

Experiment 03

Experiment 3: Implementation of combinational circuits(Realize Half adder and Full adder) and implementation of MUX and DeMUX

--

Roll No.	17
Name	Manav Jawrani
Class	D10-A
Subject	Microprocessor Lab
LO Mapped	LO2: Analyze and design combinational circuits

Aim: Implementation of combinational circuits(Realize Half adder and Full adder) and implementation of MUX and DeMUX

Introduction:

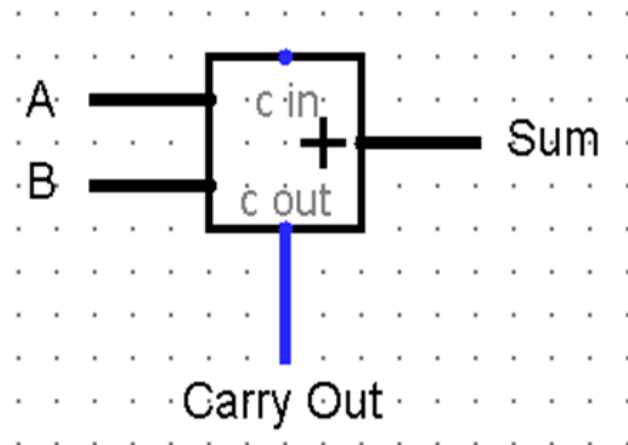
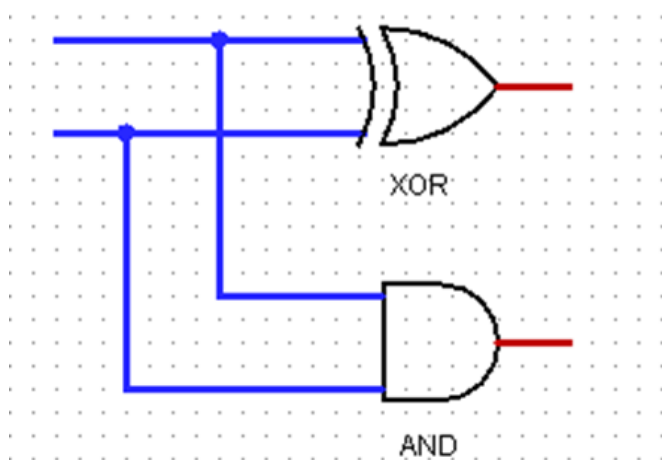
Combinational circuit is a circuit in which we combine the different gates in the circuit, for example encoder, decoder, multiplexer and demultiplexer. Some of the characteristics of combinational circuits are following –

- The output of a combinational circuit at any instant of time, depends only on the levels present at input terminals.
- The combinational circuit does not use any memory. The previous state of input does not have any effect on the present state of the circuit.
- A combinational circuit can have an n number of inputs and m number of outputs.

Theory:

A. Realization of half adder and full adder

1. Half Adder: The Half-Adder is a basic building block of adding two numbers as two inputs and producing two outputs. The adder is used to perform OR operation of two single bit binary numbers. A and B are the two input bits, and 'CARRY' and 'SUM' are two output states of the half adder. Below are the logic diagram and block diagram of Half Adder respectively:



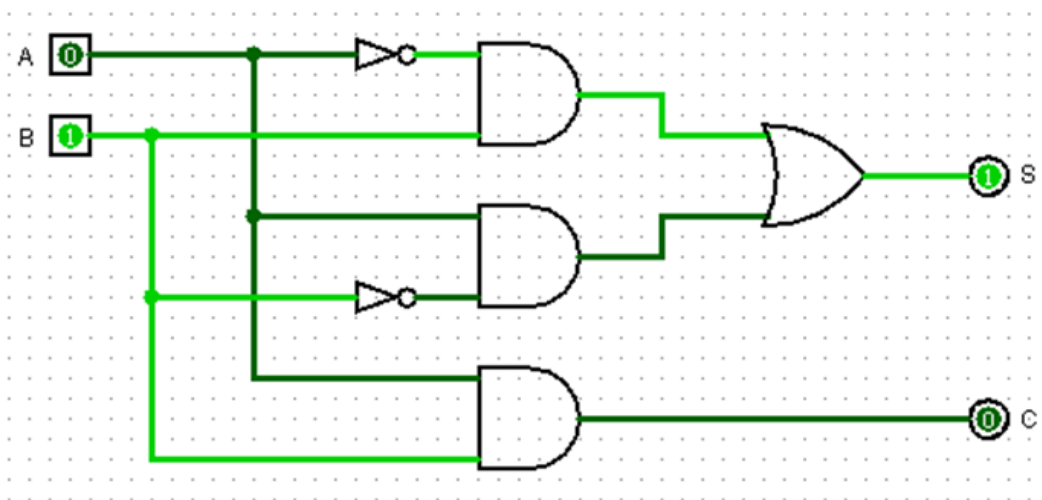
This is called Half Adder because with two inputs, it cannot consider the carry of the previous stage. Thus Half Adder is always used for LSB addition with no carry inputs forwarded.

Half Adder is required for:

1. The ALU (Arithmetic Logic Circuitry) of a computer uses half adder to compute the binary addition operation on two bits.
2. Half adder is used to make full adder as a full adder requires 3 inputs, the third input being an input carry i.e. we will be able to cascade the carry bit from one adder to the other.
3. Ripple carry adder is possible to create a logical circuit using multiple full adders to add N-bit numbers. Each full adder inputs a C-IN, which is the C-OUT of the previous adder. This kind of adder is called Ripple Carry Adder, since each carry bit "ripples" to the next full adder. Note that the first full adder (and only the first) may be replaced by a half adder.

Components required: In Half Adder, it adds two binary digits where the input bits are termed as augend and addend and the result will be two outputs one is the sum and the other is carry. To perform the sum operation, XOR gate is applied to both the inputs, and AND gate is applied to both inputs to produce carry.

Circuit Diagram: Here is the circuit diagram of half adder using basic gates.

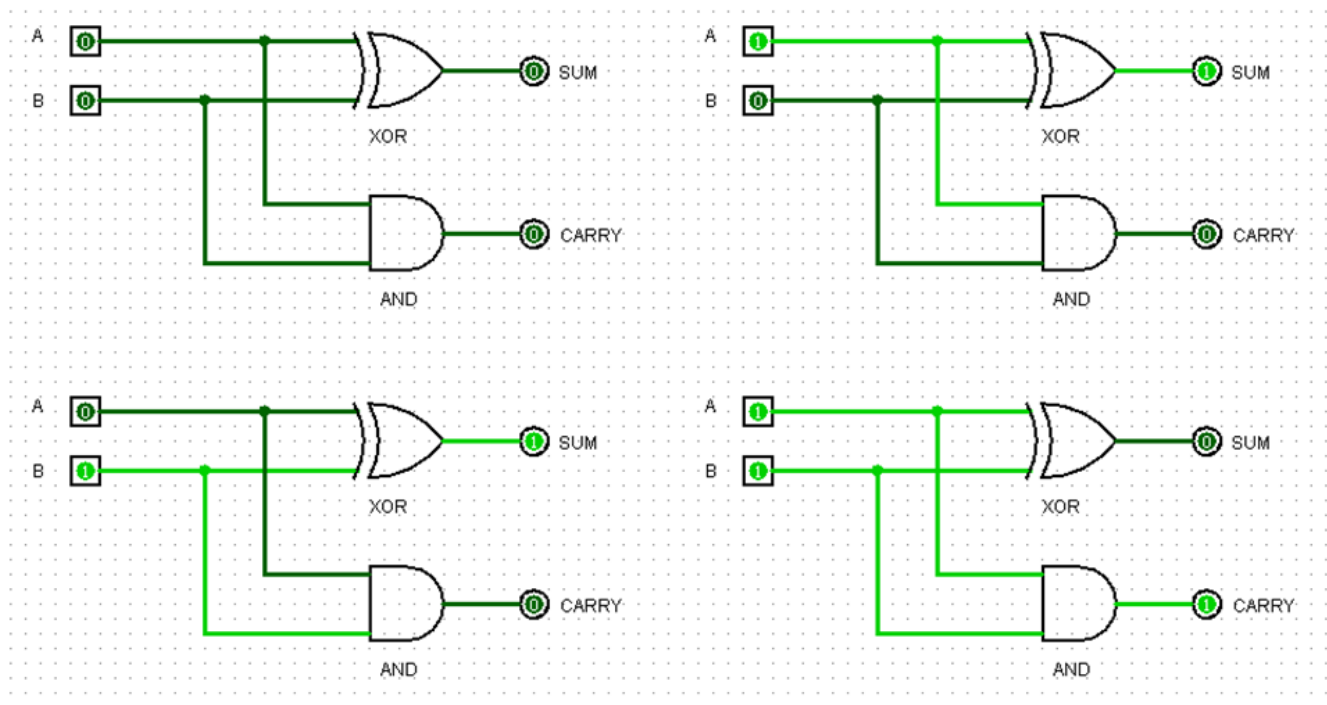


Where, A and B are the inputs given to the Half adder

S (Sum) and C (Carry) are the outputs of Half Adder

Working of Half Adder: Half Adder is a combinational arithmetic circuit that adds two numbers and produces a sum bit (S) and carry bit (C) as the output.

When both inputs are LOW then sum and carry will be logic LOW. If any one input is HIGH then Sum will be logic HIGH and carry will be logic LOW. When both inputs are HIGH then Sum becomes logic LOW and Carry becomes logic HIGH.



Statement for Half Adder:

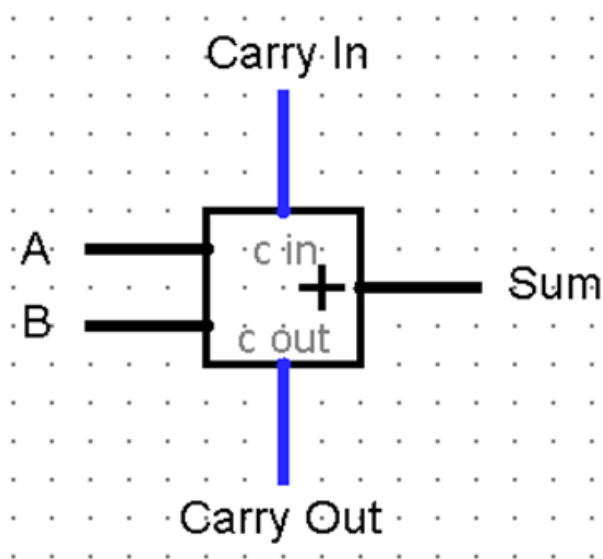
$$\text{Sum} = A' \cdot B + A \cdot B'$$

$$\text{Carry} = A \cdot B$$

Truth Table:

A	B	C (Carry)	S (Sum)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

2. Full Adder: Full Adder is the adder which adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM. Below is the block diagram of Full Adder:



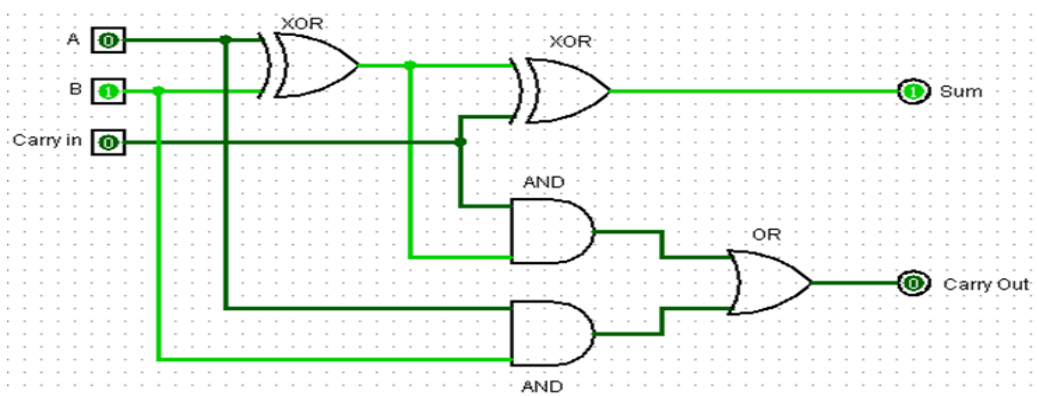
Full Adder is like 3 input adders which can generate the sum and carry.

Full Adder is required for:

1. Full Adder can be a part of Ripple carry adder which adds n-bits at a time. Ripple carry adder is possible to create a logical circuit using multiple full adders to add N-bit numbers. Each full adder inputs a C-IN, which is the C-OUT of the previous adder. This kind of adder is called Ripple Carry Adder, since each carry bit "ripples" to the next full adder.
2. The ALU (Arithmetic Logic Circuitry) of a computer uses full adder to compute the binary addition operation. It can be used for carrying out Multiplication, the dedicated multiplication circuit uses it.
3. Full Adder is used to generate memory addresses inside a computer and to make the Program Counter point to the next instruction, the ALU makes use of this adder.
4. For graphics related applications, where there is a very much need of complex computations, the GPU uses optimized ALU which is made up of full adders, other circuits as well

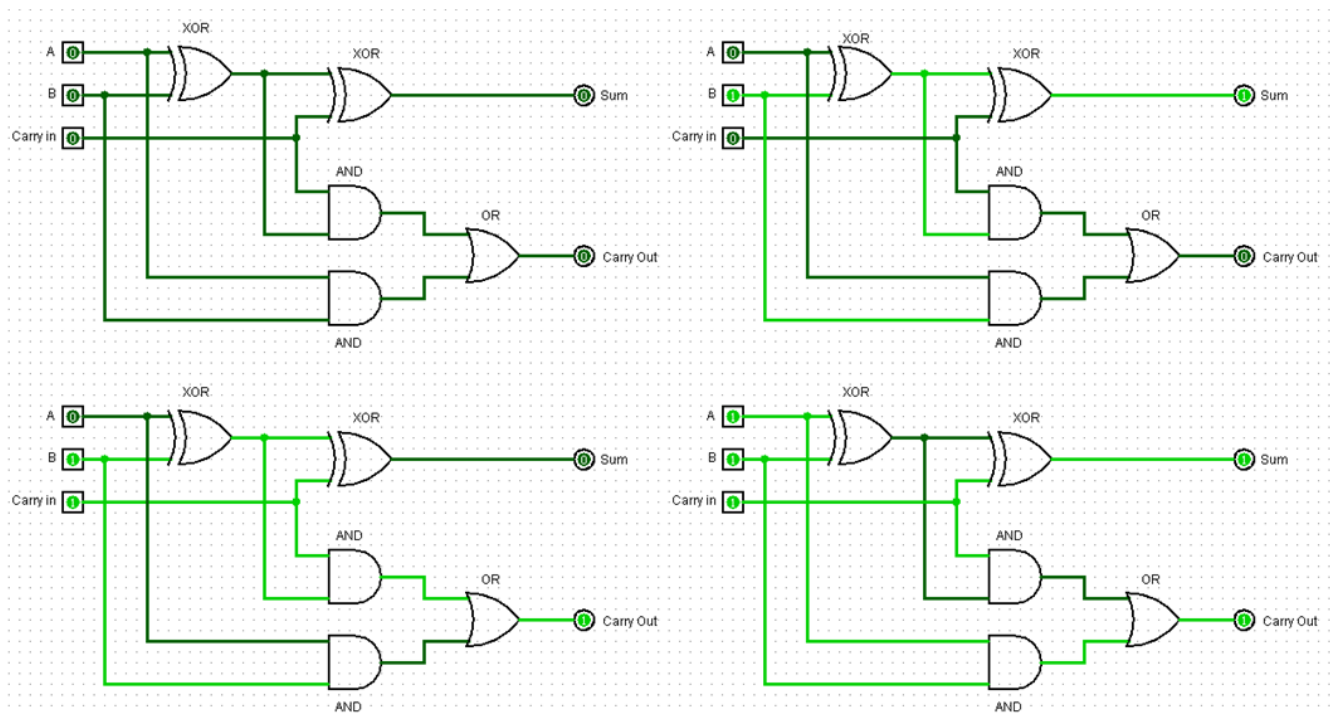
Components required: A full adder adds binary numbers and accounts for values carried in as well as out. To perform the sum operation, it includes two EX-OR gates, an OR gate, and two AND gates for its inputs and outputs.

Circuit Diagram: Here is the circuit diagram of full adder using basic and EX-OR gates.



Working of Full Adder: Full adder is a combinational circuit, which performs the addition of three bits A, B and C-IN. Where, A & B are the two parallel significant bits and C-IN is the carry bit, which is generated from the previous stage.

When both inputs are LOW then sum and carry out will be logic LOW. If any one input is HIGH then sum will be logic HIGH and carry out will be logic LOW. When two inputs are HIGH then sum becomes logic LOW and carry out becomes logic HIGH. When all inputs are HIGH the output sum and carry out will be logic HIGH.



Statement for Full Adder:

$$\text{Sum} = A' \cdot B' \cdot C + A' \cdot B \cdot C + A \cdot B' \cdot C' + A \cdot B \cdot C$$

$$\text{Carry} = A \cdot B + A \cdot C + B \cdot C$$

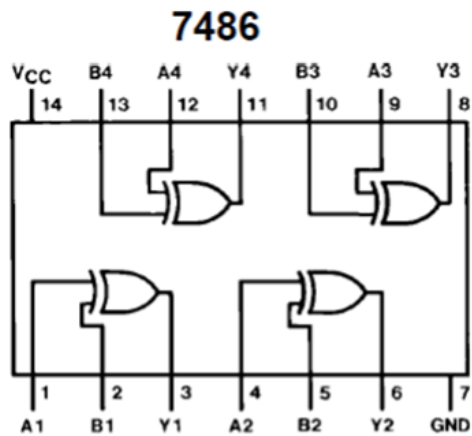
Truth Table:

A	B	Carry In	Sum	Carry Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

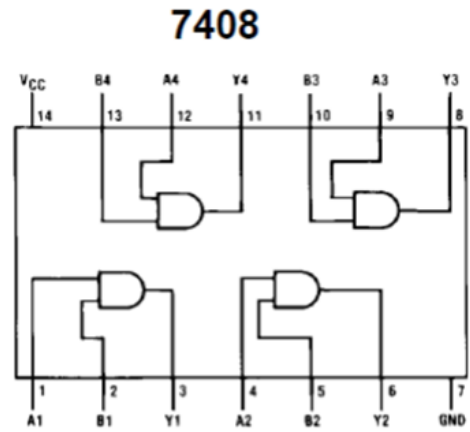
Selection of IC's for Half Adder and Full Adder :

IC for Half Adder: Here XOR gate IC 7486 and Logic AND gate IC 7408 are used to construct the half adder circuit, both are quad 2 input logic gate IC.

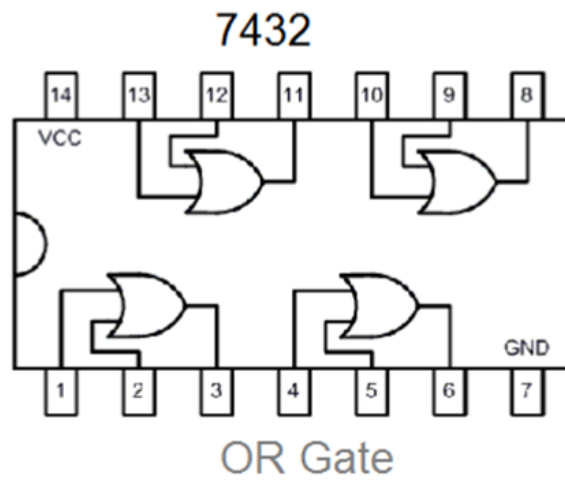
IC for Full Adder: Here XOR gate IC 7486 and Logic AND gate IC 7408 and OR gate IC 7432 are used to construct the full adder circuit, both are quad 2 input logic gate IC.



Exclusive-OR Gate
(xor)



AND Gate

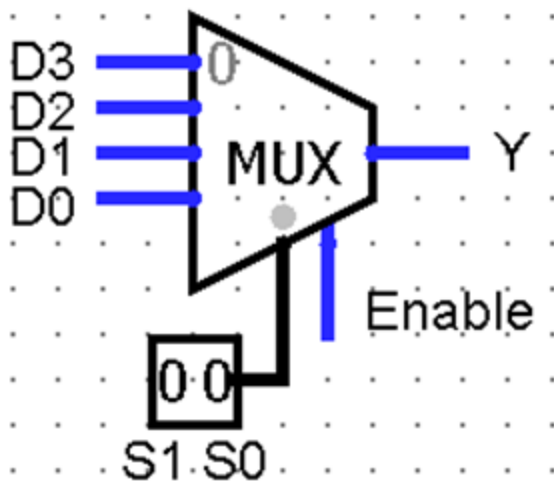


OR Gate

B. Implementation of Multiplexer (MUX) and Demultiplexer (DeMUX)

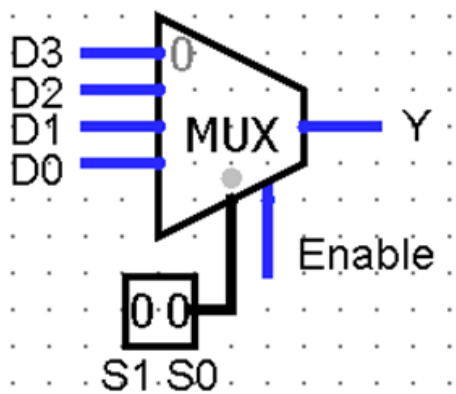
1. Multiplexers: Multiplexer is a combinational circuit that has a maximum of 2^n data inputs, 'n' selection lines and single output line. One of these data inputs will be connected to the output based on the values of selection lines. Since there are 'n' selection lines, there will be 2^n possible combinations of zeros and ones. So, each combination will select only one data input. Multiplexer is also called a Mux.

2:1 Multiplexer: A 2-to-1 multiplexer consists of two inputs D0 and D1, one select input S and one output Y. Depending on the select signal, the output is connected to either of the inputs. Since there are two input signals, only two ways are possible to connect the inputs to the outputs, so one select is needed to do these operations.



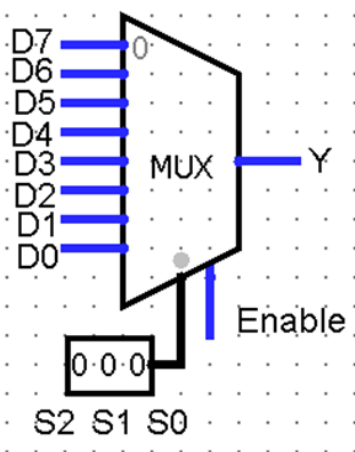
If the select line is low, then the output will be switched to D0 input, whereas if the select line is high, then the output will be switched to D1 input. The figure above shows the block diagram of a 2-to-1 multiplexer which connects two 1-bit inputs to a common destination.

4:1 Multiplexer: A 4-to-1 multiplexer consists of four data input lines as D0 to D3, two select lines as S0 and S1 and a single output line Y. The select lines S1 and S2 select one of the four input lines to connect the output line. The particular input combination on select lines selects one of input (D0 through D3) to the output.

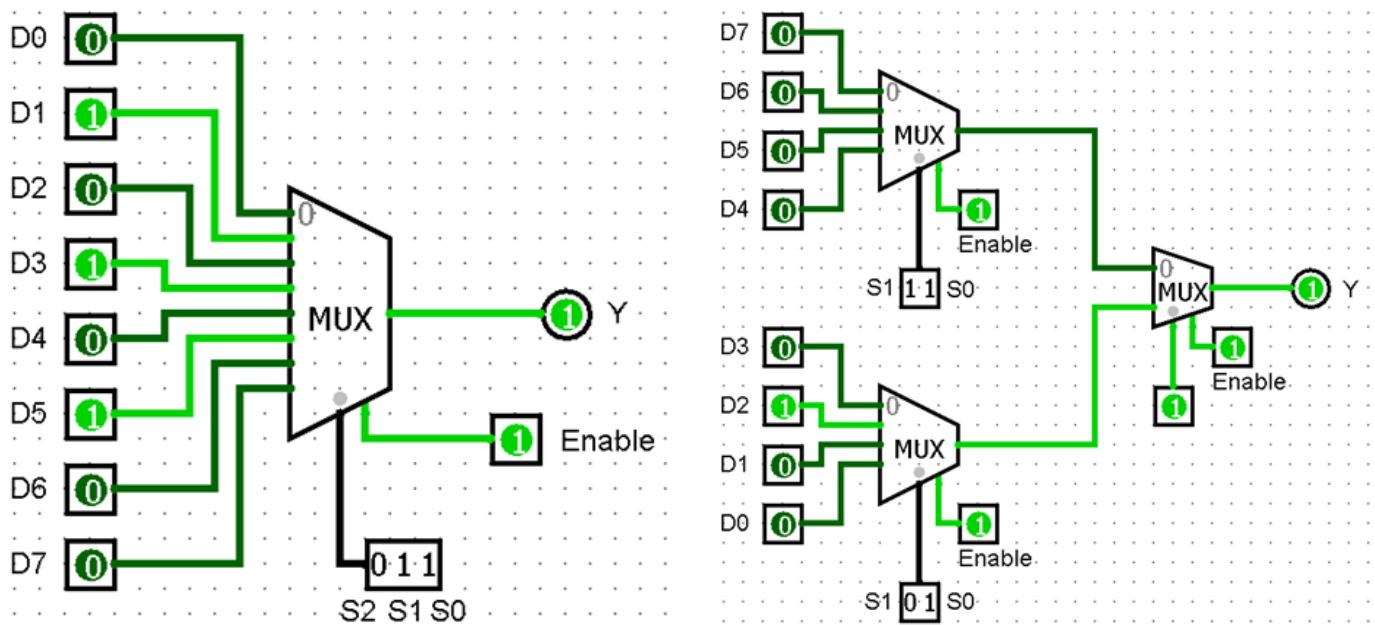


If the select line 0 and 1 is low, then the output will be switched to D0 input, if the select line 0 is high and 1 is low, then the output will be switched to D1 input, if the select line 0 is low and 1 is high, then the output will be switched to D2 input and if the select line 0 and 1 is high, then the output will be switched to D3 input. The figure above shows the block diagram of a 4-to-1 multiplexer which connects two 2-bit inputs to a common destination.

8:1 Multiplexer: An 8-to-1 multiplexer consists of eight data inputs D0 through D7, three input select lines S2 through S0 and a single output line Y. Depending on the select lines combinations, the multiplexer decodes the inputs.



The above figure shows the block diagram of an 8-to-1 multiplexer with input that enables or disables the multiplexer. Since the number data bits given to the MUX are eight then 3 bits ($2^3=8$) are needed to select one of the eight data bits.

Circuit Diagram:

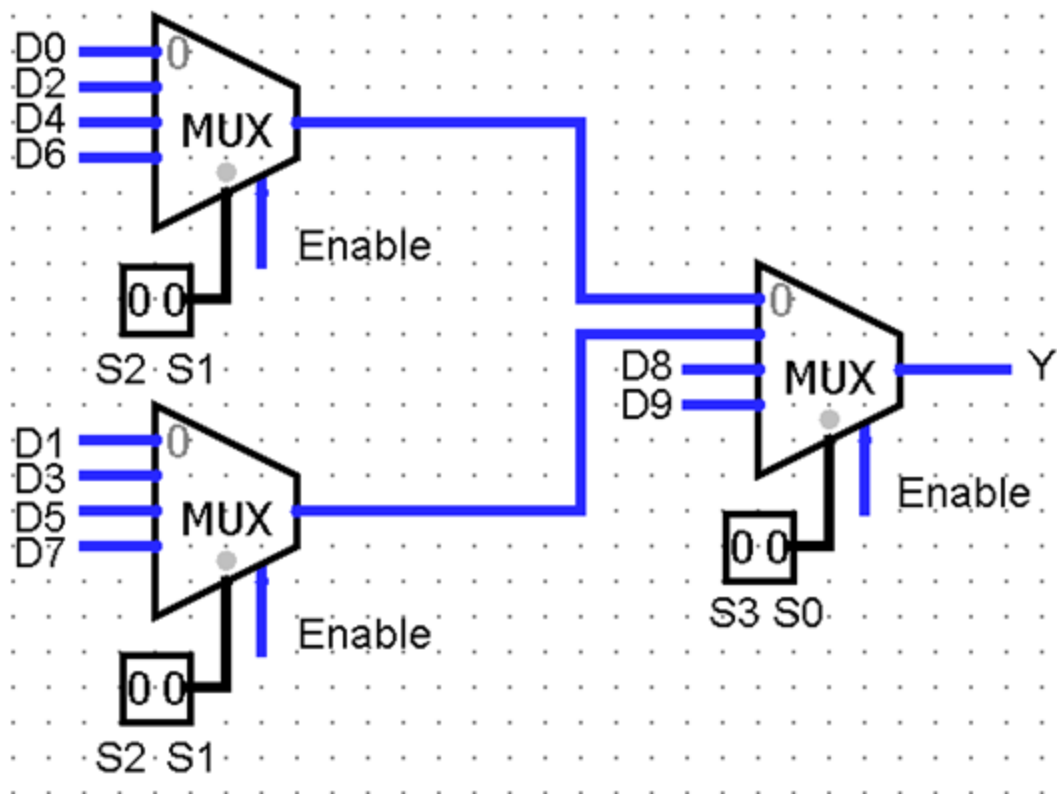
Above circuit diagrams are of 8:1 multiplexer using normal 8:1 MUX and two 4:1 MUXs with a 2:1 MUX respectively.

Truth Table:

S2	S1	S0	Y
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3
1	0	0	D4

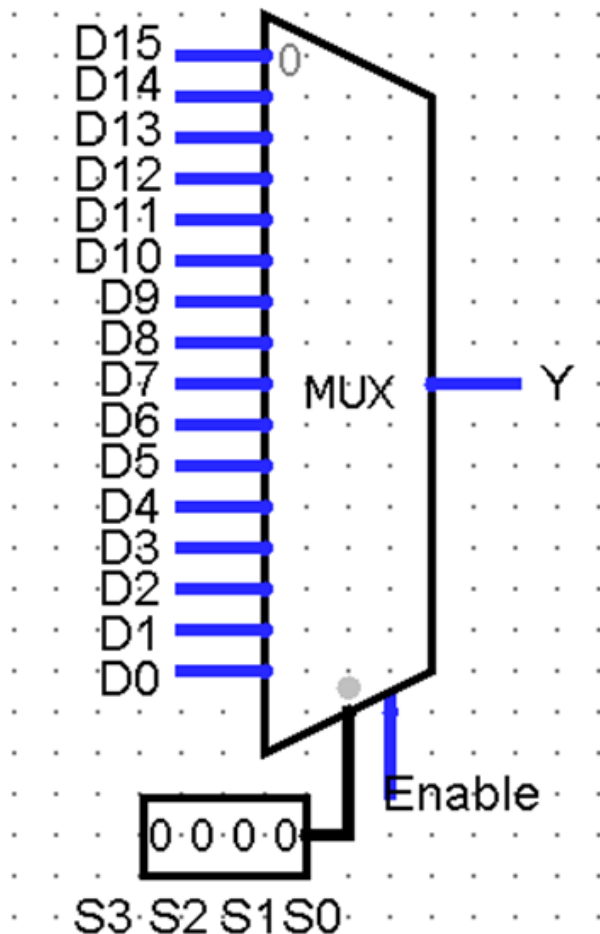
1	0	1	D5
1	1	0	D6
1	1	1	D7

10:1 Multiplexer: A 10-to-1 multiplexer consists of ten data inputs D0 through D9, four input select lines S3 through S0 and a single output line Y. Depending on the select lines combinations, the multiplexer decodes the inputs. The particular input combination on select lines selects one of input (D0 through D3) to the output.

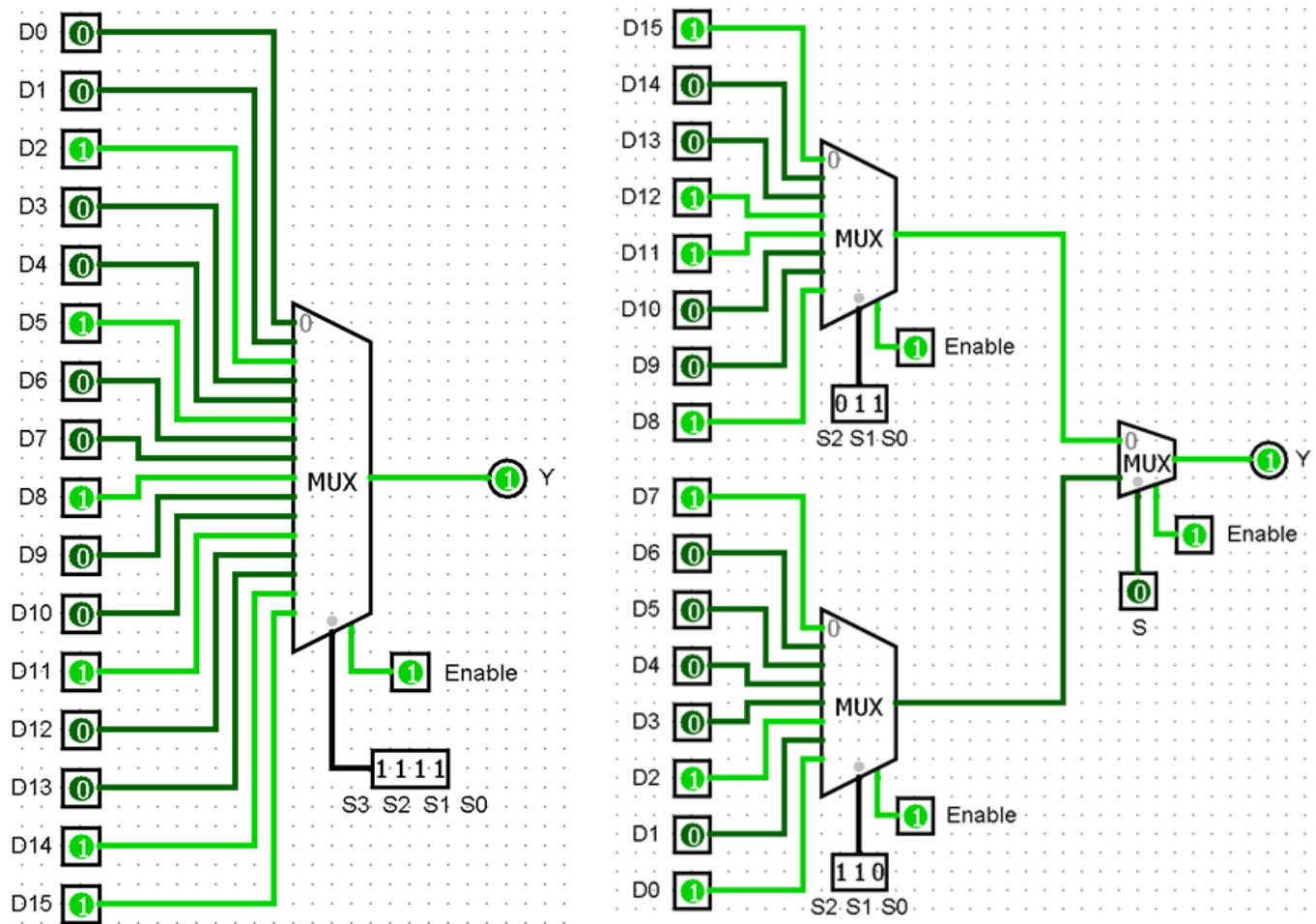


The above figure shows the block diagram of a 10-to-1 multiplexer using three 4-to-1 multiplexers with input that enables or disables the multiplexer.

16:1 Multiplexer: A 16-to-1 multiplexer consists of sixteen data inputs D0 through D15, four input select lines S3 through S0 and a single output line Y. Depending on the select lines combinations, the multiplexer decodes the inputs. The particular input combination on select lines selects one of input (D0 through D15) to the output.



The below figure shows the block diagram of a 16-to-1 multiplexer with input that enables or disables the multiplexer. Since the number of data bits given to the MUX are sixteen then 4 bits ($2^4=16$) are needed to select one of the sixteen data bits.

Circuit Diagram:

Above circuit diagrams are of 16:1 multiplexers using normal 16:1 MUX and two 8:1 MUXs with a 2:1 MUX respectively.

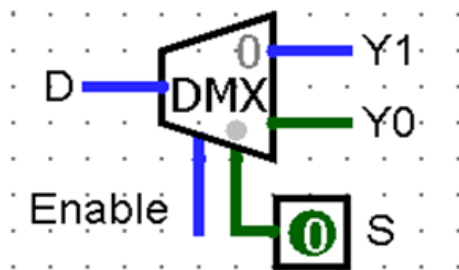
Truth Table:

S3	S2	S1	S0	Y
0	0	0	0	D0
0	0	0	1	D1

0	0	1	0	D2
0	0	1	1	D3
0	1	0	0	D4
0	1	0	1	D5
0	1	1	0	D6
0	1	1	1	D7
1	0	0	0	D8
1	0	0	1	D9
1	0	1	0	D10
1	0	1	1	D11
1	1	0	0	D12
1	1	0	1	D13
1	1	1	0	D14
1	1	1	1	D15

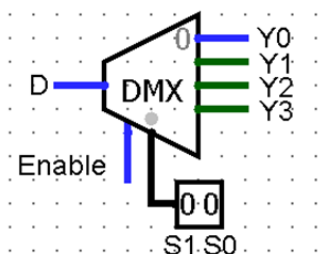
2. Demultiplexers: De-Multiplexer is a combinational circuit that performs the reverse operation of a Multiplexer. It has single input, 'n' selection lines and a maximum of 2^n outputs. The input will be connected to one of these outputs based on the values of selection lines. Since there are 'n' selection lines, there will be 2^n possible combinations of zeros and ones. So, each combination can select only one output. Demultiplexer is also called De-Mux.

1:2 Demultiplexer: A 1-to-2 demultiplexer consists of one input line, two output lines and one select line. The signal on the select line helps to switch the input to one of the two outputs. The figure below shows the block diagram of a 1-to-2 demultiplexer with additional enable input.



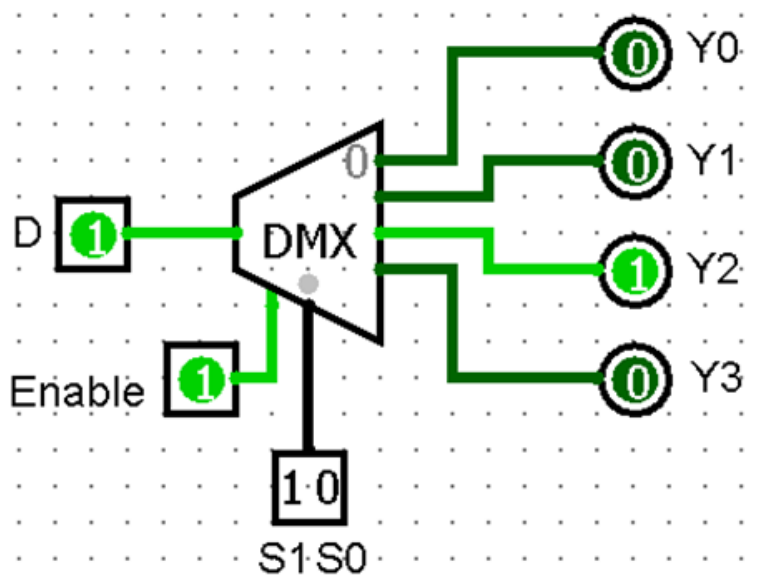
In the above figure, there are only two possible ways to connect the input to output lines, thus only one select signal is enough to do the demultiplexing operation. When the select input is low, then the input will be passed to Y0 and if the select input is high then the input will be passed to Y1.

1:4 Demultiplexer: A 1-to-4 demultiplexer has a single input (D), two selection lines (S1 and S0) and four outputs (Y0 to Y3). The input data goes to any one of the four outputs at a given time for a particular combination of select lines.



In the above figure, when both the select inputs 0 and 1 are low, then the input will be passed to Y0, if the select input 0 is high and 1 is low then the input will be passed to Y1, if the select input 0 is low and 1 is high then the input will be passed to Y2 and if both the select inputs 0 and 1 are high, then the input will be passed to Y3.

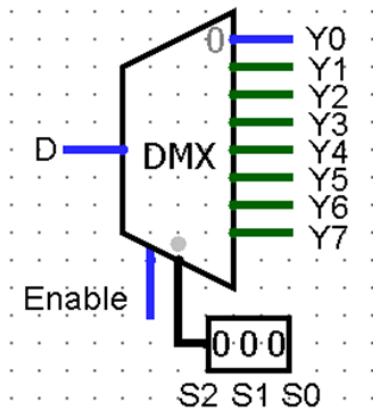
Circuit Diagram: Below is the 1:4 demultiplexer using normal 1:4 DEMUX



Truth Table:

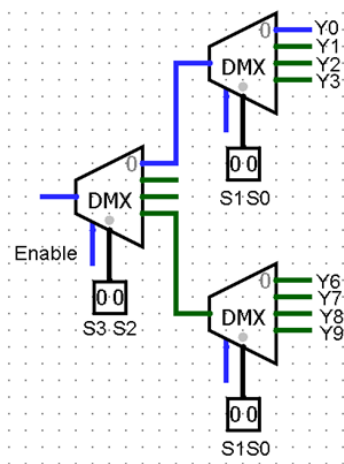
S1	S0	Y
0	0	Y0
0	1	Y1
1	0	Y2
1	1	Y3

1:8 Demultiplexer: A 1-to-8 demultiplexer has a single input (D), three selection lines (S2, S1 and S0) and eight outputs (Y0 to Y7). The input data goes to any one of the eight outputs at a given time for a particular combination of select lines.

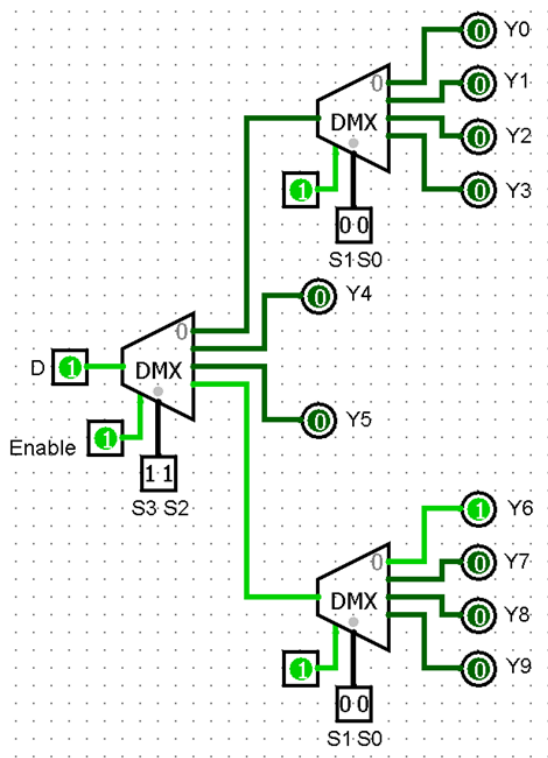


It is also called a 3-to-8 demultiplexer due to three select input lines. It distributes one input line to one of 8 output lines depending on the combination of select inputs.

1:10 Demultiplexer: A 1-to-10 demultiplexer has a single input (D), three selection lines (S2, S1 and S0) and ten outputs (Y0 to Y9). The input data goes to any one of the ten outputs at a given time for a particular combination of select lines.



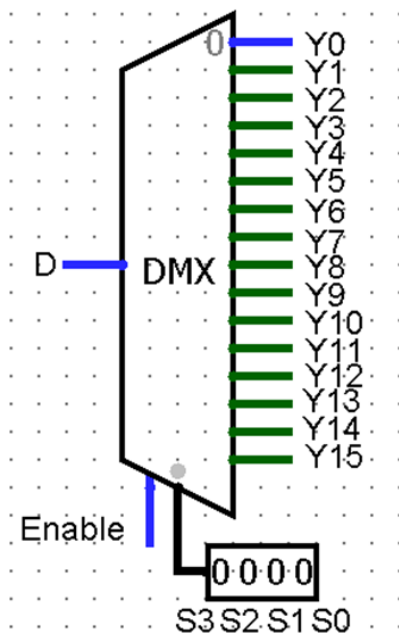
In the above figure, we have implemented a 1-to-10 demultiplexer using three 1-to-4 demultiplexers. It distributes one input line to one of 10 output lines depending on the combination of select inputs.

Circuit Diagram:**Truth Table:**

S3	S2	S1	S0	Y
0	0	0	0	Y0
0	0	0	1	Y1
0	0	1	0	Y2
0	0	1	1	Y3
0	1	0	0	Y4

1	0	0	0	Y5
1	1	0	0	Y6
1	1	0	1	Y7
1	1	1	0	Y8
1	1	1	1	Y9

1:16 Demultiplexer: A 1-to-16 demultiplexer has a single input (D), four selection lines (S3, S2, S1 and S0) and sixteen outputs (Y0 to Y15). The input data goes to any one of the sixteen outputs at a given time for a particular combination of select lines.



We can also implement a 1-to-16 Demultiplexer using 1x8 De-Multiplexers and 1-to-2 Demultiplexers. We know that the 1-to-8 Demultiplexer has a single input, three selection lines and eight outputs. Whereas, 1-to-16 Demultiplexer has single input, four selection lines and sixteen outputs.

Selection of IC's for Multiplexers and Demultiplexers:

IC's for Multiplexers:

IC	Multiplexer
74157	2:1 Multiplexer
74153	4:1 Multiplexer
74151	8:1 Multiplexer
74150	16:1 Multiplexer

IC's for Demultiplexers:

IC	Demultiplexer
74139	4:1 Demultiplexer
74138	8:1 Demultiplexer
74154	16:1 Demultiplexer

Conclusion:

Thus , we have understood the implementation of combinational circuits viz. Half adder and Full adder and also implementation of MUX and DeMUX