DSM LAB REPORT –3

Multiplexing and Demultiplexing

Macharla Harish (2020102062)

Objective:

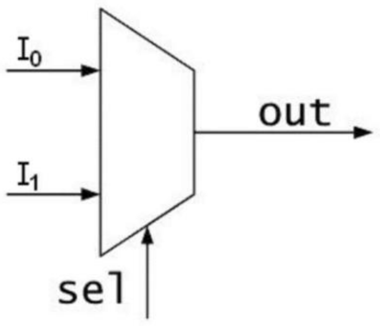
To design, assemble and test a (1:4) Multiplexer and (4:1) De-multiplexer using basic logic gates (whose select lines and inputs are through Arduino).

Description:

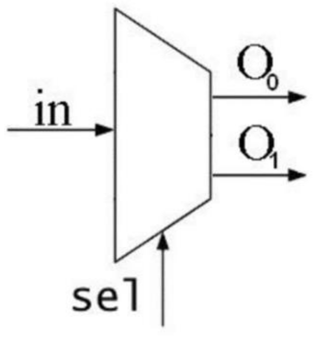
Multiplexer and Demultiplexer:

A multiplexer (or mux) is a device that selects one of several analog or digital input signals and forwards the selected input into a single line A multiplexer with 2^n inputs have n select lines, which are used to select which input line to send to the output.

Conversely, a demultiplexer (or demux) is a device taking a single input signal and selecting one of many data-output-lines, which is connected to the single input. A multiplexer is often used with a complementary demultiplexer on the receiving end.



Multiplexer



Demultiplexer

Part-A: Design 4:1 Multiplexer using basic logic gates

Objective:

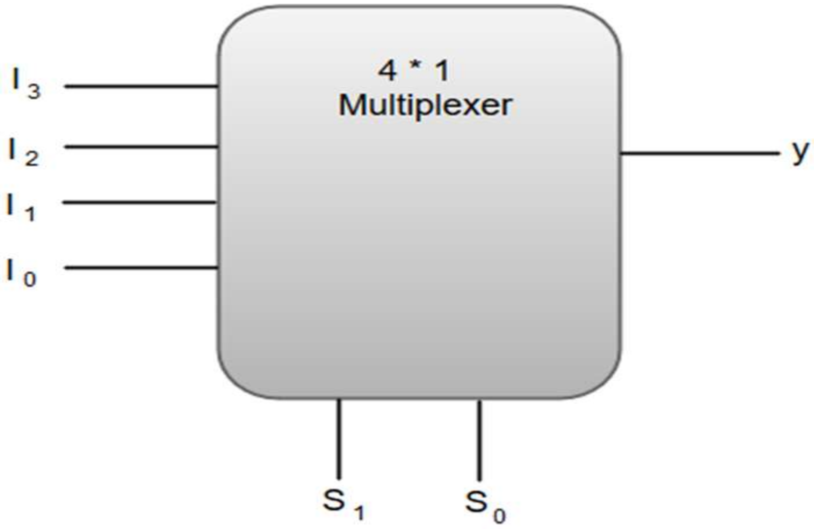
To design, assemble and test a (1:4) Multiplexer using basic logic gates (whose select lines and inputs are through Arduino).

Equipment Required:

1. LED bulb
2. Resistor
3. Arduino Uno
4. Bread board (BIG)
5. Connecting wires
6. Hex Inverter(74HC04)
7. Triple-And gate(74HC11)[2]
8. OR gate(74HC32)

Procedure:

A 4-to-1 multiplexer consists of four data input lines as I0 to I3, two select lines as S0 and S1 and a single output line Y. The select lines S0 and S1 select one of the four input lines (D0 through D3) to connect to the output line. The figure below shows the block diagram of a 4-to-1 multiplexer.



The truth table of a 4-to-1 multiplexer is shown below in which four input combinations 00, 10, 01 and 11 on the select lines respectively switch the inputs I0, I1, I2 and I3 to the output. That means when S1=0 and S0 =0, the output at Y is D0, similarly Y is D1 if the select inputs S1=0 and S0= 1 and so on.

S1	S0	Y
0	0	i0
0	1	i1
1	0	i3
1	1	i4

From the above truth table, we can write the output expressions as

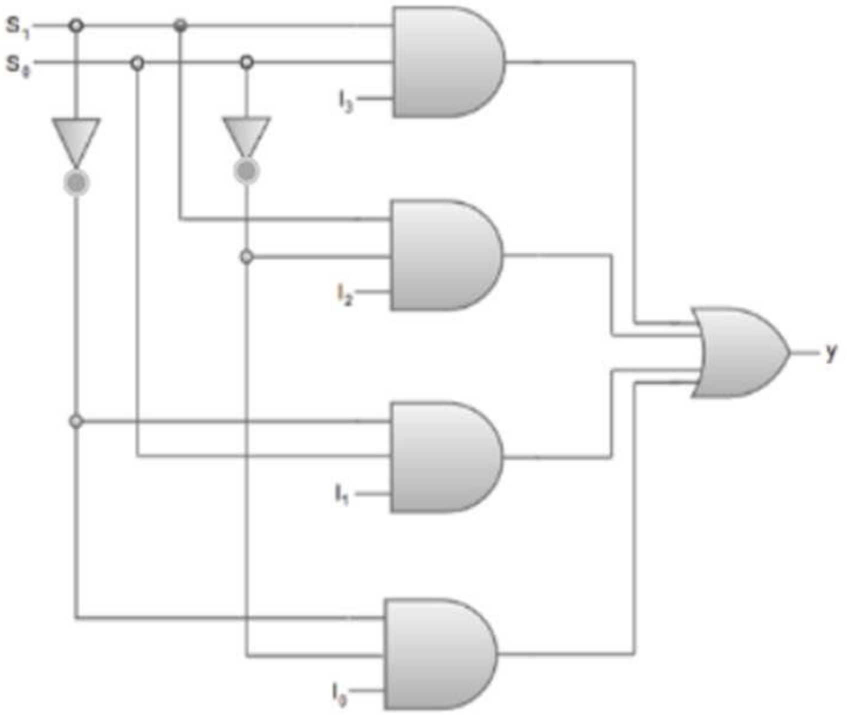
- If S1=0 and S0=0, then Y = i0 Therefore, **Y = i0 (S1)' (S0)'**
- If S1=0 and S0=1, then Y = i1 Therefore**, Y = i1 (S1)' S0**
- If S1=1 and S0=0, then Y = i2 Therefore, **Y = i2 S1 (S0)'**
- If S1=1 and S0=1, then Y = i3 Therefore, **Y = i3 S1 S0**

To get the total data output from the multiplexer, all these product terms are to be summed and then the final Boolean expression of this multiplexer is given as

Y = i0 (S1)' (S0)' + i1 (S1)' S0 + i2 S1 (S0)' + i3 S1 S0

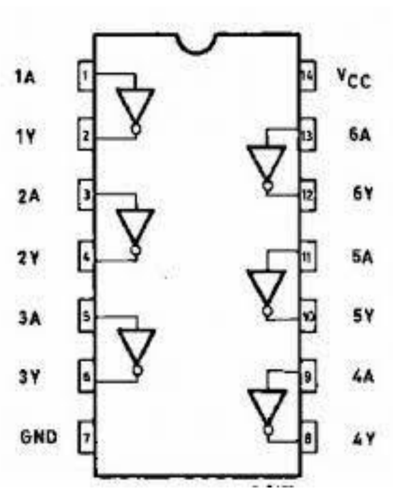
The below figure shows the logic circuit of 4:1 MUX.

Reference Circuit:

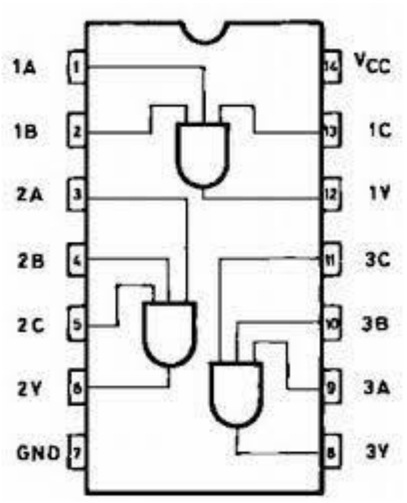


We will be using an inverter IC (74HC04), 2 AND gate ICs (74HC11) and 2 OR gate ICs (74HC32).

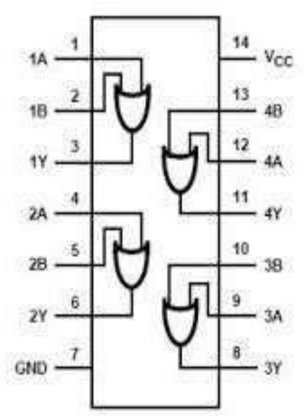
The pin connections of these ICs are as follows.



74HC04 (INVERTER)



74HC11 (AND)



74HC32 (OR)

Link:

https://www.tinkercad.com/things/4CBsvfMabCA-brilliant-hillar-bruticus/editel
sharecode=kKUdg239rC9eZWXwGUNNBg30e50_fDHvu4SGzv_Gh90

Conclusion:

We constructed 4:1 Multiplexer(combinational circuit) using basic logic gates with help of boolean functions.

Part-B: Design 1:4 Demultiplexer using basic logic gates

Objective:

De-multiplexer using basic logic gates (whose select lines and inputs are through Arduino).

Equipment Required:

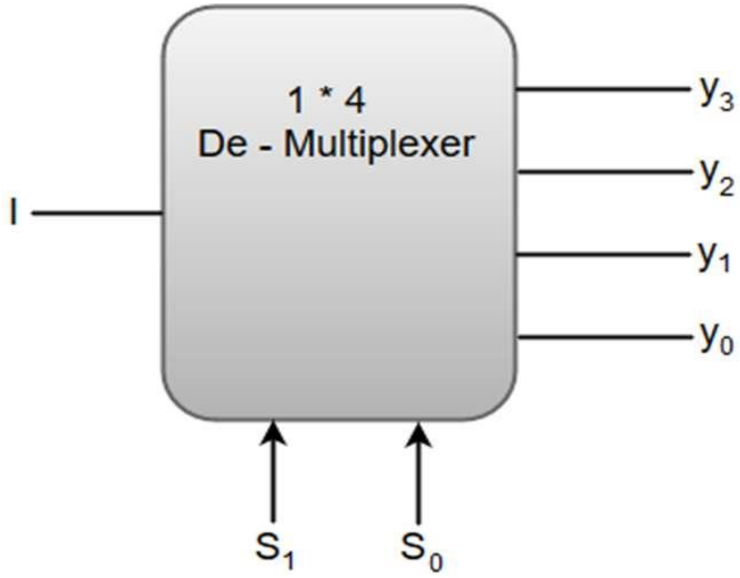
1. LED bulb(4)
2. Resistor(4)
3. Arduino Uno
4. Bread board (BIG)
5. Connecting wires

6. Hex Inverter(74HC04)

7. Triple-And gate(74HC11)[2]

Description:

A 1-to-4 demultiplexer consists of one data input line as i, two select lines as S0 and S1 and four output lines as y0, y1, y2 and y3. The select lines S0 and S1 select one of the four output lines (y0 through y3) to connect to the input line. The figure below shows the block diagram of a 1-to-4 demultiplexer.



1:4 Demultiplexer

The truth table of a 1-to-4 demultiplexer is shown below in which four input combinations 00, 10, 01 and 11 on the select lines respectively switch the input to the output lines y0, y1, y2 and y3. That means when S1=0 and S0 =0, the output at Y is y0 = i, similarly y1 is i if the select inputs S1=0 and S0= 1 and so on.

TruthTable:

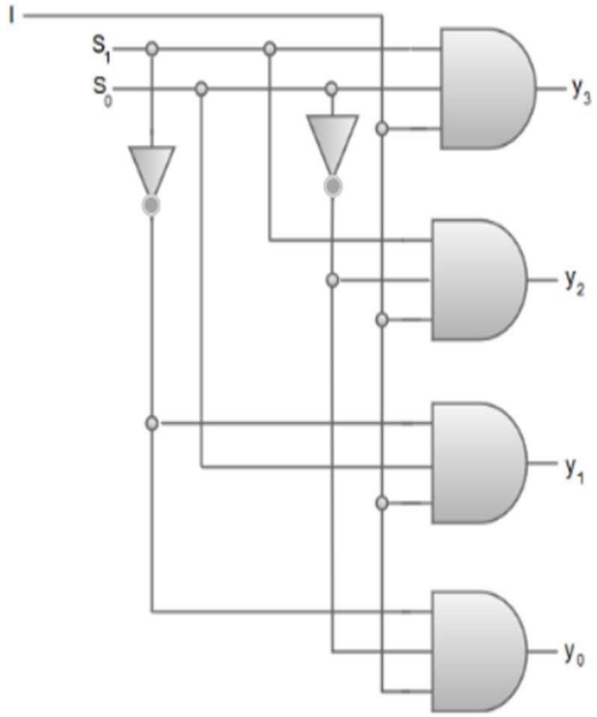
S1	S0	Y0	Y1	Y2	Y3
0	0	i	0	0	0
0	1	0	i	0	0
1	0	0	0	i	0
1	1	0	0	0	i

From the above truth table, we can write the output expressions as

- If S1=0 and S0=0, then y0 = i Therefore, **y0 = i (S1)’ (S0)’**
- If S1=0 and S0=1, then y1 = i Therefore, **y1 = i (S1)’ S0**
- If S1=1 and S0=0, then y2 = i Therefore, **y2 = i S1 (S0)’**
- If S1=1 and S0=1, then y3 = i Therefore**, y3 = i S1 S0**

From the above expressions of the outputs, a 1-to-4 demultiplexer can be implemented by using basic logic gates. The below figure shows the logic circuit of 1:4 DEMUX which is implemented by four 3- inputs AND gates and two 1-input NOT gate.

<u>Reference Circuit:</u>



<u>Link:</u>

https://www.tinkercad.com/things/7NjXBBbUrj8-ingenious-rottis-kasi/editel?sharecode=TaQ8Bg7n8WKVbpNIXe4PhbJgo40Gm1qUojE4cF8oNQQ

Conclusion:

We constructed 4:1 DeMultiplexer(combinational circuit) using basic logic gates with help of boolean functions.

Part-C: Assemble and test circuits designed in Parts A and B

Write a program to give different combinations of inputs at MUX and view them at DEMUX output using LEDs.

Code:

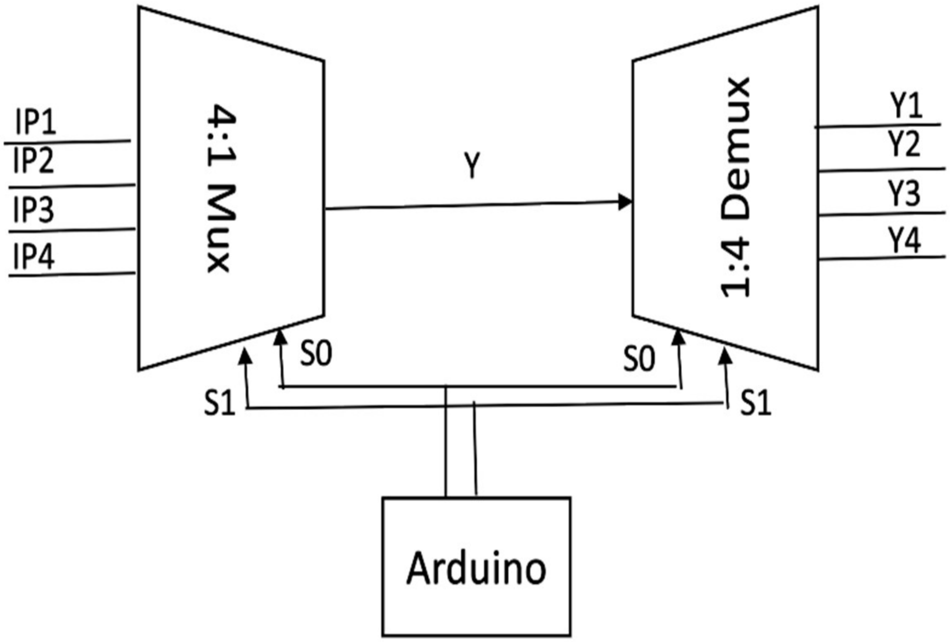
```
int S0,S1,a,b,c,d;
void setup() {
Serial.begin(9600);
  pinMode(2,OUTPUT); //S1
  pinMode(3,OUTPUT); //S0
  pinMode(8,OUTPUT); //i0
  pinMode(9,OUTPUT); //i1
  pinMode(10,OUTPUT);//i2
  pinMode(11,OUTPUT);//i3
   //opens serial port, sets data rate to 9600 bps
}
void loop() {
if(Serial.available() > 0) {
        S1=Serial.read();
        S0=Serial.read();
a=Serial.read();
b=Serial.read();
        c=Serial.read();
        d=Serial.read();
// x would be an integer between 0 and 255
// depending on the ascii value of the character read
        S1 = S1 - '0';
        S0 = S0 - '0';
a = a - '0';
        b = b - '0';
       c = c - '0';
      d = d - '0';

        // Subtracting ascii vealue of 0 from x.
}
        digitalWrite(2,S1);
        digitalWrite(3,S0);
digitalWrite(8,a);
        digitalWrite(9,b);
        digitalWrite(10,c);
        digitalWrite(11,d);

Serial.println(S1);
        Serial.println(S0);
Serial.println(a);
Serial.println(b);
          Serial.println(c);
          Serial.println(d);
}
//001000
//010100
//100010
//110001
```

Equipment Required:

1. LED bulb(1)
2. Resistor(1)
3. Arduino Uno
4. Bread board (BIG)[2]
5. Bread board(mini)
6. Connecting wires
7. Hex Inverter(74HC04)[2]
8. Triple-And gate(74HC11)[4]
9. Connecting wires 10 OR gate(74HC32)

Procedure:

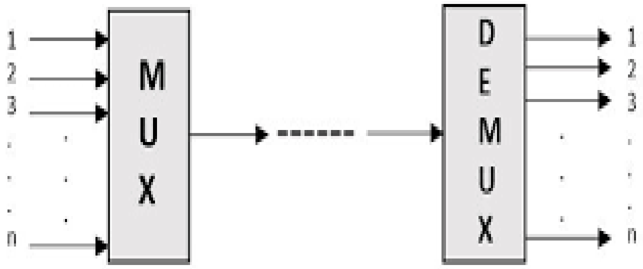
Connect VCC and GND of both mux and demux to 5V and GND pin of Arduino. Connect 4 pins of Arduino to the mux as input pins. The output of the mux should be given as an input to the demux, and the select lines of both the mux and the demux must also be taken from Arduino. Connect the output of the demux to 4 LEDs.



TruthTable:

S1	S0	Y0	Y1	Y2	Y3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Reference circuit:



Multiplexing and Demultiplexing

Link:

https://www.tinkercad.com/things/cfZIgqFZS5T-surprising-lahdi/editel?sharecode=W1lUlR6sddNoLzsK9i7IohWuA9YlriNCz4HoFrni2Hc

Conclusion:

We assembled Mux and Demux and the output of Mux acts as a input to the Demux .

------------End-----------