**Gandhi Institute of Engineering and Technology, Gunupur**

**Department of Computer Science & Engineering**

**ASSIGNMENT LIST**

**Name of the Course: Computer Organization and Architecture Lab**

**Subject Code: Semester: Fourth**

**Experiment No. 1:** To study the different components of a system.

**Experiment No. 2:** To Study of internal functional Units of Motherboard and Hard Disc Drive.

**Experiment No. 3:** To study of Hard Disc and CD Drive and its functions.

**Experiment No. 4:** To study how to assemble a computer.

**Experiment No. 5:** Design and Simulation of all logic gates through VHDL module and study the wave form.

**Experiment No. 6:** Design and Simulation of XOR and XNOR gates through VHDL module using Local Signals and Study the wave form.

**Experiment No. 7:** Design and Simulation of XOR and XNOR gates through VHDL module using Local Variables and Study the wave form.

**Experiment No. 8:** Design and Implementation of a function using MUX.

**Experiment No. 9:** Design and Implementation of a function using DEMUX.

**Experiment No. 10:** Design and simulation of the 8-bit ALU and perform the following basic logical operation.

1. Addition, 2. Substraction, 3. And, 4. Or, 5. Nand, 6. Nor, 7. Xor, 8. Xnor

**Experiment No. 11:** Design and simulation of Half and Full Adder.

**Experiment No. 12:** Design and simulation of Encoder and Decoder.

**Experiment No -1**

**AIM of theExperiment: - To study the different components of a system.**

The **hardware** components of a computer system are the electronic and mechanical parts.

The **software** components of a computer system are the data and the computer programs.

The major hardware components of a computer system are:

‑Processor

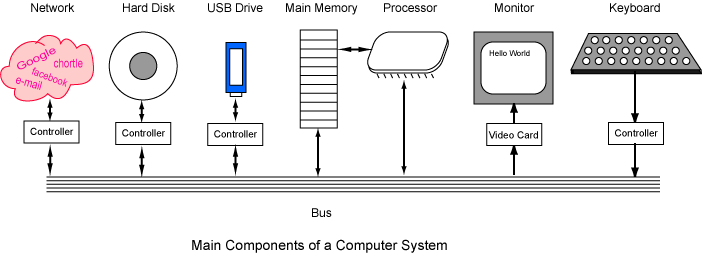
‑Main memory

‑Secondary memory

‑Input devices

‑Output devices

**Hardware Components**



A **bus** is a group of wires on the main circuit board of the computer. Most devices are connected to the bus through a **controller** which coordinates the activities of the device with the bus.

The processor is sometimes called the **Central Processing Unit** or **CPU**. A particular computer will have a particular type of processor, such as a Pentium processor or a SPARC processor.

# Memory

It has only enough memory to hold a few instructions of a program and the data they process. types: main memory, and secondary memory.

Main memory is called **volatile** because it loses its information when power is removed. Secondary memory is usually nonvolatile because it retains its information when power is removed. Main memory is sometimes called **main storage** and secondary memory is sometimes called **secondary storage** or **mass storage**.

Main memory is sometimes called **RAM.**"Random" means that the memory cells can be accessed in any order. However, properly speaking, "RAM" means the type of silicon chip used to implement main memory.

Secondary memory is where programs and data are kept on a long-term basis.

# Input and Output Devices

Input and output devices allow the computer system to interact with the outside world by moving data into and out of the system. An input device is used to bring data into the system.

‑Keyboard

‑Mouse

An output device is used to send data out of the system. Some output devices are:

‑Monitor

‑Printer

‑Speaker

Input/output devices are usually called **I/O** devices. They are directly connected to an electronic module attached to the motherboard called a **device controller**.





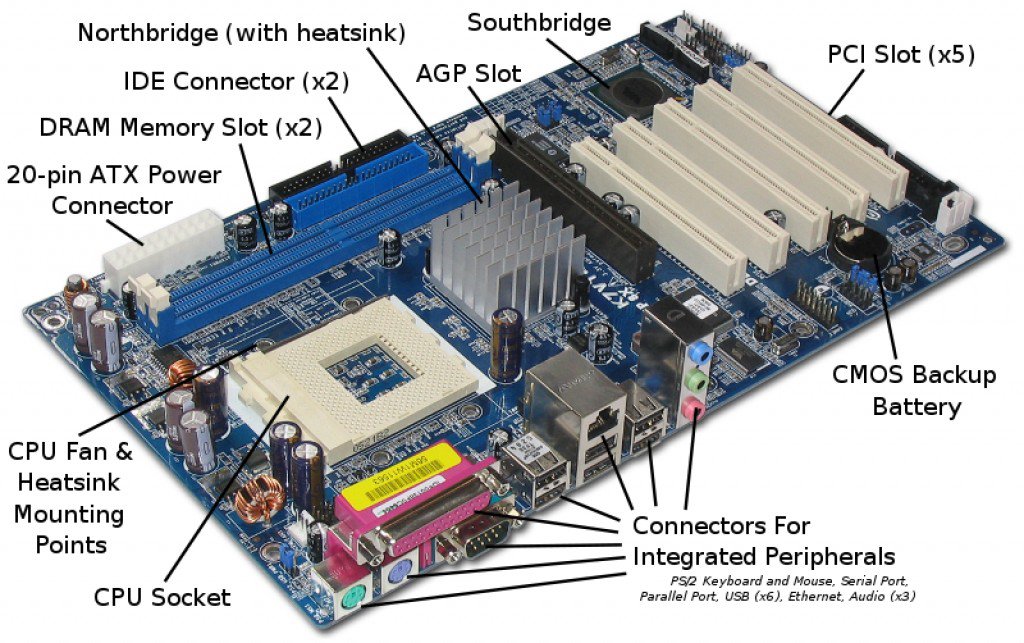
**Experiment – 2**

**AIM :ToStudy of internal functional Units of Motherboard and Hard Disc Drive.**

**Motherboard**

The main printed circuit board in a computer is known as the motherboard.

These include the processor, memory, and expansion slots. The motherboard connects directly or indirectly to every part of the PC.



**Major Motherboard Components and Their Function**

**I/O ports-** This is located in the rear panel of the CPU, this use to connect the outside hardware or peripherals of the computer.

• **PS/2 ports-** These are used to connect the mouse (green color) and the keyboard (purple color)

• **DB 25 female connector (printer port)-**This is use to connect the impact printer, one of that the DOT MATRIX printer, ex. the Epson LX 300 +

• **DB 9 male connector (Serial or com1 port)-**This use to connect a serial device of the computer, such as the serial mouse and external modem.

• **VGA port (built-in)-** This is use to interface the DB cord of the monitor screen or to connect the monitor screen of the computer. The fastest video card versions are based on the standard of the VGA slots, ex: the AGP and PCI’e card.

• **USB 2.0 × 2 ports-** These are the ports of the USB devices of the computer. Attached externally of the system unit, one of that are the USB flash drives. As of now the versions or speed of the USB are the 1.0 and 2.0 compatible.

• **LAN port-** This called as Local Area Network of the computer. This is use to connect the numbers of node or workstations (PC) to form a local area network of the computer with in their area. Using type of communication media such as the network switch and router.

• **Audio Port-** This is use to connect the speaker, mike, and MIDI (Musical Instrument Digital Interface) device of the computer.

**ATX Power connector(4 pin)-** This is use to connect the 4 pin of the power supply unit(PSU) .This is separate of the 20 or 24 pin of ATX power supply to provide DC voltage of computer processor.

**Processor’s ZIF Socket-** Part of MOBO use to hold the processor or CPU chip (Zero Insertion Force).The compatibility of the socket depends of the computer processor type. Such as the AM2 socket of the AMD and the socket 775 of the Intel company.

**Memory slot (DIMM) -** This is called as Dual Inline Memory Module, this is the slot of computer memory or what we called RAM. The memory slot depends of the type of the memory. Such as the PC 133 SDRAM and PC 400 SDRAM-DDR 1.

**ATX Power connector (24 or 28 pin**) - A group of connector of the MOBO. This is separate of the 4 pin power assignments. It provides a DC power output to the system board (SB) from the output of the Power Supply Unit.

**Name of the Vendor**- this is the manufacturer name. This is important to know the manufacturer or vendor. So you can download through the internet site the driver software of the MOBO.

**CMOS/BIOS BATTERY-** In this part, the CMOS/BIOS battery is hold, so that it can provide 3 volts direct current/voltage to the CMOS IC. To preserve the BIOS settings of the computer. Such as the time module, hardware information and settings/parameter, etc.

• **CMOS**-Complementary Metal Oxide Semiconductor

• **BIOS -** Basic Input/output system. Is a set of a program stored in CMOS IC, use to preserve the BIOS configuration of the computer

• **CMOS BATTERY PART NUMBER-** lithium dry cell CR 2032 3 volts.

**IDE connector (Integrated Drive Electronics)-** This connector is use to connect the IDE cable of the HDD(Hard Disk Drive) or CD/DVD ROM to the Mother Board IDE.

**Front panel connectors(F-Panel)-** This is use to connect the Power LED,HDD LED, Power Switch, and Reset switch of the computer system unit(CPU).

**SATA connector-**This is Serial Advance Technology Attachment. The new standard of the IDE connection array of the computer HDD (SATA HDD) to the motherboard system.

**CMOS/BIOS jumper-** This is use to disable the BIOS default setup. Such as to disabled thesupervisor password of the BIOS.

• How to disable password before the BIOS setup

• Turn off CPU

• Change jumper to clear

• Remove the CMOS battery

• Reinsert the BIOS battery

• Change jumper to normal

• Power up computer

• Press “del” or “F2” to enter BIOS setup.

**USB connector (header) -** This is use to connect the USB port of the front panel USB 2.0 x 2 port.

**FDD connector-** The connector of the FDD cable through the system board. Normally there are 34 pins out of it to connect the 3.5” 1.44mb Floppy Disk Drive.

**PCI slots-** a part of the computer motherboard use to connect the I/O cards of the computer. Such as LAN card, WIFI card, USB expansion card, and sound card.

**Motherboard Version number-** here you can get the version of the motherboard driver software. So that you can download it exactly to the manufacturer website.

**CMOS/BIOS IC-** This is the type of BIOS IC being use by the computer MOBO.A set of a program being installed in this IC to preserved the BIOS configuration settings of the computer. Such as the speed of the CPU chip and RAM, etc.

**AMR slot-** This is the AUDIO Modem Riser. Being use to demodulate and modulate the analog audio of the computer.

**PCI express slot-** A part of the computer that hold the computer video card (VGA), the PCI’e card. The latest and the fastest card being installed and most compatible with 3d and 4g gaming of the computer. The old video card type is the AGP or Accelerated Graphic Ports and the standard of the PCI’e.

**North Bridge IC-** Is a passion name as being north of the PCI buses or the PCI slots of the MOBO. The true name of the north bridge is MCI or Memory Controller Hub. Assigned to manage and control the computer memory before the processing of the CPU chip.

**South Bridge IC-** Is a passion name as being south of the PCI buses or PCI slots. The true name of the south bridge is I/O Controller HUB (ICH). It is a type of microchip task to control all the Input and output devices of the computer. Such as the keyboard and mouse. And the monitor screen of the computer that commonly found at the rear back of the CPU or system unit.

**Experiment – 3**

**AIM: To study of Hard Disc and CD Drive and its functions.**

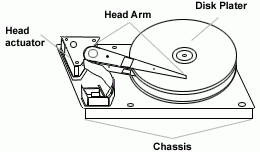
**Hard Disk Drive**

**A hard disk drive, also known as a hard drive or HDD, is used to store data long term.** Data can include the computer's operating system and applications as well as personal files including photographs, documents and music

.

**HDD Architecture**

Most hard disk drives consist of spinning platters of aluminum, glass or ceramic that are coated with a magnetic media.



**HDD Capacity and Size Information**

When purchasing a hard disk drive, the term megabytes, gigabytes or terabytes may be confusing terms. The following table gives you an example of each of these terms and how they compare to other sizes.

**Term Equal to**

Bit 0 or 1

Kb(Kilobit) 1,024 bits

Byte 8 bits (approximately one character in a Word document)

KB(Kilobyte) 1,024 bytes

MB(Megabyte) 1,024 Kilobytes or 1,048,576 Bytes

GB(Gigabyte) 1,024 Megabytes or 1,073,741, 824 Bytes

TB(Terabyte) 1,024 Gigabytes or 1,099,511,627,776 Bytes

PB(Petabyte) 1,024 Terabytes or about 1,000,000,000,000,000 Bytes

EB(Exabyte) 1,024 Petabytes or about 1,000,000,000,000,000,000 Bytes

ZB(Zeta byte) 1,024 Exabyte or about 1,000,000,000,000,000,000,000 Bytes

YB(Yottabyte) 1,024 Zeta bytes or about 1,000,000,000,000,000,000,000,000 Bytes

The capacity of a hard drive ranges from last time 10 MB to the currently up to 1 TB size or whatever latest. However, most of the available HDD on the market today are starting from 80 GB and higher (bigger).

**HDD Brand, Speed and Type**

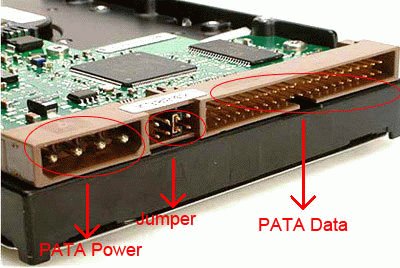
‑Brand: Seagate, Western Digital, Samsung, Hitachi, Maxtor, IBM, etc.

‑Speed: 5400 rpm, 7200 rpm and 10000 rpm (rotations per minute)

‑Type: Internal (IDE / SATA) and External (SCSI / USB)

**HDD Interfaces (Types)**

**IDE (Integrated Device Electronics) / PATA (Parallel Advanced Technology Attachment)**  
This is generally the most common interface used with old hard drives and is generally the easiest installation. When installing these types of hard drives ensure that the jumpers are correctly configured. If you have two devices connected to one IDE controller one must be set to master and the other must be set to slave.

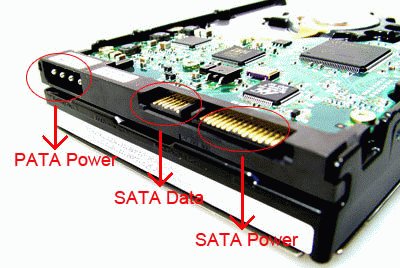


**SCSI (Small Computer System Interface)**

These require a SCSI Host adapter card connected into the system. These cards are mostly PCI bus cards. Some computer manufacturers may have the SCSI port built onto the Main board. Nowadays, most desktop computers did not use the SCSI HDD and it's more commonly found on the server PCs. SCSI HDD provide two major advantages: improved performance relative to IDE and SATA in multitasking, multiuser environments, and the ability to daisy-chain many drives on one computer.

**SATA (Serial Advanced Technology Attachment)**

This is the latest high-speed type of hard drive connectors that can transfer data at a blistering rate of 300Mb/s. Most of the latest hard drives are using this type of interface as there are faster than old IDE interface.



**USB (Universal Serial Bus)**

The Universal Serial Bus (USB) standard is a specification for an I/O connection that can support many kinds of peripheral devices including external HDD. The external HDD is make up using an internal HDD with a case with USB connection.

**SATA socket, power connector and data cable**

If you planning to buy a new SATA hard drive, make sure that your motherboard have the SATA connectors/sockets as shown below.



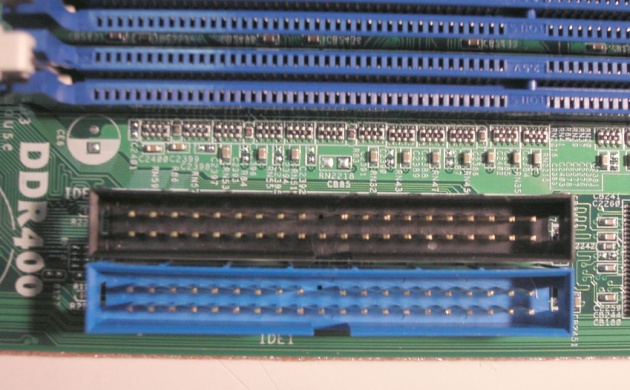
Also, you need to have a SATA power connector from the power supply unit. Else, you also can get a separate SATA power connector as an add-on to the power supply unit. This power connector will connect to the SATA HDD power connector.

Beside the power connector, you need another SATA data cable as show below. One end of the cable is connect to SATA HDD data connector and the other end connect to the main board SATA connectors/sockets.



**IDE (PATA) Socket and data cable**

Normally a computer main board has two 40-pin IDE sockets and one 32-pin FDD socket. Two 40-pin IDE sockets will connect to HDD and CD / DVD drive respectively using the IDE data cable. The only 32-pin FDD Socket is connect to the FDD drive.



**CD Drive**

Compact Disc Read Only Memory. A compact disc format that is used to hold text, graphics, and hi-fi stereo sound. The disc is almost the same as the music CD, but uses different tracks for data. The music CD player cannot play CD-ROM discs, but CD-ROM players may be able to play music CD discs and have jacks for connection to an amplifier and/or earphones. A CD-ROM player is cabled to and controlled by a card that is plugged into one of the PCs expansion slots



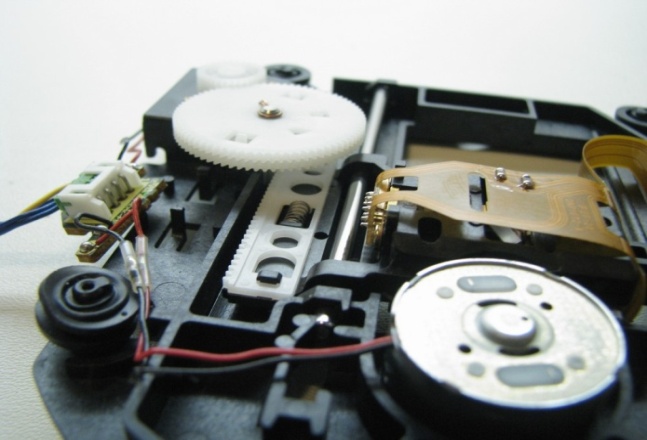
CD-ROM drives are made by many different computer hardware manufacturers.

The lens of your CD-ROM drive is responsible for reading CD data, which it does while the disc is spinning. If the lens of your CD-ROM drive becomes dirty.



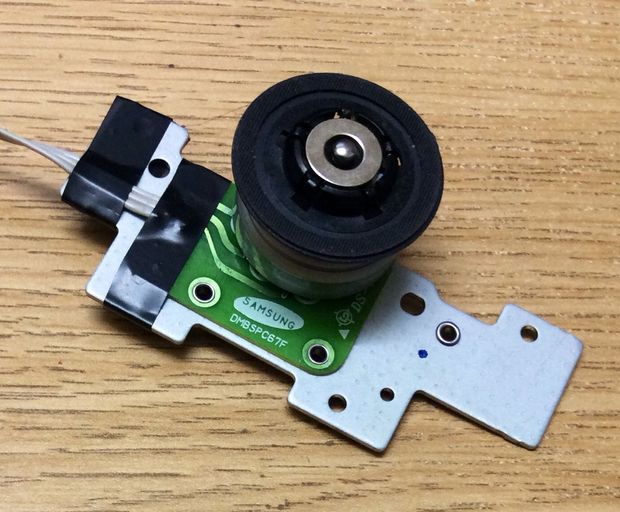
## Head Actuator

The lens of a CD-ROM driver is positioned on a mechanical actuator. This moves the lens back and forth, allowing it to read data from the surface of a CD. This is controlled by an internal servo system and a dedicated microcontroller.



## Spindle Motor

The spindle motor is responsible for spinning the CD when it is in your CD-ROM drive. The speed of a CD-ROM drive’s spindle motor is dependent on the area of the CD that is currently being read. When reading data from the inside or center of a CD, the spindle motor runs faster. As the laser starts to read data near the outside edge of a CD, the spindle motor runs slower.



## Disc Loader

Disc-loading mechanisms of CD-ROM drives vary depending on the manufacturer. The most common disc-loading mechanism in use today is the tray loader. This is a simple plastic tray that slides out of the drive, allowing you to insert the CD into the drive. The tray keeps the disc in place while it is in the drive.



## Drive Connectors

A four-pin power connector is located on the back of every CD-ROM drive; this is the same type of connector found on hard drives and most other internal devices. These drives also have an additional connector, depending on whether the drive is IDE or SCSI-based. A three- or four-wire cable is also on most CD-ROM drives, allowing for direct connection to your computer’s sound card.

## 

## Internal Logic Board: The logic board of a CD-ROM drive controls the operations of the drive itself. The logic board has no serviceable parts.



## Outputs and External Controls

Many CD-ROM drives are equipped with a headphone jack on the face of the drive, allowing you to easily plug in a pair of headphones. Other common controls include Start and Stop buttons and volume control dials.

## Drive Enclosure: A metal case is used to protect all of the components of a CD-ROM drive. Opening the drive enclosure may make it prone to damage.



**Experiment – 4**

**AIM: To study how to assemble a computer**

**Install the Power Supply**

Power supply installation steps include the following:

1. Insert the power supply into the case

2. Align the holes in the power supply with the holes in the case



**Attach Components to the Motherboard**

**CPU on Motherboard**

• The CPU and motherboard are sensitive to electrostaticdischarge.

• The CPU is secured to the socket on the motherboard with alocking assembly.

• CAUTION: When handling a CPU, do not touch the CPUcontacts.



**Thermal compound**

• It helps to keep the CPU cool.

• To install a used CPU, clean it and the base of the heat sinkwith isopropyl alcohol to remove the old thermal compound.



**Heat Sink/Fan Assembly**

• The Heat Sink/Fan Assembly is a two-part cooling device.

• The heat sink draws heat away from the CPU.



**Install RAM**

• RAM provides temporary data storage for the CPU and should be installed in the motherboard before the motherboard is placed in the computer case.



**The Motherboard**

After installing the previous components themotherboard is now ready to install in the computer case.Plastic and metal standoffs are used to mount the motherboard and to prevent it from touching the metal portions of the case.



**Install Internal Drives**

Drives that are installed in internal bays are calledinternal drives.A hard disk drive (HDD) is an example of an internal drive.

**HDD installation steps:**

1. Position the HDD so that it aligns with the 3.5-inch drive bay.

2. Insert the HDD into the drive bay so that the screw holes in the drive line up with the screw holes in the case.

3. Secure the HDD to the case using the proper screws.



**Install Drives in External Bays**

Drives in external bays allow access to the media without opening the case.

Some devices that are installed in this type of drives are:

• An optical drive is a storage device that reads and writes information to CDs or DVDs.

• A floppy disk drive (FDD) is a storage device that reads and writes information to a floppy disk.



**Install Adapter Cards**

Adapter cards are installed to add functionality to a computer.

Some examples of these adapters are:

• A NIC which enables a computer to connect to a network. It uses PCI and PCIeexpansion slots on the motherboard.

• A wireless NIC which enables a computer to connect to awireless network. Wireless NICs use PCI and PCIe expansion slots on the motherboard. Some wireless NICs are installed externally with a USB connector.

• A video adapter card is the interface between a computer and a display monitor. An upgraded video adapter card can provide better graphic capabilities for games and graphic programs.Video adapter cards use PCI, AGP, and PCIeexpansion slots on the motherboard.

**Connect Internal Cables**

Power cables are used to distribute electricity from the power supply to the motherboard and other components.



**ATX AUX SATA Molex Berg**

**Connect Internal Cables**

Data cables transmit data between the Motherboard and storage devices, such as hard drives. Some of examples of this type of cables are:

• PATA cable

• SATA cable

• Floppy drive data cable

Additional cables connect the buttons and link lights on the front of the computer case to the motherboard.

**Complete Physical Installation**

Now that all the internal components and thepower supply have been installed and connected to the motherboard, the following tasks should be completed:

•**Re-Attach the side panels:**Most computer cases havetwo panels, one on each side. Some computer caseshave one three-sided cover that slides down over thecase frame.

•**Connect External Cables:**These cables are normallyconnected to the back of the computer. Here are somecommon external cable connections: Monitor, Keyboard, Mouse, USB, Ethernet Power.

**Boot Computer for the First Time**

When the computer is booted, the basic input/output system (BIOS) will perform a power-on self test (POST) to check on all of the internal components.

The BIOS contains a setup program used to configure settings for hardware devices. The configuration data is saved to a special memory chip called a complementary metal-oxide semiconductor (CMOS).

POST checks to see that all of the hardware in thecomputer is operating correctly. If a device is

malfunctioning, an error or a beep code alerts thetechnician that there is a problem.

**Experiment No – 5**

**AIM: To Design and Simulation of all logic gates through VHDL Module and study the waveform using process.**

**NOT GATE**

entitynotgate is

Port ( X : in STD\_LOGIC;

Y : out STD\_LOGIC);

endnotgate;

architecture Behavioral of notgate is

beginprocess(X)

begin

if(X='0') then

Y<='1';

else

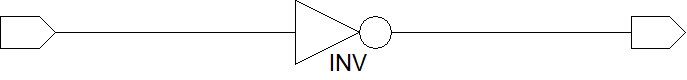
Y<='0';

end if;

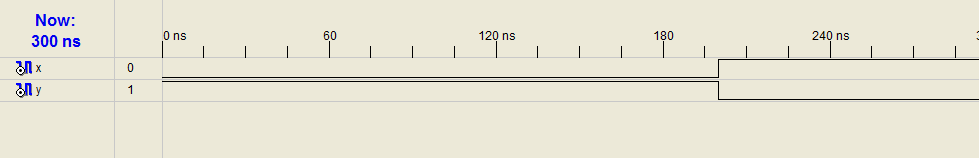
end process;

end Behavioral;

**RTL Schematic**

****

**WAVE FORM**



**AND GATE**

entityand\_gate is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

endand\_gate;

architecture Behavioral of and\_gate is

begin

process(X,Y)begin

if(X='1' and Y='1') then

Z<='1';

else

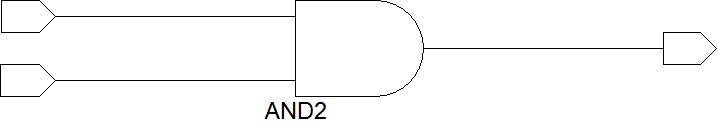
Z<='0';

end if;

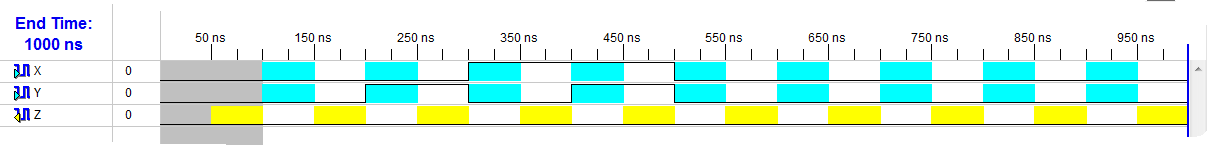
end process;

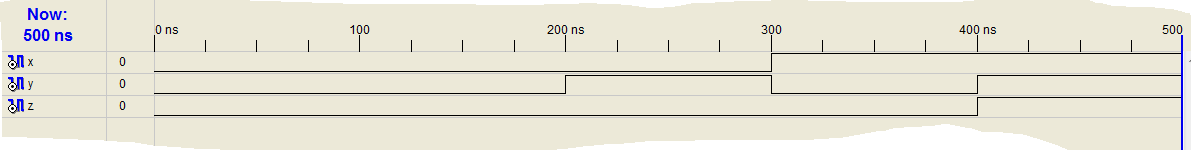
end Behavioral;

**RTL Schematic**

****

**WAVE FORM**

****



**OR GATE**

entityor\_gate is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

endor\_gate;

architecture Behavioral of or\_gate is

begin

process(X,Y)begin

if(X='0' and Y='0') then

Z<='0';

else

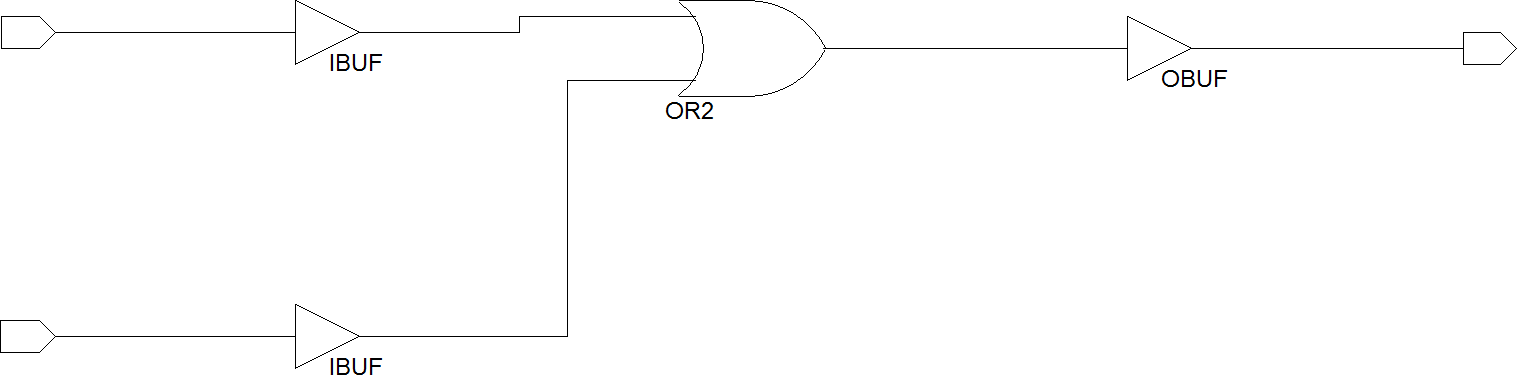
Z<='1';

end if;

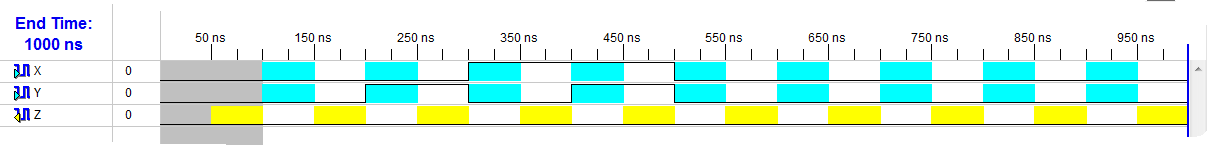
end process;

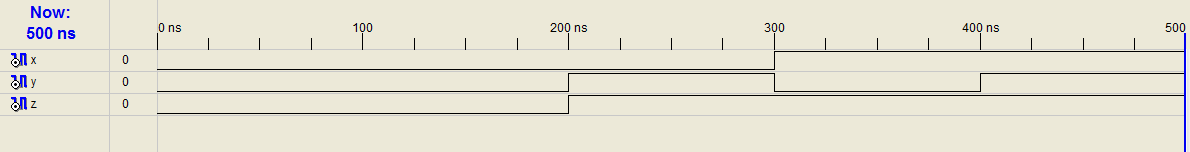
end Behavioral;

**RTL Schematic**

****

**WAVE FORM**

****



**NAND GATE**

entitynand\_gate is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

endnand\_gate;

architecture Behavioral of nand\_gate is

begin

process(X,Y)begin

if(X='1' and Y='1') then

Z<='0';

else

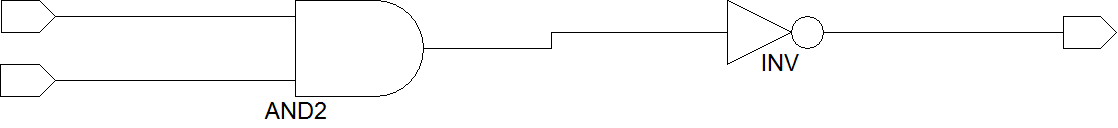
Z<='1';

end if;

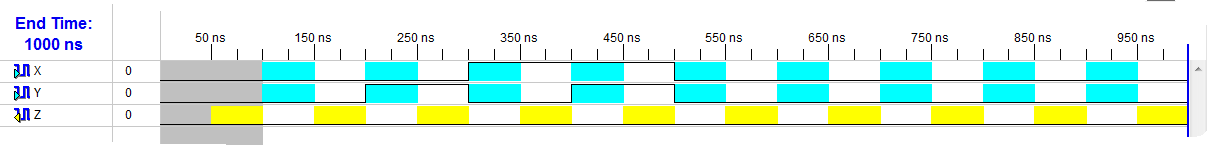
end process;

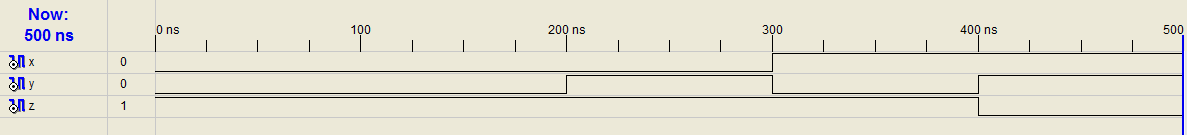
end Behavioral;

**RTL Schematic**

****

**WAVE FORM**

****



**NOR GATE**

entitynor\_gate is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

endnor\_gate;

architecture Behavioral of nor\_gate is

begin

process(X,Y)begin

if(X=’0' and Y='0') then

Z<='1';

else

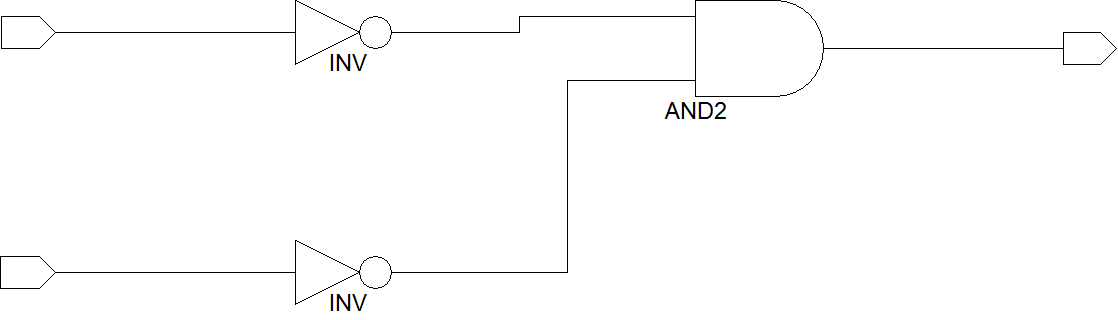
Z<='0';

end if;

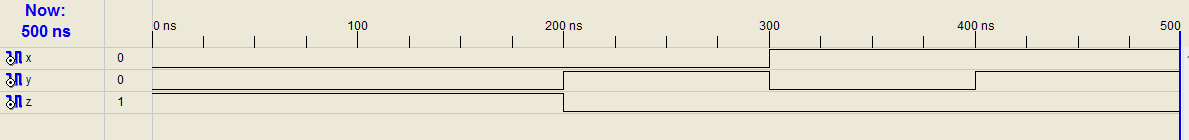
end process;

end Behavioral;

**RTL Schematic**

****

**WAVE FORM**



**XOR GATE**

entityxor\_gate is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

endxor\_gate;

architecture Behavioral of xor\_gate is

begin

process(X,Y)begin

if(X=Y) then

Z<='0';

else

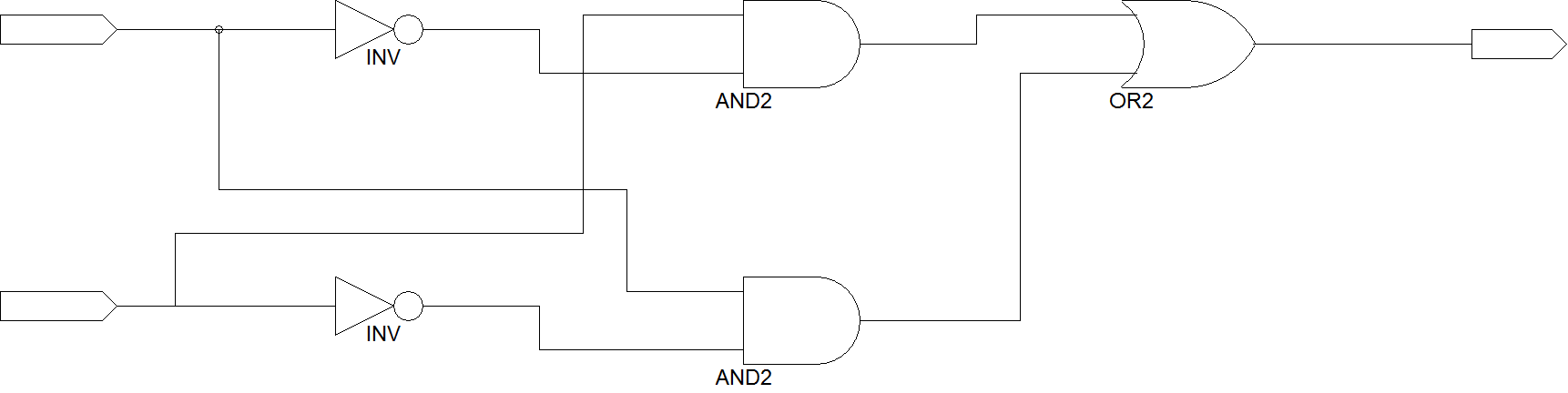
Z<='1';

end if;

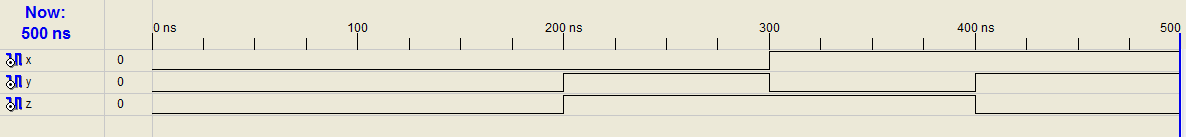
end process;

end Behavioral;

**RTL Schematic**

****

**WAVE FORM**



**XNOR GATE**

entityxnor\_gate is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end xnor\_gate;

architecture Behavioral of xnor\_gate is

begin

process(X,Y)begin

if(X=Y) then

Z<='1';

else

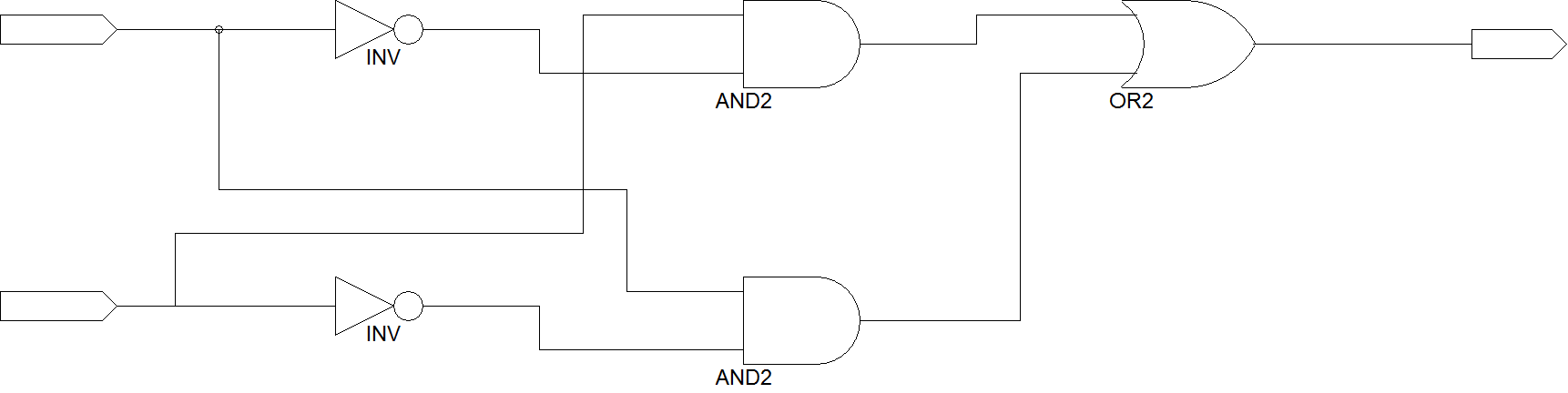
Z<='0';

end if;

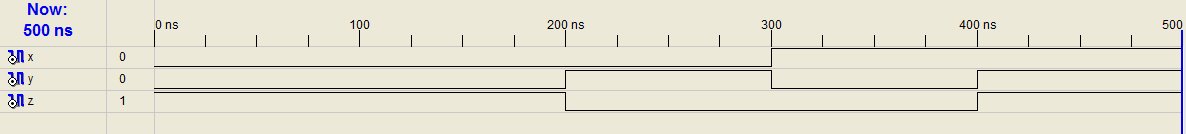
end process;

end Behavioral;

**RTL Schematic**

****

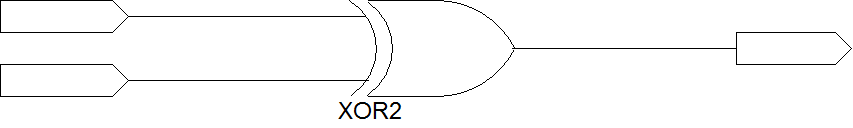
**WAVE FORM**

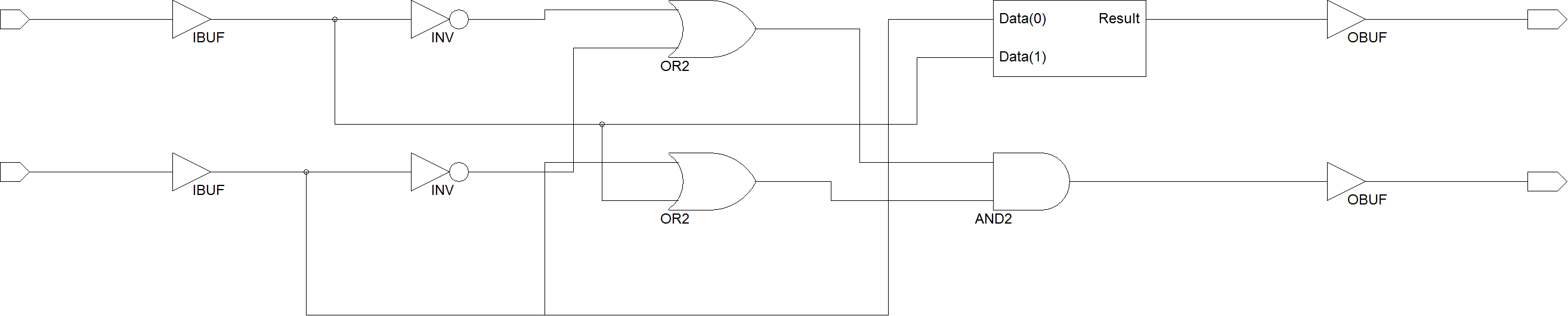


**Experiment No – 3**

**AIM: To Design and Simulation of XOR and XNOR logic gates through VHDL Module using Local Signals and study the waveform .**

**XOR GATE**

****

****

Entity signal1 is

Port (X: in STD\_LOGIC;

Y: in STD\_LOGIC;

Z: out STD\_LOGIC);

End signal1;

Architecture Behavioral of signal1 is

Signal P: STD\_LOGIC; Signal Q: STD\_LOGIC; Signal R: STD\_LOGIC; Signal S: STD\_LOGIC;

Begin

z1<= x xor y; P<=not x;

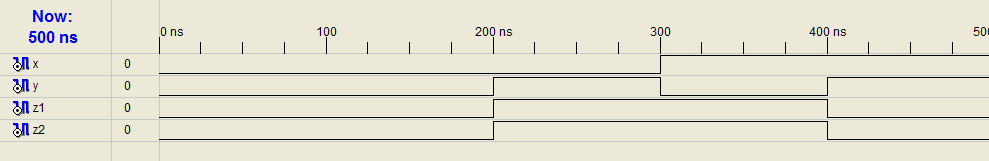
Q<=not y;

R<=P and y;

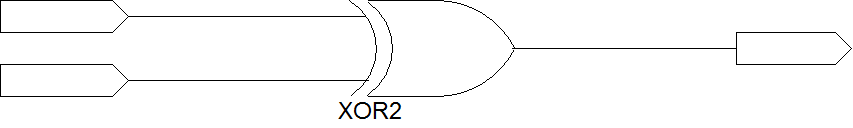
S<=Q and x;

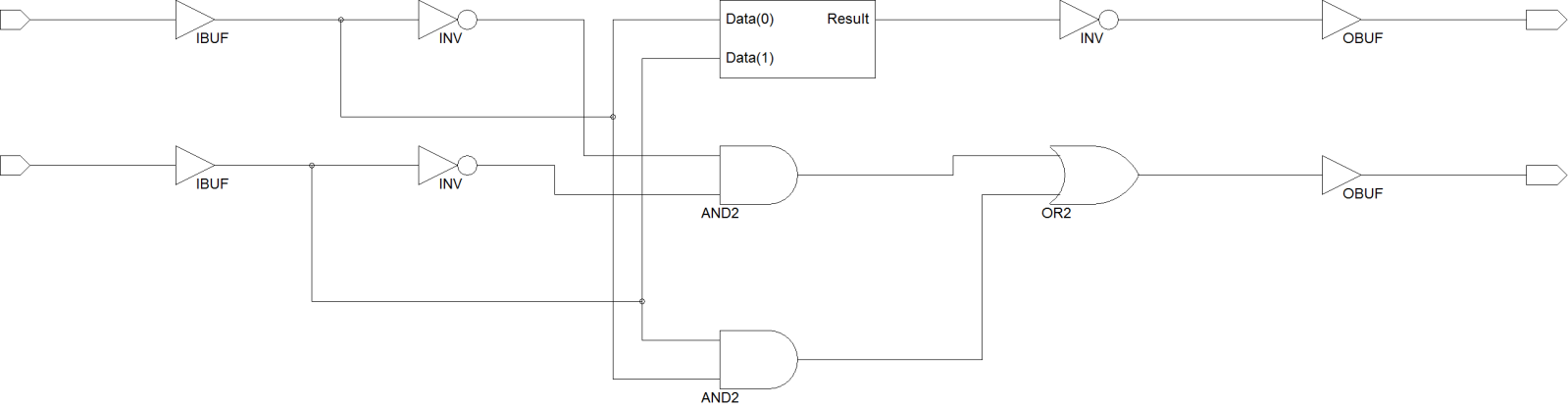
z2<=R or S;

End Behavioral;



**XNOR GATE**





entity xnor2 is

Port ( x : in STD\_LOGIC;

y : in STD\_LOGIC;

a1 : out STD\_LOGIC;

a2 : out STD\_LOGIC);

end xnor2;

architecture Behavioral of xnor2 is

signal M:STD\_LOGIC;

signal N:STD\_LOGIC;

signal O:STD\_LOGIC;

signal P:STD\_LOGIC;

begin

a1<=x xnor y;

M<=x and y;

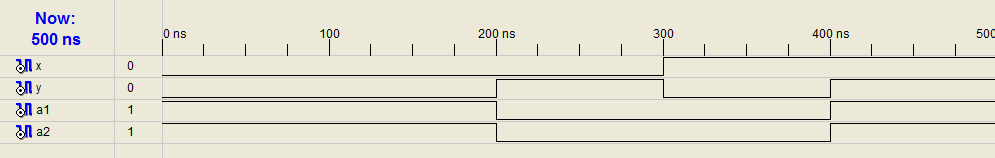
N<=not x;

O<=not y;

P<=N and O;

a2<=M or P;

end Behavioral;



**Experiment No – 4**

**AIM: To Design and Simulation of XOR and XNOR logic gates through VHDL Module using Local Variables and study the waveform using process.**

entity CIRCUIT3 is

Port (d**1** : in STD\_LOGIC;

d**2**: in STD\_LOGIC;

d**3** : in STD\_LOGIC;

f : out STD\_LOGIC);

end CIRCUIT3;

architecture Behavioral of CIRCUIT3 is

begin

process (d**1**,d**2**,d**3**)

variable var**1** : STD\_LOGIC;

variable var**2** : STD\_LOGIC;

variable var**3** : STD\_LOGIC;

variable var**4** : STD\_LOGIC;

variable var**5** : STD\_LOGIC;

variable var**6** : STD\_LOGIC;

begin

var**1** := not d**1**;

var**2** := not d**2**;

var**3** := not d**3**;

var**4** := var**1** and d**3**;

var**5** := var**2** and d**3**;

var**6** := var**3** and d**1** and d**2**;

f <= var**4** or var**5** or var**6;**

end process;

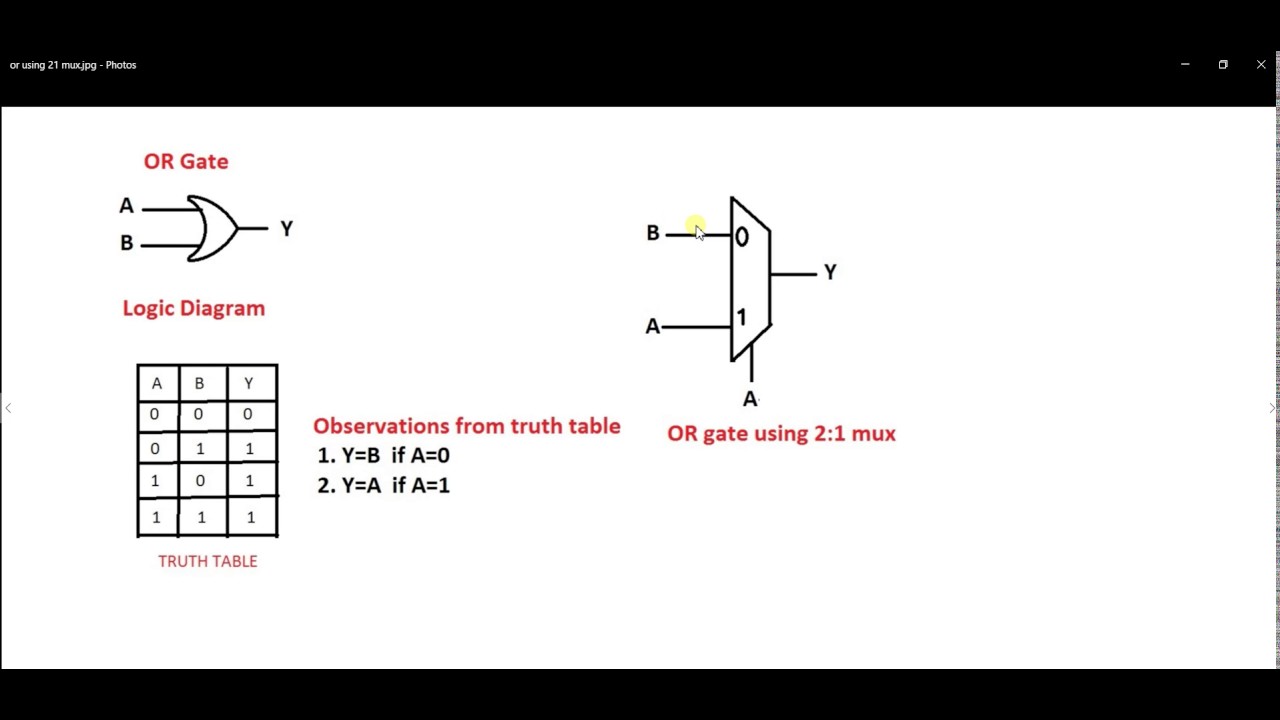
end Behavioral;

**Experiment No – 8**

**AIM: To Design and Simulation of 2:1 and 4:1 MUX (Multiplexer) and study the wave form.**

A **multiplexer** (or **mux**) is a device that selects between several analog or digital input signals and forwards it to a single output line. A **multiplexer** of inputs has select lines, which are used to select which input line to send to the output.

**2:1 MUX**



**VHDL Code**

entitymux\_2\*1 is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

S : in STD\_LOGIC;

C : out STD\_LOGIC);

endmux\_2\*1;

architecture Behavioral of mux\_2\*1 is

begin

process(A,B,S)

begin

if(S=’1’) then

C<= A;

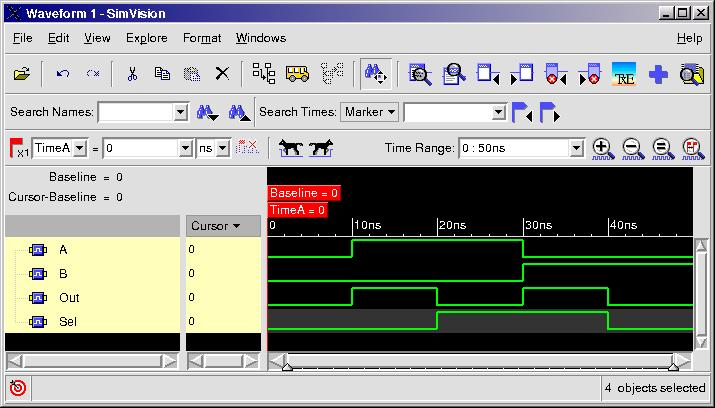
else

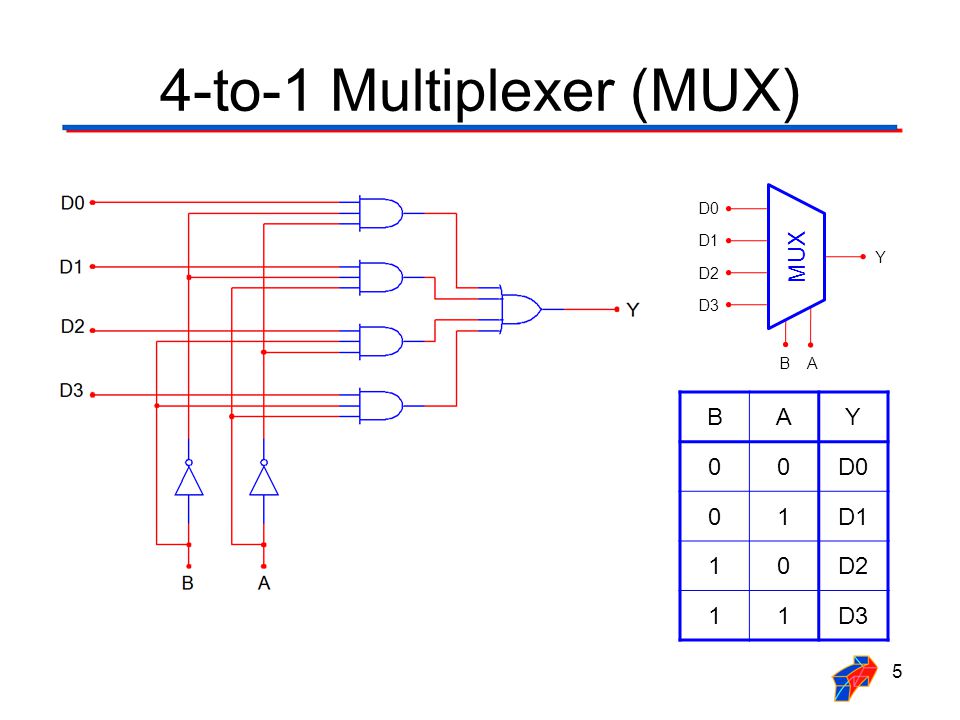
C<= B;

enf if;

end process;

end Behavioral;





**VHDL Code**

entitymux\_4\*1 is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : in STD\_LOGIC;

D : in STD\_LOGIC;

S : in STD\_LOGIC\_VECTOR(1 down to 0);

R : out STD\_LOGIC);

endmux\_4\*1;

architecture Behavioral of mux\_4\*1 is

begin

process(A,B,C,D,S)

begin

case S is

when “00” =>

R <=A;

when “01” =>

R <=B;

when “10” =>

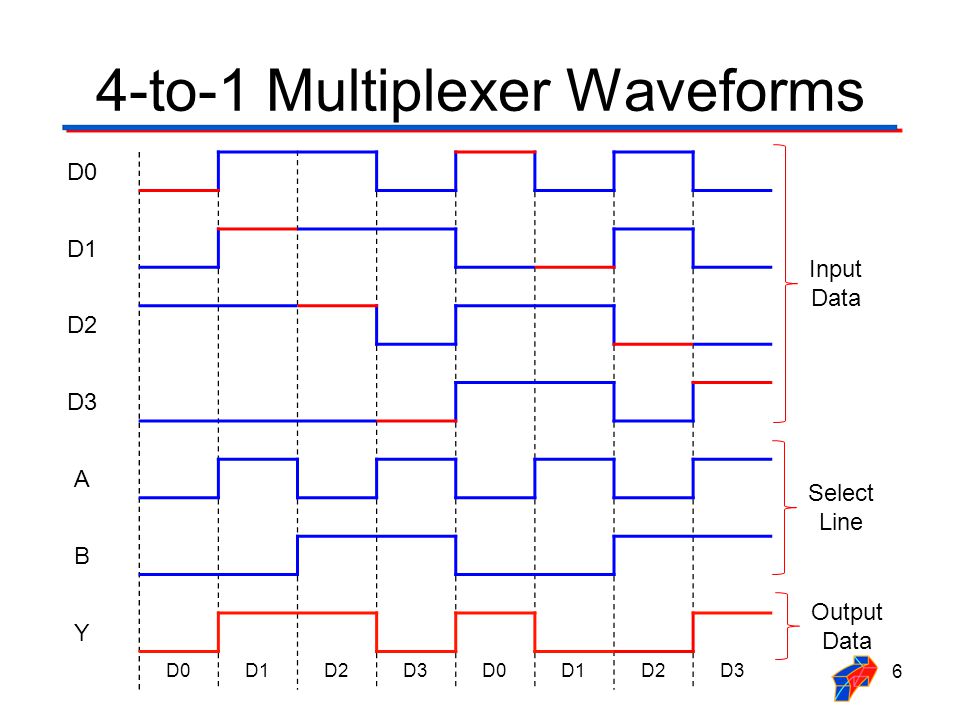
R <=C;

when others =>

R <=D;

end case;

end process;



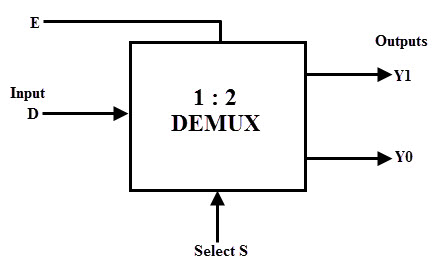
**Experiment No – 9**

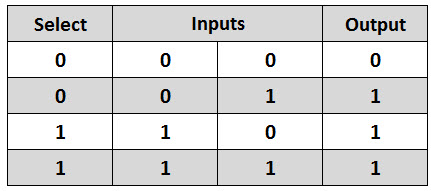
**AIM: To Design and Simulation of 1:2 and 1:4DEMUX (Demultiplexer) and study the wave form.**

A demultiplexer (or demux) is a device that takes a single input line and routes it to one of several digital output lines. A demultiplexer of 2n outputs has n select lines, which are used to select which output line to send the input. A demultiplexer is also called a data distributor.

Demultiplexers can be used to implement general purpose logic. By setting the input to true, the demux behaves as a decoder.

**1:2 DEMUX**





**1:4 DEMUX**

**VHDL CODE**

entitydemux\_1\*2 is

Port ( D : in STD\_LOGIC;

S : in STD\_LOGIC;

Y0 :out STD\_LOGIC

Y1 : out STD\_LOGIC);

enddemux\_1\*2;

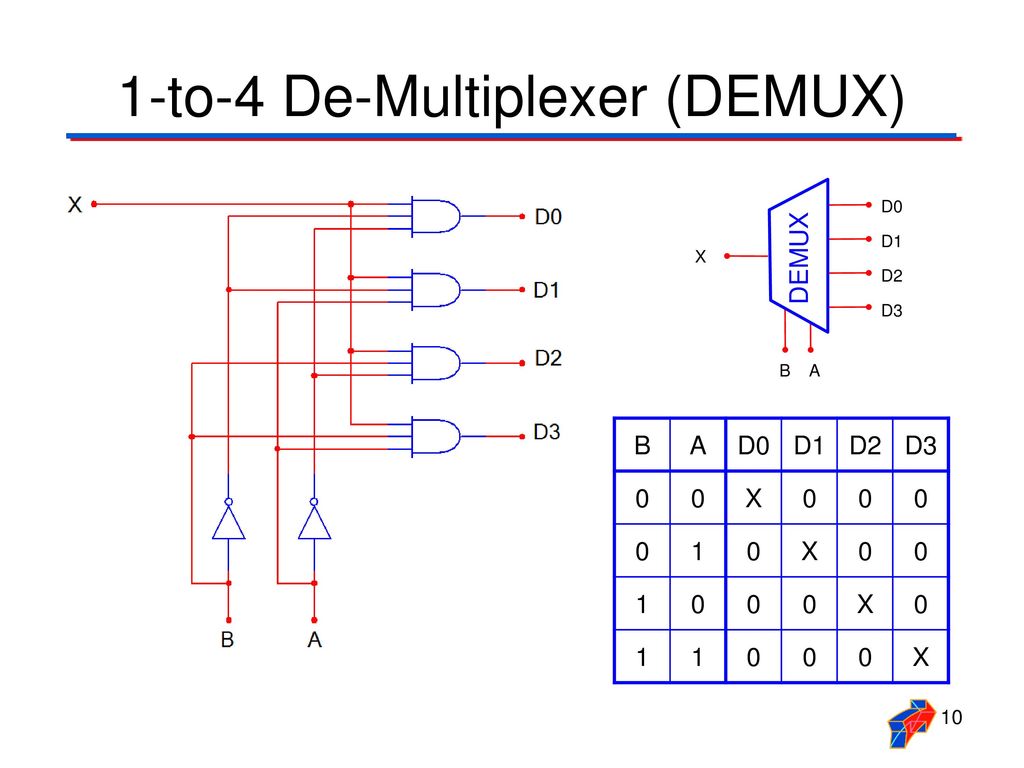
architecture Behavioral of demux\_1\*2 is

begin

Y0 <= D and (not S);

Y1 <= D and S;

end Behavioral;



**VHDL CODE**

entitydemux\_1\*4 is

Port ( A : in STD\_LOGIC;

S : in STD\_LOGIC\_VECTOR(1 downto 0);

M :out STD\_LOGIC

N : out STD\_LOGIC;

O :out STD\_LOGIC

P : out STD\_LOGIC);

enddemux\_1\*4;

architecture Behavioral of demux\_1\*4 is

begin

process(A,S)

begin

**M<=’0’;N<=’0’;O<=’0’;P<=’0’;**

case S is

when “00” =>

M <=A;

when “01” =>

N <=A;

when “10” =>

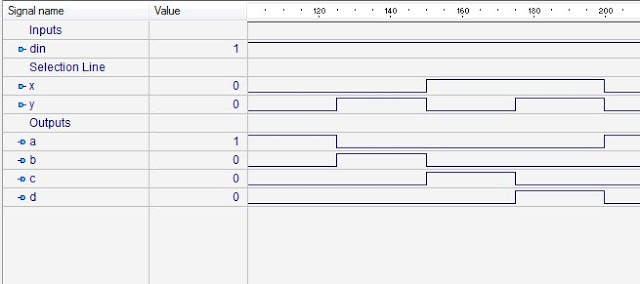
O <=A;

when others =>

P <=A;

end case;

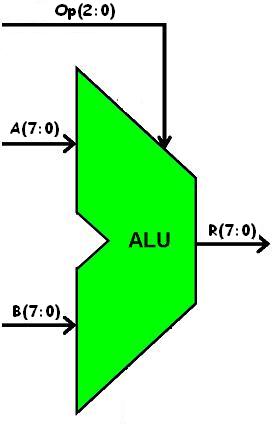
end process;



**Experiment No – 10**

**AIM: To study about the design and simulation of the 8-bit ALU and perform the following basic logical operation.**

**1. Addition, 2. Substraction, 3. And, 4. Or, 5. Nand, 6. Nor, 7. Xor, 8. Xnor**



|  |  |
| --- | --- |
| **ALU Operation** | **Description** |
| Add Signed | R = A + B : Treating A, B, and R as signed two's complement integers. |
| Subtract Signed | R = A - B : Treating A, B, and R as signed  two's complement integers. |
| Bitwise AND | R(i) = A(i) AND B(i). |
| Bitwise NOR | R(i) = A(i) NOR B(i). |
| Bitwise OR | R(i) = A(i) OR B(i). |
| Bitwise NAND | R(i) = A(i) NAND B(i). |
| Bitwise XOR | R(i) = A(i) XOR B(i). |
| Biwise NOT | R(i) = NOT A(i). |

**VHDL CODE**

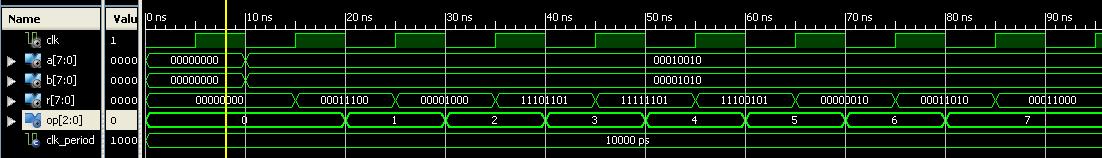
**entity** simple\_alu **is**  
**port**(   Clk : **in** std\_logic; --clock signal  
        A,B : **in** signed(7 **downto** 0); --input operands  
        Op : **in** unsigned(2 **downto** 0); --Operation to be performed  
        R : **out** signed(7 **downto** 0)  --output of ALU  
        );  
**end** simple\_alu;  
**architecture** Behavioral **of** simple\_alu **is**  
--temporary signal declaration.  
**signal** Reg1,Reg2,Reg3 : signed(7 **downto** 0) := (**others** => '0');  
**begin**  
Reg1 <= A;  
Reg2 <= B;  
R <= Reg3;  
**process**(Clk)  
**begin**  
    **if**(**rising\_edge**(Clk)) **then**

**case** Op **is**  
            **when** "000" =>  
                Reg3 <= Reg1 + Reg2;    
            **when** "001" =>  
                Reg3 <= Reg1 - Reg2;   
            **when** "010" =>  
                Reg3 <= **not** Reg1;    
            **when** "011" =>  
                Reg3 <= Reg1 nand Reg2;

**when** "100" =>  
                Reg3 <= Reg1 **nor** Reg2;   
            **when** "101" =>  
                Reg3 <= Reg1 **and** Reg2;    
            **when** "110" =>  
                Reg3 <= Reg1 **or** Reg2;    
            **when** "111" =>  
                Reg3 <= Reg1 **xor** Reg2;   
            **when** **others** =>  
                **NULL**;  
        **end** **case**;        
    **end** **if**;  
   **end** **process**;     
**end** Behavioral;

**VHDL Test Bench Code**

**ENTITY** tb **IS**  
**END** tb;  
**ARCHITECTURE** behavior **OF** tb **IS**  
   **signal** Clk : std\_logic := '0';  
   **signal** A,B,R : signed(7 **downto** 0) := (**others** => '0');  
   **signal** Op : unsigned(2 **downto** 0) := (**others** => '0');  
   **constant** Clk\_period : time := 10 ns;  
  
**BEGIN**  
    -- Instantiate the Unit Under Test (UUT)  
   uut: **entity** work.simple\_alu **PORT** **MAP** (  
          Clk => Clk,  
          A => A,  
          B => B,  
          Op => Op,  
          R => R  
        );  
  
   -- Clock process definitions  
   Clk\_process :**process**  
   **begin**  
        Clk <= '0';  
        **wait** **for** Clk\_period/2;  
        Clk <= '1';  
        **wait** **for** Clk\_period/2;  
   **end** **process**;  
     
   -- Stimulus process  
   stim\_proc: **process**  
   **begin**         
      **wait** **for** Clk\_period\*1;  
        A <= "00010010"; --18 in decimal  
        B <= "00001010"; --10 in decimal  
        Op <= "000";  **wait** **for** Clk\_period; --add A and B  
        Op <= "001";  **wait** **for** Clk\_period; --subtract B from A.  
        Op <= "010";  **wait** **for** Clk\_period; --Bitwise NOT of A  
        Op <= "011";  **wait** **for** Clk\_period; --Bitwise NAND of A and B  
        Op <= "100";  **wait** **for** Clk\_period; --Bitwise NOR of A and B  
        Op <= "101";  **wait** **for** Clk\_period; --Bitwise AND of A and B  
        Op <= "110";  **wait** **for** Clk\_period; --Bitwise OR of A and B  
        Op <= "111";  **wait** **for** Clk\_period; --Bitwise XOR of A and B  
      **wait**;  
   **end** **process**;



------------------------------------------------THE END---------------------------------------------------