Date: **25/03/2026** **CS29204**: Switching Circuits Laboratory

Assignment 8

Design of counters

Group 36
Dipam Sen (24CS10059)
Mitali Laddha (24CS10111)

Part (a)

Aim

To design a modulo-16 ripple counter using J-K flip-flops, and verify its output; to modify it to make it a modulo-12 counter, and to connect a resistive-ladder digital-to-analog converter to it and trace the output waveform.

Apparatus

- Breadboard
- Digital Storage Oscilloscope
- 74LS73 × 2 (Dual J-K flip-flop)
- 74LS00 × 1 (Quad NAND gate)
- Resistors, wires, wire cutter, tweezers

Theory

A modulo-n binary counter is a circuit which counts numbers (in their binary representation) from 0 to $n-1$ and then wraps around to 0.

A ripple counter is an asynchronous sequential circuit which comprises of $m = \lceil \log_2 n \rceil$ T flip-flops, in which the output of one gives the clock input of the next. We can design a 4-bit ripple counter using 4 T flip-flops (assumed negative-edge-triggered).

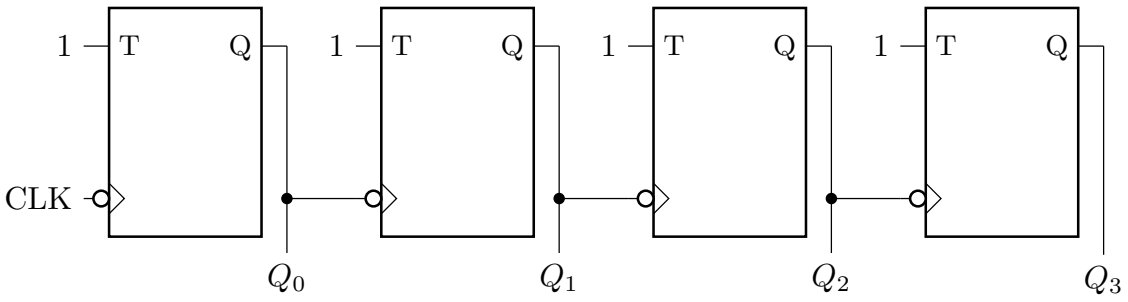


Figure 1: Modulo-16 ripple counter using negative-edge-triggered T flip-flops

We can of course replace the T flip-flops with J-K flip-flops, and connect both J and K inputs to 1, the rest of the circuit stays the same.

In order to convert this into a modulo-12 counter (which is not a power of 2), we need to reset the counter to 0 (0000), as soon as we reach the count of 12 (1100). For this, we can use the CLR inputs of the flip-flops. When the counter reaches 12, this is the first time both Q_3 and Q_2 are both on, so when this is the case, we can reset the counter by trigerring all the CLR inputs.

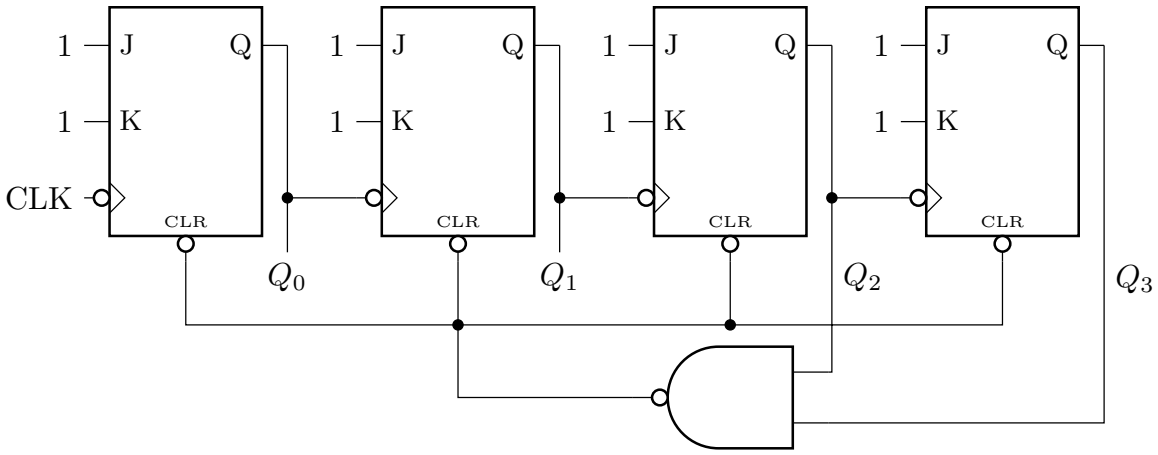


Figure 2: Design of a Modulo-12 ripple counter

Thus, we get a modulo-12 ripple counter. When the counter reaches 12, the NAND gate will output a low, which will clear all the flip-flops to 0, thus resetting the counter. The counter counts from 0 to 11, and then wraps around to 0.

We will add a resistive ladder circuit to the output of the modulo-12 ripple counter, which acts as a digital to analog converter.

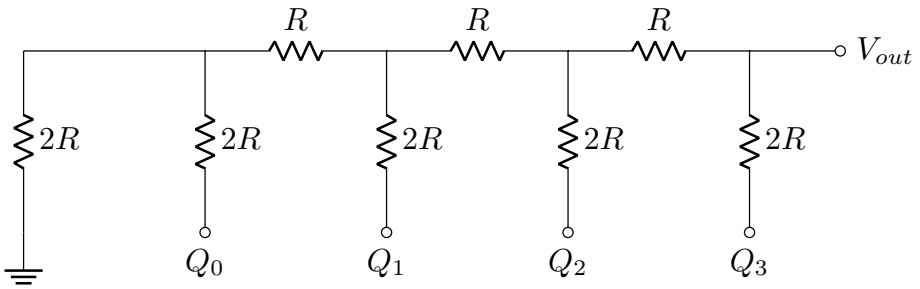


Figure 3: Resistive ladder circuit

Circuit Diagrams

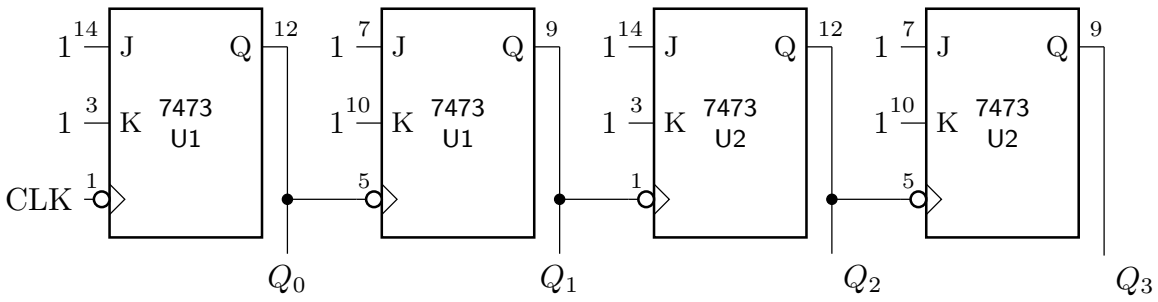


Figure 4: Circuit diagram for Modulo-16 ripple counter

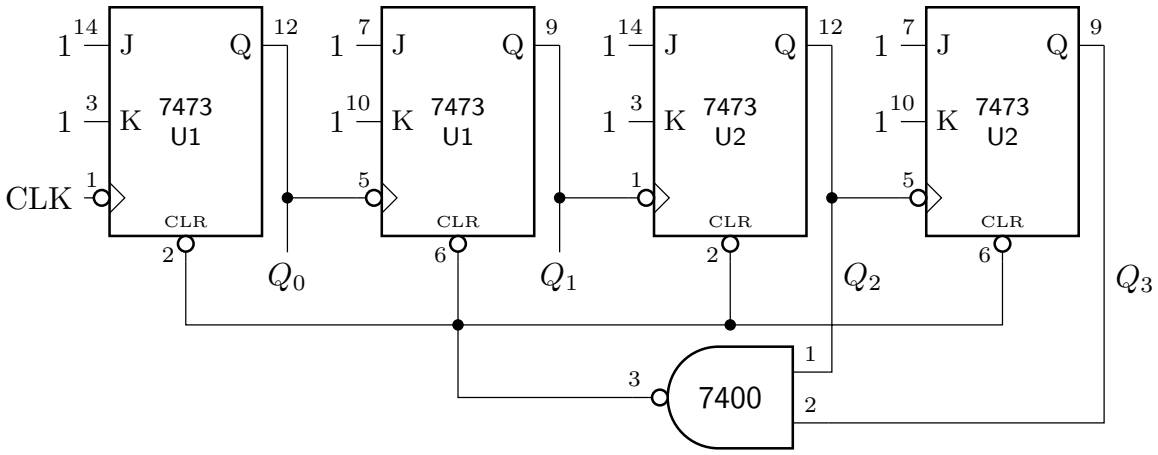


Figure 5: Circuit diagram for Modulo-12 ripple counter

Procedure

1. Realise a 4-bit ripple counter using 4 J-K flip-flops. Verify its working.
2. Connect the appropriate inputs via a NAND gate to all the CLR inputs, so that the counter resets to 0000 when it reaches 12. (Figure 5)
3. Verify that the circuit works as a modulo-12 counter.
4. Connect the output of the counter to a resistive ladder circuit (Figure 3), and observe the output waveform.

Observations

1. The modulo-16 ripple counter counts from 0 to 15, and then wraps around to 0.
2. The modulo-12 ripple counter counts from 0 to 11, and then wraps around to 0.
3. When the output of the modulo-12 ripple counter is fed to the resistive ladder circuit, the output shows a 12-step waveform. It is periodic with a period of 12 clock cycles.

Part (b)

Aim

To design a 3-bit synchronous counter using minimum number of J-K flip-flops and any required additional gates, which would count in the sequence 000, 011, 100, 110, and then back to 000.

Apparatus

- Breadboard
- Digital Storage Oscilloscope
- 74LS73 × 1 (Dual J-K flip-flop)
- 74LS02 × 1 (Quad NOR gate)
- 74LS86 × 1 (Quad XOR gate)
- Resistors, wires, wire cutter, tweezers

Theory

We are required to design a synchronous counter circuit, which counts through the sequence $\{000, 011, 100, 110\}$. We can model this as a finite state machine, without any inputs and with a 3-bit output.

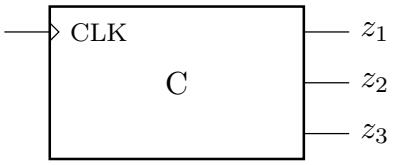


Figure 6: Schematic diagram for FSM to be designed

We can draw the state-transition diagram, since we need to count through 4 different bit sequences, we need 4 states.

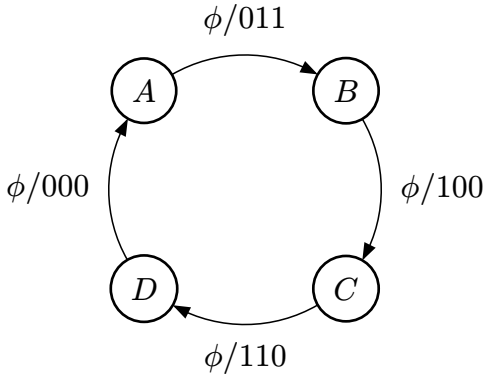


Figure 7: State transition diagram for FSM

Correspondingly, we can write the state table:

PS	NS	Output
A	B	011
B	C	100
C	D	110
D	A	000

Table 1: State table for FSM

Next, we need to assign values for the different states. Since we have 4 states, we need at least 2 bits to represent the state. We can use the following state assignment:

$$A = 00$$
$$B = 01$$
$$C = 10$$
$$D = 11$$

Thus, we can rewrite the state table using the above state assignment:

PS		NS		Output		
y_2	y_1	Y_2	Y_1	z_1	z_2	z_3
0	0	0	1	0	1	1
0	1	1	0	1	0	0
1	0	1	1	1	1	0
1	1	0	0	0	0	0

Table 2: State table for FSM after state assignment

Now, we need to choose the type of flip flop to use. In this case, we are given J-K flip-flops. So we can write down the excitation table for a J-K flip-flop:

	J	K
$0 \rightarrow 0$	0	X
$0 \rightarrow 1$	1	X
$1 \rightarrow 0$	X	1
$1 \rightarrow 1$	X	0

Table 3: Excitation table for J-K flip-flop

We need to find the combinatorial circuit which will produce the values of the flip-flop inputs and the circuit outputs.

Using the excitations for the J-K flip-flops, we can write the final excitation and output table for the FSM.

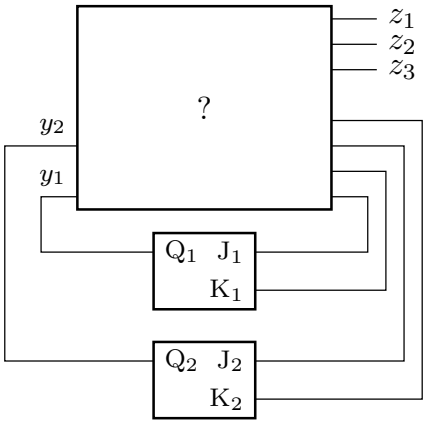


Figure 8: Schematic diagram for the FSM

y_2	y_1	J_2	K_2	J_1	K_1	z_1	z_2	z_3
0	0	0	X	1	X	0	1	1
0	1	1	X	X	1	1	0	0
1	0	X	0	1	X	1	1	0
1	1	X	1	X	1	0	0	0

Table 4: Final excitation and output table for FSM

Thus, we have the values that our circuit should produce for J_1, K_1, J_2, K_2, and the output bits z_1, z_2, z_3.

To realise a circuit that implements the above table, we can draw Karnaugh maps for these outputs and obtain their minimal switching expressions.

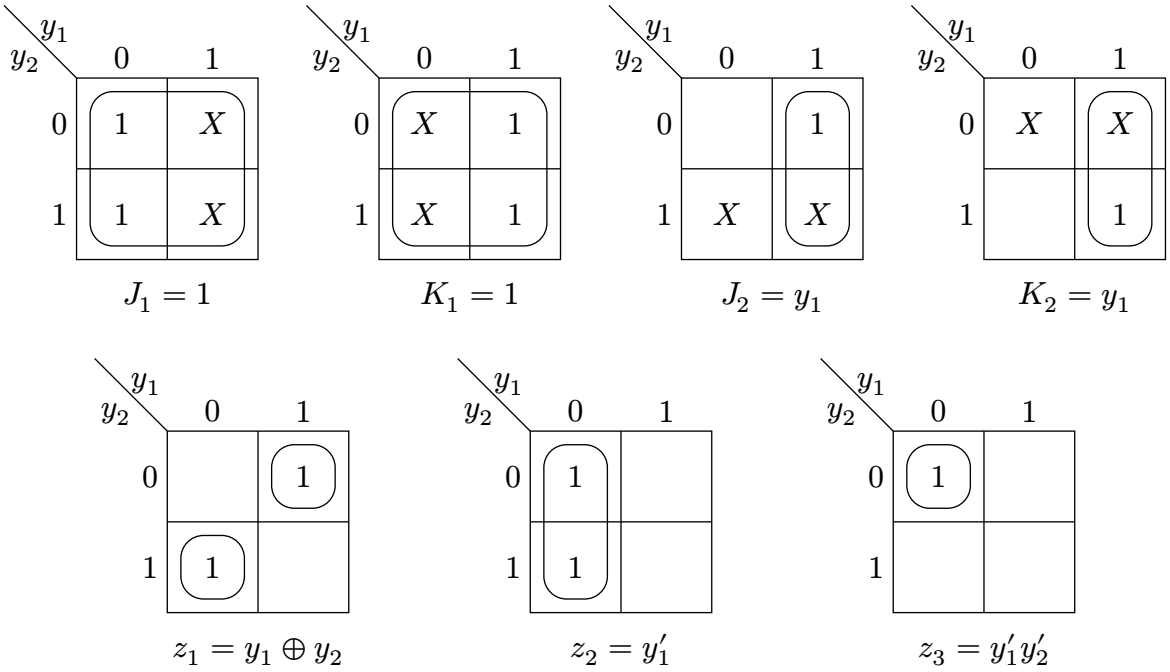


Figure 9: Karnaugh maps for all the output signals of the FSM

Thus, we can design the combinational circuit for the FSM using NOR and XOR gates.

Circuit Diagrams

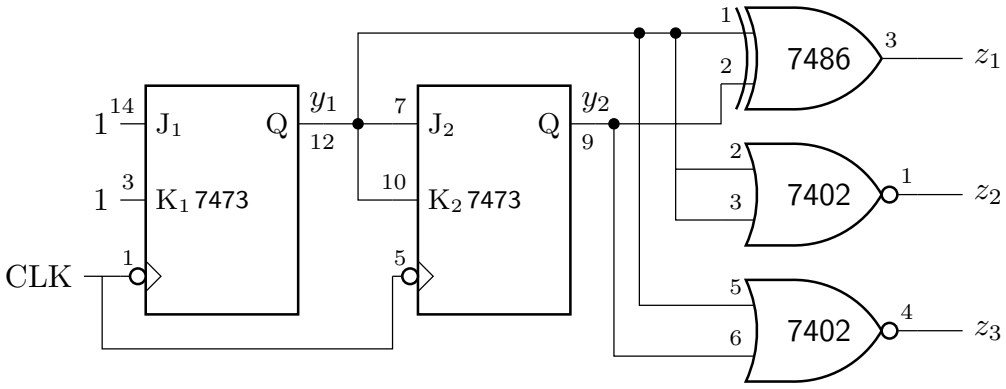


Figure 10: Circuit diagram for the FSM

Procedure

Realise the final designed circuit (Figure 10) for the FSM. Verify that it operates correctly and counts in the sequence $\{000, 011, 100, 110\}$.

Observations

The circuit is designed correctly, and works as a counter which counts in the sequence $\{000, 011, 100, 110\}$.

Part (c)

Aim

[Design Only] To modify the design of the counter as designed in part (b) such that if the counter starts with any initial state (other than the four 3-bit states), it will move to state 000 in the next clock.

Theory

We need to modify the previous design, so that if the counter starts in an invalid state, it should move the the 000 state in the next clock cycle.

However, note that in the previous design, the FSM has 4 states, and we only use 2 flip-flops to store the state, so we can only represent 4 states in total, all of which are valid. Thus it is not possible for the circuit to ever start with an invalid state.

Thus, to fulfill the requirements in this part, we need to add another flip-flop to the circuit (or equivalently, change the state assignments to 3-bits). Then, we will have some invalid states, which we can use to transition to the 000 state in the next clock cycle.

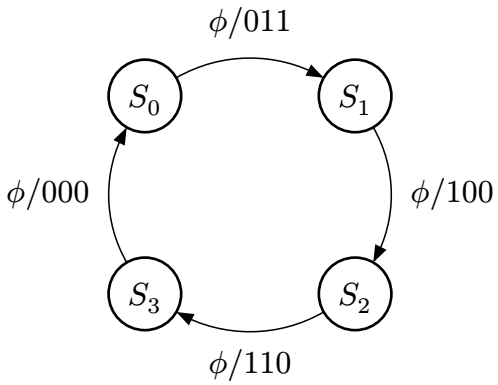


Figure 11: State transition diagram for the FSM

We can use the following state assignment:

$$S_0 = 000$$
$$S_1 = 001$$
$$S_2 = 010$$
$$S_3 = 011$$

But now observe that some of the state assignments are invalid, so we can assign them as well to some states:

$$S_4 = 100$$
$$S_5 = 101$$
$$S_6 = 110$$
$$S_7 = 111$$

And, we can modify the state transition diagram to include these invalid states, so that the circuit can transition to the S_0 state in the next clock cycle.

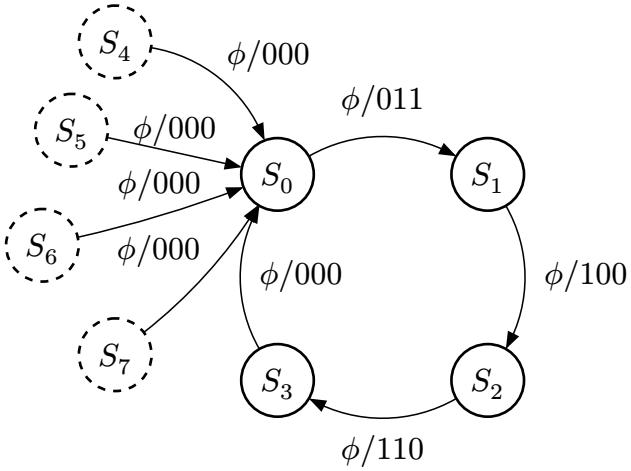


Figure 12: Modified state transition diagram for the FSM

To design a circuit corresponding to this FSM, first we need to write the state table for the FSM.

PS	NS	Output
S_0	S_1	011
S_1	S_2	100
S_2	S_3	110
S_3	S_0	000
S_4	S_0	000
S_5	S_0	000
S_6	S_0	000
S_7	S_0	000

Table 5: State table for the FSM

We can rewrite it using our state assignment:

PS			NS			Output		
y_3	y_2	y_1	Y_3	Y_2	Y_1	z_1	z_2	z_3
0	0	0	0	0	1	0	1	1
0	0	1	0	1	0	1	0	0
0	1	0	0	1	1	1	1	0
0	1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0

Table 6: State table for FSM after state assignment

Once again, we choose to use J-K flip-flops. We can continue the design same as before.

	J	K
$0 \to 0$	0	X
$0 \to 1$	1	X
$1 \to 0$	X	1
$1 \to 1$	X	0

Table 7: Excitation table for J-K flip-flop

y_3	y_2	y_1	J_3	K_3	J_2	K_2	J_1	K_1	z_1	z_2	z_3
0	0	0	0	X	0	X	1	X	0	1	1
0	0	1	0	X	1	X	X	1	1	0	0
0	1	0	0	X	X	0	1	X	1	1	0
0	1	1	0	X	X	1	X	1	0	0	0
1	0	0	X	1	0	X	0	X	0	0	0
1	0	1	X	1	0	X	X	1	0	0	0
1	1	0	X	1	X	1	0	X	0	0	0
1	1	1	X	1	X	1	X	1	0	0	0

Table 8: Final excitation and output table for FSM

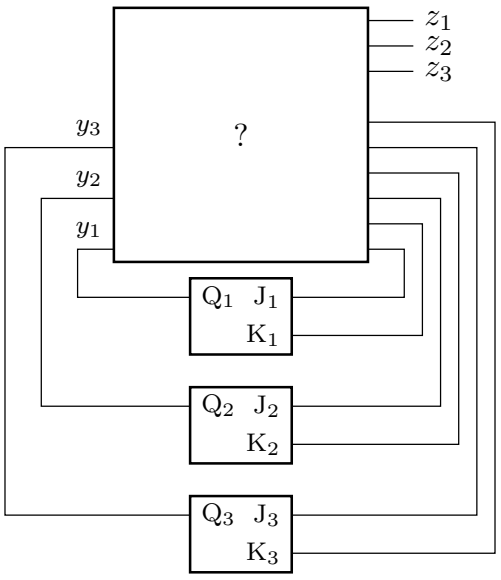
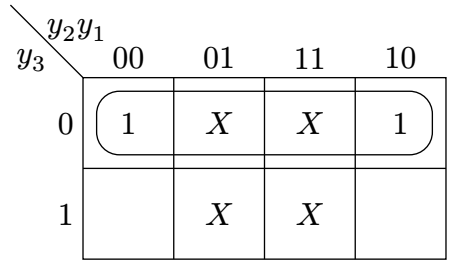
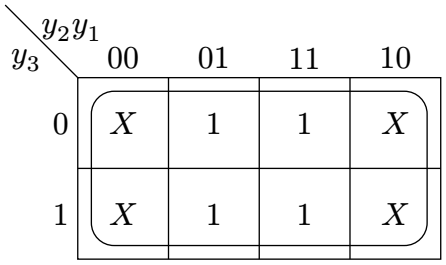


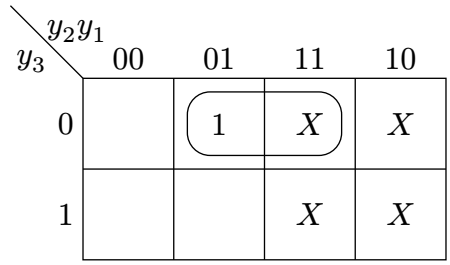
Figure 13: Schematic diagram for the FSM



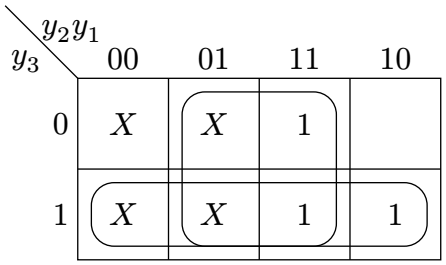
$J_1 = y_3'$



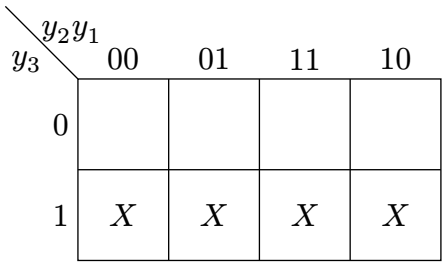
$K_1 = 1$



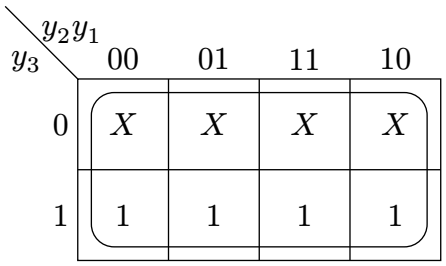
$J_2 = y_3'y_1$



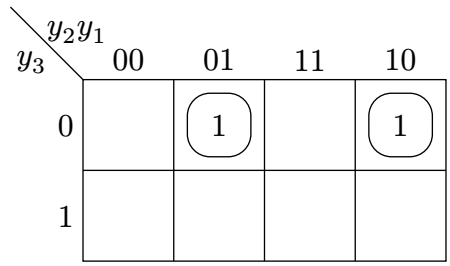
$K_2 = y_3 + y_1$



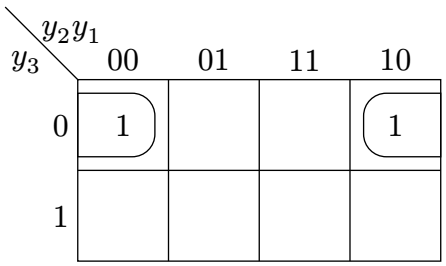
$J_3 = 0$



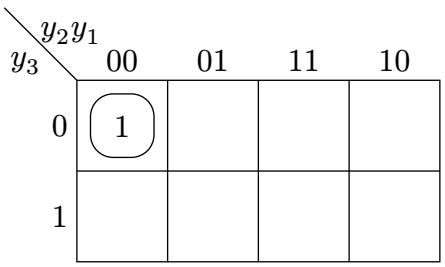
$K_3 = 1$



$z_1 = y_3'(y_1 \oplus y_2)$



$z_2 = y_3'y_1'$



$z_3 = y_1'y_2'y_3'$

Figure 14: Karnaugh maps for all the output signals of the FSM

Circuit Diagram

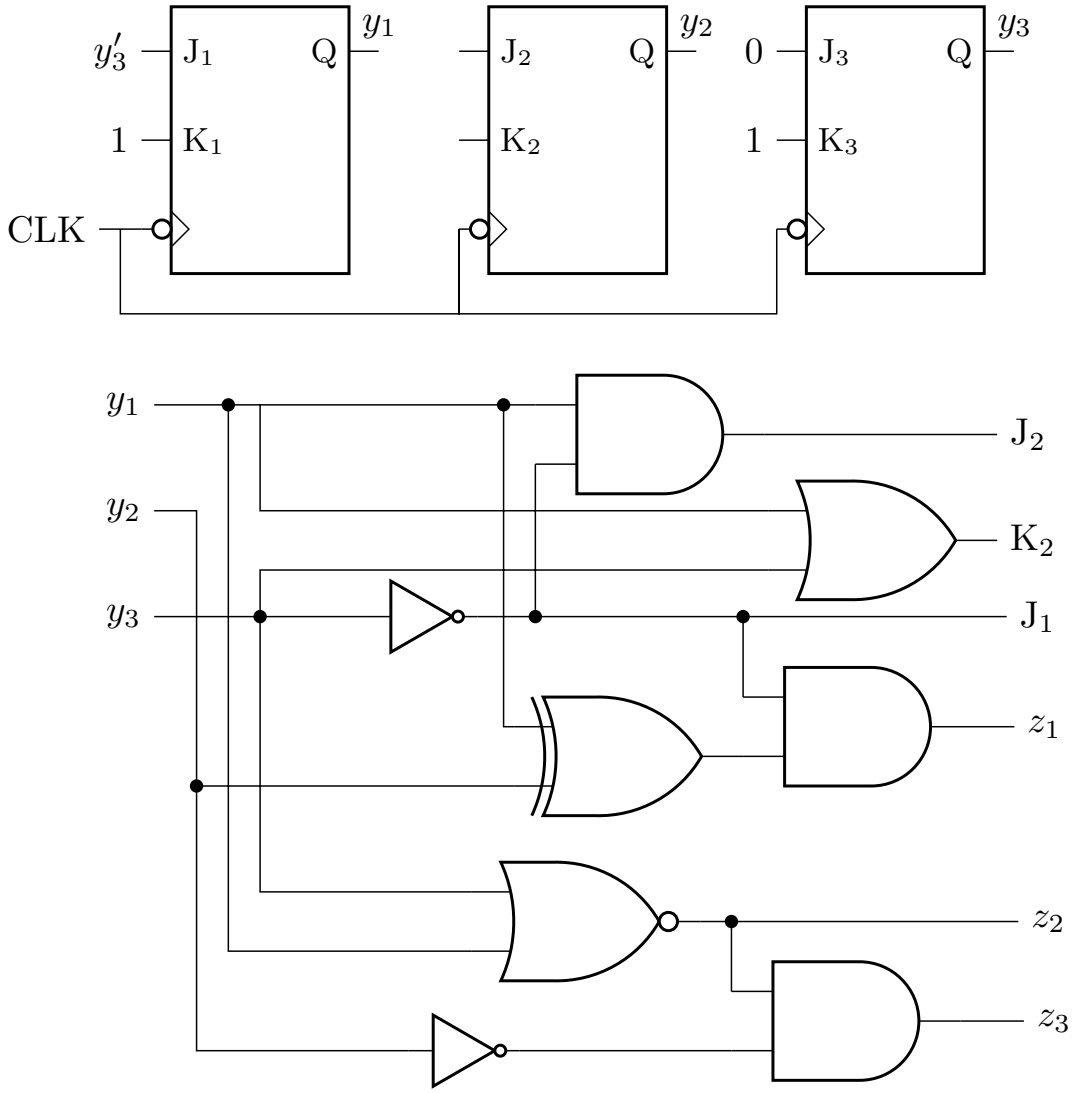


Figure 15: Final circuit diagram for the FSM