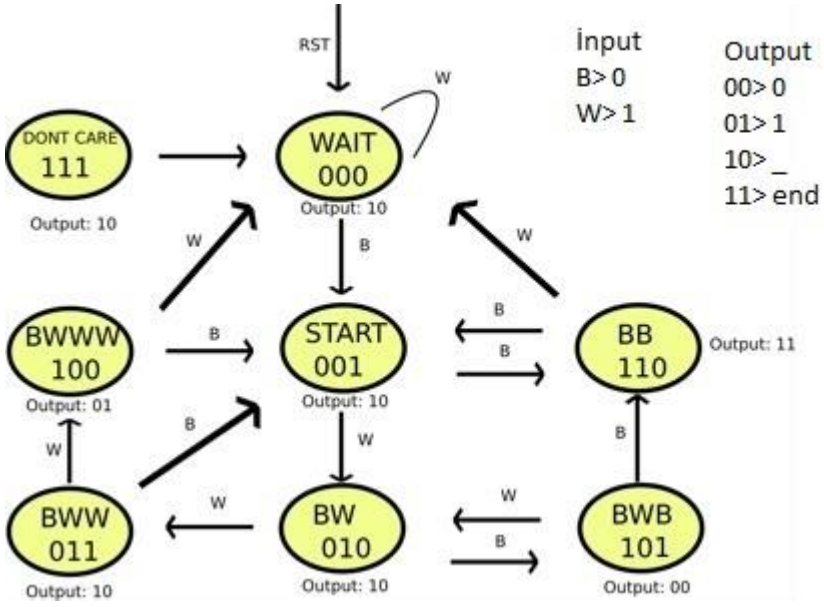
Student Names:
Student IDs:
Group ID:
Session:

ÖMER FARUK ÖZDEMİR SECTİON: FF12
MUSTAFA FATİH KURT GROUP ID: 6

CMPE 240 2018 Experiment 4 Preliminary Work

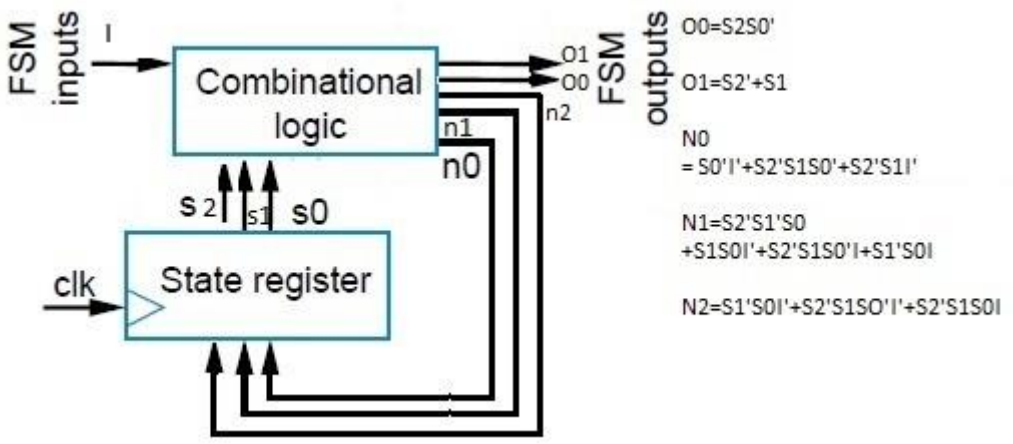
(For illustrations you can use any drawing tool that you want including Microsoft Word Shapes. Do not use scanned images of hand drawn state machines and architecture diagrams.)
(For tables please use insert table feature of Microsoft Word)

Step 1: Capture the FSM: Create and draw the finite state machine that describes the desired behavior of the controller.



Student Names:
Student IDs:
Group ID:
Session:

Step 2: Create the architecture: Create and draw standard architecture by a using state register of the appropriate width and combinational logic. Refer to book or lecture slides. Use the same convention.



Student Names:
Student IDs:
Group ID:
Session:

Step 3: Encode the states: Assign a unique binary number to each state. Each binary number representing a state is known as an encoding. Any encoding will do as long as each state has a unique encoding. (The content of the following table is an example. Rename the states according to the ones you specified in the first section. Also use any encoding you want.)

STATE NAME	ENCODING
WAIT	000
START	001
BW	010
BWW	011
BWWW	100
BWB	101
BB	110
DON'T CARE	111

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1		S2	S1	S0	I	N2	N1	N0	O1	O0					I			Output	O1	O0
2	0	0	0	0	0	0	0	1	1	0	-		WAIT	0,0,0	0	B		0	0	0
3	1	0	0	0	1	0	0	0	1	0	-		START	0,0,1	1	W		1	0	1
4	2	0	0	1	0	1	1	0	1	0	-		BW-	0,1,0				-	1	0
5	3	0	0	1	1	0	1	0	1	0	-		BWW	0,1,1				end	1	1
6	4	0	1	0	0	1	0	1	1	0	-		BWWW	1,0,0						
7	5	0	1	0	1	0	1	1	1	0	-		BWB	1,0,1						
8	6	0	1	1	0	0	0	1	1	0	-		BB	1,1,0						
9	7	0	1	1	1	1	0	0	1	0	-		DONTCARE	1,1,1						
10	8	1	0	0	0	0	0	1	0	1	1									
11	9	1	0	0	1	0	0	0	0	1	1		O0	S2S0'						
12	10	1	0	1	0	1	1	0	0	0	0		O1	S2'+S1						
13	11	1	0	1	1	0	1	0	0	0	0		N0	S0'I'+S2'S1S0'+S2'S1I'						
14	12	1	1	0	0	0	0	1	1	1	end		N1	S2'S1'S0+S1S0I'+S2'S1S0'I+S1'S0I						
15	13	1	1	0	1	0	0	0	1	1	end		N2	S1'S0I'+S2'S1SO'I'+S2'S1S0I						
16	14	1	1	1		0	0	0	1	0	-									
17	15	1	1	1		0	0	0	1	0	-									

Student Names:
Student IDs:
Group ID:
Session:

Step 4: Create the state table: Create a truth table for the combinational logic such that the logic will generate the correct FSM outputs and next state signals. Ordering the inputs with state bits first makes this truth table describe the state behavior, so the table is a state table. (Update the table according to the number of state variables that you have used.)

CURRENT STATE	INPUTS	NEXT STATE	OUTPUTS
WAIT	B	START	-
WAIT	W	WAIT	-
START	B	BB	-
START	W	BW	-
BW	B	BWB	-
BW	W	BWW	-
BWW	B	START	-
BWW	W	BWWW	-
BWWW	B	START	1
BWWW	W	WAIT	1
BWB	B	BB	0
BWB	W	WAIT	0
BB	B	START	End
BB	W	WAIT	End
DON'T CARE	B	WAIT	-
DON'T CARE	W	WAIT	-

Student Names:
Student IDs:
Group ID:
Session:

Step 5: Draw the combinational logic: Implement the combinatorial logic using any method (You do not need to draw the inside circuit of multiplexers or decoders if you are using any. You can show those as blocks).