DSM LAB REPORT – 7

Name: **M.Harish**

Id no: **2020102062**

Lab 7 - Binary cell for RAM

Objective:

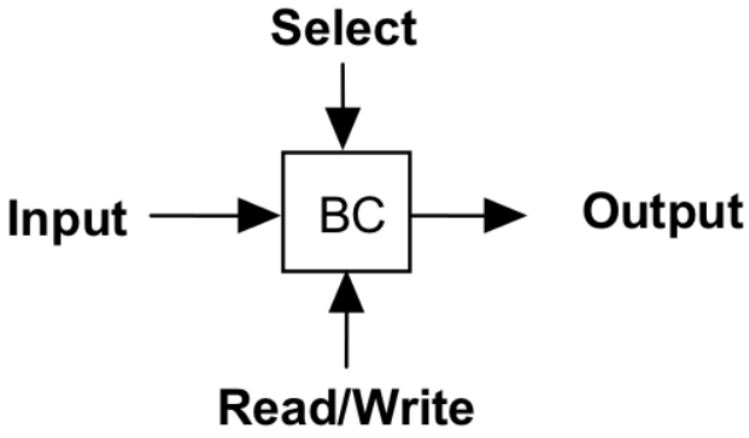
To implement and verify the operation of a Binary cell for RAM based on RS flipflop.

Electronics components Required:

- Arduino
- Bread Board
- 74HC11 - (Tripple 3-Input AND gate)
- 74HC02 - (Quad NOR gate)
- 74HC04 - (HEX inverter)
- Led's, Resistors, Connecting wires and push button.

Theory:

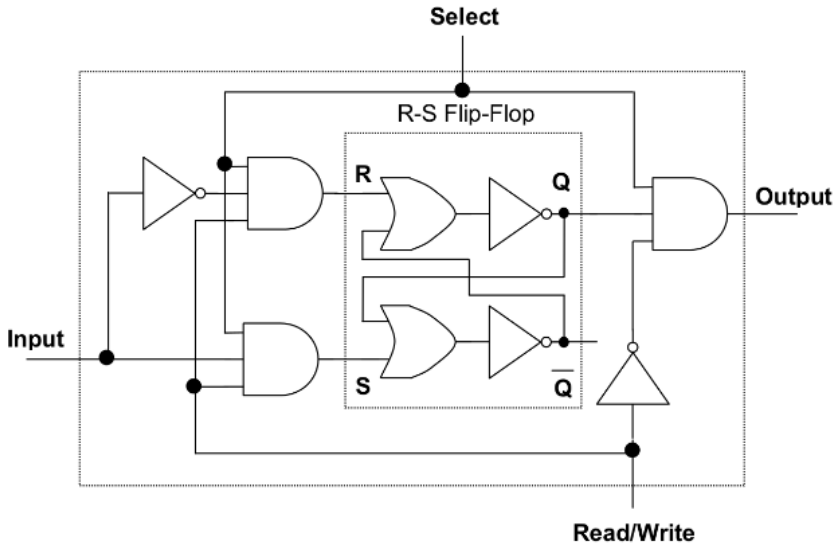
The cell has three inputs and a single output. The inputs are labeled "Select", "Read/write", and "Input". The output line is labelled "Output".



Working of a Binary cell:

The schematic of a binary cell is shown in the following figure.

Reference circuit:



- The “select” input is used to access the cell, either for reading or writing.
- When the select line is high, “1”, then a memory operation can be performed on this cell.

- When the select line of the binary cell is low, “0”, then the cell is not being read from or written to i.e., if “select” is low, the inputs to the R-S flip-flop will stay low (its stored value will not change) and the output produced by the cell will be low regardless of whether the actual bit held in the flip-flop is “0” or “1”.
- Now we’ll see how this block acts as a memory by doing read/write operations on the cell. This depends on the value at read/write signal. A low, “0”, will signify “read” while a high, “1”, will signify “write”. One point to be noted is that both the operations cannot be done simultaneously.
- For a memory operation to be performed on the cell, the “select” should be high.

Reading the contents:

- If the clock value on the “Read/write” line is low (which makes the “negated Read/write” high) indicating the cell contents are to be read. In this case, the value output by the cell will depend solely on the Q value of the flip-flop.
- If Q is low, the cell outputs a “0”, if Q is high, the cell outputs a “1”. This is because the and gate attached to the cell’s output line has three inputs: “select”, “negated Read/write”, and Q; and both “select” and “negated Read/write” are currently high.
- When the cell is being read its contents cannot be modified. The reason for this is that the same low value on the “Read/write” line that allows the cell to be read, is fed into the and gates guarding the inputs to the flip-flop. Thus, during reads, the inputs to R and S are guaranteed to be low preventing the value of the flip-flop from being modified.

Writing contents on to the cell:

- When the cell is selected and the “Read/write” line is set to high, signifying a “write” operation, the value placed into the cell will depend solely on the state of the “Input” line.
- The reason for this is that the and gates that guard the R and S inputs of the flip-flop will both have two of their inputs set high: the “select” and “Read/write” inputs.
- Thus, if “Input” is high, S (set) will receive a high and the flip-flop will store a “1”. If, on the other hand, “Input” is low, then R (reset) which receives a negated version of “Input” will go high and the flip-flop will reset to “0”.

NOTE: Having a negated version of the input line run into R prevents the R-S flip-flop from ever entering its invalid state.

Procedure:

- Place a breadboard and an Arduino onto the workspace and connect the 5V pin of the Arduino to the positive strip of the breadboard both at the top and bottom.
- Similarly connect the GND pin of the Arduino to the negative strip of the breadboard both at the top and bottom.
- Place the necessary ic’s on the bread board and connect the power terminals of ic’s to 5v and ground terminals to ground.
- Now consider the D13 pin of the Arduino to be the input, the D12 pin to be the read/write point and the D11 pin to be the select lines.
- Make the necessary connections as shown in the reference circuit diagram.
- Write the code accordingly.
- Note down the observations.

Code:

```
int r_w=0;
int i=0;
int s=0;
void setup()
{
        Serial.begin(9600);
        pinMode(11, OUTPUT);
        pinMode(13, OUTPUT);
        pinMode(12, OUTPUT);
}
void loop()
{
        if(Serial.available()==3)
        {
                s=Serial.read();
                s=s-'0';
                Serial.print("SELECT:");
                Serial.println(s);
                i=Serial.read();
                i=i-'0';
                Serial.print("INPUT:");
```

```
            Serial.println(i);
            r_w=Serial.read();
            r_w=r_w-'0';
            Serial.print("R/W:");
            Serial.println(r_w);
            if(s==1)
            {
                  if(r_w==0)
                  {
                        Serial.print("READING\n ");
}
            }
            else if(r_w == 1)
            {
                  Serial.print("WRITING\n");
            }
            else if(s==0)
            {
                  Serial.print("the device is reset\n");
            }
      }
digitalWrite(12, s);
```

```
digitalWrite(13, i);

digitalWrite(11, r_w);

}
```

Observations:

- When all the select line, Input and R/W are set to 1 the circuit will be in the write state and it stores 1 in it.
- Now when Select line = 1 Input = 1 R/W = 0

The circuit is in the Read mode and the Output LED starts glowing since the previously stored value in the circuit was 1. The same thing happens when the input is 0.

Now if the select line is 0 then the circuit gets turned off.

When the select line is zero the circuit is in reset state.

A continuous series of inputs can be given to analyze the circuit.

Select	input	R/W	Q	Q'	OUTPUT
0	0	0	1	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	0	1	0
1	1	0	1	0	1
1	1	1	1	0	0

Link for tinkercad simulation:

https://www.tinkercad.com/things/5lnaPEvrtTI2020102062lab7/editel?sharecode=wG6ErKhafTFCUjBhveHRK7XD9xpsFlulpONChKcSNmY

Conclusion:

I understand the working of Binary cell using this experiment. Using this experiment, I understood the functionality of select lines, read/write and their applications in real life and how they act in hardware.