Project Report on
Implementing MUX based logic and JK synchronous counter

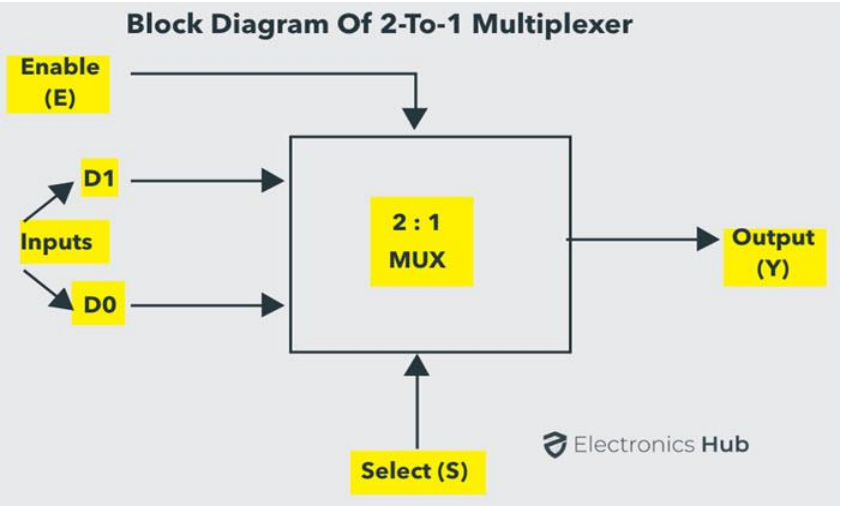
Submitted to:
Prof: Rik Dey.
IIT Kanpur.

By,
Krishna Satpute -22104134
Akash Kumar - 22104006
Neelesh Kumar – 22104064
Kumar Suryamauli Shah-22104052

Implementing MUX based logic

2 to 1 Multiplexer:

- A 2 to 1 multiplexer (MUX) is a combinational logic circuit that selects one of two input signals and forwards it to a single output line, based on a select input signal.
- The basic logic gates used to implement a 2 to 1 MUX are AND, OR, and NOT gates.
- The circuit diagram of a 2 to 1 MUX using logic gates is shown below:



Source- https://www.electronicshub.org/multiplexerandmultiplexing/

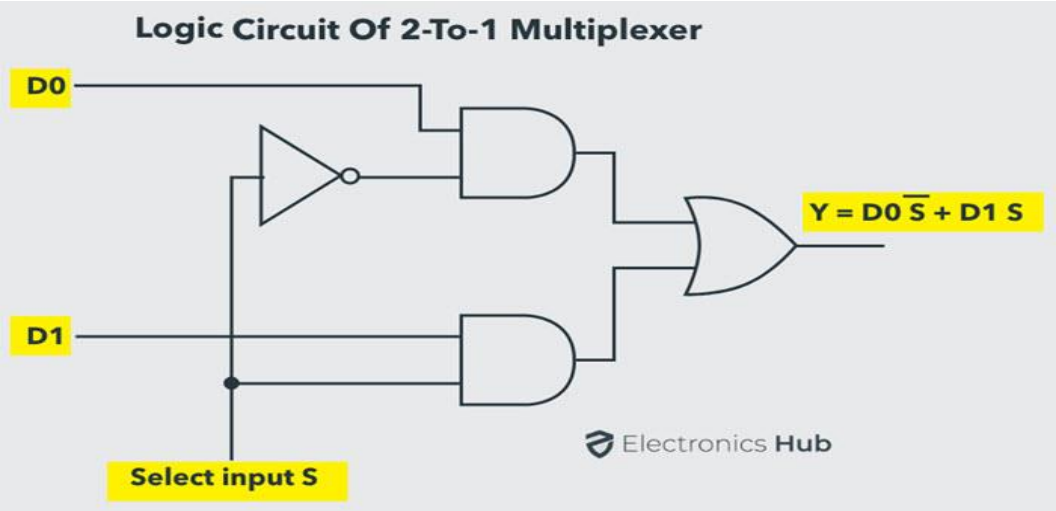
- From the above circuit according to select line input signals are selected and transferred to the output.

- Truth table for 2 to 1 MUX:

Select line	Input D0	Input D1	Output Y
0	0	X	0
0	1	X	1
1	X	0	0
1	X	1	1

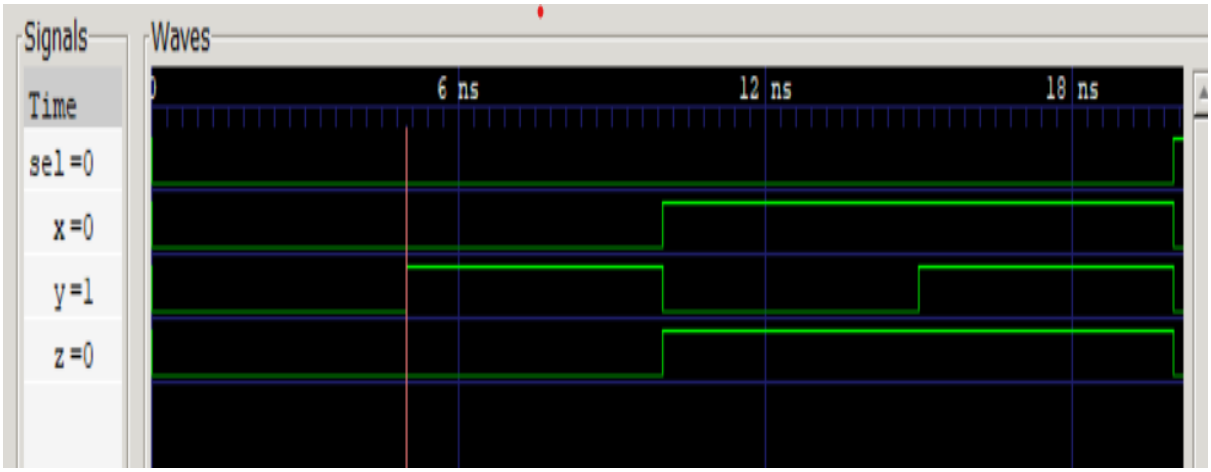
- Implementation using logic gate

- The basic logic gates used to implement a 2 to 1 MUX are AND, OR, and NOT gates



Source- https://www.electronicshub.org/multiplexerandmultiplexing/

- Output Wave for 2 to 1 MUX:



1. At t=0 all the input are sel =0,X=0, Y=0, Z=0
2. At t=5 all the input are sel=0,X=0, Y=1, Z=0
3. At t=10 all the input are sel =0,X=1, Y=0, Z=1
4. At t=15 all the input are sel=0,X=1, Y=1, Z=1
5. At t=20 all the input are sel=1, X=0, Y=0, Z=0

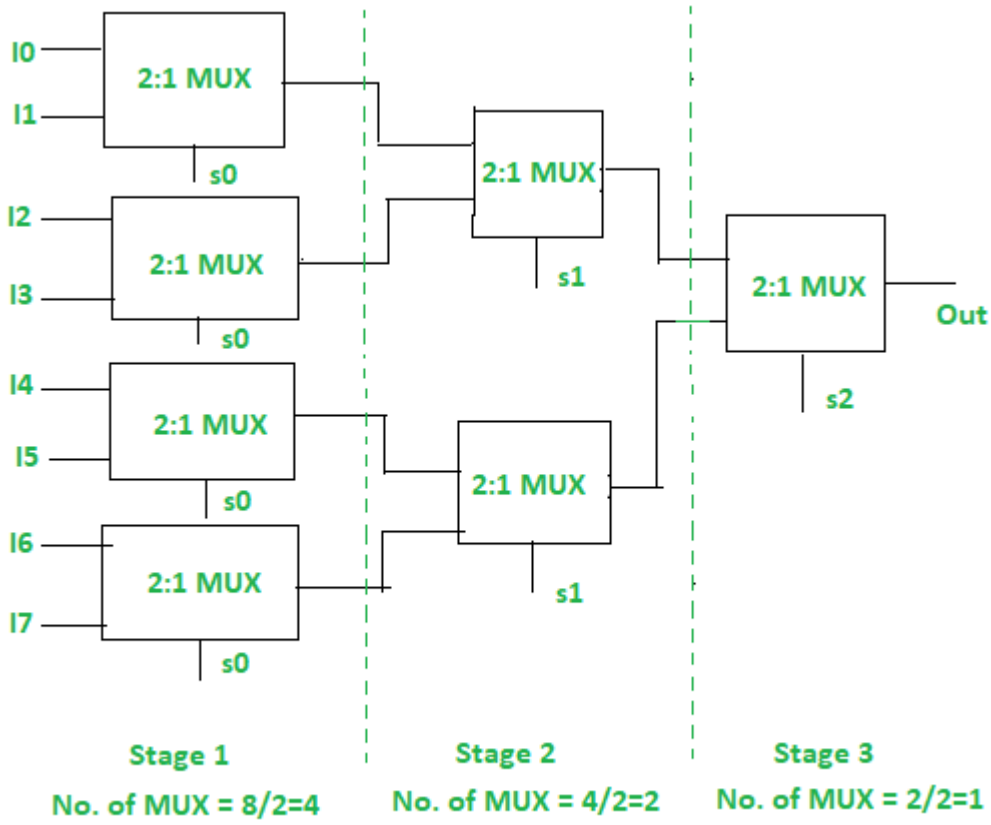
32 to 1 Multiplexer:

- A 32-to-1 multiplexer (mux) is a combinational logic circuit that has 32 input signals, one select signal, and a single output. The output of the mux is one of the 32 input signals, determined by the value of the select signal.

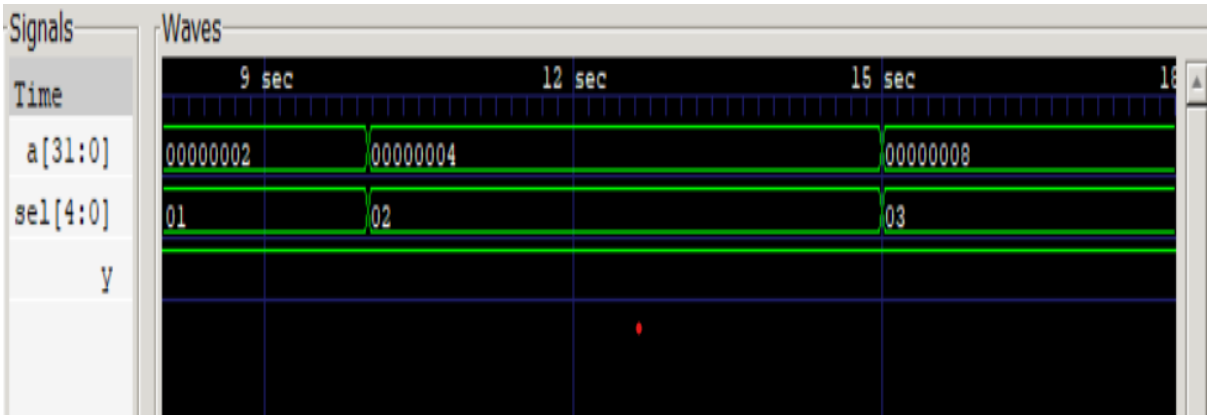
- The select signal is typically a 5-bit signal, since it takes 5 bits to represent 32 possible input selections. Each input signal is connected to one of the inputs of an AND gate, and the select signal is connected to the other input of each AND gate. The outputs of all the AND gates are then connected to the inputs of a 32-input OR gate.

- The select signal is used to enable one of the AND gates, which in turn enables the corresponding input signal to pass through to the output. All other input signals are blocked from passing through to the output, as their corresponding AND gates are disabled. The output of the 32-input OR gate represents the selected input signal.

- The logic diagram for a 32-to-1 mux is shown below:



- Output waveform for given question



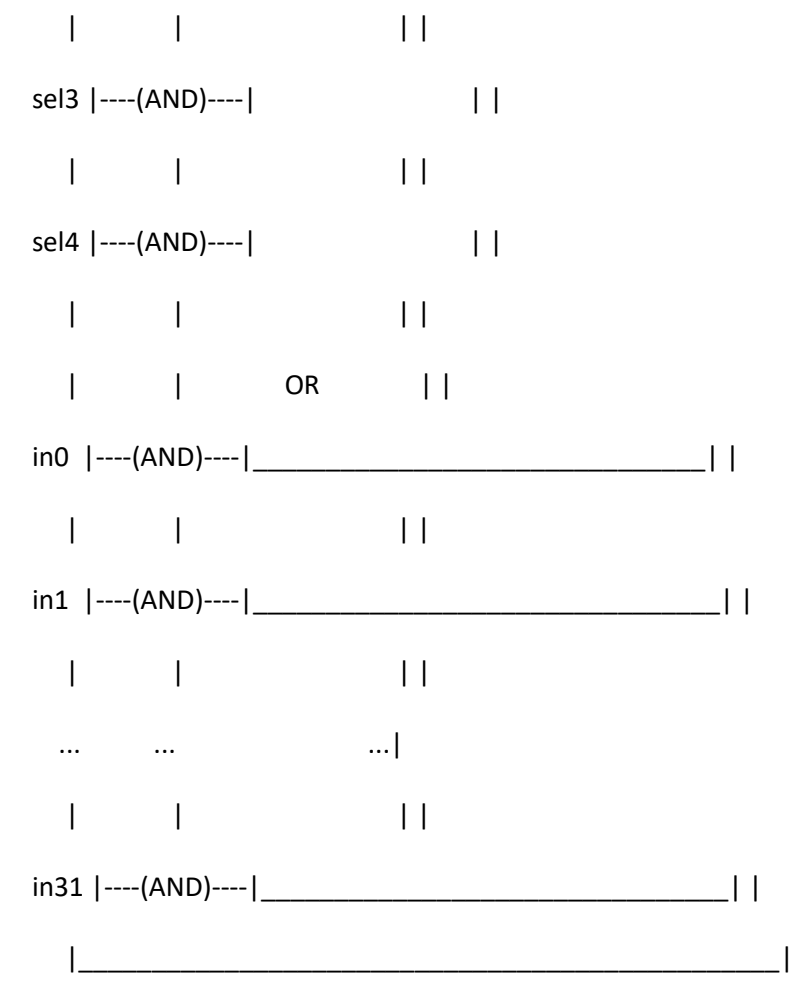
A 32-to-1 multiplexer (mux) is a combinational logic circuit that has 32 input signals, one select signal, and a single output. The output of the mux is one of the 32 input signals, determined by the value of the select signal.

The select signal is typically a 5-bit signal, since it takes 5 bits to represent 32 possible input selections. Each input signal is connected to one of the inputs of an AND gate, and the select signal is connected to the other input of each AND gate. The outputs of all the AND gates are then connected to the inputs of a 32-input OR gate.

The select signal is used to enable one of the AND gates, which in turn enables the corresponding input signal to pass through to the output. All other input signals are blocked from passing through to the output, as their corresponding AND gates are disabled. The output of the 32-input OR gate represents the selected input signal.

The logic diagram for a 32-to-1 mux is shown below:

```
      _____________________________________________
     |          ___________________________________ |
     |          |                  | |
  sel0 |----(AND)----|                  | |
     |          |                  | |
  sel1 |----(AND)----|                  | |
     |          |                  | |
  sel2 |----(AND)----|                  | |
```




```

```

The 5 select signals (sel0-sel4) are used to enable one of the 32 AND gates, which in turn enables the corresponding input signal (in0-in31) to pass through to the output. The output of the 32-input OR gate is the selected input signal.

To implement a 32-to-1 mux, one can use any combination of logic gates such as AND, OR, and NOT gates. The size and complexity of the circuit will depend on the specific implementation and design goals, such as minimizing area, power consumption, or delay.