**Switching Theory & Logic Design**

**UNIT -3**

**Half Adder:**

With the help of half adder, we can design circuits that are capable of performing simple addition with the help of logic gates.

**Truth table**

|  |  |
| --- | --- |
| **INPUTS** | **OUTPUTS** |
| **A** | **B** | **SUM** | **CARRY** |
| 0 | 0 | 0 | 0 |
| 0   | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

From the equation it is clear that this 1-bit adder can be easily implemented with the help of EXOR Gate for the output ‘SUM’ and an AND Gate for the carry. Take a look at the implementation below



## ****Full Adder:****

## This type of adder is a little more difficult to implement than a half-adder. The main difference between a half-adder and a full-adder is that the full-adder has three inputs and two outputs. The first two inputs are A and B and the third input is an input carry designated as CIN. When a full adder logic is designed we will be able to string eight of them together to create a byte-wide adder and cascade the carry bit from one adder to the next.

TRUTH TABLE:

**INPUTS                 OUTPUTS**

**A             B             CIN         COUT    S**

0              0              0              0              0

0              0              1              0              1

0              1              0              0              1

0              1              1              1              0

1              0              0              0              1

1              0              1              1              0

1              1              0              1              0

1              1              1              1              1

From the above truth-table, the full adder logic can be implemented. We can see that the output S is an EXOR between the input A and the half-adder SUM output with B and CIN inputs. We must also note that the COUT will only be true if any of the two inputs out of the three are HIGH.



Though the implementation of larger logic diagrams is possible with the above full adder logic a simpler symbol is mostly used to represent the operation. Given below is a simpler schematic representation of a one-bit full adder.

 Single-bit Full Adder

With this type of symbol, we can add two bits together taking a carry from the next lower order of magnitude, and sending a carry to the next higher order of magnitude. In a computer, for a multi-bit operation, each bit must be represented by a full adder and must be added simultaneously. Thus, to add two 8-bit numbers, you will need 8 full adders which can be formed by cascading two of the 4-bit blocks. The addition of two 4-bit numbers is shown below.



 Multi-Bit Addition using Full Adder

BINARY SUBTRACTOR:

Subtractor circuits take two binary numbers as input and subtract one binary number input from the other binary number input. Similar to adders, it gives out two outputs, difference and borrow (carry-in the case of Adder). There are two types of subtractors.

1. Half Subtractor
2. Full Subtractor

Half Subtractor :The half-subtractor is a combinational circuit which is used to perform subtraction of two bits. It has two inputs, X (minuend) and Y (subtrahend) and two outputs D (difference) and B (borrow). The logic symbol and truth table are shown below.

![Description: [clip_image003%255B4%255D.jpg]]()

|  |  |
| --- | --- |
| **INPUTS** | **OUTPUTS** |
| **A** | **B** | **DIFFERENCE** | **BORROW** |
| 0 | 0 | 0 | 0 |
| 0   | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

Difference = A'B+AB'=AB
Borrow=A'B

The logic Diagram of Half Subtractor is shown below.



Parallel Adders

1. Introduction

The saying goes that if you can count, you can control. Addition is a fundamental operation for any digital system, digital signal processing or control system. A fast and accurate operation of a digital system is greatly influenced by the performance of the resident adders. Adders are also very important component in digital systems because of their extensive use in other basic digital operations such as subtraction, multiplication and division. Hence, improving performance of the digital adder would greatly advance the execution of binary operations inside a circuit compromised of such blocks. The performance of a digital circuit block is gauged by analyzing its power dissipation, layout area and its operating speed.

4. Parallel Adders

Parallel adders are digital circuits that compute the addition of variable binary strings of equivalent or different size in parallel. The schematic diagram of a parallel adder is shown below

4.1 Ripple-Carry adder

The ripple carry adder is constructed by cascading full adders (FA) blocks in series. One full adder is responsible for the addition of two binary digits at any stage of the ripple carry. The carryout of one stage is fed directly to the carry-in of the next stage.

A number of full adders may be added to the ripple carry adder or ripple carry adders of different sizes may be cascaded in order to accommodate binary vector strings of larger sizes. For an n-bit parallel adder, it requires n computational elements (FA). Figure 4 shows an example of a parallel adder: a 4-bit ripple-carry adder. It is composed of four full adders. The augends’ bits of x are added to the addend bits of y respectfully of their binary position. Each bit



Figure 4: Parallel Adder: 4-bit Ripple-Carry Adder Block Diagram

addition creates a sum and a carry out. The carry out is then transmitted to the carry in of the next higher-order bit. The final result creates a sum of four bits plus a carry out (c4).Even though this is a simple adder and can be used to add unrestricted bit length numbers, it is however not very efficient when large bit numbers are used. One of the most serious drawbacks of this adder is that the delay increases linearly with the bit length. As mentioned before, each full adder has to wait for the carry out of the previous stage to output steady-state result. Therefore even if the adder has a value at its output terminal, it has to wait for the propagation of the carry before the output reaches a correct value.

*Multiplexer*

Multiplexer, we can simply say that a circuit which can deliver single output from multiple inputs. It can often refer as data selector or mux. The inputs to this circuit may be analog or Digital. It is very useful in sending large amount of data over a network with decrease in bandwidth and time. A single pole multi-positioned switch is a plain example of a multiplexer which is not having an electronic circuit or components. But for high speed switching, automatically selecting electronic multiplexers are implemented. Multiplexers that are built from [transistors](http://www.electrical4u.com/bipolar-junction-transistor-or-bjt-n-p-n-or-p-n-p-transistor/) and relays are employed for analog applications. In digital applications, standard logic gates are used to build it. They are also termed as digital multiplexer.



This circuit selects one of the inputs with the help of control signals and delivers that particular input into a single line as output. Therefore, it is also termed as data selector. The figure below shows the pin diagram of multiplexer.



4 to 1 Multiplexer

For understanding the multiplexer further, we are selecting a 4-to-1 Multiplexer. There are many others like 2-to-1, 8-to-1, 16-to-1 multiplexers etc. In this multiplexer, the details are the following:

* Inputs As the name indicates, there are 4 input bits. They are D0, D1, D2 and D3.
* Output The number of output is one and is referred as Y.
* Control Bits Two control bits are used here, A and B. This will decide which input bit should have to be selected. The output clearly depends on this control bits.



If the condition AB = 00, the top most [AND gate](http://www.electrical4u.com/logical-and-gate/) is enabled (shown in figure above). At this time, all the other three AND gates are in disabled condition. So, input bit D0 is selected and transmitted as output. Thus, Y = D0. If the condition AB = 11, every other AND gates are disabled excluding the bottom most AND gate. So, input bit D3 is selected and transmitted as output. Thus, Y = D3. The examples of multiplexers are IC 74153, IC 45352 (4-to-1 multiplexers), IC 74150 (16-to-1 multiplexer)

Applications of Multiplexer

Multiplexers are implemented in several fields where there is a necessity of transmitting large amount of data with use of single line

Computer Memory

In computer, the huge quantity of memory is implemented by means of multiplexers. It also has advantage of reduction in number of copper lines which are used for the connection of memory to other parts in the computer.

Communication System

Multiplexer is implemented in this system to increase efficiency. Using a single [transmission line](http://www.electrical4u.com/electrical-power-transmission-system-and-network/), various types of data (video, audio etc) are transmitted at the same instant.

Telephone Network

Here, the multiple audio signals are brought into a single line and transmitted with the implementation of multiplexer. By this method, the numerous audio signals are made isolated and ultimately the recipient will receive the required audio signals.

Computer System of a Satellite Transmission

Multiplexers are implemented for the data signals to be transmitted from space craft or computer system of satellite to the earth by means of GPS.

Demultiplexer:

*Demultiplexer* is in fact a circuit which can distribute or deliver multiple outputs from a single input. It can often refer as data distributor or demux. It can perform as single input many output switch. The output lines of demultiplexer are ‘n’ in number, select line number is ‘m’ and n = 2m. The control signal or select input code decides the output line which the input has to be transmitted. The demux can also perform as binary to decimal decoder. For this, the data input line should be at logic 1 level and the binary input is given to the select input lines. The corresponding line will give the output. In the designing of multiple combinational circuits, this circuit is really useful. Since, the need of package count is least for demultiplexer. The function of this circuit is the reverse of the [multiplexer](http://www.electrical4u.com/multiplexer/).



The pin diagram of demultiplexer is in figure below.



1 to 4 Demultiplexer

Now, we can select a **1 to 4 Demultiplexer**. There are many other types like 1-to-2, 1-to-8, 1-to-16 demultiplexers etc. The details of this type are the following:

Input: 1 input bit is present. Here it is Data D.

Outputs: The number of outputs is four. They are Y0, Y1, Y2 and Y3.

Control Bits: Two control bits are used here. They are A and B. The input data bit is send to the data bit of the output lines depending on the value of the select input or control bit

If the condition AB = 01; the second [AND gate](http://www.electrical4u.com/logical-and-gate/) from the top is enabled (shown in figure above). At this moment, all the other three AND gates are in disabled condition. So, input bit Data D is delivered to the output. Thus, Y1= Data. When the input is set as 0, Y1 will be low that is 0 and when input is 1 (high), Y1 will be 1(high). Thus, we can say that the Y1 value will directly depend on input D. The other three outputs are 0 (low state). If the condition AB = 10, every other AND gates are disabled excluding the second AND gate from the bottom. So, input bit D is sent to Y2 output. Thus, Y2 = Data D.

Truth Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Data D** | **Control** | **Input** | **Outputs** |
|  | A | B | Y0 | Y1 | Y2 | Y3 |  |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |  |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |  |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |  |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |  |

Applications of Demultiplexer:

Demultiplexers are implemented in several fields where there is a necessity of connecting single source to several destinations.

### Communication System of Demultiplexer

Generally the communication system includes transmission and reception of the signals. For these purposes, the multiplexer in the transmitting end and the demultiplexer in the receiving end have to work simultaneously. Transmission is done with the implementation of multiplexer. The output of the multiplexer; which is the single output and it is given as the input of the demultiplexer. The demultiplexer in the receiver will change this given input to the original signals.

### Arithmetic and Logic Unit (ALU) of Demultiplexer

With the implementation of demultiplexer, the output of the ALU can keep in storage units or multiple registers. Here, the ALU output is given to the demultiplexer as its input. The outputs of the demultiplexer are given to the multiple registers where the data is stored

### Serial to Parallel Converter of Demultiplexer

This converter can give (recreate) the parallel data from the received serial data. Here, the input serial data is set as the input of demultiplexer. This is done at regular intervals. At the control input of demultiplexer, a counter is attached. The data signal is directed to the demultiplexer output with the help of this counter. After every bit of data signals is stored, the demultiplexer output can be extracted. It can be read out in parallel.