

EECS 151/251A Homework 3

Due Friday, September 23rd, 2022 11:59PM

Problem 1: Simplifying with Karnaugh Maps

Use the following truth table to answer the questions.

A	B	C	D	output
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

(a) Write the sum of products expression for the truth table.

(b) Use a Karnaugh map to write a simplified boolean expression for the truth table.

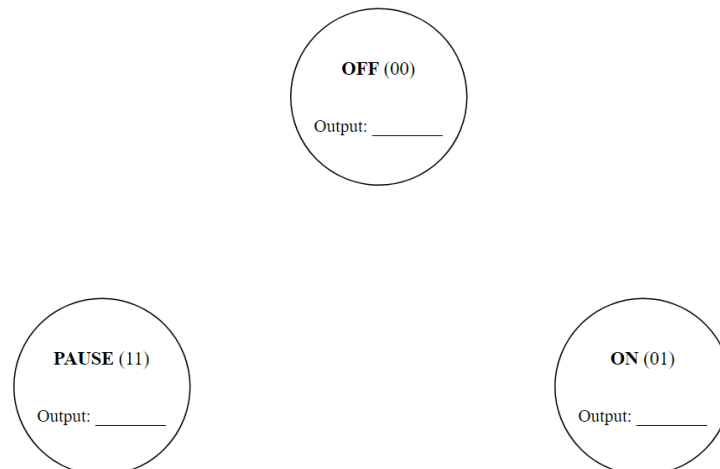
- (c) [251A Only] Using boolean algebra, prove that your Karnaugh map expression is equivalent to your sum of products expression.

Problem 2: Microwave FSM

Read the following microwave FSM specification and answer the questions.

The microwave has two buttons: a stop button and a start button. When the microwave is off, the lights are off and the food is not heated. Pressing the stop button when the microwave is off does nothing. Pressing the start button when the microwave is off turns the microwave on. When the microwave is on, the lights are on and the food is heated. Pressing the start button again when the microwave is already on does nothing. Pressing the stop button when the microwave is on pauses the microwave. The lights stay on when the microwave is paused, but the food is no longer heated. When the microwave is paused, pushing the start button turns the microwave back on, and pushing the stop button turns the microwave off.

- (a) Complete the Moore FSM diagram for the microwave FSM. Let bit 1 of the output represent the microwave lights and bit 0 of the output represent the microwave heat, and let pressing the start button give an input of 1 and pressing the stop button give an input of 0.

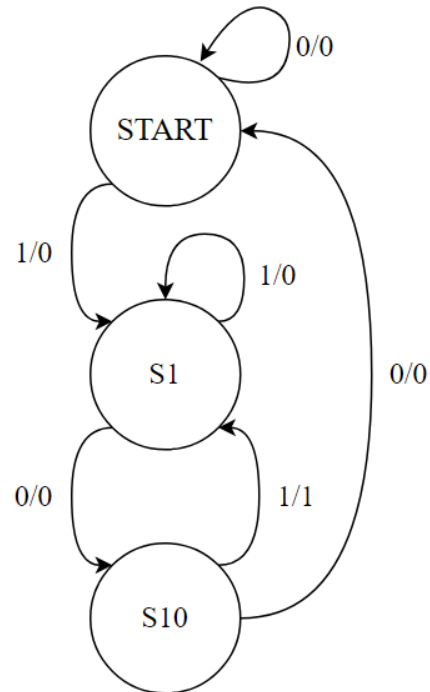


- (b) Write the truth table for this FSM.

- (c) Using Karnaugh maps, compute a boolean expression for each bit of the next state.

Problem 3: FSMs in Verilog

Use the following FSM diagram for this question.



(a) The verilog starter code for this FSM is shown below:

```

module fsm(
    input clk,
    input rst,
    input in,
    output out
);

parameter START = 2'b00,
           S1   = 2'b01,
           S10  = 2'b10;

reg [1:0] current_state;
reg [1:0] next_state;
reg out_state;

assign out = out_state;

always @(posedge clk) begin
    //Synchronous state update logic
    //TODO: Part (i) code goes here
end

```


(b) Describe the function of this FSM.

(c) Is this a Moore state machine or a Mealy state machine? Explain your reasoning.

(d) Redraw this FSM as the other type of state machine (ex. Moore to Mealy, Mealy to Moore).

- (e) [251A Only] Write a verilog module that implements the FSM in your diagram from part (d).