

Digital Logic Design

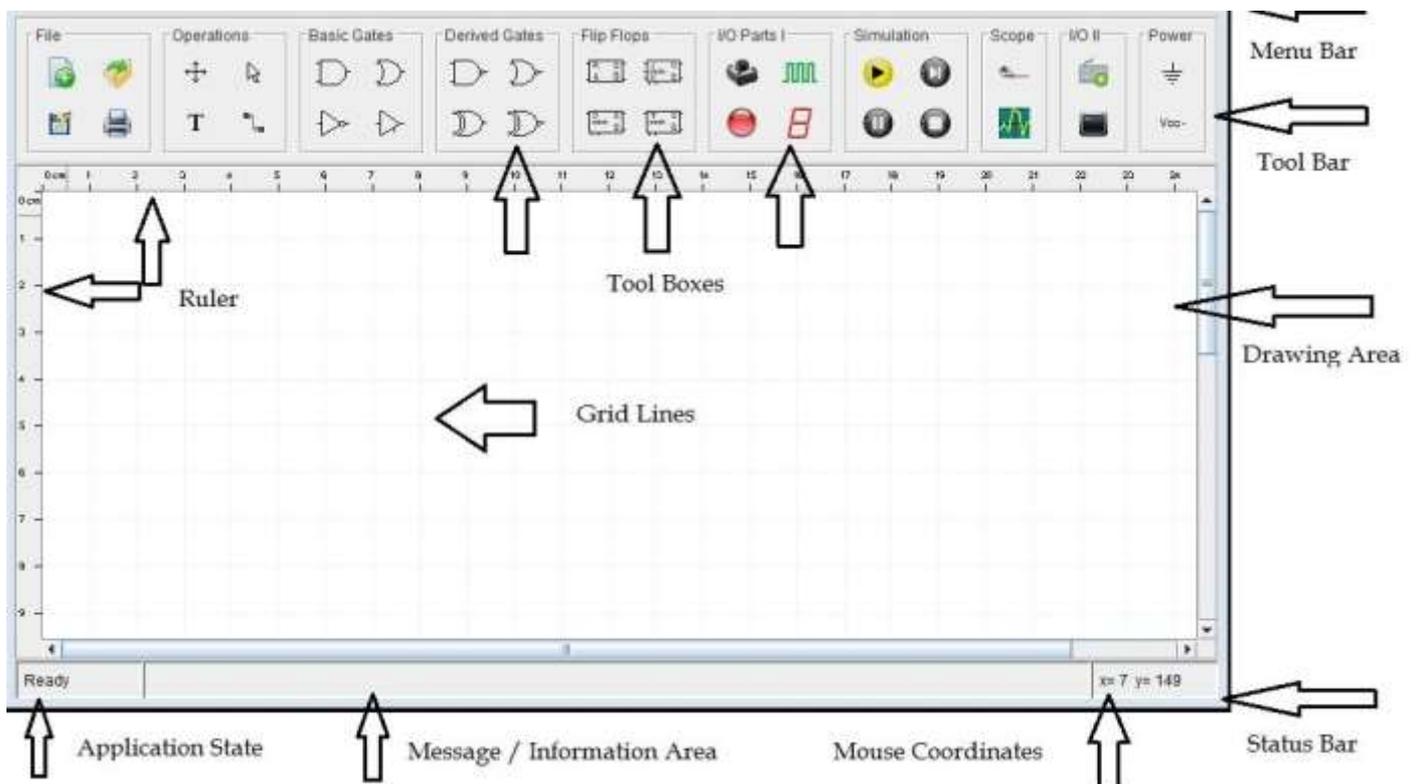
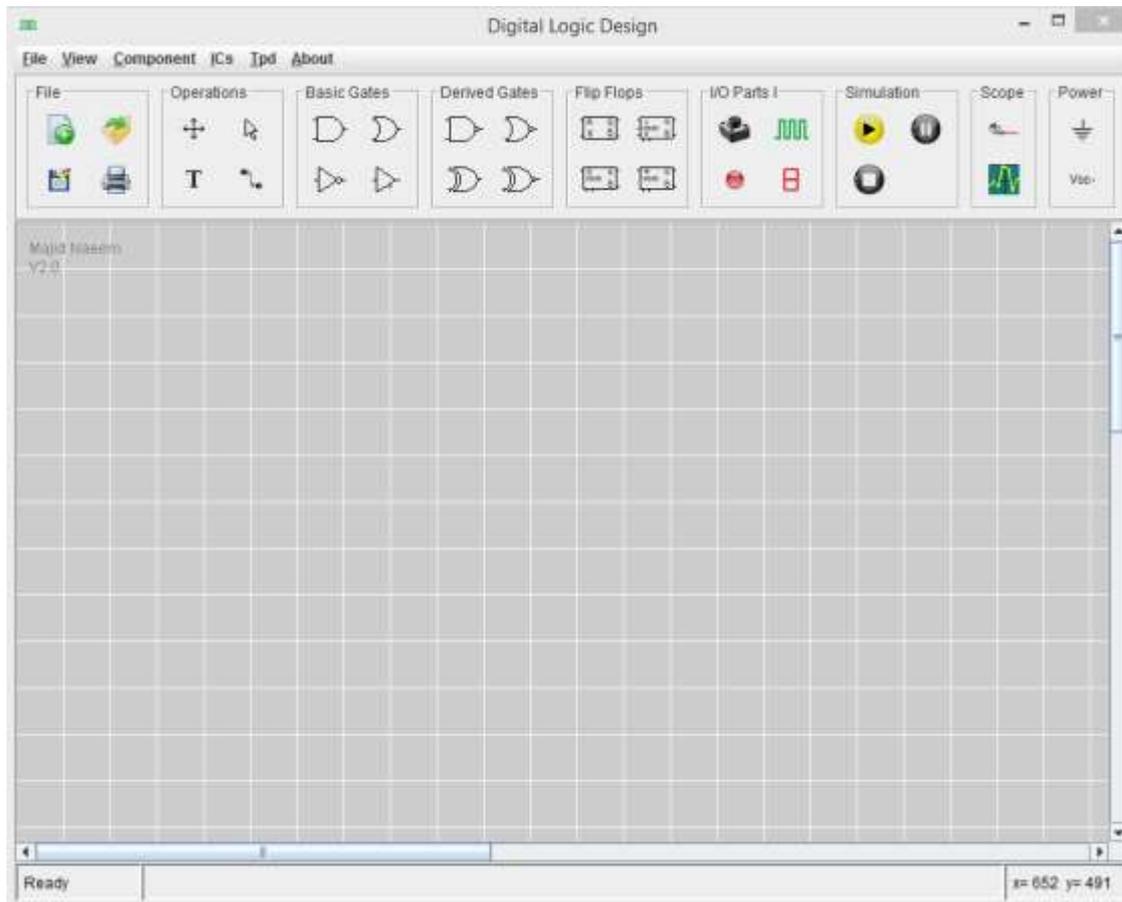
Version 2.0

29-05-2023

CONTENTS

1. Application GUI	1
2. Tool Bar Parts and their Function	5
3. Basic Input and Output Parts	8
4. Advanced Input and Output Parts	9
5. Digital Oscilloscope Parts	10
6. Logic Gates and Flip-Flops	11
7. Power Parts	12
8. Track or Wire	13
9. Simulation	16
10. How to use Digital Scope	17
11. How to make a custom ICs/Modules	20
12. How to make a State Machine	25
13. Generating ICs from Boolean Expressions	26
14. Adding Control Pins in ICs	27
15. Memory	29
16. Components Parameters	30

1 - Application GUI



The GUI toolbar can be customized by selecting or deselecting the options available in the View Menu. Deselecting an item hides it from the toolbar.

You can change the drawing size by selecting the "Drawing Area Size" option as well as drawing background color by selecting "Background Color" option.

There are two special options, Component Name Display and Component Id Display. These allow the component Name and Number to be displayed above the respective components in the circuit drawing.



2 - Tool Bar Parts and their Function

File	File Related Operations
	Create New Circuit Drawing and Close all other circuit drawings
	Open an Existing Circuit Drawing Close all other circuit drawings
	Save Current Circuit Drawing. If name of the file is not assigned, ask for the name
	Print currently open circuit drawing.

Operations	Set Application State to perform various Operations on Circuit Drawing
	Drag the Whole Circuit drawing to some new position
	Ready State – Perform various Editing Operations on Components and Tracks and Place and Remove Components, Track and Text Nodes in the Circuit Drawing
T	Text Node – Add Text in the Circuit Drawing
	Track Mode – Draw Track to connect components

Basic Gates	Place Logic Gate on Circuit Drawing
	Logical AND Gate
	Logical OR Gate
	Logical NOT Gate
	TRI State - May not be treated as Basic Gate but required for building useful circuits

Derived Gates	Gates derived from Basic Gates; can also be designed using Basic Gates
	Logical NAND Gate
	Logical NOR Gate
	Logical XOR Gate
	Logical XNOR Gate

Flip Flops	Sequential Circuit Designing Basic Components; May be designed using Basic and Derived Gates
	SR Flip Flop
	JK Flip Flop with reset/clr and set pin which may be left unconnected if not needed
	D Flip Flop with reset/clr and set pin which may be left unconnected if not needed
	T Flip Flop with reset/clr and set pin which may be left unconnected if not needed

I/O Parts I	Input Out Parts
	BUTTON – Input Part - Click On Click OFF
	CLOCK – Input Part - Provided continuous train of On-Off square wave signals
	LED – Output Part - show signal state
	SSD – Seven Segment Display – Output Part

Simulation	For Circuit Simulation
	Run the Circuit Simulation
	Move the circuit simulation in Single Step on each click on this button
	Stop the Simulation

Scope	Digital Oscilloscope and its Probe to monitor Digital Signals at various multiple points in the circuit during Simulation
	Probe – Use to mark the points in the circuit where signal monitoring is required
	Oscilloscope – Opens up in a separate window to Display the selected digital signals

I/O II	Advanced Input / Output Parts
	7-Bits ASCII Keypad as Input Part
	CRT Display to display 7-Bits ASCII data and characters – Opens in a separate window

Power	Power Parts
	Input Part –Provide constant LO signal
Vcc-	Input Part – Provides constant HI Signal

3 - Basic Input and Output Parts



Button



This is an input part and used to provide HI and LO input. In normal state it provides LO input and when pressed during running of simulation, provides HI input.

LED



This is an output part and use to display the signal level. This part has single connection.

The above two parts have two parameters "Name" and "Number" to be assigned appropriate values by the user. If you are converting a circuit into a re-usable Module, it is strongly recommended that you should assigned values to these two parameters as explained in Module section.

Clock



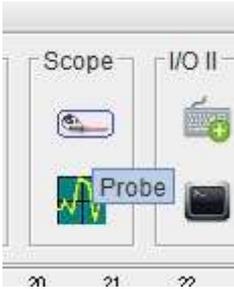
This is an input part and use to provide continuous train of HI and LO signal. The speed of the clock can be set from the **Clock** menu. It provides a single pin connection.

SSD – Seven Signal Display



This is an output component. It has seven input connections each connected with one segment of the part. The pins are treated as a...g from top to bottom. These pins are active HI.

4 - Advanced Input and Output Parts



Keypad

1	2	3	4	5	6	7	8	9	0
A	B	C	D	E	F	G	H	I	J
K	L	M	N	O	P	Q	R	S	T
U	V	W	X	Y	Z		Sh		/n



This Keypad generates ASCII code for the characters shown on the pad. There are 7 output pins which provide the corresponding ASCII value. The pin numbers are from 0...6 from top to bottom. The "Sh" key is "Shift" key and once pressed remained active until pressed again. When pressed, provides capital letters ASCII value. The Enter or Return key is represented by "/n" on the keypad. Blank key acts as a "Space Bar".

Display



The Display acts as a CRT. It gets 7-Bit ASCII value and displays it as decimal or ASCII character. The input pins are numbered 0...6 from left to right. The right most pin is an "Enable" pin which is active on Rising Edge "RE" and causes the value available on input pins to be displayed on the display. During simulation, by clicking on the Display, a new window will appear as display. This part has one important parameter named "Data Type" which is used to choose whether data is displayed as decimal value or ASCII character.

5 - Digital Oscilloscope Parts



There are two parts associated with it:

Probe

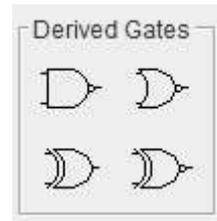
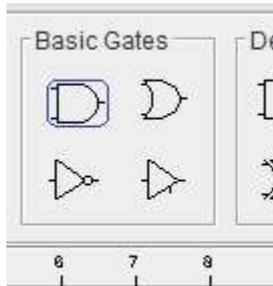


It acts just like a probe for a digital oscilloscope. It is connected to the point of observation through a track. You may connect as many probes in a circuit as needed. It is also recommended that you assign suitable values to Name and Number parameters of a probe. The Name will appear as a signal name in the Oscilloscope display while Number will set the position of the signal among other signals currently displayed in the oscilloscope.

Scope

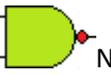
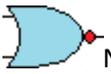
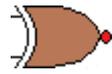
It appears as a separate window. It visually displays the signal data at various points in a circuit as determined by the probes. This window allows saving the displayed signal data and viewing the already stored data. It also allows setting the starting and ending trigger point of the scope. The scope can be set to store a predetermined number of observations from the start or from the start trigger point.

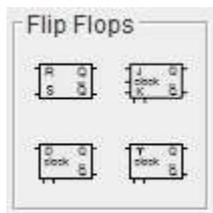
6 - Logic Gates and Flip-Flops



Logic Gates

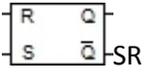
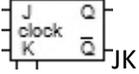
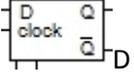
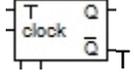
These are standard logic gates use to perform logic operations and built combinational circuits. The TRI state gate is active HI.

 AND	 OR	 NOT	 TRI
 NAND	 NOR	 XOR	 XNOR



Flip Flops

These are four standard flip-flops to design sequential circuits. T and D flip-flops have two more pins at the bottom for clear or reset and set the flip-flop respectively. These two pins are active HI and may be left un-connected.

 SR	 JK	 D	 T
--	--	---	---

Each of these parts has three parameters may be defined by the user:

Name: A Single word name of the part

Number: An integer Number

Tpd : Propagation delay

7 - Power Parts



There are two power parts, Gnd and Vcc, provided in this software. These parts are only meant to provide logic LO and logic HI values respectively.

Ground



This is an input part and provides logic LO value. It is connected at those places which requires constant logic LO value.

VCC

Vcc-

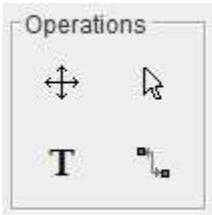
Vcc-

This is an input part and provides logic HI value. It is connected at those places which require constant logic HI value.

8 - Track or Wire



Tool Bar -> Operation Tool Box



The word “Track” is borrowed from PCB (Printed Circuit Board) terminology which is used to assemble a circuit. A Track is a printed wire on a circuit board which is used to connect components through their pins. Track or wire is used to connect components with each other to form a circuit. A circuit is a collection of tracks and components. A Track plays the same role in this software as it plays in a PCB. It is used to connect pins of various components in the circuit.

How to Connect Two pins via Track or Wire

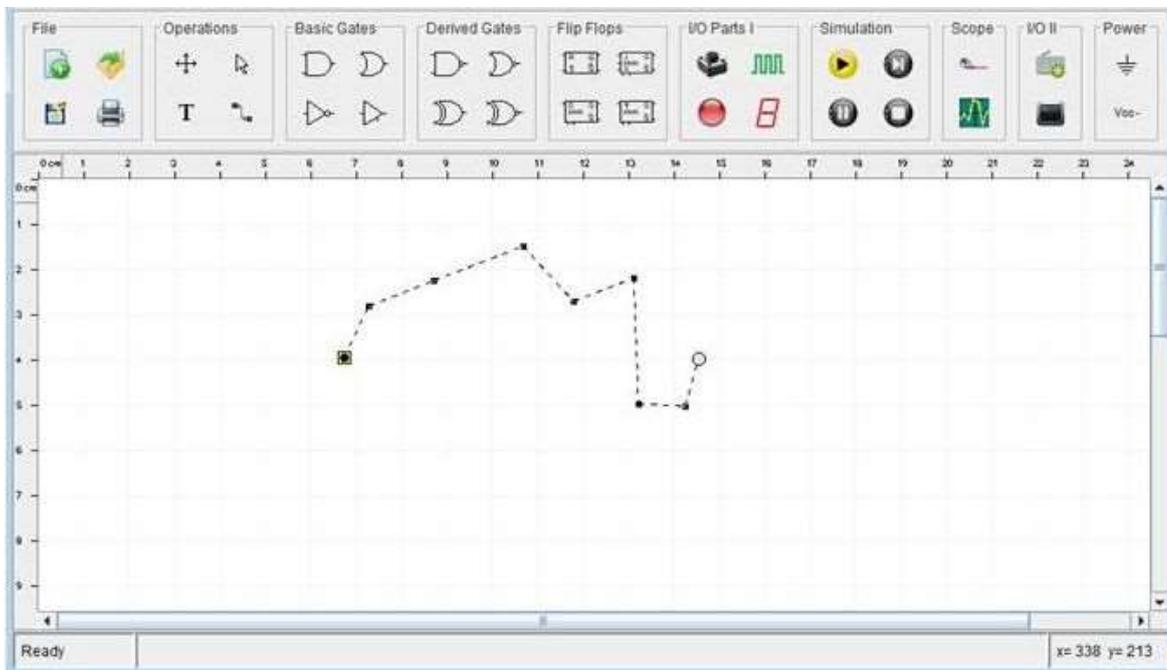
1. Click the Track button in the Tool Bar in Operation Tools Box.
2. Move to the drawing area. The cursor will change its shape to a Track drawing and connecting tool (Soldering Iron) .
3. Click the one of the pins you want to connect and move to the other pin. A Track is start drawing as you move the mouse.
4. Click the other pin you want to connect. A straight line or Track will be drawn between the two pin. i.e., both pins are now connected via this Track or wire.

Important Notes

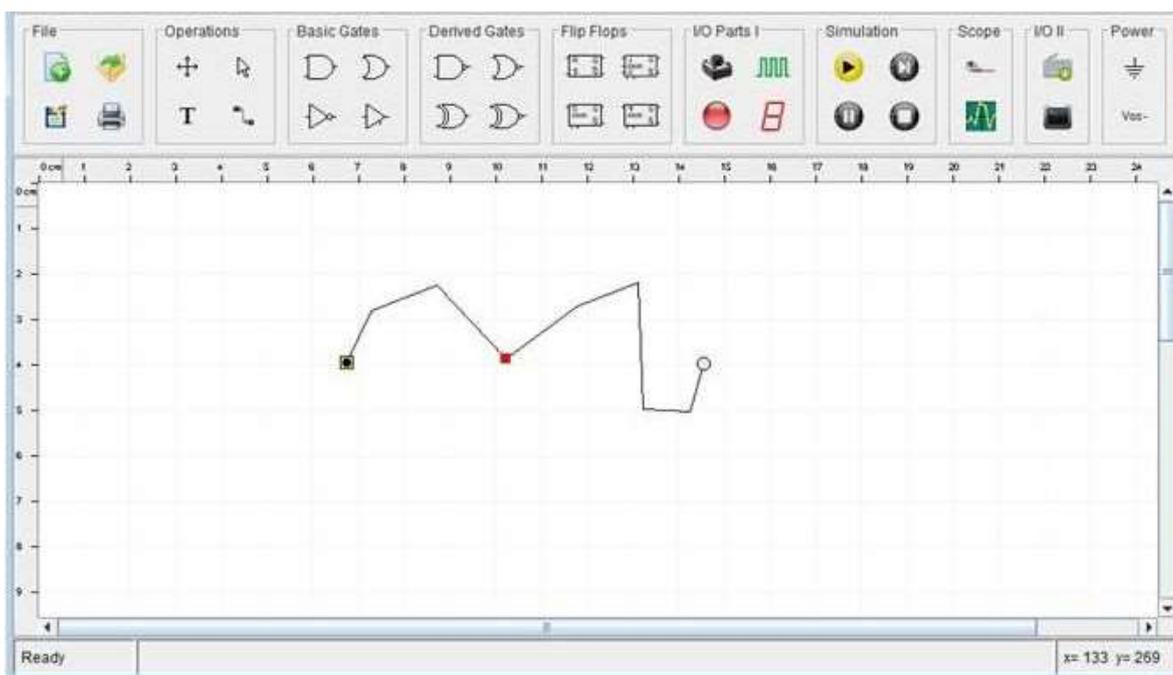
1. When you bring the Track drawing cursor new to a pin or a Track, the tool may change its color from “Yellow” to “Green” . It means that you are allowed to connect to this Pin or Track. If it does not change its color, you are not allowed to connect with this pin or Track, because:
 - a. You cannot connect two output pins to the same Track. A Track connects to only one output pin at a time.
 - b. When a pin is connected to a Track, you cannot directly make any new connection to this pin but only to the track connected to this pin. If you want to connect more than two pin, first connect any two pins through a Track, then connect all other pins to this Track by drawing Track between each Pin and the Track.

Track Drawing and Editing

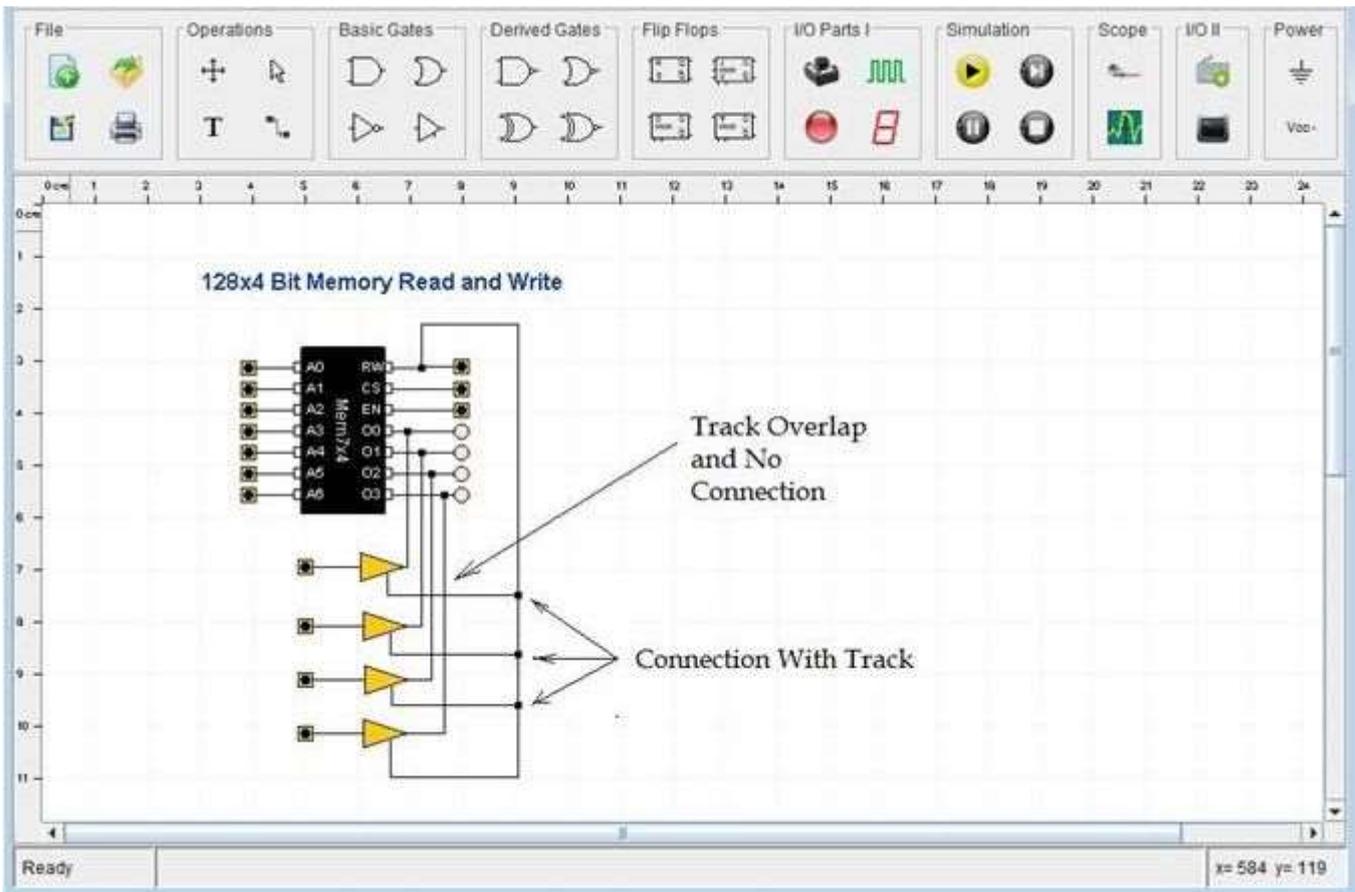
You may draw straight line Track or you may draw segmented Track. During Track drawing if you left-click the mouse, the previous Track line will stop there and a new Track line will start from this point onward. The click point becomes the drag-point of the Track to change its shape. In this way you can draw Track of any shape composed of a number of straight line segments.



If you want to align the Track, Click the Ready button  in the Operation Tool Box in the Tool Bar and then click the Track. The Track will show all the drag-able points, in BLACK boxes, on the Track which you can use to change the shape or alignment of the Track. Use the mouse for dragging or just click the drag-able point, and it will change its color to RED, on the Track and use arrow key to change the alignment of the Track.



It is recommended that instead of drawing long straight Track; draw the track in a number of small segments, by clicking at various places along the path of the Track to introduce drag-able points in the Track, which will be helpful to adjust the shape and alignment of the Track at later stage of circuit design.



When a Track makes a connection with another Track, a BLACK box will appear at the point of connection and if a Track just passes over another Track, no such box will appear.

Component PIN Types

There may be four different types are pins in a Component:

- Input** Used to receive data from other components.
- Control** These are Input Pins but trigger special operation inside the component
- Output** Used to send data out of the component to other component.
- TRI-State** These are Output Pin but they can be put in High Impendence mode by EN pin. In this mode it is like cut-off from all other components. All those components which have EN or output enable control pin, their output pins become TRI-State pins.

Note: A Track can have only one Output pin but can have any number of Input and TRI-State pins.

9 - Simulation



After completing a circuit, you can run the simulation to check the working of the circuit. In simulation you may change or enter values in your circuit through input parts and observe the behavior of your circuit through output parts or on a digital Oscilloscope provided in this software. There are four buttons associated with the simulation of any circuit.

 **Run** the simulation

 **Stop** the simulation

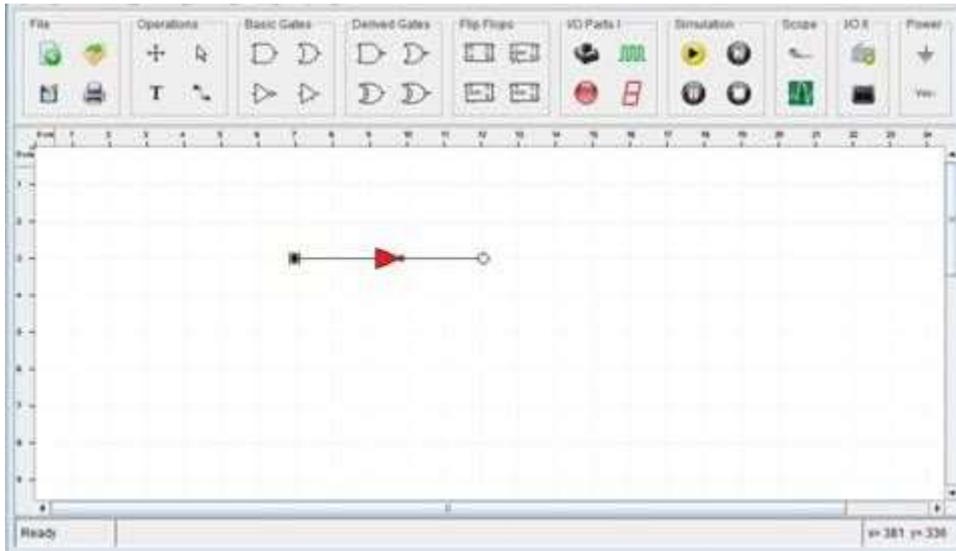
 **Pause** and **Resume** the simulation

10 - How to use a Digital Scope

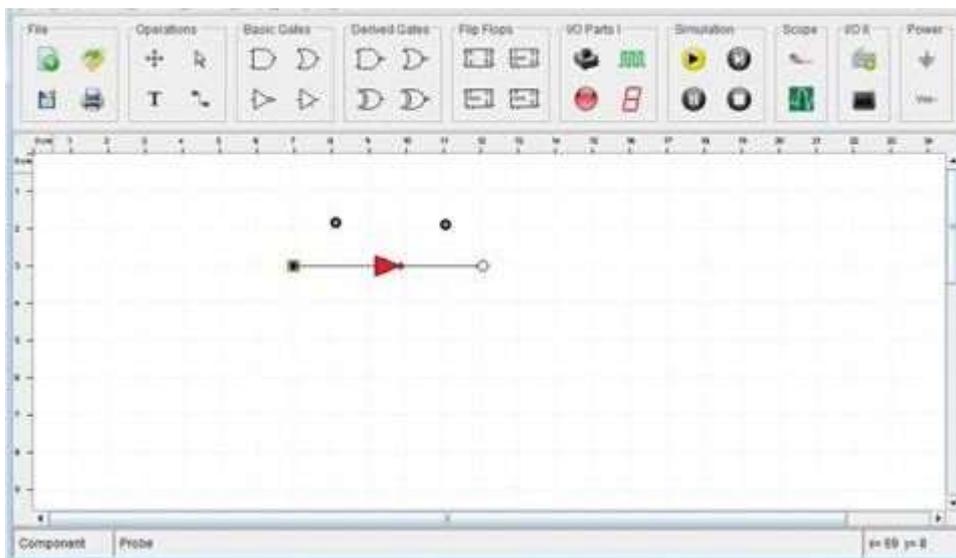
The Digital Oscilloscope or Scope provides the visual information regarding signal level at different points in a circuit. The Scope works in combination with the probe which marks the places where we monitor the signal level during running of the circuit.

Step-by-Step

1. Design the circuit or open already designed circuit.

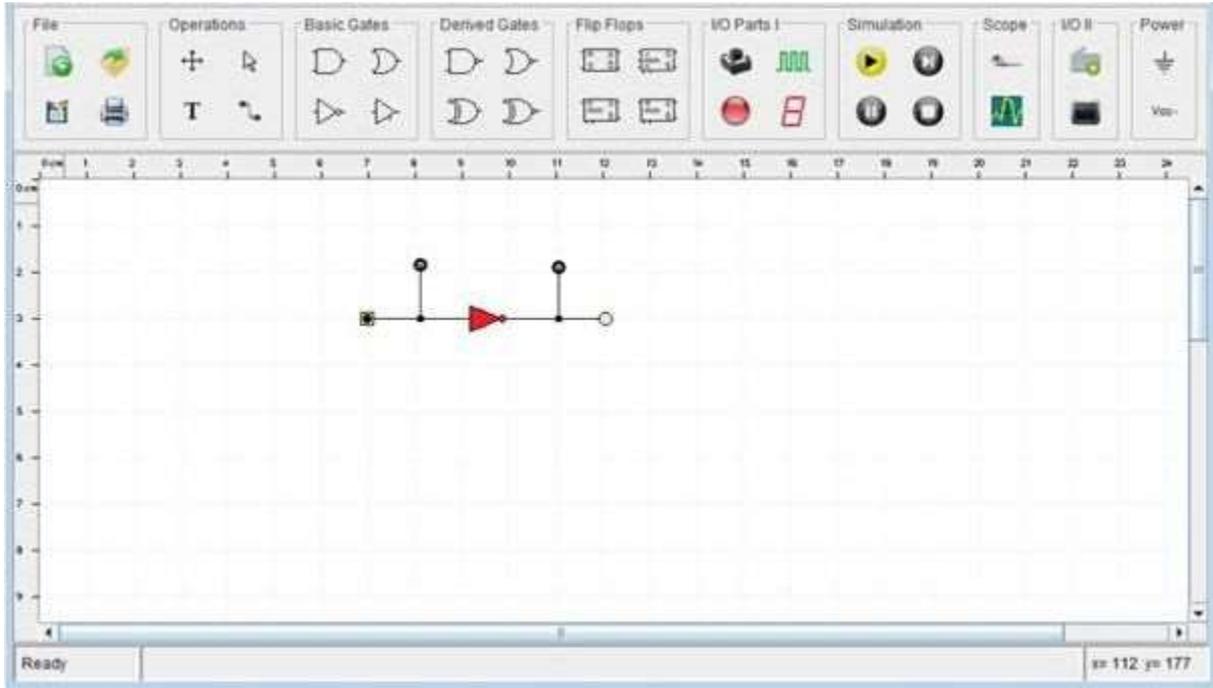


2. Decide about the points where you want to monitor the signal level in the circuit during its operation.
3. Now click the “Probe”  icon the “Scope” tool box in the tool bar at the top of the window.
4. Now move to the drawing area. The cursor will change its shape to a Black circle .
5. Click the mouse near the place where you want to monitor the signal level. A back Circle is placed at that place.
6. Mark and place the probe at all the places in the same way

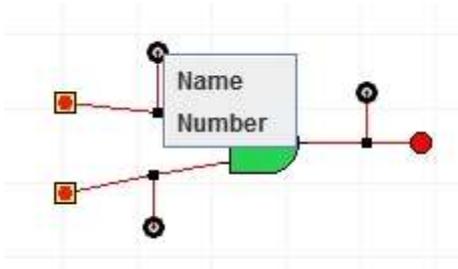


7. Now click the “Track”  icon in the “Operation” tool box in the tool bar at the top of the window.

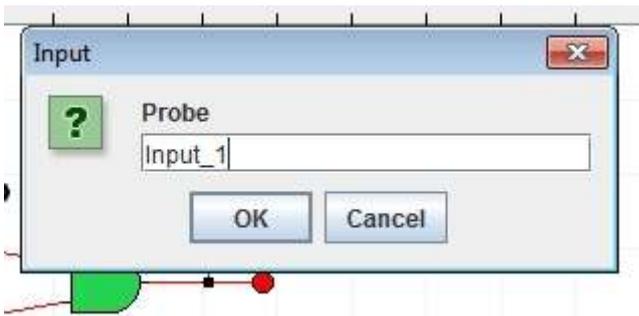
8. Move to the drawing area and the cursor shape will be changed to the track drawing tool .
9. Drawing the track between the "Probe" marking and the monitoring places.
10. Connect all the "Probe" markings in the same way.



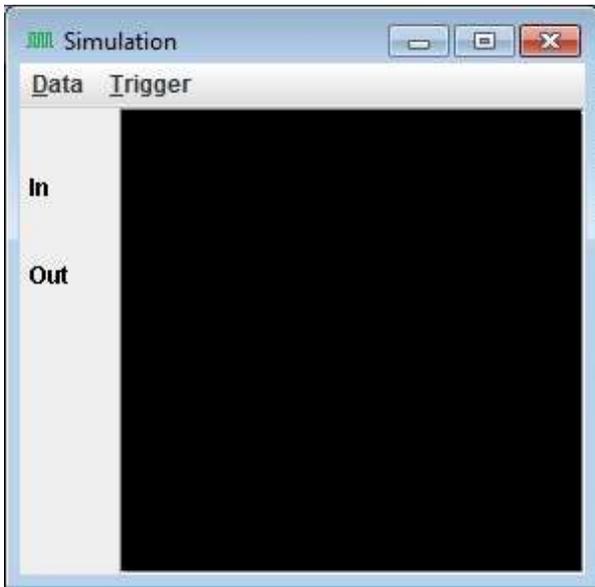
11. Assign names to the "Probe" marking, by using each Probe drop-down menu.



12. A dialog will open to ask the name of probe. You may assign any meaningful name to the probe.

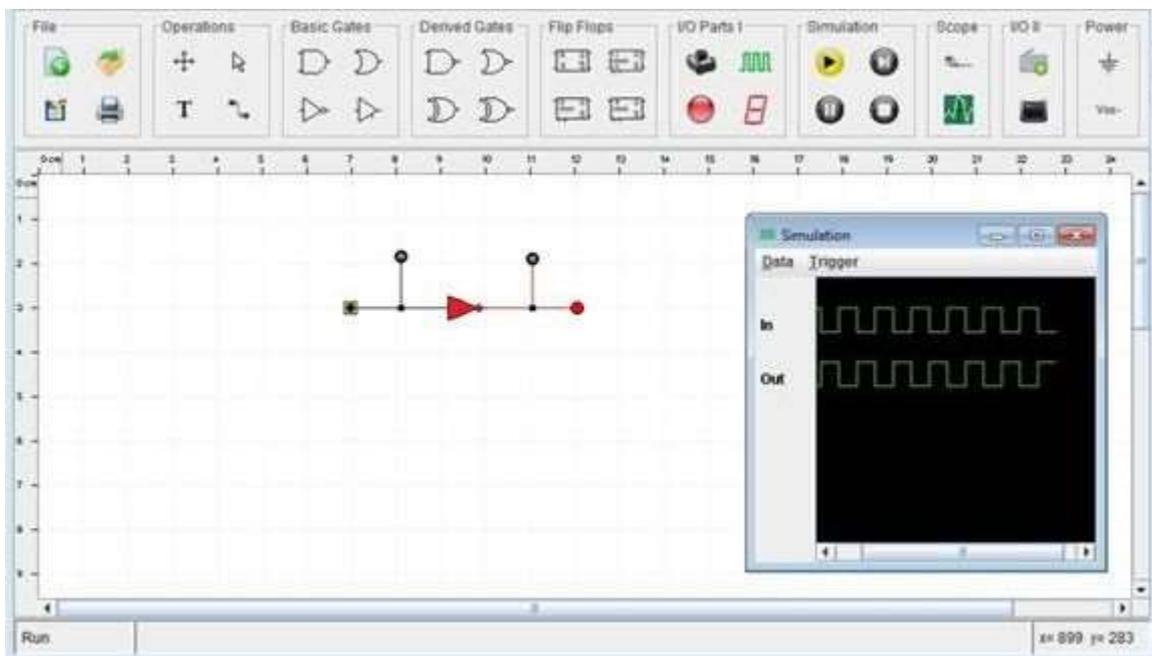


13. By using the drop-down menu of each Probe, assign the Id number to each "Probe" marking according to the order in which you want them to appear in the "Scope" display.
14. Now click the "Scope"  icon in the "Scope" tool box in the tool bar at the top of the window.
15. A new Scope Window will be opened showing all the signal names assigned to the "Probe" marking and as per their assigned number value in ascending order.



16. Now run the simulation.

17. All the signal information will start displaying in the Scope window.



18. You can save the scope data appearing in the scope by selecting menu Data->Save Data.

19. You can also view the already save scoped data in the Scope by selecting menu Data->View Saved Data.

20. You can also set that at which value the scope "Start recording the data", "Stop recording the data", and "How many values" it must record before stop recording and displaying the signal information. These settings are available in the "Trigger" menu of the simulation window.

11 - How to make a custom Modules or ICs (Integrated Circuits)

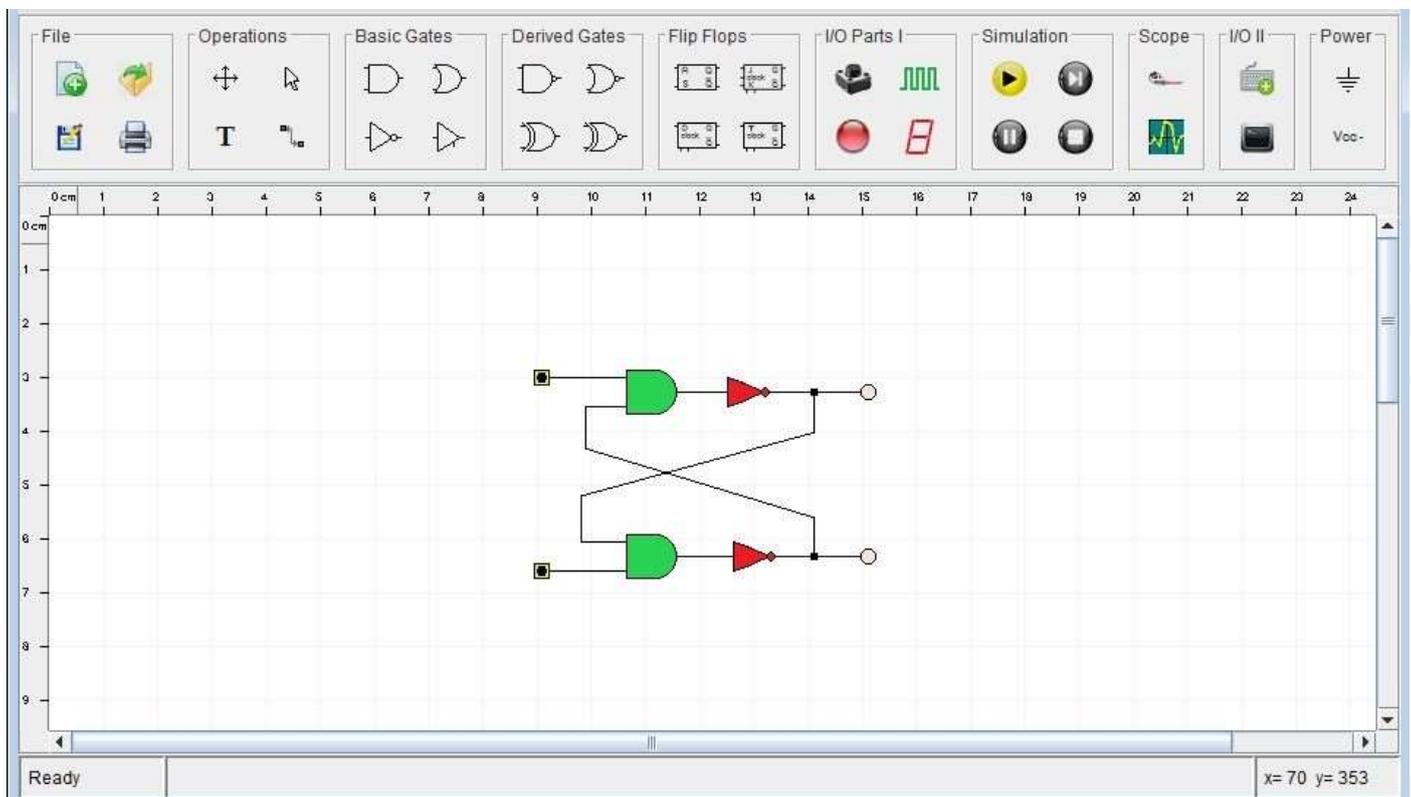
A circuit can easily be converted into reusable Module which can be used to build more complex circuits. They behave like ICs in a circuit.

How to convert a circuit into a Module

When a circuit is converted into a module, the Button and LED present in the circuit are replaced by input and output pins of the Module. The name of these parts becomes the name of the pin while their Number becomes the pin number which determines the position of this pin in the image of the newly created Module.

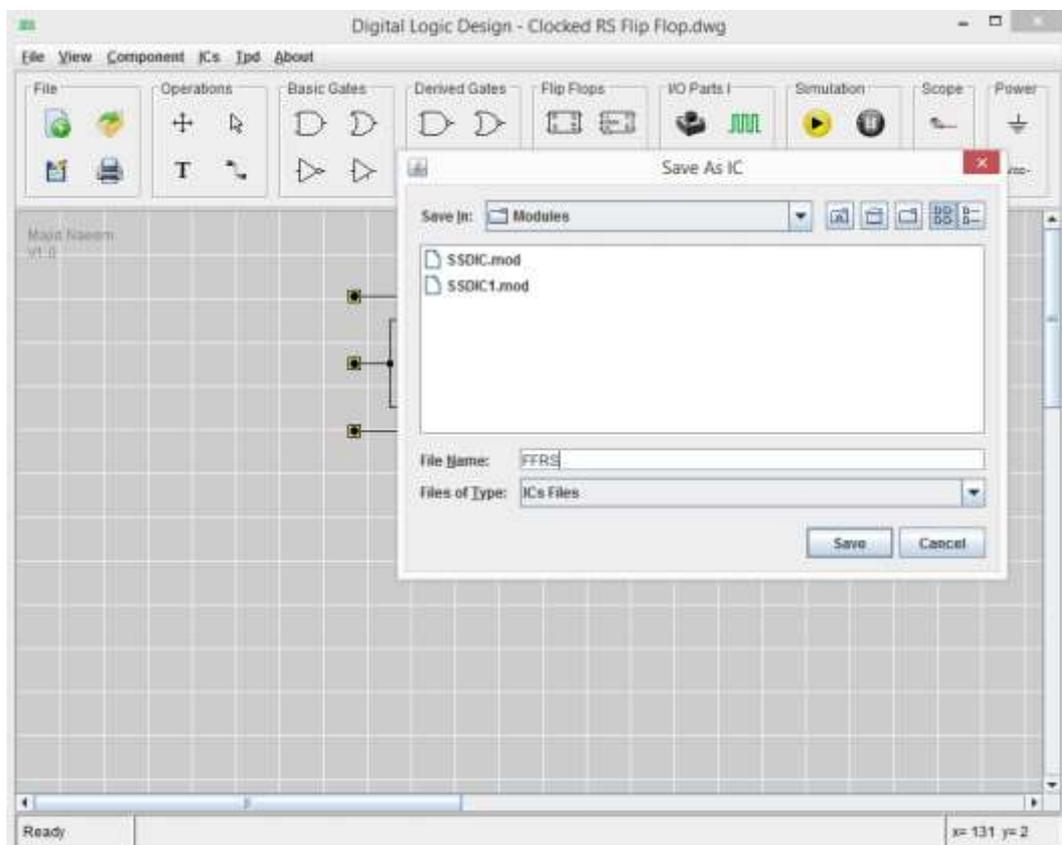
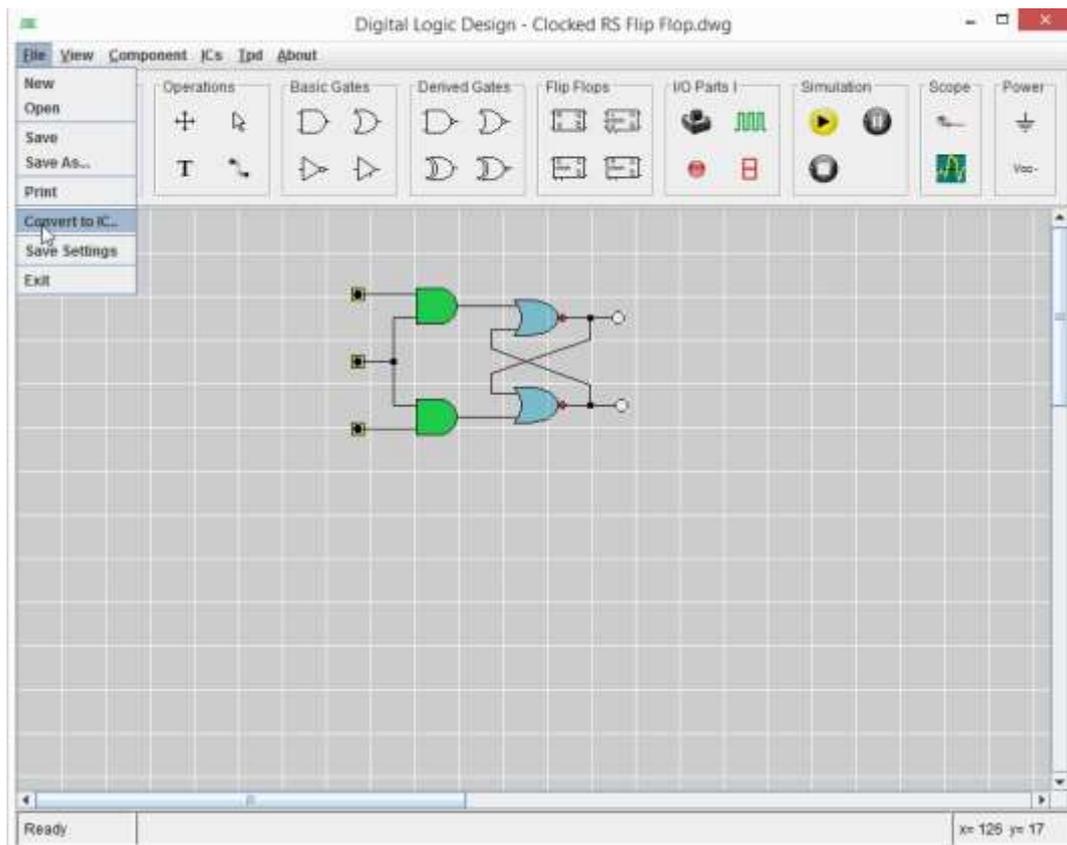
Step-by-Step:

1. Design the circuit for the Module.
2. Place and connect Button on all the places which will appear as input pins of the Module.
3. Place and connect LED on all the places which will appear as Output pins of the Module.
4. Assign proper name, using drop-down menu, at most three characters long, to these Button and LED.
5. Assign Number to these Button and LED, using drop-down menu, according to their placement on the Module image.

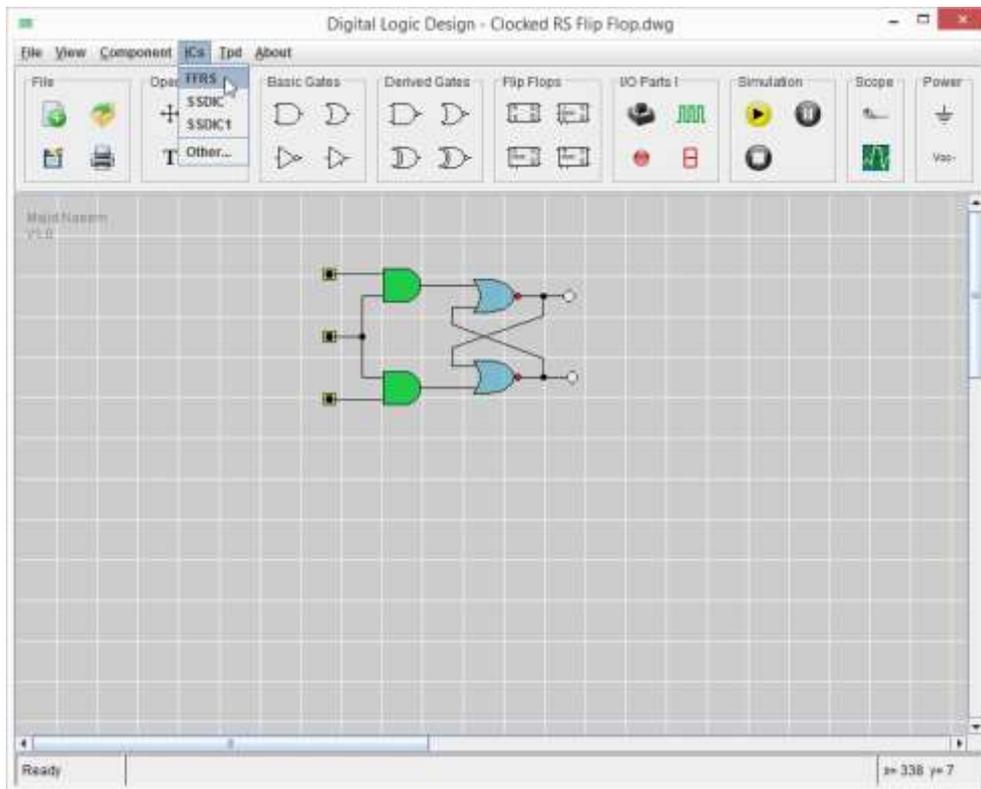


6. If you want to place a “Not Connected” or NC pin, just place a Button or LED in the circuit diagram and assign it name “NC”, and a number as per its position but do not connect it to any other Track or pin.

- Now select menu File-> Convert to IC... and select the name and folder for the newly generated Module. The new Module will be generated.

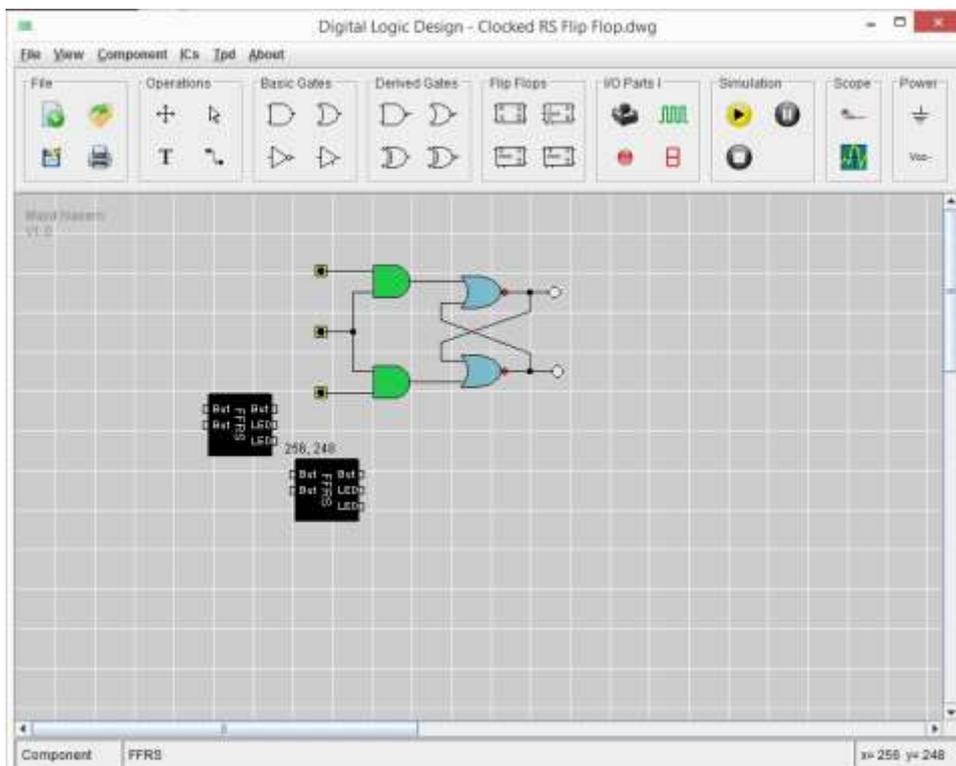


- All the Modules which are placed in /Module folder will appear in Module menu of the program.



9. The Modules which are stored in any other folder can be access using “Other” option in Module menu.

A module can be used as many times as needed in a circuit. All such instances of a module behave as standalone and separate parts. Any modification in one instance of a module does not affect any other instance of same module and, therefore, later changes in the module design will not affect the already used instances of such module, until and unless, manually replaced with the modified module.



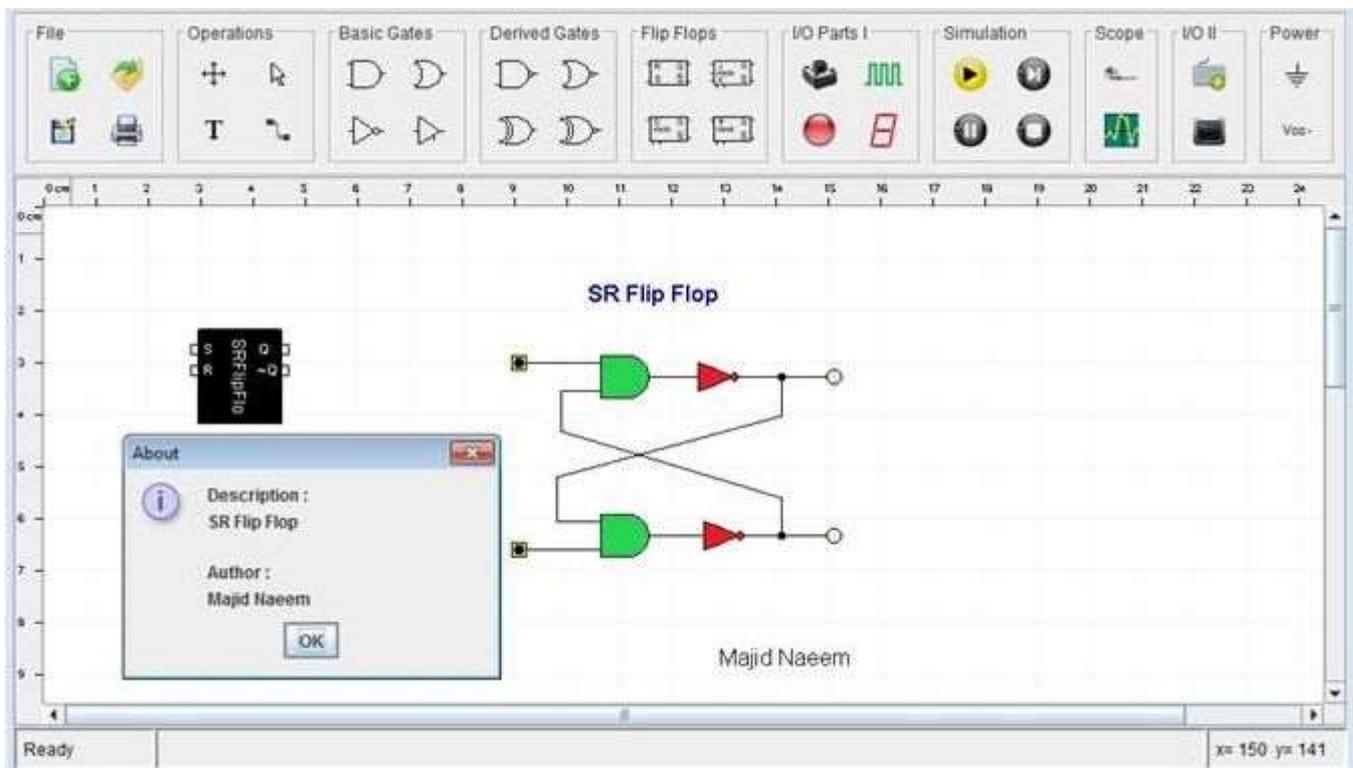
Suggestions

The following suggestions are not bindings but useful for designing better reusable Modules.

- If there is a “Clock” component in the circuit, replace it with a Button and provide “Clock” signals externally to the Module.
- Remove / replace all other Input/output parts from the circuit as they serve no useful purpose in a module.
- When making such changes, always modify a copy of that circuit file and not the original circuit diagram.
- If there is some component already available in the software, like flip-flops, use these parts instead of designing new module for same functionality.
- The Designer name and Module Description appears in the context menu about of the Module. If the module uses any other module, the name of the Designer of that Module will also be shown. The Description is added by introducing a Text field, named D into the circuit diagram of the Module.

Adding Description in Module

The Designer name and Module Description appears in the context menu “About” of the Module. If the module uses any other module, the name of the Designer of that Module will also be shown. The Description is added by introducing a Text field, named “D” into the circuit diagram of the Module.



Step-by-Step:

1. Insert a Text field into the circuit diagram and enter the Description into it.
2. Right-click this Text field and assign the Text field Title as “H”

3. This Text field can be placed anywhere on the circuit diagram.
4. Save the circuit and generate the Module.

12 - How to make a State Machine

It is a component whose next output depends upon its current input value and on current output value.

(Current_Input, Current_Output) -> Next_Output.

A user must provide a data file to provide conversion value. This file may be placed in /Data folder under the application folder. This is a simple text file having **.smd** extension. The data is provided in the following format in this file:

```
<Input_Value><Space><Output_Value><Space><Next_Output_Value>  
..  
..  
..  
END
```

Input value, output values and next output values are integer values separated by single space. If "clr" or Clear or Reset pin is available, the first Output_Value in the file is treated as reset or clear value.

13 - Generating ICs from Boolean Expressions

User can also generate a custom component / IC by providing Boolean expressions for the required custom components. The Boolean expression is provided in Sum-or-Product form for each output pin of the component.

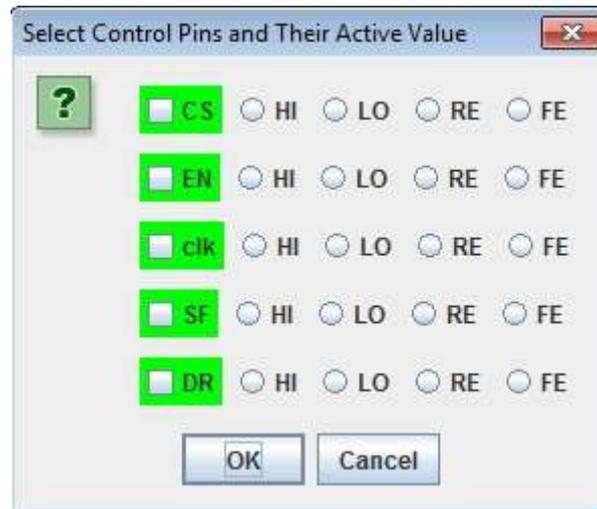
User must provide Boolean Expression(s) in a data file. This file may be placed in /Data folder under the application folder. This is a simple text file having .exp extension. The format of the file is given below:

```
<Name_of_The_Component>
<Number_of_Input_Pins><Space><Number_of_Output_Pins>
<Name_of_Input_Pins_Separated_by_Space>
<Name_of_Output_Pins_Separated_by_Space>
<Minterms_of_Boolean_Expression_Separated_By_Space_for_Output_Pin_0>
<Minterms_of_Boolean_Expression_Separated_By_Space_for_Output_Pin_1>
...
<Minterms_of_Boolean_Expression_Separated_By_Space_for_Output_Pin_n>
END
```

Minterm are provided in decimal form separated by single spaces. The Boolean Expression in the form of Minterms will be provided for each output pin in ascending order on separate line. Pin names may be three characters long which may include the pin number if the designer does so. The input and output pin quantity, i.e., number of input pins and number of output pins, are provided as integer values separated by space. Name of the component should be a single word name.

14 - Adding Control Pins in ICs

Control pins add special features in a component. Control Pins can be added and removed from a component through its context or drop down menu under Control Pins option by selection and deselecting the relevant checkbox. The active value for these pins is also set in the same menu.



The types and purpose of three common control pins are given below:

- EN** Output enable pin. It is used to enable or disable output of a component. When output is disabled, the component output goes into high-impedance (Z) mode.
- CS** Chip select pin. It is used to select or deselect a component. If a component CS is disabled, the component performs no operation.
- clk** Clock pin. The purpose of this pin is to synchronize a component operation with other components. In Counter components it is used to increment or decrement a counter. A counter already has "clk" pin which is "RE" or rising-edge active

There are some special control pins specific to different components as given below:

- clr** Clear or reset pin which is active "HI". In Counters, it resets counter value to 0.
- DN** Down-counting pin which is active "HI" and it is available for Counter components to enable down counting of each clock pulse.
- LD** Load Counter pin. It is used with Counters and Registers to directly load some data into the component. It is usually set as active on "RE" or "FE".
- SF** Shift control pin adds data shift property in the Register.

DR Direction control pin determines the direction of data shift as left shift or right shift in a Register.

Every control pin can be **set active** on any of the following signal values:

HI High value, 1.

LO Low value, 0.

RE Rising Edge, when value is changing from LO to HI.

FE Falling edge, when value is changing from HI to LO.

15 - Memory

The user must provide the length of a word in Bits as Data size and also the size of the memory in number of locations. These locations are accessed through address pins whose name starts with “A” followed by the pin number. The data appears on output pins whose name starts with “O” followed by pin number.

Data can also be load into the memory using a file in text format. The file may be placed in /Data folder. This is a simple text file having .mem extension. The format of the file is given below:

```
<Memory_Location><Space><Data_Value>>
..
..
END
```

Memory Location and Data value integer values separated by single space.

Memory Read Cycle

1. CS->LO , RW->LO Set CS and RW control line to LO, i.e., their de-active value
2. Set Memory address
3. CS->HI, EN->HI Set CS and EN control line to HI, i.e., their active value
4. CS->LO

Data from the given address will be appear on output pins

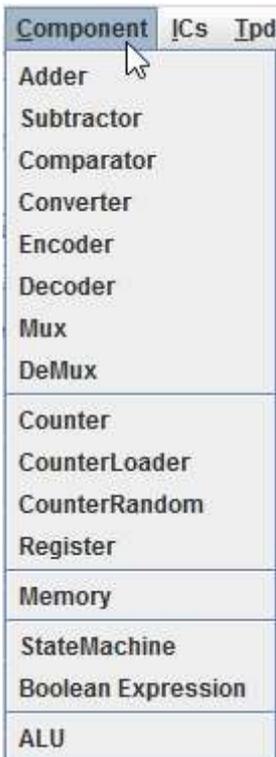
Memory Write Cycle

1. CS->LO
2. Set Memory Address
3. Set Data on Data Line through TRI State Gate
4. RW->HI, CS->HI Set CS and RW control line to LO, i.e., their active value
5. CS->LO

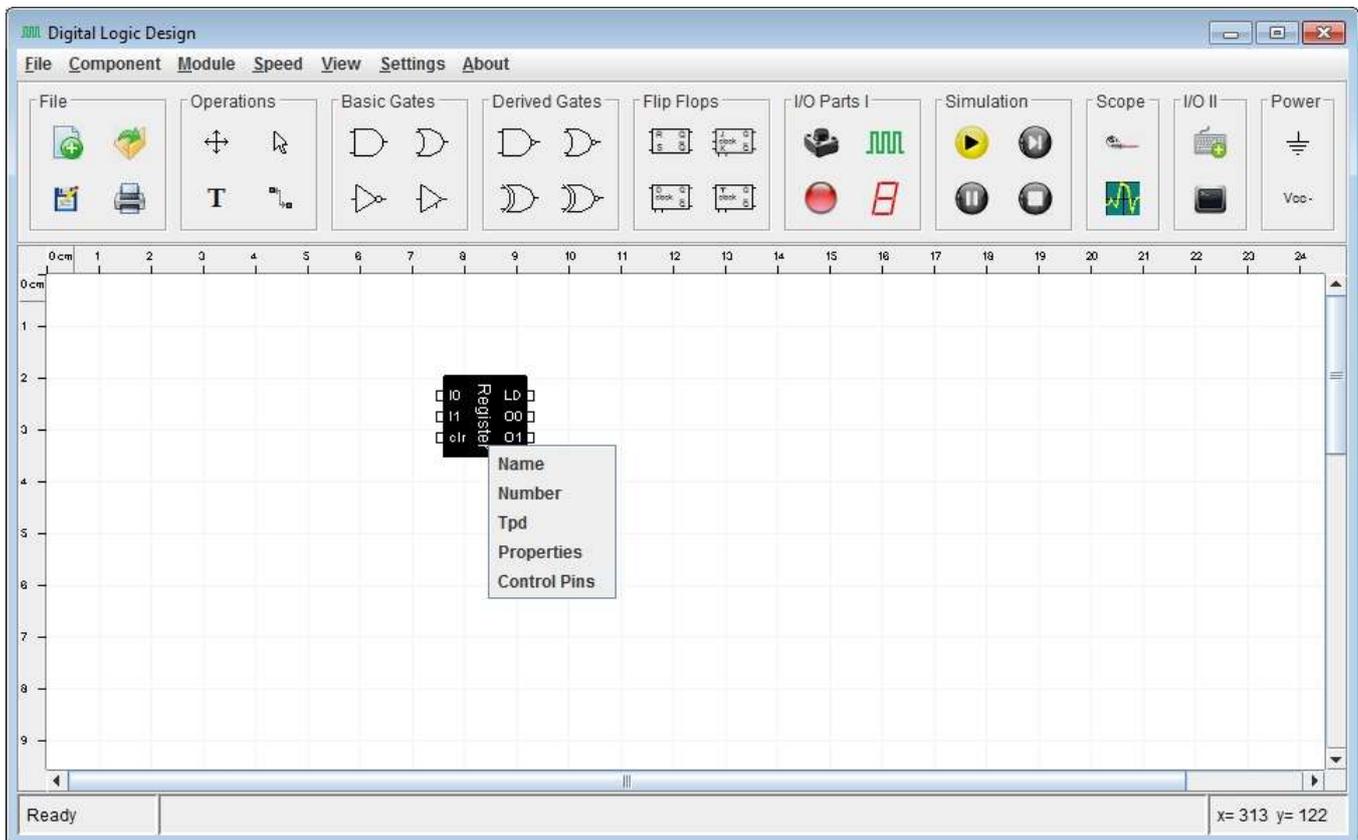
Data will be written in the given location

16 - Components Parameters

Component Menu provide a list of all available standard parts or component, other than that available in the Tool Bar, which can be customized and used for the circuit designing.



A component may have a number of parameters, like name and id number, which may be set by using component context or drop down menu in a circuit drawing. This menu appears when a component is right-clicked with the mouse. Components may have different parameters depending upon their type. Some components have more parameters which may be set while some has less. Some parameters like name and id are common for all the components while some are available for specific components. This section provides the details regarding parameters of components which may be set by the user.



There are three common parameters for each component:

- Name** It is single word meaningful Name of the Component assigned by the designer. It is optional but required for Probe, Button and LED component.
- Number** It is an integer number, starting from 1, assigned to a component. It is a good practice to use unique numbers for every component or at least the number should not be repeated for same type of components. It is optional for most of the components but required for Probe, Button and LED component. When converting a circuit into a Module, it is strongly recommended that Button and LED should be assigned unique numbers in ascending order according to their pin position in the newly created Module.
- Tpd** Propagation delay. This value is option for all the components and only set when some critical timing analysis of a circuit is required. It is needed for very advance level design. A propagation delay of 1 equals to one clock cycle, i.e., the value of 5 means a propagation delay of 5 clock cycles. It is an advance concept and should used with lot of care. Normally components have no propagation delay in this software.

There are three more parameters specific to components:

Control Pins	You can add or remove control pins in a component. The control pins are discussed in a separate section.
Properties	It varies from one type of component to other. It is used to set number of input and output pins or size of value in bits according to the component type.
Load	It is used to load some data into the component. This feature is discussed in relevant component section.

The parameters specific to components are discussed below with relevant components:

Adder

Properties: User is required to set the size of the operands, in Bits, which are going to be added. A size of 4 means, there will be two operands of 4 bits each and the result is also of 4 bits long. Operands are represented by pins whose name started with “a” and “b” while output pins have name stated with “O”. ($O = a + b$). The overflow or Carry pin “C” is **HI** when there is some carry.

Subtractor

Properties: User is required to set the size of the operands, in Bits, which are going to be subtracted. A size of 4 means, there will be two operands of 4 bits each and the result is also of 4 bits long. Operands are represented by pins whose name started with “a” and “b” while output pins have name stated with “O”. ($O = a - b$). The sign bit “s” represent the sign. If $b > a$ then sign bit is set **HI** and remaining output pins show the magnitude.

Comparator

Properties: User is required to set the size of the operands, in Bits, which are going to be compared. A size of 4 means, there will be two operands of 4 bits each and the result is of 3 bits long. Operands are represented by pins whose name started with “a” and “b” while output pins have name stated with “O”. The result of comparison is shown by the **HI** value on output pin named as “a > b”, “a=b” and “a<b” for respective result.

Converter

Properties: User is required to set the number of input and output pins.

Load: A user must provide a data file to provide conversion value. This file may be placed in /Data folder under the application folder. This is a simple text file having **.con** extension. The format of the file is given below:

<Input_Value><Space><Output_Value>

..

..

END

Input value and output values are integer values separated by single space.

Encoder

Properties: User is required to set the number of input pin.

Decoder

Properties: User is required to set the number of input pin.

Mux

Properties: User is required to set the size of an input Chanel, in Bits, and also set the number of channels required in the Multiplexer. A channel represents a value or operand. Input channels are represented by pins whose names are started with **A, B, C...** followed by pin number. The output channel pins names are started with **"O"** followed by pin number. The channel selection pins names are started with **"s"**.

DeMux

Properties: User is required to set the size of an output Chanel, in Bits, and also set the number of channels required in the Multiplexer. A channel represents a value or operand. Output channels are represented by pins whose names are started with **A, B, C...** followed by pin number. The channel selection pins names are started with **"s"**. The input channel pins names are started with **"i"** followed by the pin number.

Counter

Properties: User is required to provide the size of the counter in Bits.

CounterLoader

Data is provided on the data input pins, whose name started with **"D"** followed by the pin number, to be loaded into the counter.

Properties: User is required to provide the size of the counter in Bits.

CounterRandom

Properties: User is required to provide the size of the counter in Bits.

Load: A user must provide a data file to provide random values in the same order as may be appear on the counter output. This file may be placed in /Data folder under the application folder. This is a simple text file having **.crd** extension. The format of the file is given below:

<Random_Value>

..

..

END

Random value is an integer value.

Register

Properties: User is required to provide the size of the Register in Bits.

ALU

Properties: User is required to provide the size of the ALU in Bits.

ALU Operation

S0-S3

**Value in
Decimal**

Operations

0	NOP	
1	Add	
2	Subtract	
3	Increment	A
4	Decrement	B
5	~A	
6	~B	
7	A << B	
8	A >> B	
9	AND	A&B
10	OR	A B
11	XOR	A^B
12	XNOR	~(A^B)
13	NAND	~(A & B)
14	NOR	~(A B)
15	ALL High	

