Project 2: Digital Electronics using HDL

Nahin Ul Sadad
Lecturer
CSE, RUET

প্রশ্নঃ Real life এ Digital IC (like Microprocessor/Network Switch) কিভাবে ডিজাইন করা হয়?

উত্তরঃ HDL এ Code করে Digital IC ডিজাইন করা হয়। কারণ হাতে Schematics draw/Breadboard এ Prototype করা impractical।

প্রশ্নঃ HDL কি?

উত্তরঃ In computer engineering, a hardware description language (HDL) is a specialized computer language used to describe the structure and behavior of electronic circuits, and most commonly, digital logic circuits. - Wikipedia

প্রশ্নঃ HDL এবং High level programming language যেমন C এর মধ্যে পার্থক্য কি?

উত্তরঃ HDL is not a programming language. It just describes logic circuits. But C itself a programming language.

HDL code works concurrently. But C code works sequentially.

প্রশ্নঃ HDL শিখার সুবিধা কি?

উত্তরঃ Knowing HDL is the first requirement of almost every hardware engineer job.

প্রশ্নঃ HDL code এ লিখা digital circuit সরাসরি run করার উপায় আছে?

উত্তরঃ হ্যা আছে। FPGA ব্যবহার করে।

প্রশ্নঃ FPGA কাকে বলে?

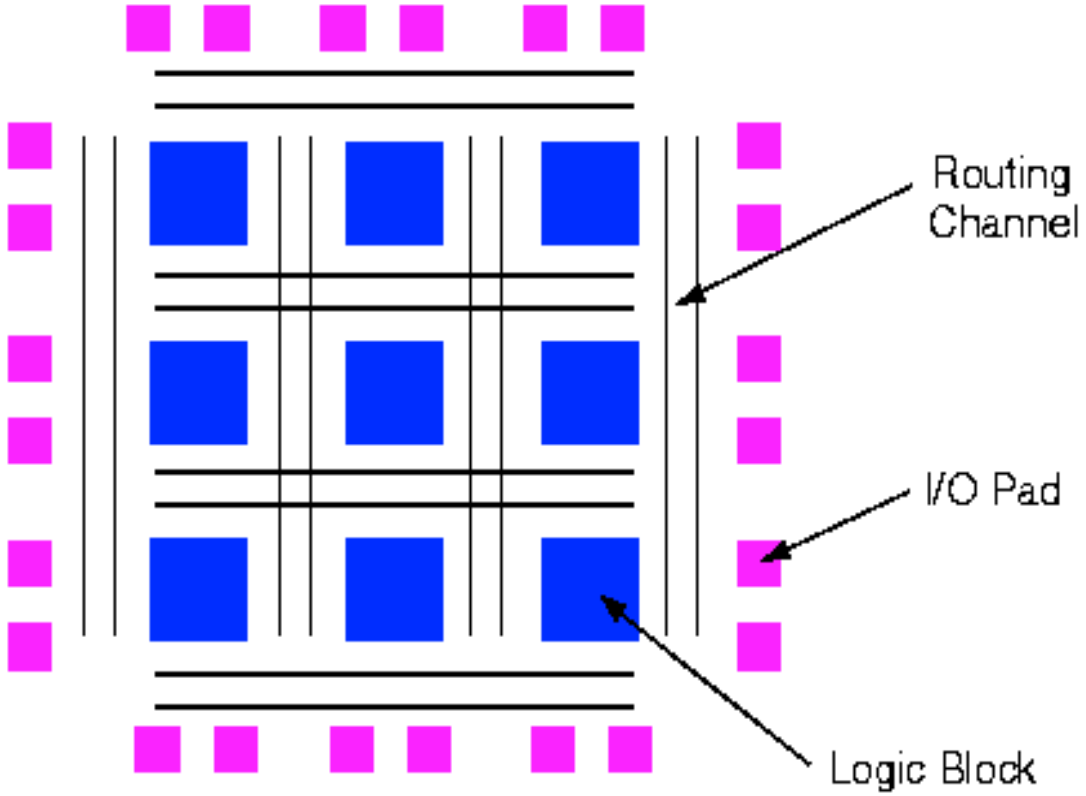
উত্তরঃ A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence the term "field-programmable".

The FPGA configuration is generally specified using a hardware description language (HDL).

–Wikipedia

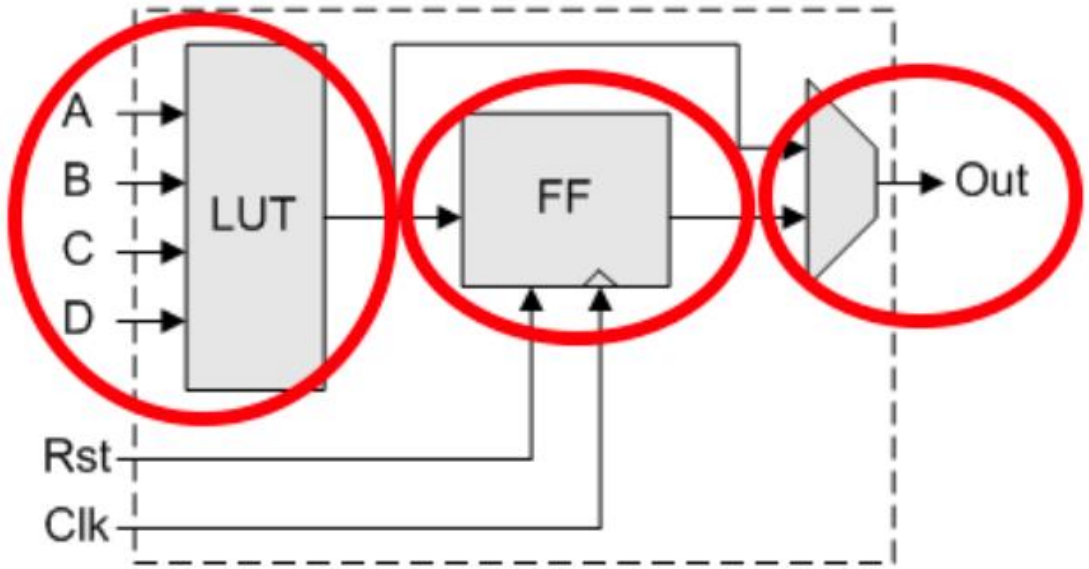
Field-Programmable Gate Array (FPGA)

- First introduced by Xilinx in 1985
- Arrays of logic blocks (to implement logic functions)
- Lots of programmable wiring in routing channels
- Very flexible I/O interfacing logic core to outside world
- Two dominant FPGA makers:
  - Xilinx and Altera
- Other specialist makers e.g. Actel and Lattice Logic

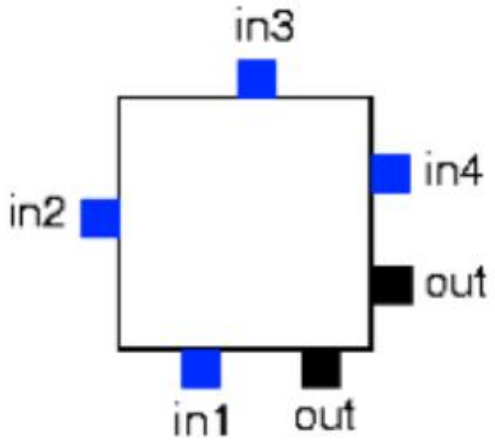


Configurable Logic Block (CLB)

- Based around Look-up Tables (LUTs), most common with 4-inputs
- Optional D-flipflop at the output of the LUT
- 4-input LUT can implement ANY 4-input Boolean equation (truth-table)
- Special circuits for cascading logic blocks (e.g. carry-chain of a binary adder)



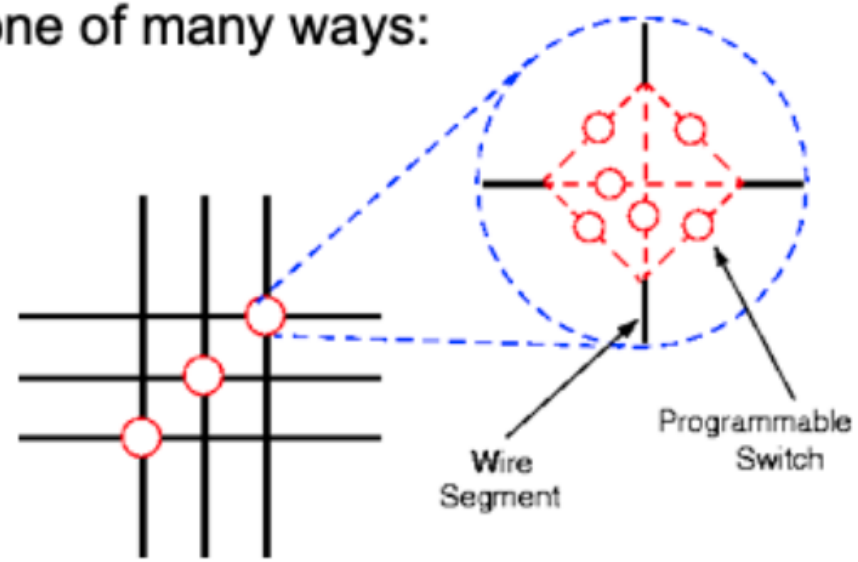
Logic Element
(LE)

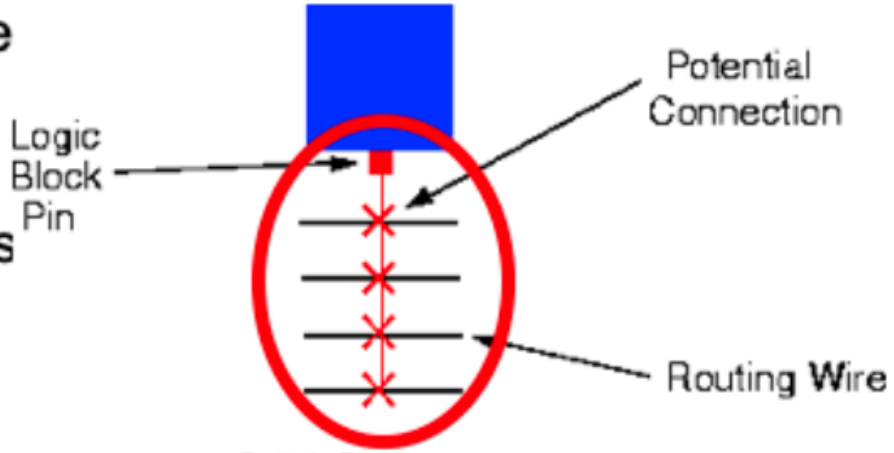


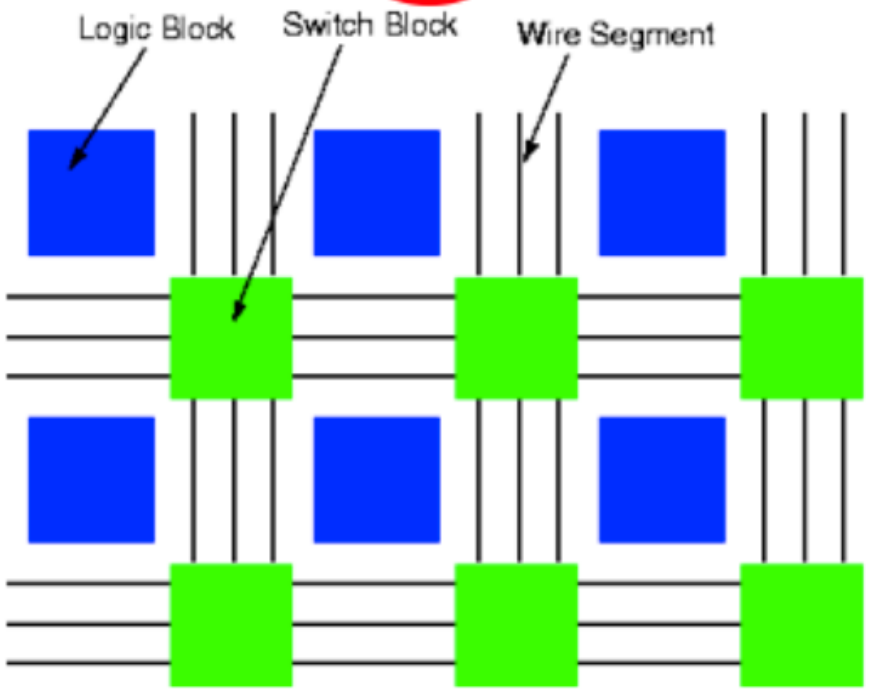
- Each logic block has pins located for easy access:

Programmable Routing

- Between rows and columns of logic blocks are wiring channels
- These are programmable – a logic block pin can be connected to one of many wiring tracks through a programmable switch
- Xilinx FPGAs have dedicated switch block circuits for routing (more flexible)
- Each wire segment can be connected in one of many ways:

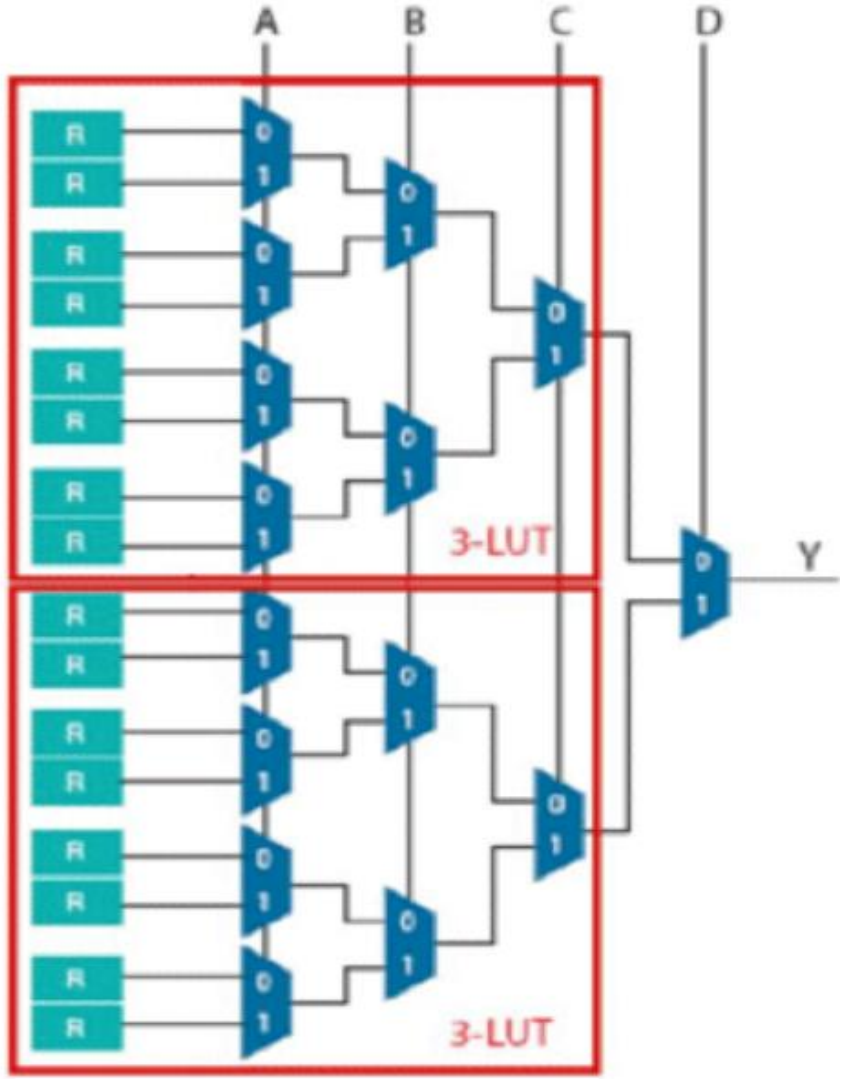




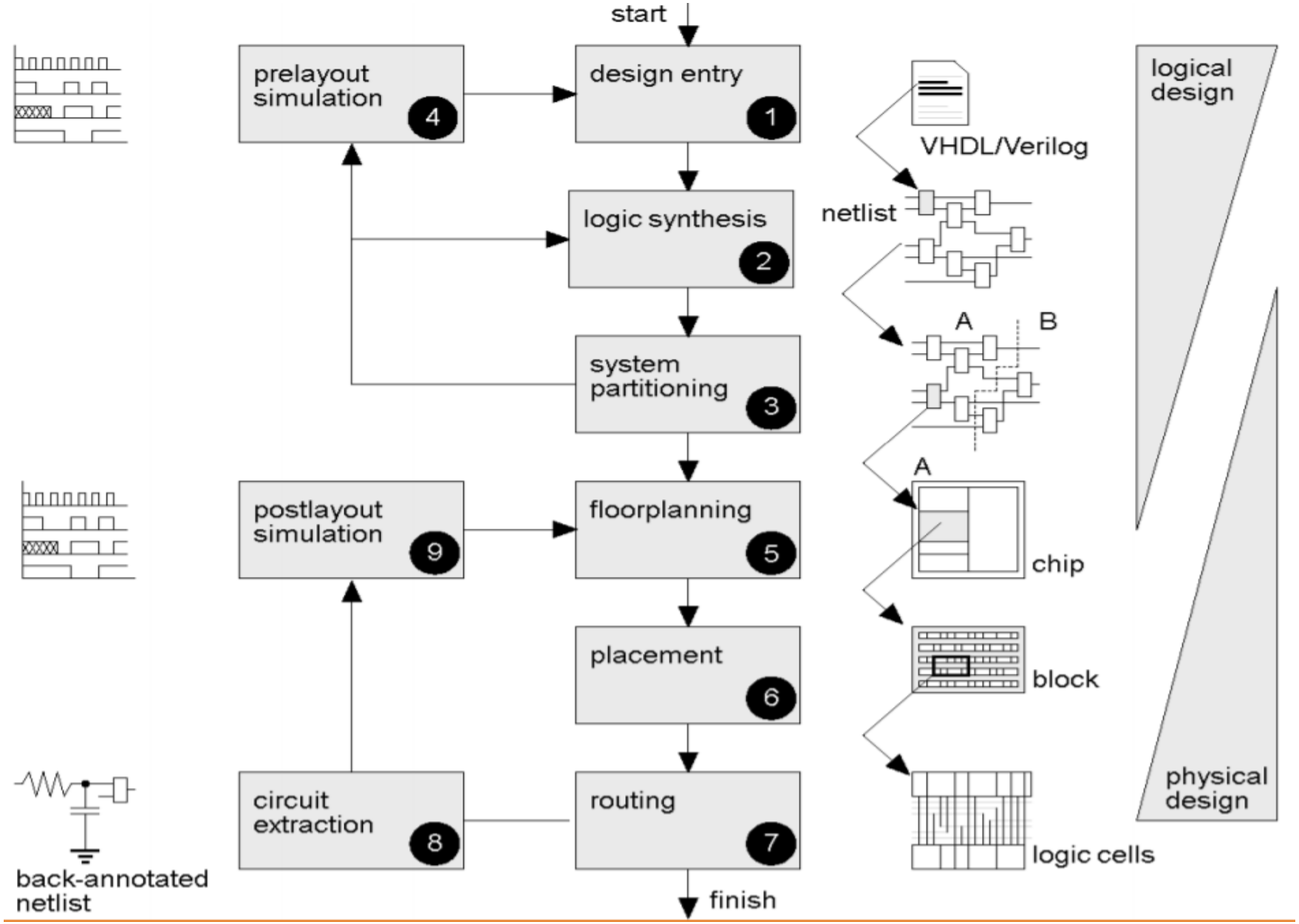


The idea of configuring FPGA

- Programming an FPGA is NOT the same as programming a microprocessor
- We download a **BITSTREAM** (not a program) to an FPGA
- Programming an FPGA is known as **CONFIGURATION**
- All LUTs are configured using the BITSTREAM so that they contain the correct values to implement the Boolean logic
- Shown here is a typical implementation of a 4-LUT circuit
  - ABCD are the FOUR inputs
  - There is four level of 2-to-1 multiplexer circuits
  - The 16-inputs to the mux tree determine the Boolean function to be implemented as in a truth-table
  - These 16 binary values are stored in registers (DFF)
  - Configuration = setting the 16 registers to 1 or 0



Typical VLSI Design Flow



প্রশ্নঃ What are names of some HDL language?

উত্তরঃ Verilog, VHDL etc.

প্রশ্নঃ How to install Verilog?

উত্তরঃ To download and install Iverilog, Goto this site:
http://bleyer.org/icarus/

Follow this YouTube tutorial:
https://www.youtube.com/watch?v=hg9splN_83Y

Make sure to click “Add Icarus Verilog executables folder to the system PATH”

উত্তরঃ For Linux installation,

```
sudo apt-get update
sudo apt-get install iverilog
sudo apt-get install gtkwave
```

প্রশ্নঃ What should be IDE to write Verilog code?

উত্তরঃ My suggestion is VSCode. Goto to this link to download: https://code.visualstudio.com/download

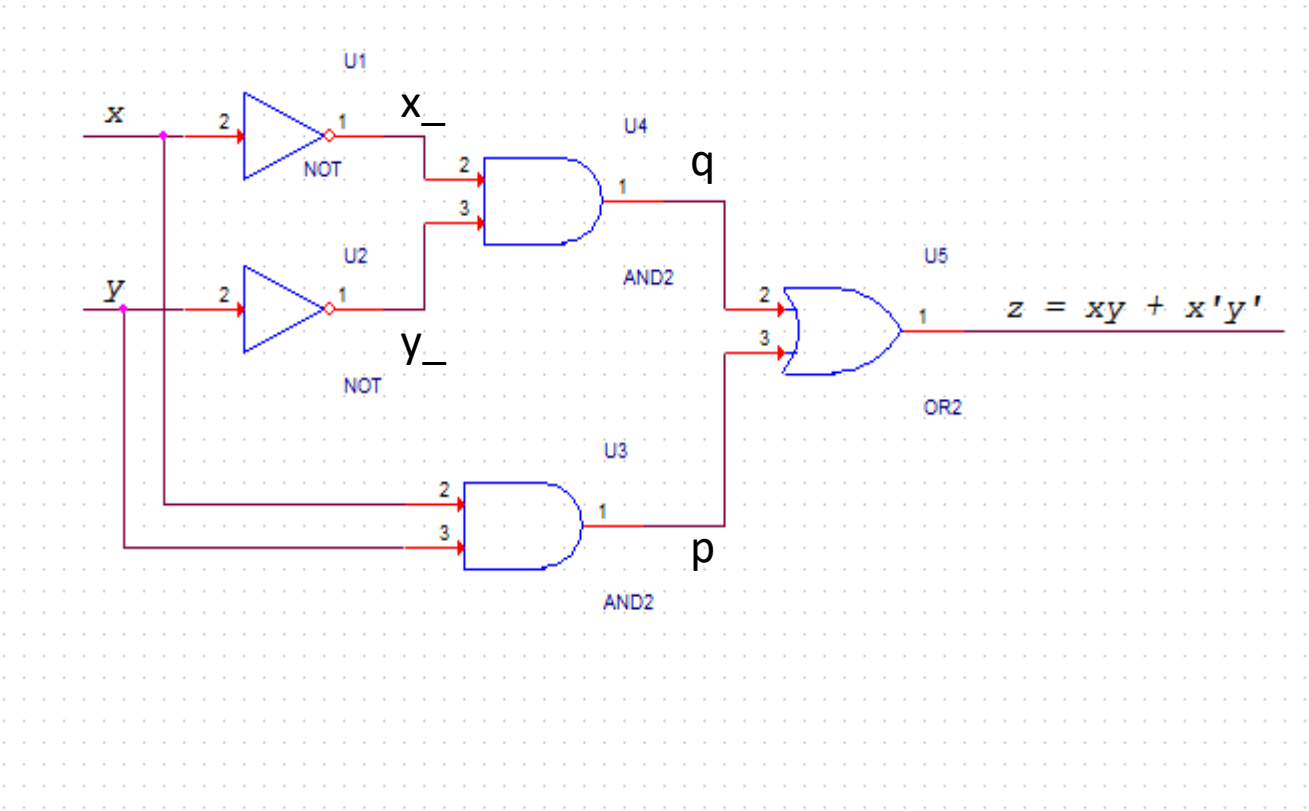
Follow this YouTube tutorial: https://www.youtube.com/watch?v=rwVFDfy2xVI

Make sure to install Verilog addon for syntax highlighting.

প্রশ্নঃ Write a simple comparator logic circuit in Verilog?

Input x	Input y	Output z
0	0	1
0	1	0
1	0	0
1	1	1

উত্তরঃ


U1
x
2
1
NOT
x_
U4
2
3
1
q
AND2
U2
y
2
1
y_
NOT
U5
2
3
1
z = xy + x'y'
OR2
U3
2
3
1
p
AND2

উত্তরঃ

```
module comparator
(
input x,
input y,
output z
);

wire x_, y_, p, q;
not(x_, x);
not(y_, y);
and(p, x, y);
and(q, x_, y_);
or(z, p, q);

endmodule
```

comparator.v

উত্তরঃ

```
module comparator
(
input x,
input y,
output z
);

wire x_, y_, p, q;
not(x_, x);
not(y_, y);
and(p, x, y);
and(q, x_, y_);
or(z, p, q);

endmodule
```

Logic function name

Input port names

Output port names

Internal wires

NOT/AND/OR gate (output, input)

Here we have used Gate Level Modelling.

প্রশ্নঃ Verify a simple comparator logic circuit written in Verilog?

Input x	Input y	Output z
0	0	1
0	1	0
1	0	0
1	1	1

উত্তরঃ Verilog এ লেখা কোড verify করতে আমরা test bench লিখি।

```
`timescale 1ns / 1ps
module stimulus;
// Inputs
reg x;
reg y;
// Outputs
wire z;
// Instantiate the
//Unit Under Test
//(UUT)
comparator uut
(
.x(x),
.y(y),
.z(z)
);
    initial begin
        // Initialize Inputs
    x = 0;
    y = 0;

    #20 x = 1;
    #20 y = 1;
    #20 y = 0;
    #20 x = 1;
    #40;
    end

        initial begin
        $monitor("x=%d,y=%d,z=%d \n",x,y,z);
         end

endmodule
```

stimulus.v

উত্তরঃ Verilog এ লেখা কোড verify করতে আমরা test bench লিখি। এই test benchকে পরে simulation করা হয়।

Timescale
Simulation will run in steps of 1ns and has a precision value of 1ps.

Test bench name

Register
Inputs will be defined as reg.

Instantiation
It creates an instance of the comparator module.

```
`timescale 1ns / 1ps
module stimulus;
// Inputs
reg x;
reg y;
// Outputs
wire z;
// Instantiate the
//Unit Under Test
//(UUT)
comparator uut
(
.x(x),
.y(y),
.z(z)
);
```

```
initial begin
// Initialize Inputs
x = 0;
y = 0;
#20
x = 1;
#20
y = 1;
#20
y = 0;
#20
x = 1;
#40;
end

initial begin
$monitor("x=%d,y=%d,z=%d \n",x,y,z);
end
endmodule
```

Initializing inputs at the beginning of simulation.

 # 20 means
Wait for 20ns.

x = 1 means
X value will be changed to 1 after waiting 20ns.

$monitor means
It will display the values of x, y and z.

প্রশ্নঃ How to simulate Verilog program with test bench.

উত্তরঃ Go to terminal in VSCode and write:

```
iverilog -o out.vpp comparator.v stimulus.v
# comment: out.vpp is the name output file after
# compilation. "-o" defines output parameter.
vvp out.vvp
# comment: Show the output of the stimulus
```

```
#output:
x=0,y=0,z=1
x=1,y=0,z=0
x=1,y=1,z=1
x=0,y=1,z=0
```

প্রশ্নঃ How to show graphical simulation/waveforms of Verilog program with test bench.

উত্তরঃ We will be using gtkwave.

First, add these two new lines in test bench:

```
`timescale 1ns / 1ps
module stimulus;
// Inputs
reg x;
reg y;
// Outputs
wire z;
// Instantiate the
//Unit Under Test
//(UUT)
comparator uut
(
.x(x),
.y(y),
.z(z)
);
```

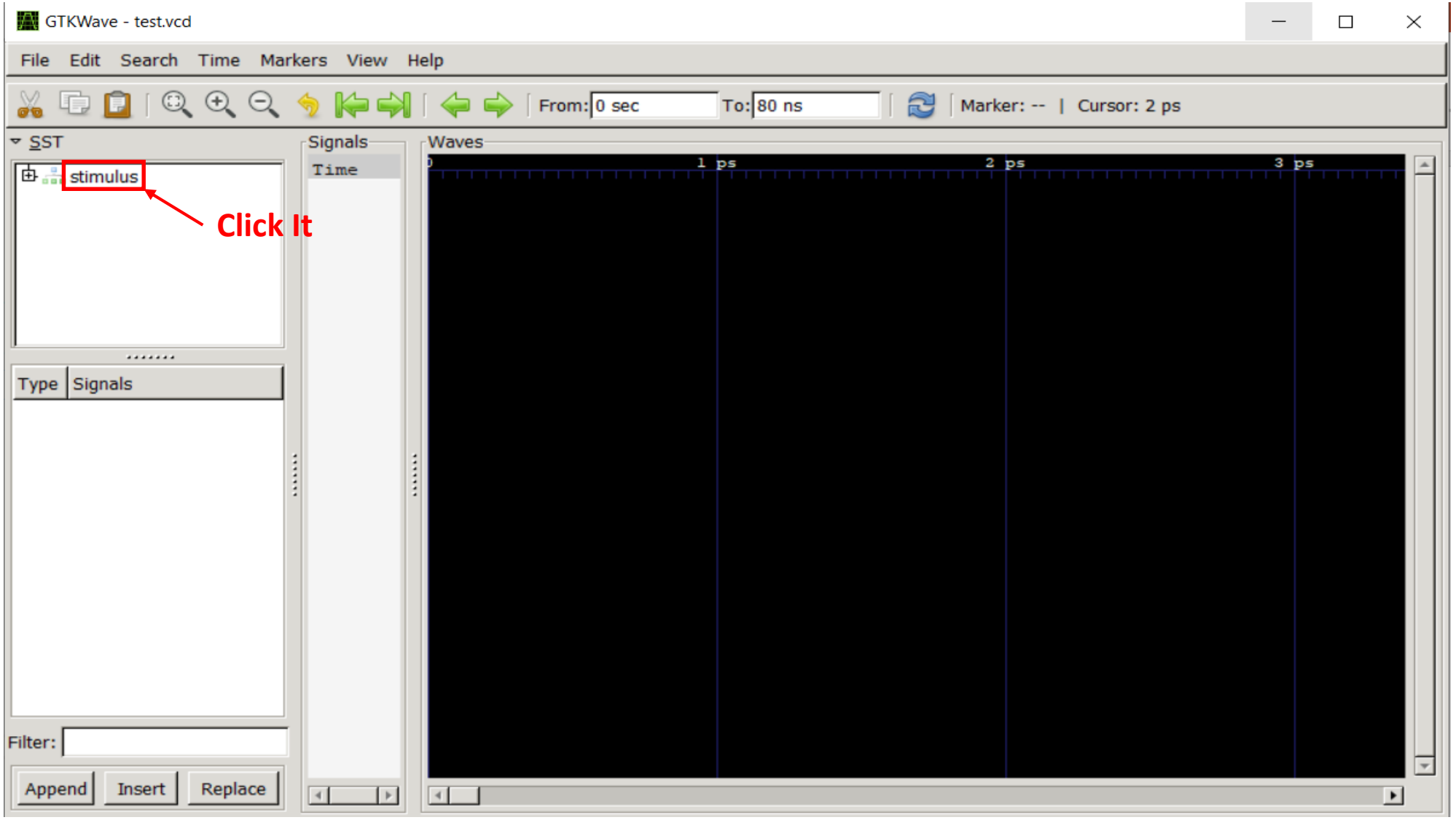
```
initial begin
$dumpfile("test.vcd");
$dumpvars(0,stimulus);
// Initialize Inputs
x = 0;
y = 0;

#20 x = 1;
#20 y = 1;
#20 y = 0;
#20 x = 1;
#40;
end

initial begin
$monitor("x=%d,y=%d,z=%d \n",x,y,z);
end

endmodule
```

stimulus.v

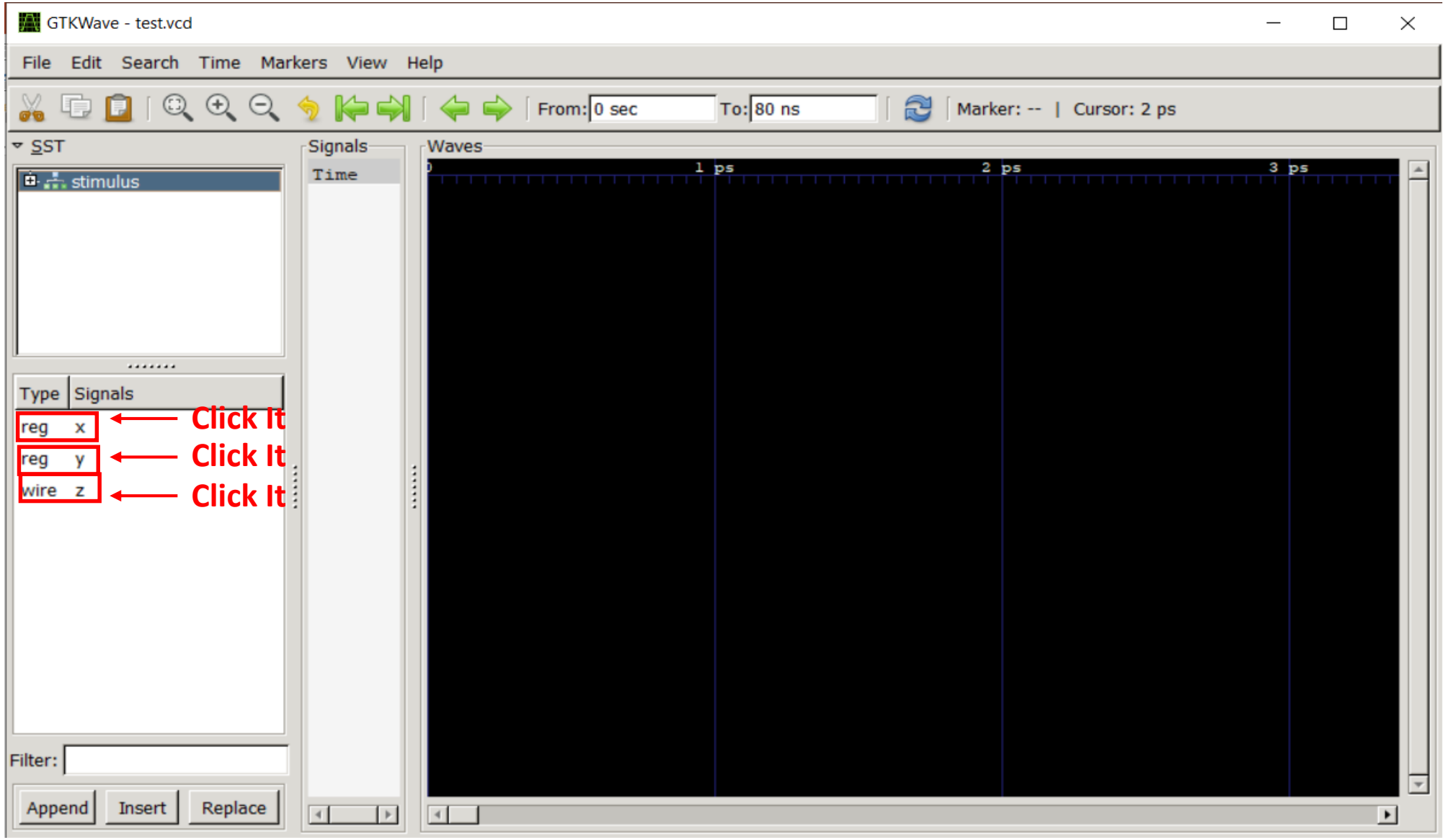
উত্তরঃ Go to terminal in VSCode and write:

```
iverilog -o out.vpp comparator.v stimulus.v
# comment: out.vpp is the name output file after
# compilation. "-o" defines output parameter.
vvp out.vvp
# comment: Show the output of the stimulus
gtkwave test.vcd &
# It will show graphical simulation.
```

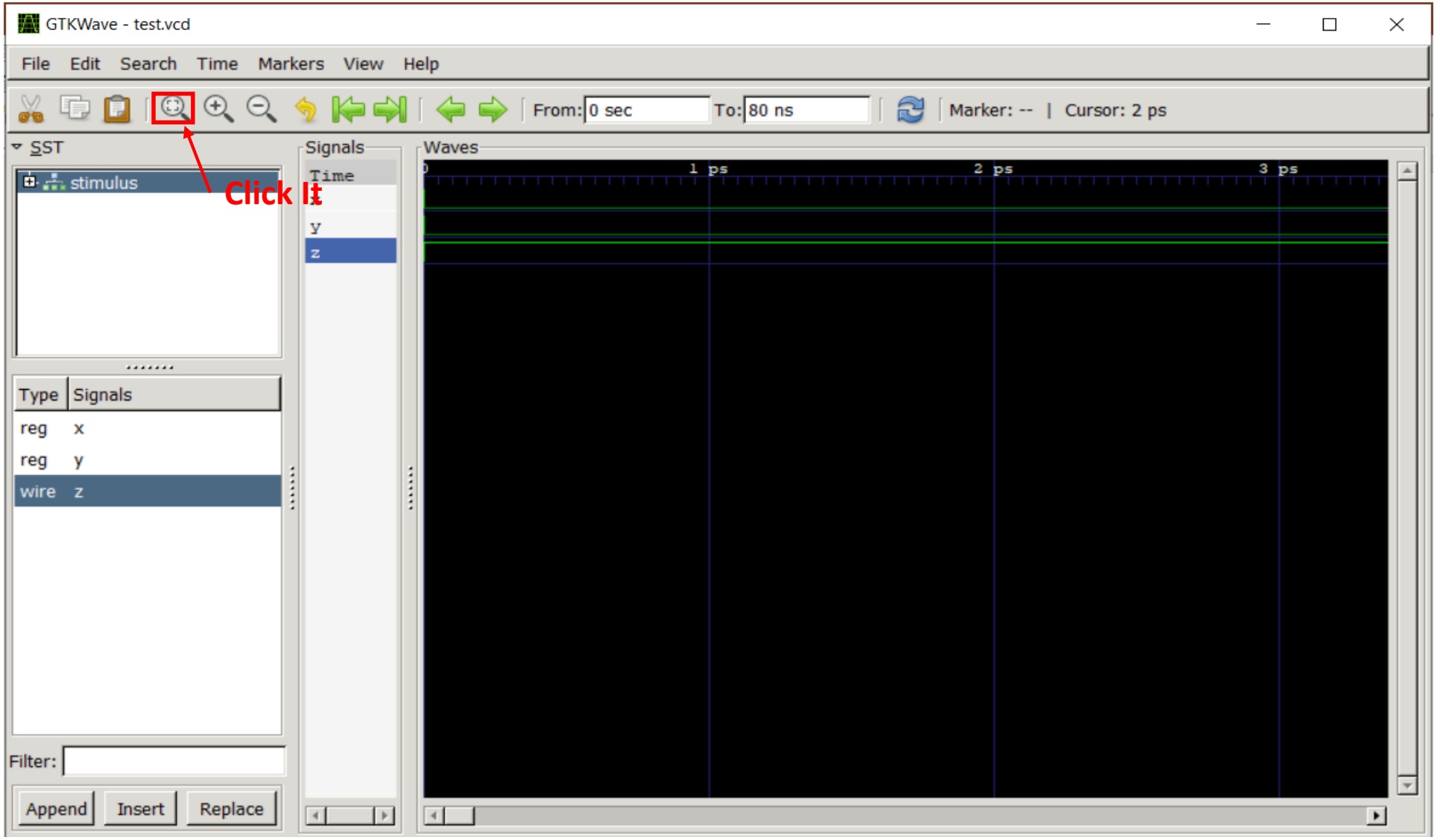
উত্তরঃ


GTKWave - test.vcd
File
Edit
Search
Time
Markers
View
Help
From: 0 sec
To: 80 ns
Marker: --
Cursor: 2 ps
SST
stimulus
Click It
Type
Signals
Filter:
Append
Insert
Replace
Signals
Time
Waves
1 ps
2 ps
3 ps

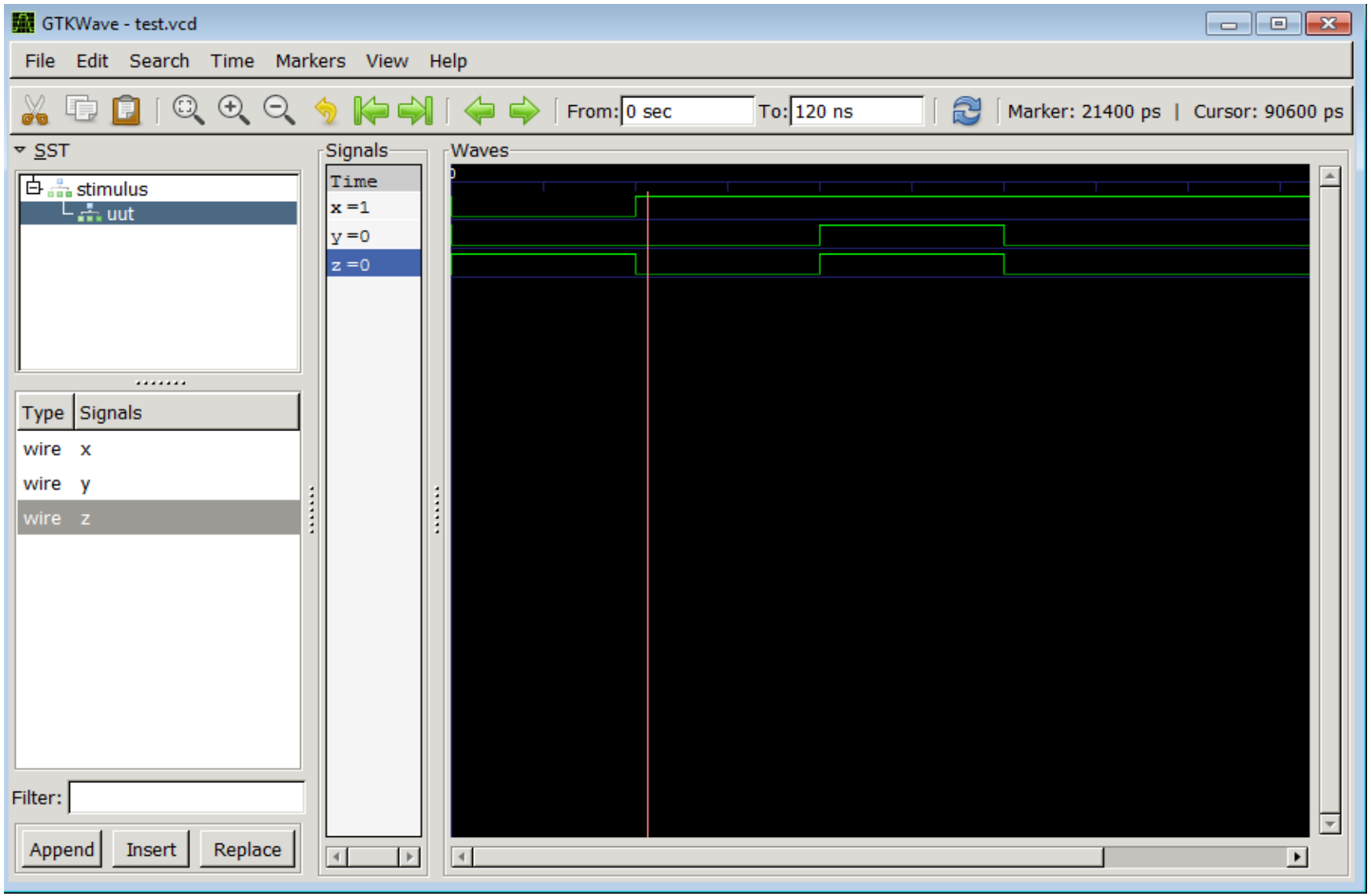
উত্তরঃ


GTKWave - test.vcd
File Edit Search Time Markers View Help
From: 0 sec
To: 80 ns
Marker: -- | Cursor: 2 ps
SST
stimulus
Type Signals
reg x
reg y
wire z
Click It
Click It
Click It
Filter:
Append Insert Replace
Signals
Time
Waves
1 ps
2 ps
3 ps

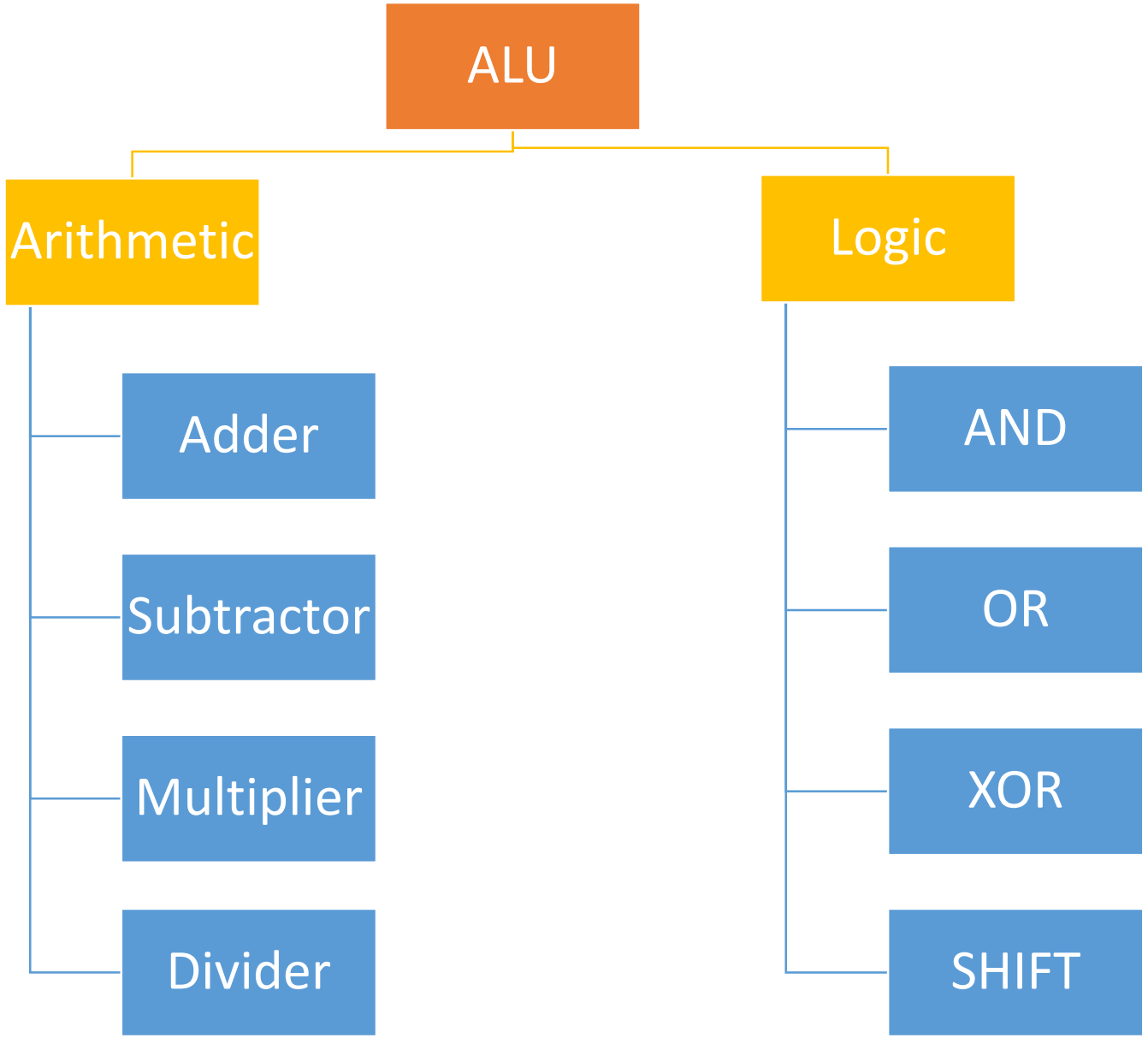
উত্তরঃ

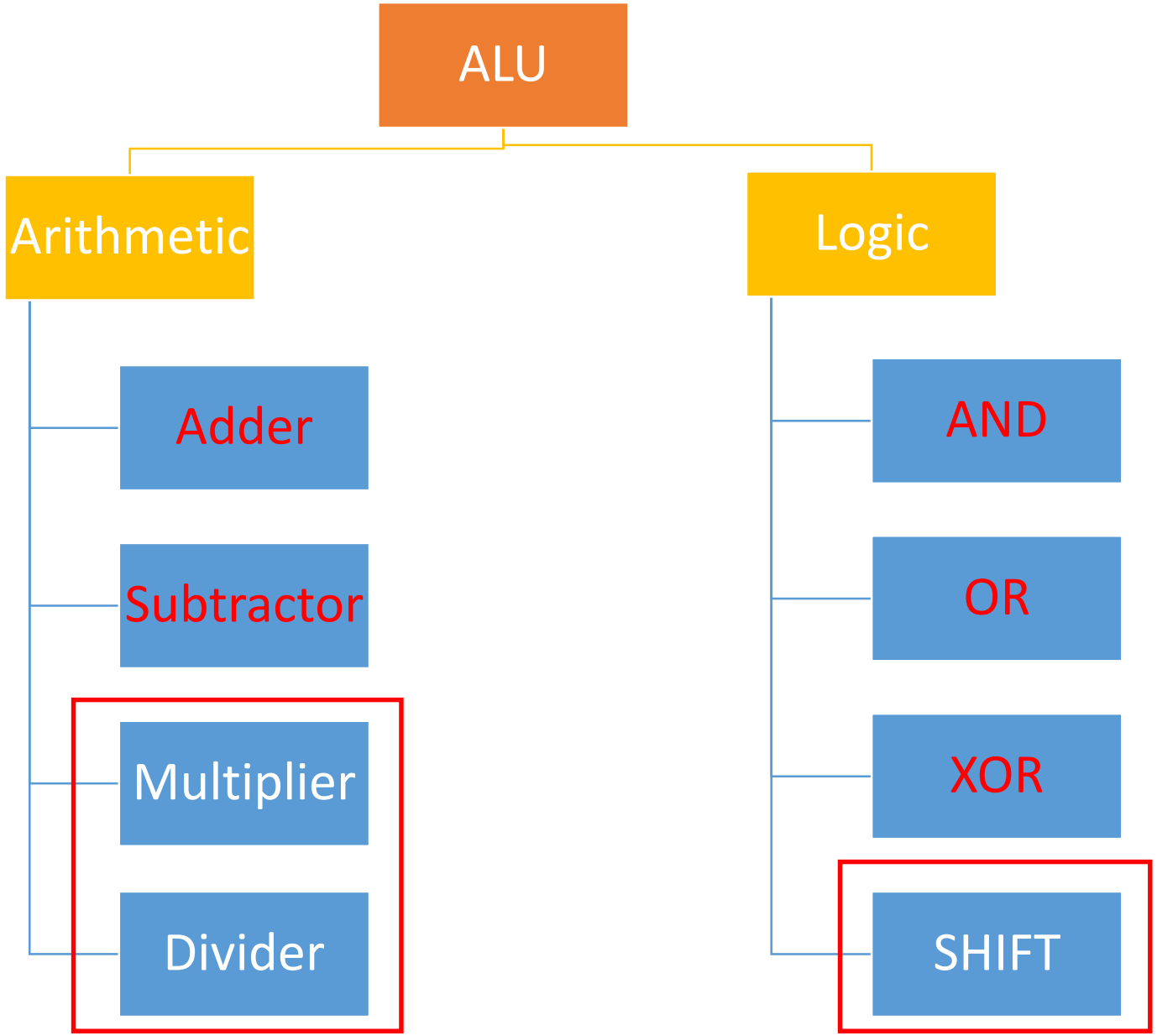

GTKWave - test.vcd
File Edit Search Time Markers View Help
From: 0 sec
To: 80 ns
Marker: -- | Cursor: 2 ps
SST
stimulus
Click It
Signals
Time
x
y
z
Waves
1 ps
2 ps
3 ps
Type Signals
reg x
reg y
wire z
Filter:
Append
Insert
Replace

উত্তরঃ


GTKWave - test.vcd
File Edit Search Time Markers View Help
From: 0 sec
To: 120 ns
Marker: 21400 ps | Cursor: 90600 ps
SST
stimulus
uut
Signals
Time
x =1
y =0
z =0
Waves
Type Signals
wire x
wire y
wire z
Filter:
Append Insert Replace

Next Day:
ALU Design I: Shifter & Multiplier


ALU
Arithmetic
Logic
Adder
Subtractor
Multiplier
Divider
AND
OR
XOR
SHIFT


ALU
Arithmetic
Logic
Adder
Subtractor
Multiplier
Divider
AND
OR
XOR
SHIFT

Thank You ☺