Exam - Computer Architecture Unit I [08/02/2023] (A)

Surname: _____ Name: _____

Student ID Number (Matricola): _____

DSA Students should solve only the first 4 exercises (grade will be scaled accordingly)

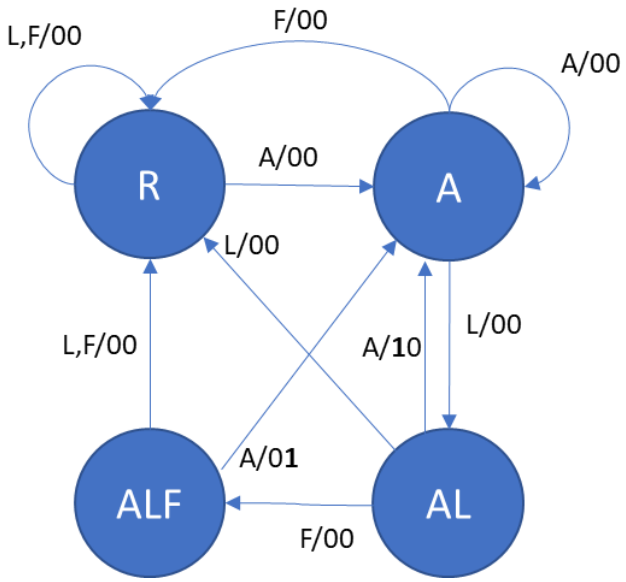
Exercise 1 (8 points)

Design a sequential circuit with two inputs x1,x0 that encode the characters A, L, F as follows:

x1, x0	character
00	A
01	L
1-	F

The circuit has 2 outputs z1 and z0. The state machine outputs z1=1 when it receives on the input the sequence ALA, and outputs z0=1 when it receives on the input the sequence ALFA. Overlaps are allowed. Draw the circuit.

Solution:



State encoding:

R	**00**
A	**01**
AL	**10**
ALF	**11**

Next state and output table:

CS	S_1	S_0	x_1	x_0	NS	S_1'	S_0'	z1	z0
R	0	0	0	0	A	0	1	0	0
R	0	0	0	1	R	0	0	0	0
R	0	0	1	-	R	0	0	0	0
A	0	1	0	0	A	0	1	0	0
A	0	1	0	1	AL	1	0	0	0

A	0	1	1	-	R	0	0	0	0
AL	1	0	0	0	A	0	1	**1**	0
AL	1	0	0	1	R	0	0	0	0
AL	1	0	1	-	ALF	1	1	0	0
ALF	1	1	0	0	A	0	1	0	**1**
ALF	1	1	0	1	R	0	0	0	0
ALF	1	1	1	-	R	0	0	0	0

Derive next state and output equations, and then draw the circuit (with gates, ROM, PLA, or whatever you prefer)

Exercise 2 (6 points) A combinational circuit has a 4 bit input $A=a_3a_2a_1a_0$ representing a two's complement number, and outputs $Z=z_1z_0$ as an unsigned binary number, such that:

Z=0 if $3 \leq A \leq 7$
Z=1 if $-3 \leq A < 3$
Z=2 if $-7 \leq A < -3$
Z=3 if $A=-8$

Write down:

- The corresponding truth table
- The minimal POS form for z_1
- The all-NAND and all-NOR form for z_1 (NOT gates are allowed, i.e., you don't need to translate NOT with NAND/NOR)
- z_1 using only three 2:1 multiplexers

Solution:

Truth Table:

a_3	a_2	a_1	a_0	z1	z0
0	0	0	0	0	1
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	1	1
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	1	0
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	0	1

Minimal POS form for z_1:

a1 a0 \ a3 a2	00	01	11	10
00	0	0	1	1
01	0	0	0	1
11	0	0	0	1
10	0	0	0	1

z1=(~a2+~a0)*(~a2+~a1)*(a3)

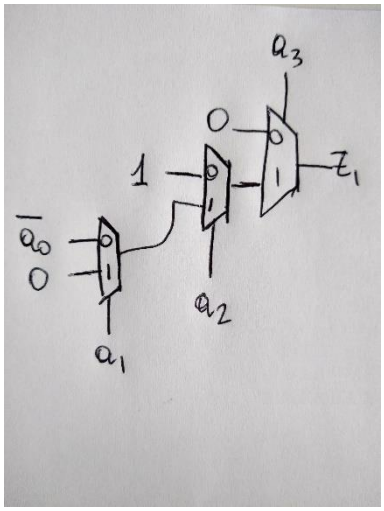
all-NAND form for z_1:

z1 = ~(~(~(a2*a0)*~(a2*a1)*(a3)))

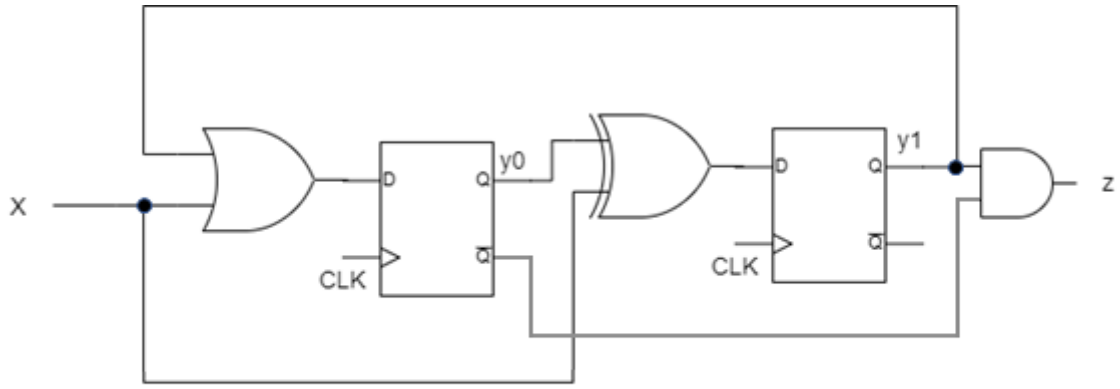
all-NOR form for z_1:

z1 = ~(~(~a2+~a0)+~(~a2+~a1)+(~a3))

z1 using MUX:



Exercise 3 (5 points)
Analyze the state machine shown in the figure. Write down next state and output tables and draw the state transition diagram.



Solution:

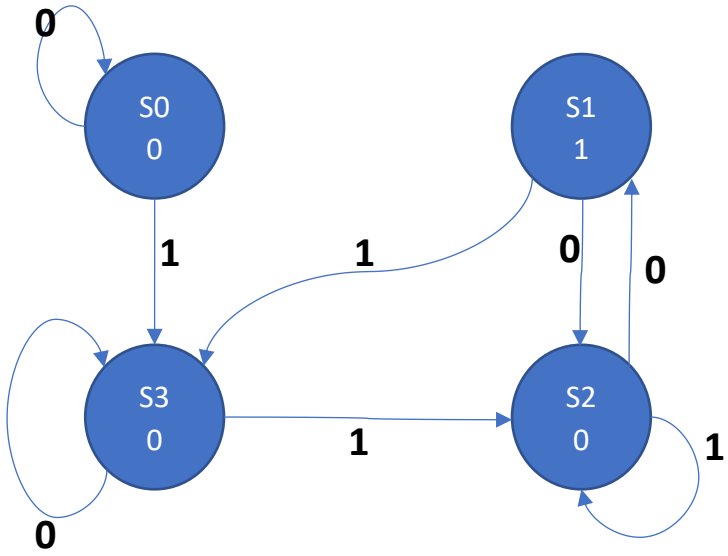
It is a Moore FSM (output does not depend on the input).

Outputs table + state encoding

y0	y1	z	Stato
0	0	0	S0
0	1	1	S1
1	0	0	S2
1	1	0	S3

Next state table

y0	y1	CS	x	y0'	y1'	NS
0	0	S0	0	0	0	S0
0	0	S0	1	1	1	S3
0	1	S1	0	1	0	S2
0	1	S1	1	1	1	S3
1	0	S2	0	0	1	S1
1	0	S2	1	1	0	S2
1	1	S3	0	1	1	S3
1	1	S3	1	1	0	S2

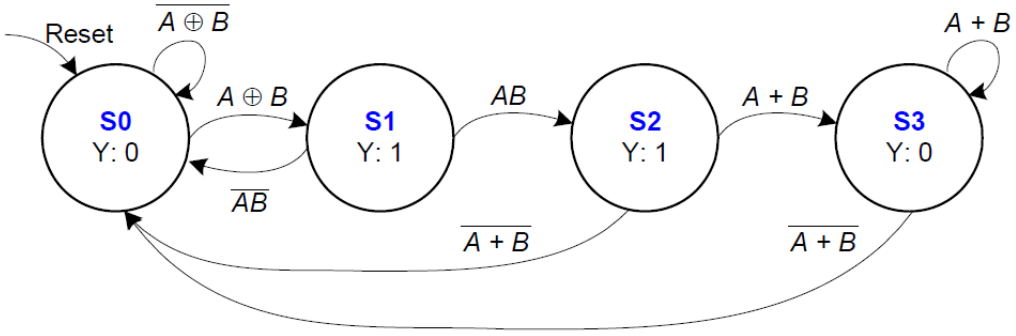


Exercise 4 (3 points)

Draw the state transition diagram described by the following SystemVerilog code:

```
module fsm2(input logic clk, reset,
            input logic a, b,
            output logic y);
   logic [1:0] state, nextstate;
   parameter S0 = 2'b00;
   parameter S1 = 2'b01;
   parameter S2 = 2'b10;
   parameter S3 = 2'b11;
   always_ff @(posedge clk, posedge reset)
     if (reset) state <= S0;
     else state <= nextstate;
   always_comb
     case (state)
       S0: if (a ^ b) nextstate = S1;
           else nextstate = S0;
       S1: if (a & b) nextstate = S2;
           else nextstate = S0;
       S2: if (a | b) nextstate = S3;
           else nextstate = S0;
       S3: if (a | b) nextstate = S3;
           else nextstate = S0;
     endcase
   assign y = (state== S1) | (state== S2);
endmodule
```

Solution:



Exercise 5 (4 points)

Given X = -1614 and Y = 675 represented in base 10:

- Convert them to two's complement numbers using the minimum number of bits necessary to represent both values
- Compute X+Y and X-Y
- Convert the results to base 10 number and check their correctness
- Convert X and Y to IEEE 754 half-precision representation

Solution:

Base 2 conversion:

1614 = 011001001110 => -1614 = 100110110010
675 = 001010100011

Sum, subtraction, and base 10 conversion:

X+Y = 100110110010 +
001010100011 =

110001010101 = -939 = -1614+675

-675 = 110101011101

X-Y = 100110110010 +
110101011101 =

011100001111 = **Overflow!** -1614 – 675 = -2289 (11 bit are not enough, they can only represent up to -2048)

Half-precision conversion

1614 = 011001001110.0 = $1.1001001110 * 2^{10}$
X = -1614:
Sign = 1
Exponent = 10_{10}
Exponent + bias = $10_{10}+15_{10} = 25_{10} = 11001_2$
Fraction = 1001001110_2
IEEE: 1 11001 1001001110

Y = 675 = 001010100011 = $1.010100011 * 2^9$
Sign = 0
Exponent = 9
Exponent + bias = 9+15 = 24 = 11000
Fraction = 010100011
IEEE = 0 11000 0101000110

Exercise 6 (4 points)

Given the expression $f = \left(\overline{a(a + \overline{bcd})} + \bar{e}\right) \oplus (e + cd)$ simplify it and bring to minimal POS form.

Solution:

$$f = \left(\overline{a(a + \overline{bcd})} + \bar{e}\right) \oplus (e + cd)$$

$$= \left(\overline{a + a\overline{bcd}}\right) + \bar{e}\Big) \oplus (e + cd)$$
$$= (\bar{a} + \bar{e}) \oplus (e + cd)$$

$$= \overline{(\bar{a} + \bar{e})}(e + cd) + (\bar{a} + \bar{e})\overline{(e + cd)} =$$
$$= ae(e + cd) + (\bar{a} + \bar{e})\bar{e} \cdot \overline{cd} = ae + \bar{e} \cdot \left(\bar{c} + \bar{d}\right) = ae + \bar{c}\bar{e} + \bar{d}\bar{e} =$$
$$\left(a + \bar{c} + \bar{d}\right)(a + \bar{c} + \bar{e})\left(a + \bar{d} + \bar{e}\right)(a + \bar{e})\left(e + \bar{c} + \bar{d}\right) = (a + \bar{e})\left(a + \bar{c} + \bar{d}\right)\left(\bar{c} + \bar{d} + e\right) =$$
$$(a + \bar{e})\left(\bar{c} + \bar{d} + e\right)$$