

COA Virtual Lab adapted for web based usage

1. Overview

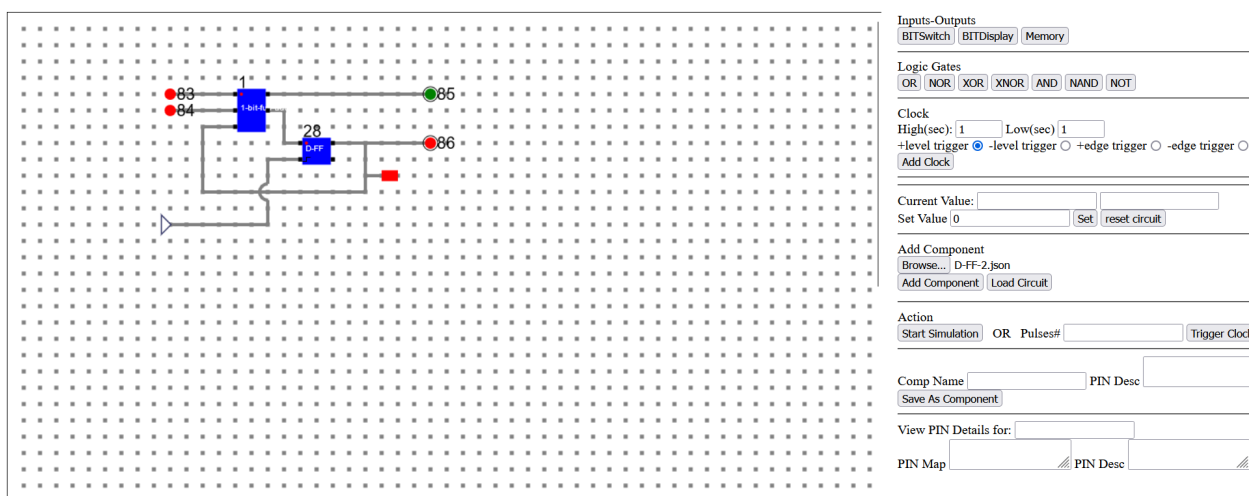
Web Based Virtual Lab (WBVL) developed so far is ready to perform any experiment with combinational as well as sequential logic. User needs to add necessary logic GATEs, connections and bit-switches (to represent input, output and memory) in the circuit and then click 'Simulate' button. User can also design sequential circuit and add clock and the click 'Trigger Clock' button to simulate the circuit with one clock pulse.

2. Technical Insight

The Virtual Lab has been developed using HTML 5 CANVAS and JavaScript. Two overlaid canvases have been used to provide the main framework of the simulator. The bottom layer canvas is mainly generating the breadboard. The top layer is generating the circuit. The circuit needs to be redrawn every time a new element is added/deleted or state of the circuit is changed. But the breadboard part remains intact throughout the experiment. This design strategy provides a faster response with even a complex circuit.

3. Manual

A breadboard like panel is given with dotted horizontal and vertical lines to draw the digital circuit. The control panel for the digital circuit is provided at the right hand side of the breadboard panel. The interface looks like the below picture.



The screenshot displays a digital circuit simulator interface. On the left, a breadboard is shown with a circuit diagram. The circuit includes a 7401 NAND gate (pin 14) and a DFF (Data Flip-Flop) component (pin 28). Inputs are connected to pins 83 and 84, and outputs are connected to pins 85 and 86. A clock pulse is applied to the DFF. On the right, the control panel is visible, featuring several sections:

- Inputs-Outputs:** Includes buttons for BITSwitch, BITDisplay, and Memory.
- Logic Gates:** Includes buttons for OR, NOR, XOR, XNOR, AND, NAND, and NOT.
- Clock:** Includes fields for High(sec) and Low(sec) (both set to 1), and radio buttons for +level trigger (selected), -level trigger, +edge trigger, and -edge trigger. An Add Clock button is also present.
- Current Value:** Includes a text input field and a Set button, along with a reset circuit button.
- Add Component:** Includes a Browse... button (set to D-FF-2.json), an Add Component button, and a Load Circuit button.
- Action:** Includes a Start Simulation button, an OR button, a Pulses# input field, and a Trigger Clock button.
- Comp Name:** Includes a text input field and a Save As Component button.
- View PIN Details for:** Includes a text input field.
- PIN Map:** Includes a table with columns for PIN Map and PIN Desc.

Please find below the detail of each section of the control panel, mouse and how to design a circuit with this tool.

1. Inputs-Outputs

With this section user can add any number of input bits, output bits and memory bits as required in the circuit. At each click of mouse on BITSwitch, one input bit with default value 1 (green) is added to the circuit. With a double click on the input bit the value will be toggled between 1 and 0. Similarly, at each click of mouse on BITDisplay, one output bit with default 'No Value' (grey) is added to the circuit. User can drag the bit to any position in the breadboard to position it as required.

A memory bit is required to hold an output from a device to be used in next clock pulse. This memory bit is very useful to simulate sequential circuits where output and next state depends on the input and previous state. With memory bit we can store the state of the circuit.

2. Logic Gates

With this section user can mouse click on any logic gate to add to the breadboard. Then drag the gate to any place in the breadboard wherever required.

3. Clock

User can design and add any clock to the breadboard as required using this section. User can provide duration for the high level and low level of the pulse in millisecond. User can choose type of the clock as edge trigger or level trigger. Then can click the 'Add Clock' button to add the clock.

Limitations: As of now followings are the limitations.

1. There is no difference between edge trigger and level trigger clock
2. No more than one clock is supported by a circuit.

4. Connection ID and Value Check

If user double click on any connection, the Current value field will show the id and current value of the connection in the section provided for 'Current value'. User can set any initial value of a connection if required using 'Set Value' text box and 'Set' button. User can forcibly reset value of all connections and pins using 'reset circuit' button.

5. Add Component

This section is used to add any previously created and saved circuit as a component or as a full circuit. After selecting the saved file, user needs to click 'Add Component' button to add any complex circuit as a simple 'Chip' with necessary PINs. Thus any previously tested logic circuit could be used in another circuit to build more complex logic. User can click 'Load Circuit' button to change and retest any previously generated and saved circuit.

PINs on the left side of the component are input pins and PINs on the right side of the components are output pins. The RED dot on the component indicate the starting pin (PIN no – 1). The PIN numbers increments vertically down starting from RED dot on the left side and then again increments vertically down on the right side.

The PIN with a pulse symbol is for clock input.

6. Action

User can click 'Simulate' button to start the simulation of a circuit. Or user can provide number of clock pulses as input and click 'Trigger Clock' to start a clock if exists in the circuit. As of now no more than one pulse is tested with this system.

7. Component Save

After designing a circuit, user can give a name of the circuit and purpose for each input, output nodes and then click 'Save as Component' button. This will save the current circuit in a json file in download folder. This file could be used later as module in another circuit. Could be load using 'Add Component' section.

8. View PIN Detail of a Component

After loading a component, if the component is double clicked, the PIN to Bit mapping along with description of the bits what user provided during the saving of the component will be displayed in this section. With this information user will be able to properly connect the PINS of the component with relevant connections.

9. Draw a connection

To draw a connection, hold mouse left button pressed and drag the mouse from one breadboard point to another breadboard point horizontally or vertically.

10. Jump over a connection

To draw connection over another connection without connecting, user can draw a jumper by pressing the 'Shift' key or 'Alt' key while drawing a line from one breadboard point to next third breadboard point horizontally or vertically.

11. Select Circuit Element

To select a circuit element, user need to press 'Ctrl' key and single click on the element.

12. Delete Circuit Element

To delete an element from the circuit, user should select the element as mentioned in above step and then press delete key.

13. Delete Last Element

User can just press 'Ctrl-z' key to delete lastly added element.

14. Designing and managing a circuit

To design a new circuit follow the below instructions

- a. Add necessary basic gates and components using the above instructions. Place the gates and components at suitable positions by dragging them with mouse.
- b. Hold the mouse left button clicked and drag it from one breadboard point to another breadboard point either horizontally or vertically to draw a straight line segment. You can draw multiple such straight line segments one after another to join them and create a long

- connection. Thus connect logic gates and components using these connections. The connections can connect PIN to PIN, Bit to PIN, Pin to Bit.
- User can jump over any connection by creating jumper. Press 'Shift' key or 'Alt' key, hold mouse left button and drag from one point to next third point in horizontal or vertical direction to create a jumper. Likewise user can create connection crossing other connections with help of jumpers.
 - Add input bits to provide binary inputs and output bits to display binary output as required. Place the input bits and output bits on connections as required to provide inputs to the connections and get outputs from the connections.
 - Similar to input bit, add clock as described above and connect it to the specific connection as required.
 - Do Hold State/Clear State setup as required. Default is clear state.
 - Click 'Start Simulation' button or Provide number of pulses and click 'Trigger Clock' as applicable to simulate the circuit and see the output.
 - Do delete any element from the breadboard hold 'Ctrl' button and single click on the element. Thus the element will be selected. Press delete button to delete the selected element from the breadboard/circuit.
 - Pressing 'Ctrl-z' will delete the lastly added element from the circuit.

Once the circuit is completed and tested, the circuit could be saved as a json file using 'Save Component' section. Please do not forget to provide accurate purpose of each input and output bits of the circuit. That will be used when the circuit will be used in another circuit as a component.

4. Sample Experiments

1. Combinational Circuit: 2 input multiplexer (MUX)

Truth Table

	input-1	Input-2	Control Switch	Output
Case 1	0	0	1	0
Case 2	0	1	1	0
Case 3	1	0	1	1
Case 4	1	1	1	1
Case 5	0	0	0	0
Case 6	0	1	0	1
Case 7	1	0	0	0
Case 8	1	1	0	1

A 2 input multiplexer has been designed and simulated as below using all NAND Gates. (Green-1, RED-0)

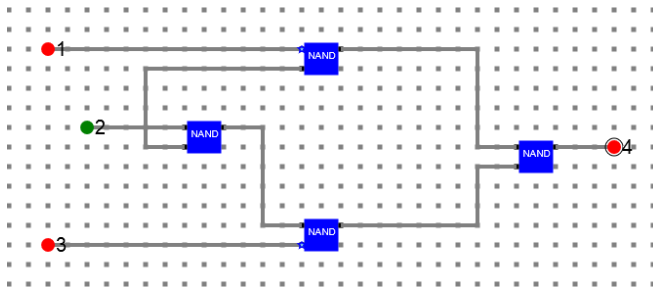
Input-1: Bit-1

Input-2: Bit-3

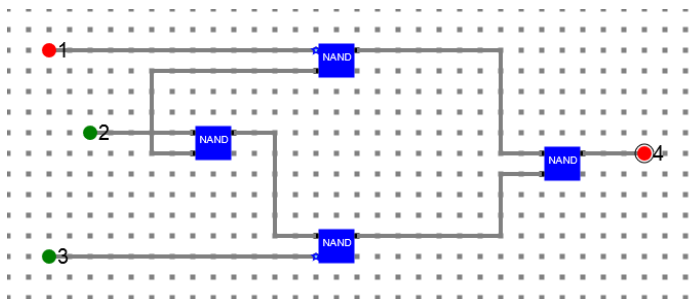
Control Switch: Bit-2

Output: Bit-4

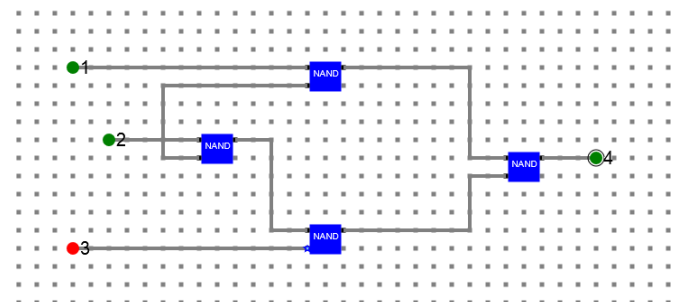
Case 1:



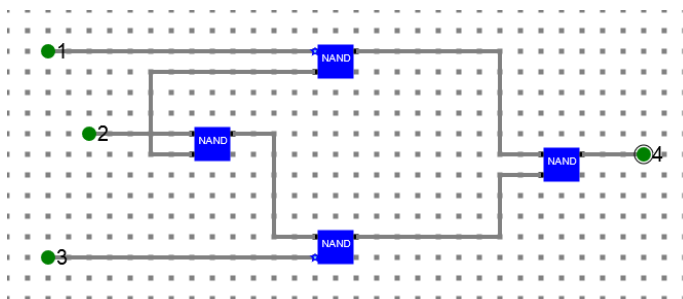
Case 2:



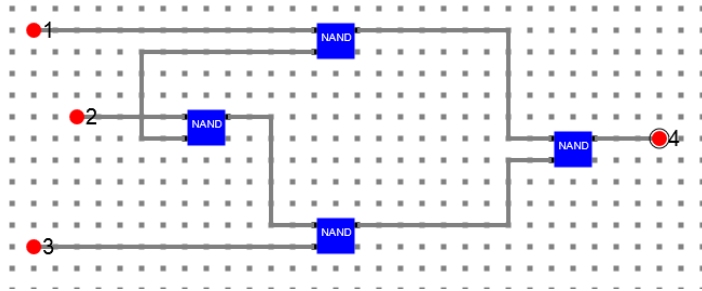
Case 3:



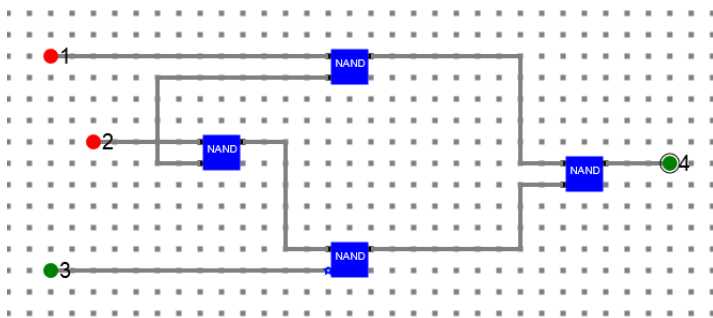
Case 4:



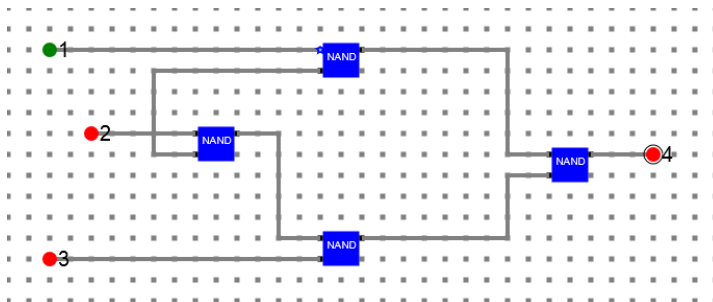
Case 5:



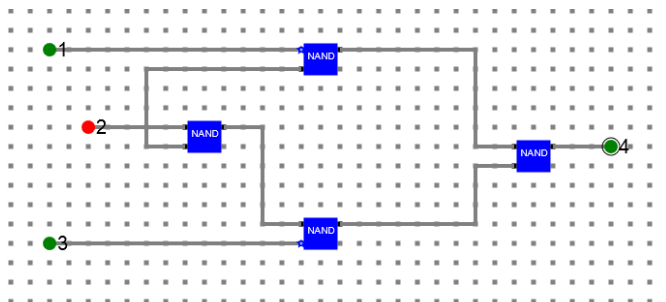
Case 6:



Case 7:



Case 8:



2. Combinational Circuit: Half Adder

Truth Table

	Inputs		Output	
	A	B	S	C
Case-1	0	0	0	0
Case-2	0	1	1	0
Case-3	1	0	1	0
Case-4	1	1	0	1

A half adder has been designed and simulated as below. (Green-1, RED-0)

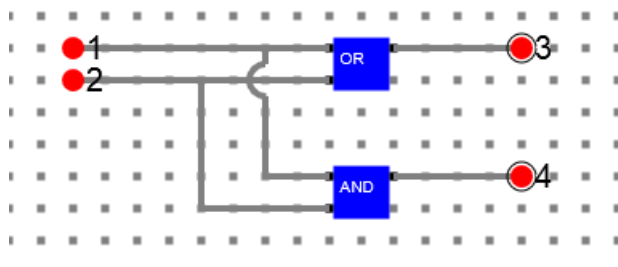
A: Bit-1

B: Bit-2

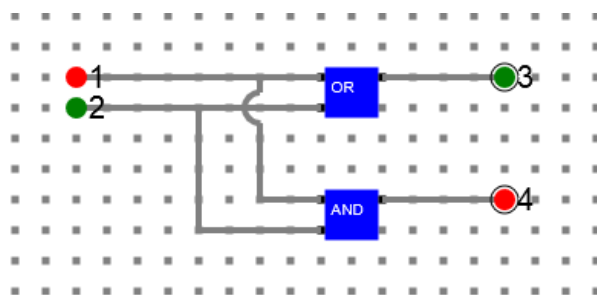
S: Bit-3

C: Bit-4

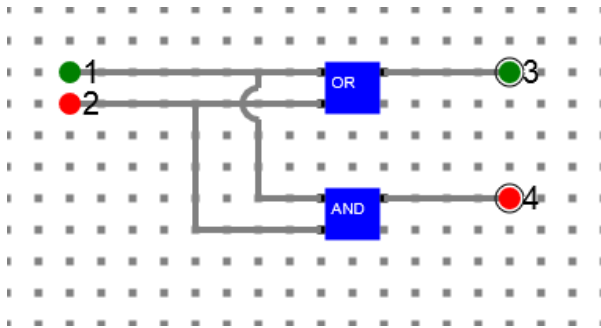
Case 1:



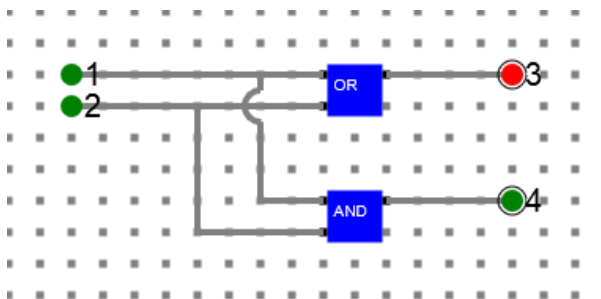
Case 2:



Case 3:

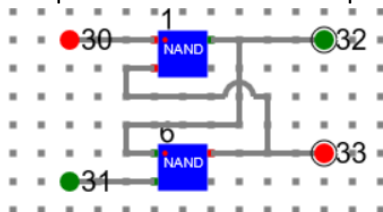


Case 4:

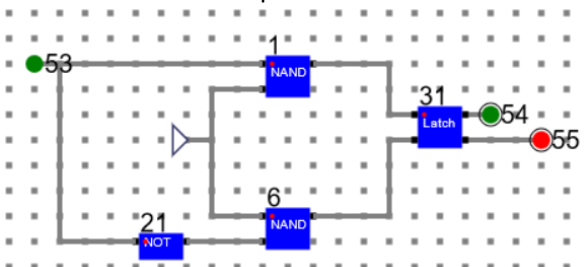


5. D-FlipFlop

To design a D-FlipFlop, first we need a LATCH circuit. Design the Latch circuit as below and make a component. 30 and 31 are input data bits. 32 and 33 are output data bits.



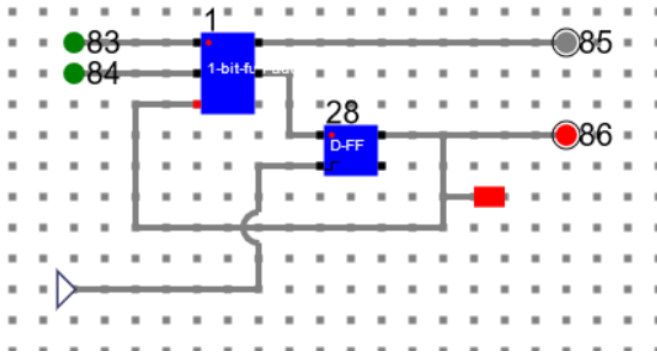
Using the LATCH component, design a D-FF and make corresponding component. Here 53 is input bit. 54 and 55 are output bits.



If 53 is set, the FlipFlop will also set (54:1). If 53 is reset, the FlipFlop will also reset (54:0).

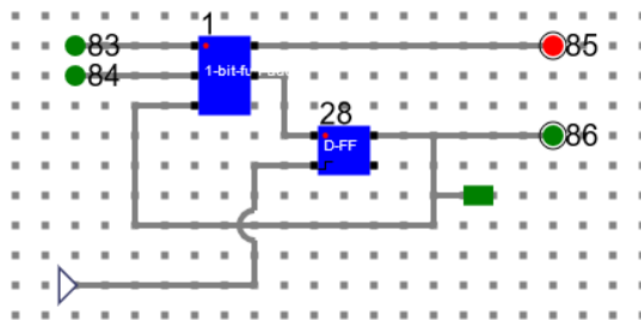
Sequential Circuit: Serial Adder

83, 84 are input bits. 86 is carry bit. 85 is the output bit.

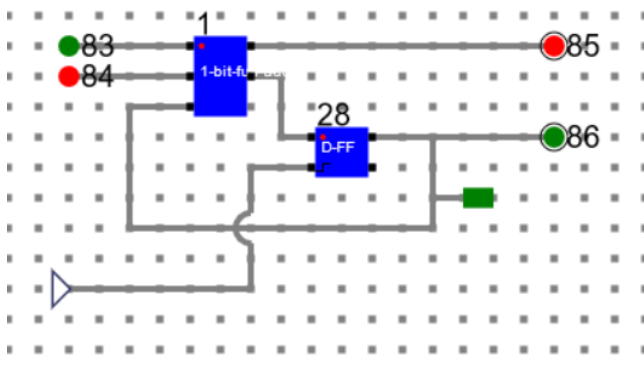


Initially the carry-in is set to 0.

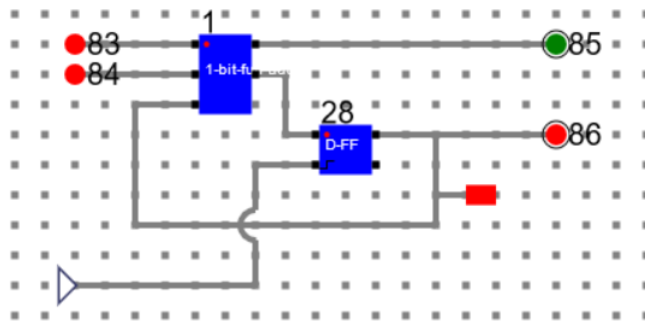
At the application of clock on the above circuit, the output will be 0 and carry-out will be 1.



Let's change the input to 10. Now the carry-in is 1. If clock applied, the output will be 0 and carry-out will be 1.



Let's change the input to 00. Now the carry-in is 1. If clock applied, the output will be 1 and carry-out will be 0.



6. Conclusion

With the current development of the Virtual Lab, any complex combinational and sequential logic could be simulated. There are some limitations on clock design as of now. As of now there is no difference between an edge triggered and a level triggered clock. Also the flow of the circuit is kind of left to right. As the elements are not rotatable at this point. Also no asynchronous FSM yet tested with this simulator. More development and fine tuning of the simulator can improve the user experience with this simulator.