for above state diagram, draw the k-map, and then implement the circuit use MUX of your choice and decoder.
Using the mux below, complete the 4 input truth table.
Give answer to following Two’s Complement numbers:

0100 + 0001 =

0111 + 1010 =

1011 + 1001 =

1111 – 0110 =

0001 – 1111 =

1110 – 1010 =

0100 – 0010 =
Question:

Draw the circuit represented by this code. What is its counting sequence?

```verilog
module Hsr (Clock, R, L, Q);
    input logic Clock, L;
    input logic [2:0] R;
    output logic [2:0] Q;
    always @ (posedge Clock) begin
        if (L)
            Q <= R;
        else begin
            Q[0] = Q[2];
            Q[1] = Q[0];
        end
    end
endmodule
```
Design a circuit to determine if a given number is odd using a MUX and Basic gates.
a) Produce the minimum sum of products of the following K-map

```
  | 1 | 0 | 1 | X |
---|---|---|---|---|
 0 | I | C | 0 | X |
 1 | X | X | X | 1 |
 0 | 0 | 1 | 1 | 1 |
```

b) Draw the Circuit Diagram for the minimum sum of product equation above using only NANDs, NORs and Inverters.
EE 271 HW 6a
Design a 4:3 encoder. It will take inputs D4, D3, D2, D1 and produce A2, A1 and A0. It will count how many inputs are true. (0 true inputs will give 000, 1 true input will give 001, 2 true inputs will give 010, 3 true inputs will give 011, 4 true inputs will give 100).
Write the Verilog for a 3 digit input. Output is TRUE when odd number of inputs are true; Output is FALSE when even number of inputs are true.
For the circuit below, explain whether or not output “OUT” will ever be true, given an input “IN”.

\[ T_{\text{setup}} = T_{\text{hold}} = 0.2 \text{ ns}, \quad \text{CLK} \rightarrow Q = 3 \text{ ns}, \text{ gate delays are 1 ns, and clock period is } 50 \text{ ns.} \]
Implement the following logic function using the four methods described

\[ F = AB + B\overline{C} + \overline{A} C + \overline{B} C \]

a. 4:1 mux
b. K map
c. 3:8 decoder
d. Using only NAND/NOR gates
Perform the following Base Conversions:

a) \((271)_{10}\) to Binary:

b) \((1101011)_{2}\) to Hexadecimal:

c) \((48879)_{10}\) to Hexadecimal:

d) \((ACE)_{16}\) to decimal:
Convert the Verilog code into circuit using basic gates and muxes

```verilog
logic [1:0] a;
logic [2:0] b, c, d;

always_comb begin
    case (a)
        2'b00: b = c;
        2'b01: b = d;
        2'b10: b = 2'b11;
        2'b11: b = 2'b01;
    endcase
end
```
Implement $F = ABD + BCD$ using 3:8 decoders

Build a 32x4 RAM from $16 \times 4$ RAMs.
Build a circuit diagram for a 3 bit 2's comp subtractor.
Design a mediocre 4-bit calculator using circuit and block diagrams. It has 3 buttons: "+1", "x2", and "clear". Whenever "+1" is pressed, the display (output) should go up by 1. Whenever "x2" is pressed, the output should double. If "clear" is pressed, the output goes to "0000". Ignore overflow.
If gate delays are 1.0 ns, Tsetup is .5 ns, clk->q is .25 ns, Thold is 1.5 ns. Find the smallest legal clock period. Is the given Thold the largest legal Thold? If it is not, find the largest legal Thold. Redesign the circuit if the given Thold is not met in any paths taken in the circuit.
Both 0101, the output is true, but 1010, and 0 is false. The output otherwise, the output will be false. For example, it is a and b are always true.

Sequence is found, the system will always output true. Occurs in both inputs at the same time. It a 4-bit matching another or not true is a 4-bit sequence that occurs in two inputs and b and are determined.

Draw the state diagram for the following.
Build a full adder using only two 4:1 muxes and as few inverters as possible.