIT$ is asserted, Z is continuously 0. Once $INIT$ is negated, Z should remain 0 until X has been 0 for two successive ticks and 1 for two successive ticks, regardless of the order of occurrence. Then Z should go to 1 and remain 1 until $INIT$ is asserted again.
Hint: No more than ten states are required.



Output Z is 1 when the sequence 0011 or 1100 is detected.

Input format: $INIT\ X$
At all states, $INIT = 1$, i.e., the input 1x, takes the FSM to the initial state $S0$.

Problem 5:

(i) A ring counter is a shift register with serial output connected to the serial input.

(a) Starting from an initial state of 1000, list the sequence of states of the four flip-flops after each shift.

1000, 0100, 0010, 0001, 1000

(b) Beginning in state 10...0, how many states are there in the count sequence of an n-bit ring counter? How many unused states are there?

Number of states: n, Unused states: $2^n - n$

(ii) A switch-tail ring counter (Johnson counter) uses the complement of the serial output of a right shift register as its serial input.

(a) Starting from an initial state of 0000, list the sequence of states of the four flip-flops after each shift until register turns to 0000.

0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001, 0000

(b) Beginning in state 00...0, how many states are there in the count sequence of an n-bit switch-tail counter? How many unused states are there?

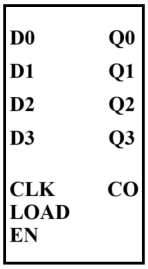
Number of states: $2n$, Unused states: $2^n - 2n$

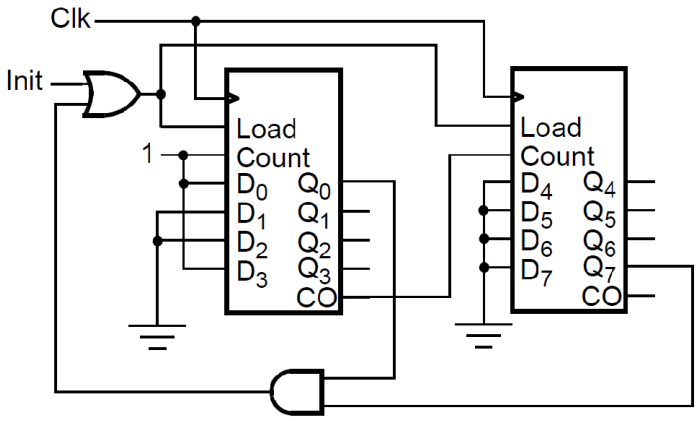
(iii) How many flip-flop values are complemented in an 8-bit binary ripple counter to reach the next count value after,

(a) 11101111: 11110000 $\rightarrow$ 5 bits are complemented

(b) 01111111: 10000000 $\rightarrow$ 8 bits are complemented

Problem 6: Using two binary counters of the type shown in figure and logic gates, construct a binary counter that counts from decimal 9 through decimal 129. Add an additional input to the counter that initializes it synchronously to 9 when the signal $INIT$ is 1.



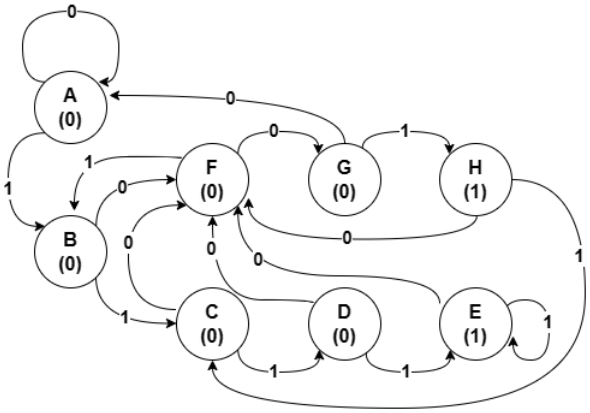


Problem 7:

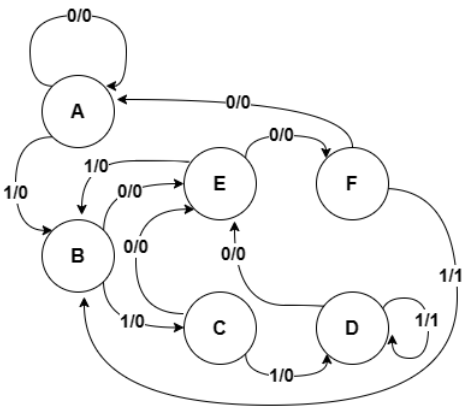
(a) Derive the state diagram for an FSM that has an input W and output Z. The machine has to generate $Z = 1$ when the previous four values of W were 1001 or 1111; otherwise, $Z = 0$. Overlapping input patterns are allowed. An example of the desired behavior is:

W : 010111100110011111
Z : 000000100100010011

Moore:



Mealy:



(b) Write the VHDL code for the FSM you derived.

```
-- VHDL code for Mealy type implementation:
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY prob7 IS
    PORT(Clock  IN:  STD_LOGIC;
         Resetn IN:  STD_LOGIC;
         W      IN:  STD_LOGIC;
         Z      OUT: STD_LOGIC);
END prob7;

ARCHITECTURE Behavior OF prob7 IS
    TYPE State_type IS (A, B, C, D, E, F);
    SIGNAL Y: State_type;
BEGIN
    PROCESS(Resetn, Clock)
    BEGIN
        IF Resetn = '0' THEN
            Y <= A;
        ELSIF rising_edge(Clock):
            CASE Y IS
                WHEN A =>
                    IF W = '0' THEN Y <= A;
                    ELSE Y <= B;
                    END IF;
                WHEN B =>
                    IF W = '0' THEN Y <= E;
                    ELSE Y <= C;
                    END IF;
                WHEN C =>
                    IF W = '0' THEN Y <= E;
                    ELSE Y <= D;
                    END IF;
                WHEN D =>
                    IF W = '0' THEN Y <= E;
                    ELSE Y <= D;
                    END IF;
                WHEN E =>
                    IF W = '0' THEN Y <= F;
                    ELSE Y <= B;
                    END IF;
                WHEN F =>
                    IF W = '0' THEN Y <= A;
                    ELSE Y <= B;
                    END IF;
            END CASE;
        END IF;
    END PROCESS;
    PROCESS(Y, W)
    BEGIN
        IF (Y = D AND W = '1') OR (Y = F AND W = '1') THEN
            Z <= '1';
        ELSE
            Z <= '0';
        END IF;
    END PROCESS;
END Behavior;
```