**Name:** Anirudh Kaushik

**Roll No:** 2020111015

**Group Number:** 5

**Experiment Part A**

**Objective**

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

**Theory:**

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

|  |
| --- |
|  |

**Working of a Binary cell:**

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.

(Basically the read/write will act as clk here)

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.

|  |
| --- |
|  |

**Experiment setup/ procedure**

**Materials required:**

|  |  |
| --- | --- |
| 1 | Arduino Uno R3 |
| 1 | Quad NOR gate |
| 2 | Triple 3-Input AND gate |
| 1 | Hex Inverter |
| 3 | 1 kΩ Resistor |
| 1 | Red LED |
| 1 | Pushbutton |
| 1 | Green LED |
| 1 | Blue LED |

**Procedure:**

Construct the circuit as shown in the circuit diagram.

**Code:**

int latchPin = 13;

//Pin connected to SH\_CP of 74HC595

int clockPin = 12;

////Pin connected to DS of 74HC595

int dataPin = 11;

void setup() {

//set pins to output because they are addressed in the main loop

pinMode(latchPin, OUTPUT);

pinMode(clockPin, OUTPUT);

pinMode(dataPin, OUTPUT);

Serial.begin(9600);

}

int val;

int A;

int B,C;

void loop()

{

if(Serial.available() > 0)

{

while(!Serial.available())

{}

A = Serial.read(); // x would be an integer between 0 and 255

// depending on the ascii value of the character read

A = A - '0'; // Subtracting ascii value of 0 from x.

if(A == 0)

{

digitalWrite(13,LOW);

Serial.print("Select = ");

Serial.println(A);

}

if(A == 1)

{

digitalWrite(13,HIGH);

Serial.print("Select = ");

Serial.println(A);

}

while(!Serial.available())

{}

B = Serial.read(); // x would be an integer between 0 and 255

// depending on the ascii value of the character read

B = B - '0'; // Subtracting ascii value of 0 from x.

if(B == 0)

{

digitalWrite(12,LOW);

Serial.print("Read/Write = ");

Serial.println(B);

}

if(B == 1)

{

digitalWrite(12,HIGH);

Serial.print("Read/Write = ");

Serial.println(B);

}

while(!Serial.available())

{}

C = Serial.read(); // x would be an integer between 0 and 255

// depending on the ascii value of the character read

C = C - '0'; // Subtracting ascii value of 0 from x.

if(C == 0)

{

digitalWrite(11,LOW);

Serial.print("Input = ");

Serial.println(C);

}

if(C == 1)

{

digitalWrite(11,HIGH);

Serial.print("Input = ");

Serial.println(C);

}

}

}

**Observations**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Select** | **Read/Write** | **Input** | **Q’** | **Q** | **Output** |
| 0 | X | X | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | X | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | X | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | X | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | X | 0 | 1 | 1 |
| 0 | X | X | 0 | 1 | 0 |

**Conclusion**

Successfully constructed a Binary cell based on an RS flip-flop with select line and Read/Write function and verified it’s working.

**Tinkercad link with circuit**

<https://www.tinkercad.com/things/lUyTrcG16of-grand-robo-hillar/editel?sharecode=dgb7ZZFT8DA_u-SuYzNPtrgIIhToFvAmYt5sA1CAkns>

**Circuit Diagram**

**Please scroll down**

|  |
| --- |
|  |