Experiment 3: Multiplexing and Demultiplexing

Digital Systems and Microcontrollers | Spring 21

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).

In the previous experiment, you learned the working of the basic logic gates. In this experiment we are going to use those logic gates to design multiplexer (or mux) and demultiplexer (or demux)

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 $2^{\wedge n}$ inputs has 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.

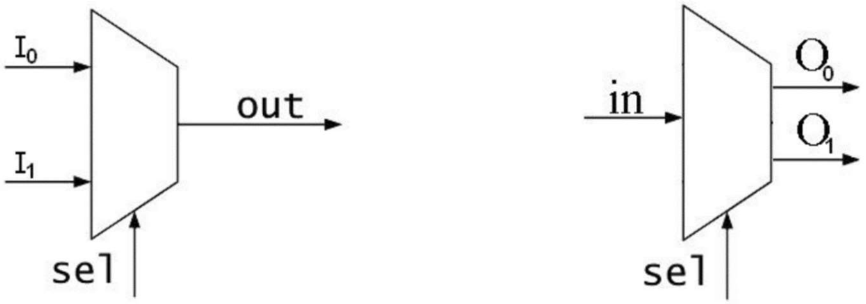


Fig.1 Multiplexer and Demultiplexer

Experiment:

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

A 4-to-1 multiplexer consists of four data input lines as I_0 to I_3, 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.

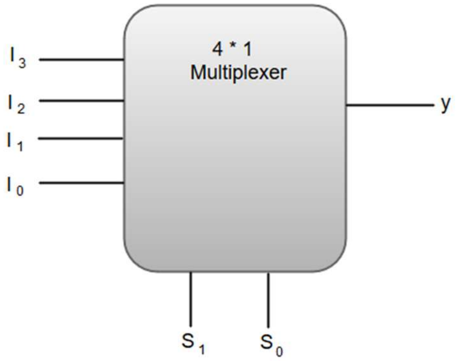


Fig.3 4:1 Multiplexer

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 I_0, I_1, I_2 and I_3 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	I_0
0	1	I_1
1	0	I_2
1	1	I_3

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

If S1=0 and S0=0 then Y = I_0

Therefore, Y = I_0 (S1)' (S0)'

If S1= 0 and S0=1, the Y = I_1

Therefore, Y = I_1 (S1)' S0

If S1=1 and S0=0, then Y = I_2

Therefore, Y = I_2 S1 (S0)'

If S1=1 and S0=1 the Y = I_3

Therefore, Y = I_3 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 = I_0 (S1)' (S0)' + I_1 (S1)' S0 + I_2 S1 (S0)' + I_3 S1 S0

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

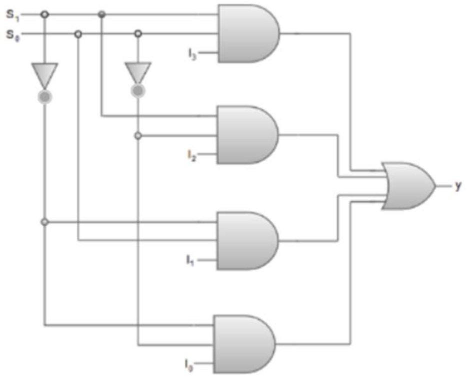
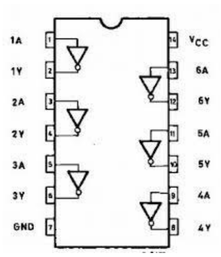
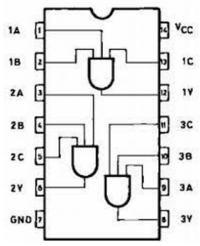


Fig.4 4:1 Multiplexer using logic gates

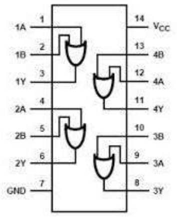
We will be using an inverter IC (74HC04), 2 AND gate ICs (74HC11) and 2 OR gate ICs (74HC32). The pin connections of these ICs are as follows.



74HC04 (INVERTER)



74HC11 (AND)



74HC32 (OR)

Fig.5 Logic gates pin diagrams

1) For the IC 74HC04 (Inverter) the data inputs are denoted by 1A, 2A, 3A, 4A and the data outputs by 1Y, 2Y, 3Y, 4Y.
2) For the IC 74HC11 (AND) and 74HC32 (OR), the data inputs are denoted by 1A, 1B, 1C and so on and the data outputs by 1Y, 2Y, 3Y.
3) Write an Arduino code to give different combinations of inputs at input and select lines and view them using LED at the output line.
4) Verify the multiplexer function by tabulating the values of the output(s) for all input combinations.

The below figure shows the final circuit of 4:1 MUX in Tinkercad:

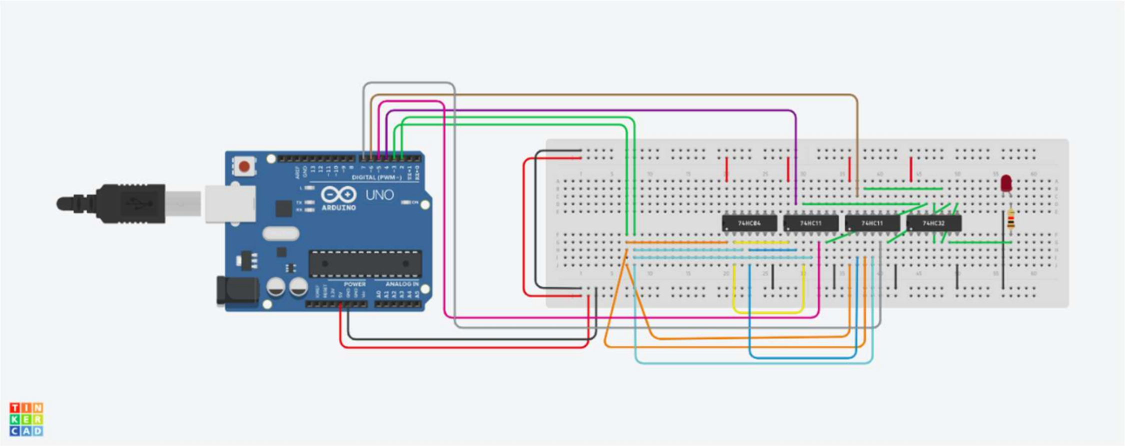


Fig.6 4:1 Multiplexer circuit

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

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.

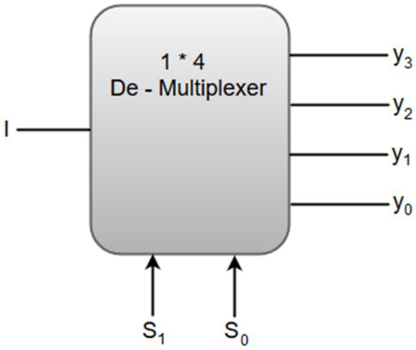


Fig.7 1:4 Demultiplexer

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.

S1	S0	y3	y2	y1	y0
0	0	0	0	0	i
0	1	0	0	i	0
1	0	0	i	0	0
1	1	i	0	0	0

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, the 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 the 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.

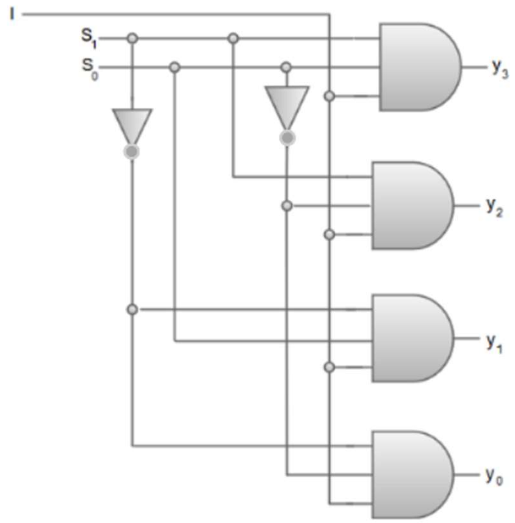


Fig.8 Demultiplexer using logic gates

Like the 4:1 MUX, we will be using an inverter IC (74HC04) and 2 AND gate ICs (74HC11). For the pin connections of these ICs, please refer to figure 5.

1) Write an Arduino code to give different combinations of inputs and select lines and view them using LEDs at the output lines.
2) Verify the demultiplexer function by tabulating the values of the output(s) for all input combinations.

The below figure shows the final circuit of 1:4 DEMUX in Tinkercad:

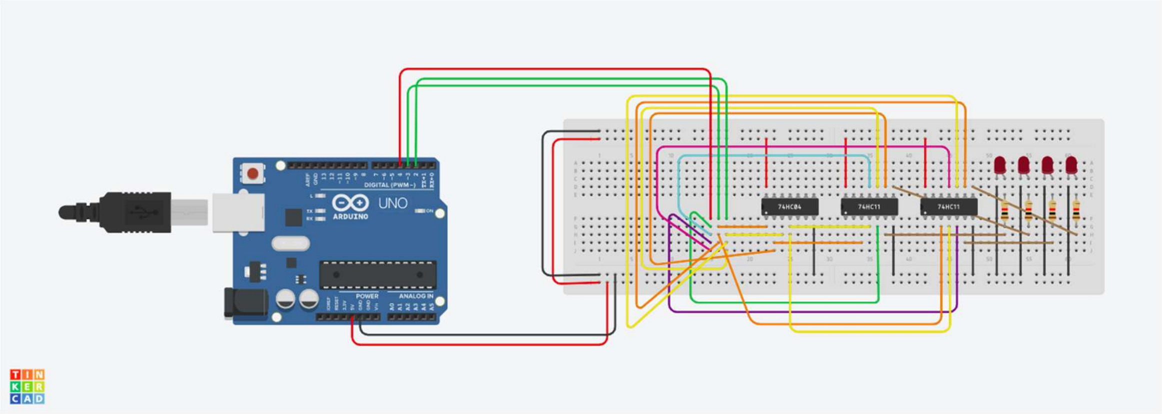


Fig.9 1:4 Demultiplexer

➢ 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.

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 demox to 4 LEDs. The block diagram and Tinkercad circuit diagram of the final circuit are given below.

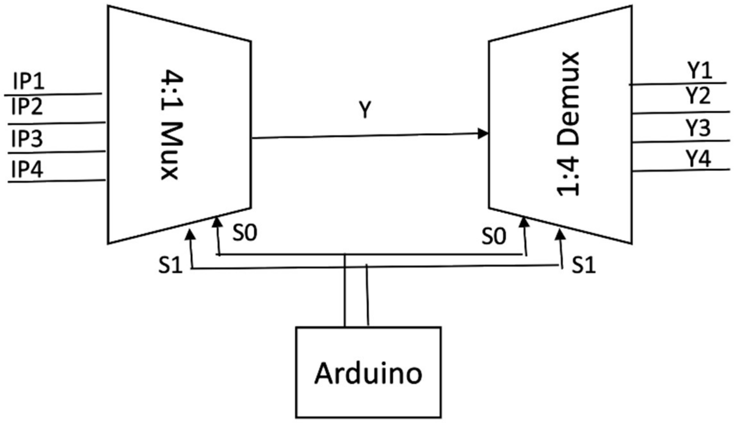


Fig.10 Block diagram

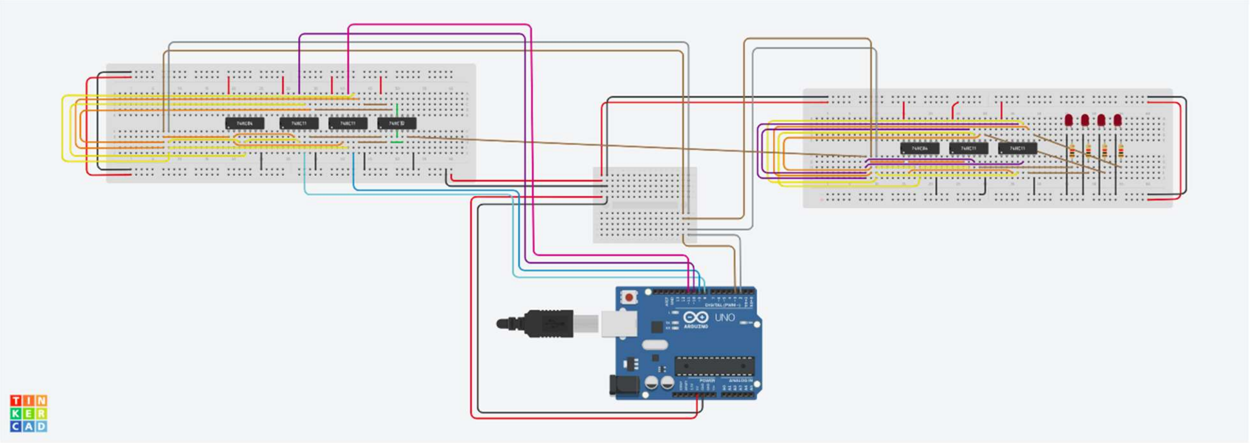


Fig.11 Final Circuit diagram