Exam - Computer Architecture Unit I [16/01/2024] (A)

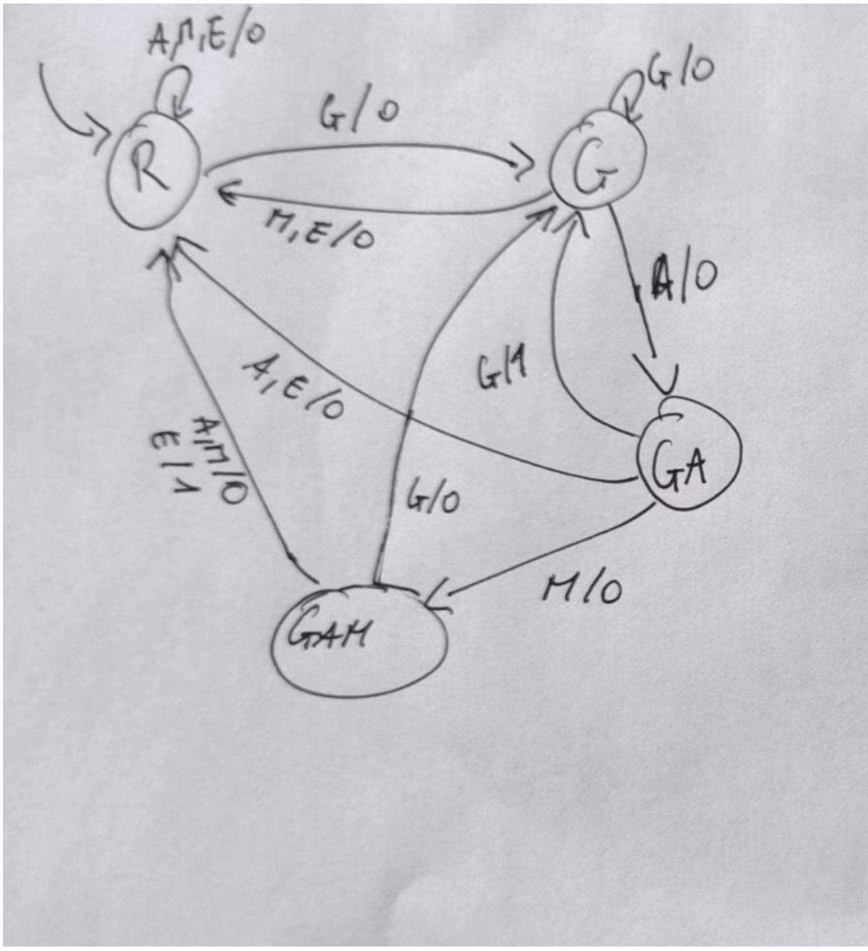
Surname: ______________________________ **Name:** __________________________

Student ID Number (Matricola): ____

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

Exercise 1 (7 points) Design a sequential circuit that receives on input a sequence of characters from the alphabet {A, E, G, M}, and produces an output equal to 1 every time that the sequence GAG or GAME is detected. Consider overlaps. Design and draw the finite state machine. Then derive the minimal SOP and POS forms for the combinational part.

Solution:



INPUTS

	X_1	X_0
A	0	0
E	0	1
G	1	0
M	1	1

STATE ENCODING

	S_1	S_0
R	0	0
G	0	1
GA	1	0
GAM	1	1

matricola_____________

CS	S_1	S_0	X_1	X_0	NS	S_1'	S_0'	Z
R	0	0	0	0	R	0	0	0
R	0	0	0	1	R	0	0	0
R	0	0	1	0	G	0	1	0
R	0	0	1	1	R	0	0	0
G	0	1	0	0	GA	1	0	0
G	0	1	0	1	R	0	0	0
G	0	1	1	0	G	0	1	0
G	0	1	1	1	R	0	0	0
GA	1	0	0	0	R	0	0	0
GA	1	0	0	1	R	0	0	0
GA	1	0	1	0	G	0	1	1
GA	1	0	1	1	GAM	1	1	0
GAM	1	1	0	0	R	0	0	0
GAM	1	1	0	1	R	0	0	1
GAM	1	1	1	0	G	0	1	0
GAM	1	1	1	1	R	0	0	0

S_1'

$x_1x_0 \backslash S_1S_0$	00	01	11	10
00	0	1	0	0
01	0	0	0	0
11	0	0	0	1
10	0	0	0	0

$SOP = \bar{x}_1\bar{x}_0\bar{S}_1S_0 + x_1x_0S_1\bar{S}_0$

$POS = (S_1+S_0)(\bar{x}_1+x_0)(\bar{S}_1+x_1)(\bar{S}_0+x_0)$

S_0'

$x_1x_0 \backslash S_1S_0$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	1
10	1	1	1	1

$SOP = (S_1\bar{S}_0x_1) + (x_1\bar{x}_0)$

$POS = (x_1)(S_1+\bar{x}_0)(\bar{S}_0+\bar{x}_0)$

Z

$x_1x_0 \backslash S_1S_0$	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	0	0
10	0	0	0	1

$SOP = (S_1S_0\bar{x}_1x_0) + (S_1\bar{S}_0x_1\bar{x}_0$

$POS = (S_1)(x_1+x_0)$
$(S_0+\bar{x}_0)(\bar{S}_0+\bar{x}_1)$

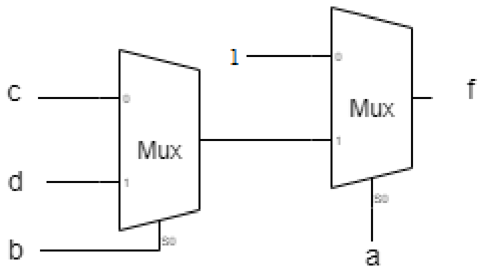
matricola____________

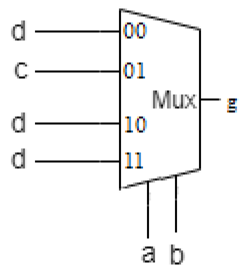
Exercise 2 (4 points) The function $f(a,b,c,d)$ is equal to 0 if $a\bar{b}\bar{c} = 1$ or $ab\bar{d} = 1$. Otherwise, it is equal to 1. The function $g(a,b,c,d)$ is equal to 1 if $a + \bar{b} + \bar{c} = 0$ or $cd = 1$, it is not specified if $c + \bar{d} = 0$, and it is equal to 0 in all the other cases.

Write down the truth table and design the circuits that implements the function *f* using 2:1 multiplexers, and the function *g* using a single 4:1 multiplexer.

Solution:

a	b	c	d	*f*	*g*
0	0	0	0	1	0
0	0	0	1	1	X
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	1	0
0	1	0	1	1	X
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	0	X
1	0	1	0	1	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	1	1	X
1	1	1	0	0	0
1	1	1	1	1	1





matricola_____________

Exercise 3 (4 points)

- Convert the base 10 numbers X=103 and Y=-68 to 8-bits two's complement number.
- Compute Z=X-Y, W=X+Y (specify if Z and W overflow or not), and convert the results to base 16 numbers.
- Compute R=S+T where S=1AB and T=2B7 (in base 16). Convert the result to base 10 and check the correctness of the result by converting to base 10 the original operands.

Solution:

X = 103_{10} = 01100111_2 68_{10} = 01000100 => Y = -68_{10} = 10111100

```
Z = X-Y
----------
01100111 +
01000100 =
---------------
```

10101011_2 = 0xAB

This is an overflow (sum of two positive numbers gives us a negative number)

```
W = X+Y
------------
 01100111 +
 10111100 =
----------------
100100011 = 0x23
```

This is not an overflow (sum of two numbers with opposite sign never overflows)

```
S+T =
1AB +
2B7 =
------
462
```

0x1AB = 427

0x2B7 = 695

427 + 695 = 1122 = 0x462

matricola_____________

Exercise 4 (5 points)

Consider the expression $f = (a \oplus b)(a \oplus c) + bc$. Simplify it to minimal POS and SOP forms using Boolean algebra theorems and axioms. Then, write down the all-NAND and all-NOR forms.

Solution:

$$f = (\bar{a}b + a\bar{b})(\bar{a}c + a\bar{c}) + bc = \bar{a}bc + a\bar{b}\bar{c} + bc = a\bar{b}\bar{c} + bc$$

POS:

$$f = (a + b)(a + c)(\bar{b} + c)(b + \bar{c})$$

matricola____________

Exercise 5 (6 points)

Describe in SystemVerilog the finite state machine corresponding to the following state transition and output tables.

Current State	Input	Next State
A	0	B
A	1	C
B	0	A
B	1	C
C	0	D
C	1	E
D	0	D
D	1	A
E	0	E
E	1	B

Current State	Output
A	0
B	0
C	1
D	1
E	0

Solution:

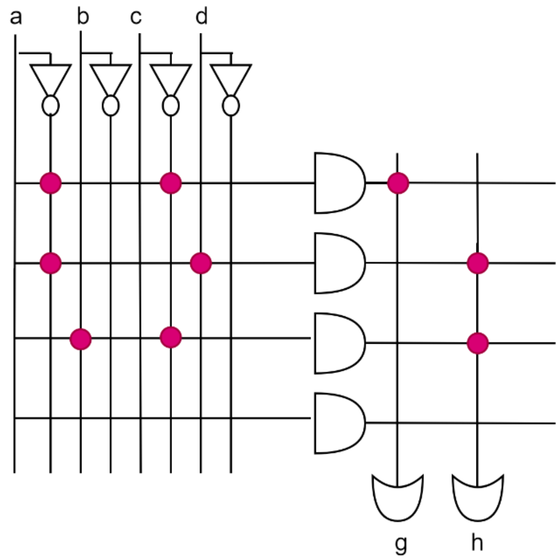
```
module ex5(input logic clk,
           input logic reset,
           input logic x,
           output logic y);
    typedef enum logic [2:0] {A, B, C, D, E} statetype;
    statetype state, nextstate;

    // state register
    always_ff @(posedge clk, posedge reset)
        if (reset) state <= A;
        else state <= nextstate;

    // next state logic
    always_comb
        case (state)
            A: if(x) nextstate = C;
               else nextstate = B;
            B: if(x) nextstate = C;
               else nextstate = A;
            C: if(x) nextstate = E;
               else nextstate = D;
            D: if(x) nextstate = A;
               else nextstate = D;
            E: if(x) nextstate = B;
               else nextstate = E;
            default: nextstate = A;
        endcase

    // output logic
    assign y = (state == C | state == D);
endmodule
```

matricola____________

Exercise 6 (4 points)

Consider the following PLA and write down:

- The expression for functions *g* and *h*
- Transform the expression *f=g* ⊕ *h*, using Boolean algebra axioms and theorems, in canonical SOP form



Solution:

$$g = \bar{a}\bar{c}$$
$$h = \bar{a}d + b\bar{c}$$
$$f = \bar{a}\bar{c} \oplus (\bar{a}d + b\bar{c}) = ab\bar{c} + \bar{a}cd + \bar{a}\bar{b}\bar{c}\bar{d}$$
$$f = \bar{a}\bar{c} \oplus (\bar{a}d + b\bar{c}) = ab\bar{c}d + ab\bar{c}\bar{d} + \bar{a}bcd + \bar{a}\bar{b}cd + \bar{a}\bar{b}\bar{c}\bar{d}$$ **(canonical)**