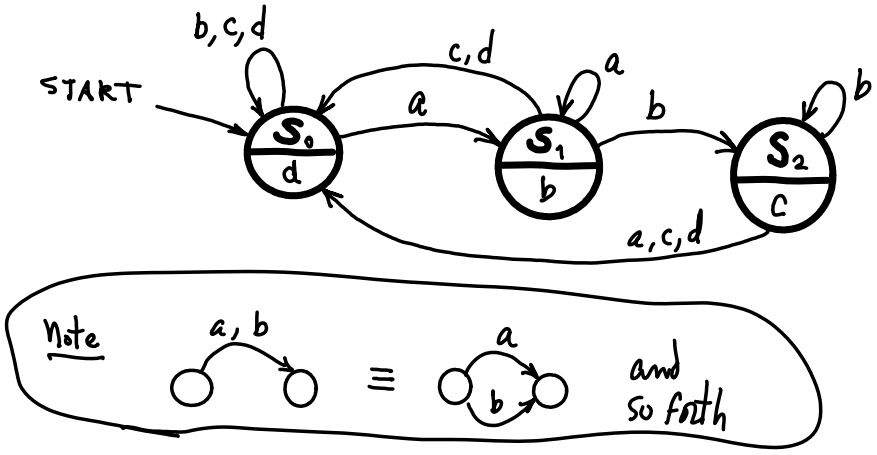
Implement a FSM for the Moore machine described by the state-transition diagram below.
Use 1-bit D flipflops and 2-input Boolean logic gates {AND, OR, NOT}.



a Moore machine: output is only dependent on state

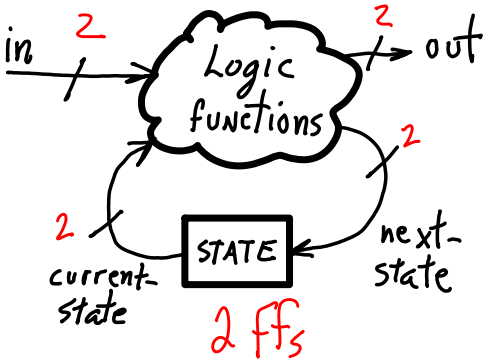
Preliminary steps:

Q. Define a Boolean encoding of the input symbol set {a, b, c, d}; that is, show how to implement a 4-symbol set using 2-bits. Also, define a binary encoding of the machine's states. Complete the tables at right.

symbol	code $b_1\ b_0$
a	0 0
b	0 1
c	1 0
d	1 1

state	code $Q_1 Q_0$
S_0	0 0
S_1	0 1
S_2	1 0

Q. Complete the general FSM implementation diagram at right. Specify the number of wires (bits) on each signal paths, "in", "out", "current-state", and "next-state". Specify the number of state elements (1-bit D FFs).



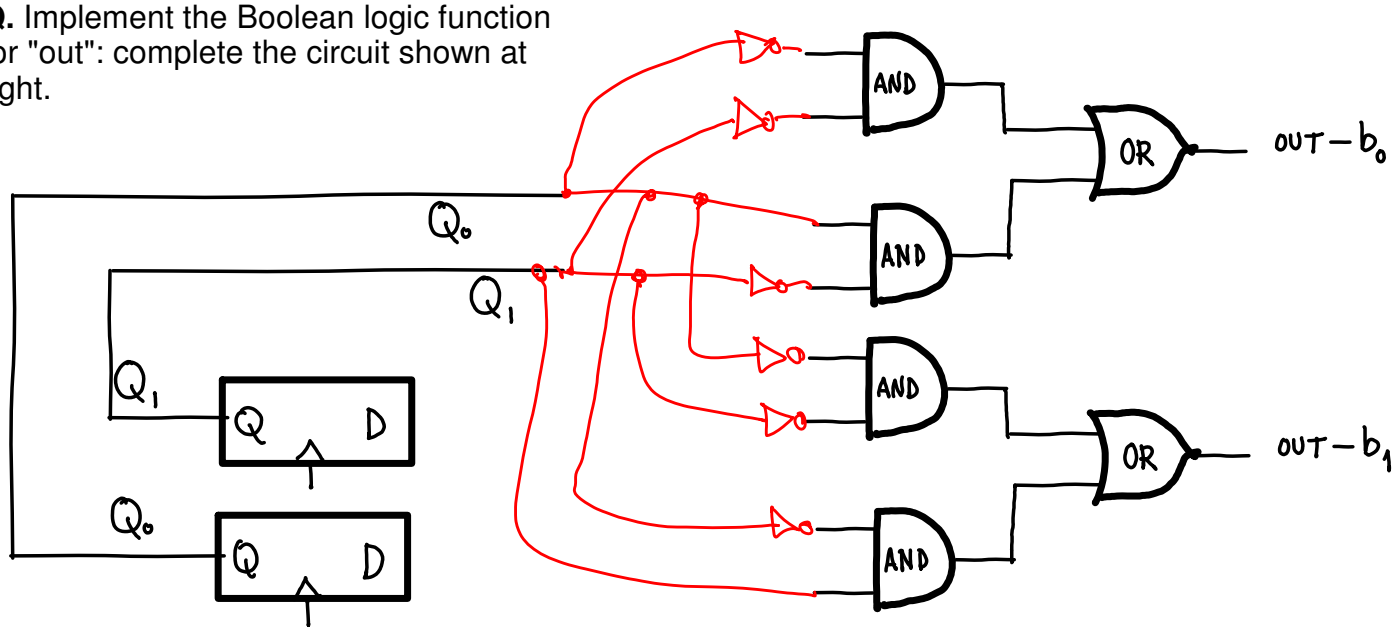
Q. Specify the next-state and output logic functions: complete the tables below.

in $b_1 b_0$	current-state $Q_1 Q_0$	next-state $Q_1^+ Q_0^+$
0 0	0 0	0 1 ← m_0
0 1	0 0	0 0
1 0	0 0	0 0
1 1	0 0	0 0
0 0	0 1	0 1 ← m_1
0 1	0 1	1 0
		m_5
X X	1 1	X X (Don't care)

in $b_1 b_0$	current-state $Q_1 Q_0$	next-state $Q_1^+ Q_0^+$
1 0	0 1	0 0
1 1	0 1	0 0
0 0	1 0	0 0
0 1	1 0	1 0 ← m_6
1 0	1 0	0 0
1 1	1 0	0 0
X X	1 1	X X

current-state $Q_1 Q_0$	out $b_1 b_0$	
S_0 0 0	1 1	$\overline{Q_1}\,\overline{Q_0}$
S_1 0 1	0 1	$\overline{Q_1} Q_0$
S_2 1 0	1 0	
X 1 1	X X	Don't care

Q. Implement the Boolean logic function for "out": complete the circuit shown at right.



Q. Implement the Boolean logic function for "next-state": complete the circuit shown below.