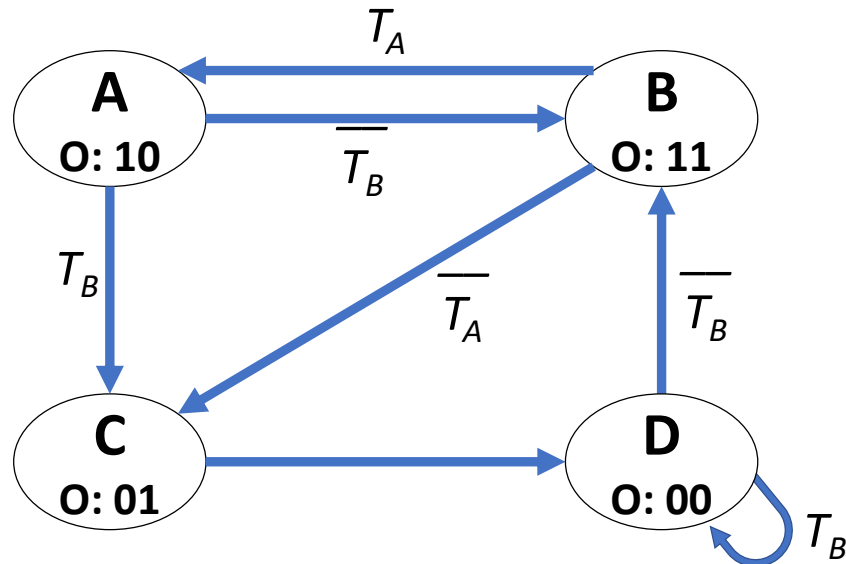


## 4 Finite State Machine [50 points]

You are given the following FSM with two one-bit input signals ( $T_A$  and  $T_B$ ) and one two-bit output signal ( $O$ ). You need to implement this FSM, but you are unsure about how you should encode the states. Answer the following questions to get a better sense of the FSM and how the three different types of state encoding we discussed in the lecture (i.e., one-hot, binary, output) will affect the implementation.



- (a) [3 points] There is one critical component of an FSM that is *missing* in this diagram. Please write what is missing in the answer box below.

The reset line or indication for initial state.

- (b) [2 points] Of the two FSM types, what type of an FSM is this?

Moore

(c) [5 points] List one major advantage of each type of state encoding below.

One-hot encoding	reduces next-state logic
------------------	--------------------------

Binary encoding	reduces FFs to hold state
-----------------	---------------------------

Output encoding	reduces the output logic
-----------------	--------------------------

(d) [10 points] Fully describe the FSM with equations given that the states are encoded with **one-hot** encoding. Assign state encodings such that numerical values of states increase monotonically for states A through D while using the **minimum** possible number of bits to represent the states with one-hot encoding. Indicate the values you assign to each state *and* simplify all equations:

State assignments: A: 0001, B: 0010, C: 0100, D: 1000

$$NS[3] = TB * TS[3] + TS[2]$$

$$NS[2] = TB * TS[0] + \overline{TA} * TS[1]$$

$$NS[1] = \overline{TB} * (TS[0] + TS[3])$$

$$NS[0] = TS[1] * TA$$

$$O[1] = TS[0] + TS[1]$$

$$O[0] = TS[1] + TS[2]$$

- (e) [10 points] Fully describe the FSM with equations given that the states are encoded with **binary** encoding. Assign state encodings such that numerical values of states increase monotonically for states A through D while using the **minimum** possible number of bits to represent the states with binary encoding. Indicate the values you assign to each state *and* simplify all equations:

State assignments: A: 00, B: 01, C: 10, D: 11

$$NS[1] = \overline{TS[1]} * (\overline{TS[0]} * TB + TS[0] \overline{TA}) + TS[1] * (\overline{TS[0]} + TS[0] * TB)$$

$$NS[0] = \overline{TS[1]} * \overline{TS[0]} * \overline{TB} + TS[1]$$

$$O[1] = TS[1]$$

$$O[0] = TS[1] \text{ XOR } TS[0]$$

- (f) [10 points] Fully describe the FSM with equations given that the states are encoded with **output** encoding. Use the **minimum** possible number of bits to represent the states with output encoding. Indicate the values you assign to each state *and* simplify all equations:

State assignments: A: 10, B: 11, C: 01, D: 00

$$NS[1] = TS[1] * \overline{TS[0]} * \overline{TB} + TS[1] * TS[0] * TA + \overline{TS[1]} * \overline{TS[0]} * \overline{TB}$$

$$= \overline{TS[0]} * \overline{TB} + TS[1] * TS[0] * TA$$

$$NS[0] = TS[1] * \overline{TS[0]} + TS[1] * TS[0] * \overline{TA} + \overline{TS[1]} * \overline{TS[0]} * \overline{TB}$$

$$= TS[1] * (\overline{TS[0]} + TS[0] * \overline{TA}) + \overline{TS[1]} * \overline{TS[0]} * \overline{TB}$$

$$O[1] = TS[1]$$

$$O[0] = TS[0]$$

(g) [10 points] Assume the following conditions:

- We can only implement our FSM with 2-input AND gates, 2-input OR gates, and D flip-flops.
- 2-input AND gates and 2-input OR gates occupy the *same* area.
- D flip-flops occupy 3x the area of 2-input AND gates.

Which state encoding do you choose to implement in order to **minimize the total area** of this FSM?

one-hot: 10 logics 4 FFs binary: 16 logics. 2 FFs output: 10 logics. 2 FFs  
Output encoding has the least amount of circuitry elements.