Circuit Simulator

User Guide

If, at any point, you'd like some more in-depth help, there will always be a button in the top right of the screen which will provide help.

View 1: Editor

If you've written out the circuit:

Click on "Browse Files" and load in your text file.

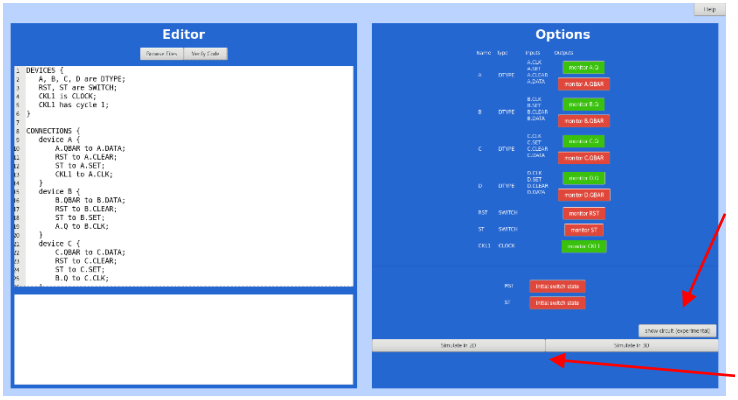
If not:

Write in your circuit to your specifications into the big white box. If you need help with that, just look at our handy guide on the help screen.

Click on "Verify Code". If there are any issues, you'll be told where they are in the bottom white box. Once you're happy that they're fixed, click on it again.

Once there are no issues, you'll be taken to the next screen.

View 2: Options



Here you'll be able to choose which signals you'd like to monitor and at what level you'd like the switches to be set.

Green means being monitored or high (switches), red means not being monitored of low (switches).

You will now be able to confirm everything in your circuit. If you are concerned at all, click "show circuit (experimental)" and you will be shown a diagram of the circuit you have built.

Once you're happy with everything, click "Simulate in" however many dimensions you would like to see. A new window will open with the simulation.

View 3: The Simulation

You've now got to the simulation itself!

The majority of the screen will be taken up by the signal trace itself. On the right you'll also see some options whose meanings are explained below:

Number -+ = This is how many cycles the simulation will run for every time you click continue. You can change this number either by typing in the box or pressing the plus and minus buttons.

Continue = Runs the simulation for however many cycles you've deemed necessary. If no change has been made, it will default to 5.

Switch Name On/Off = Toggles switch positions in the middle of the simulation

Show/Hide 2/3D = Toggles the view of the 2D and/or 3D simulations

Reset Scene = Takes simulation back to initial state, resets everything

View Start/End – Takes the view to the beginning or end

Close Simulation – Closes the window, returns to Options

GF2 Logic Simulator Detail and Sample Files (Shared)

The circuit file is made up of three parts:

- DEVICES
- CONNECTIONS
- MONITORS

In terms of a regular circuit: DEVICES is to select how many of each device type as well as the specifications and names of each device, CONNECTIONS is the wiring up of the different devices together and MONITORS is the decision of where to place the voltmeter to find out what's going on inside the circuit.

Each part starts with the name, followed by an open curly bracket "{", and ends with a close curly bracket "}".

DEVICES

Each line must end in a colon ";". All device names must consist of an integer number of alpha-numeric characters and must start with a letter. Spaces are not allowed.

Each device must first be defined. This can be done in several ways:

- *Device name* is/is a *device type*
- *Multiple device names separated by a comma (e.g. A, B, C)* are/are some *device types*
- *Range of devices* are/are some *device types*

A range consists of a series of devices which end in numbers and have the same string preceding the final numbers. The beginning and end will be separated by the symbol "=>". As such, C1 => C3 will consist of C1, C2 and C3 but C1 => Dandelion3 will be invalid.

Each device must then have its number of inputs defined. This can be done in similar ways to the definitions:

- *Device name* has *number* inputs
- *Multiple device names separated by a comma (e.g. A, B, C)* have *number* inputs
- *Range of devices* have *number* inputs

DTYPES have pre-ordained inputs (clock, reset, clear and data) so there is no need to define those.

For clocks, there is no need to define inputs, as they have no inputs. Simply write the following:

Clock name has cycle *clock cycle time*

CONNECTIONS

Each device must have its inputs done separately. Start talking about a particular device by writing:

Device *device name* {

What will then follow will be a list of everything connecting to the inputs of this device. Input ports for DTYPES are called CLK (clock), SET, CLEAR and DATA. All other input ports will be labelled "I" followed by a number, e.g. I1, I2, I3.

Connections are made in the following way:

*Device name at start of connection** to *device name at end of connection . device port at end of connection* ;

e.g. “Andrew to Fantastic.I1;” will connect the output of Andrew to the first input port of Fantastic.

*For DTYPES, add either “.Q” or “.QBAR” depending on which output being connected.

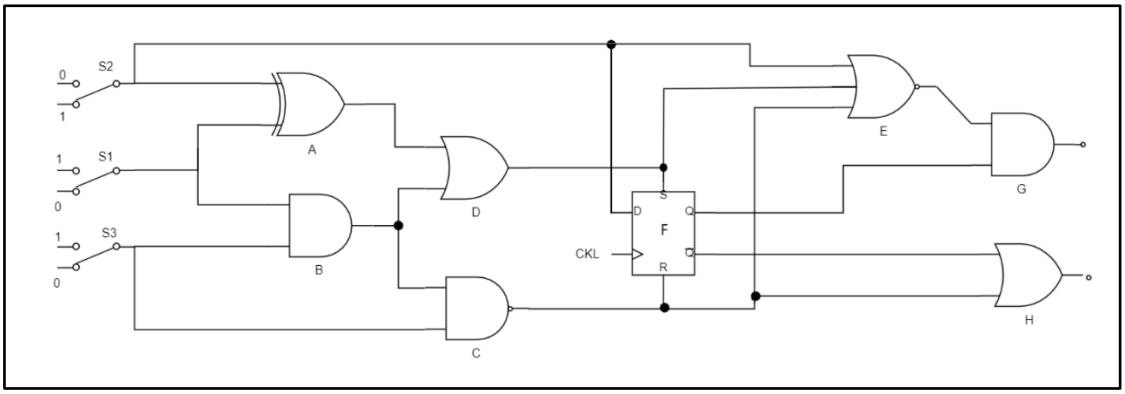
Finish talking about a specific device by writing a closing curly bracket “}”.

MONITOR

Simply list the device names that should be monitored followed by a colon at the end of each line. As with connections, simply add “.Q” or “.QBAR” at the end of the device name.

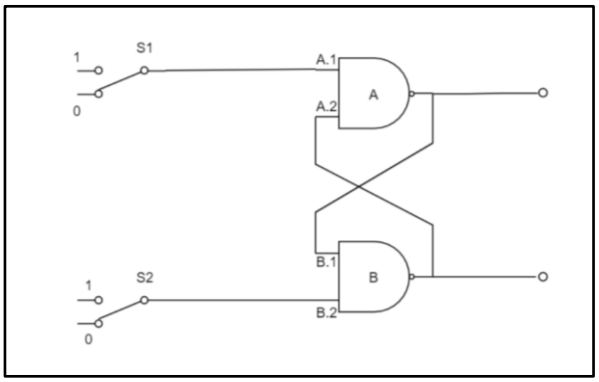
Worked Example 1 – Complex Circuit

```
DEVICES {
  A is XOR;
  B is AND;
  C is NAND;
  D is OR;
  E is NOR;
  F is DTYPE;
  G is NAND;
  H is OR;
  A,B,C,D,G,H have 2 inputs;
  E has 3 inputs;
  S1 => S3 are SWITCH;
  S2 set 1;
  CKL1 is CLOCK;
  CKL1 has cycle 5;
}

CONNECTIONS {
  device A {
     S1 to A.I1;
     S2 to A.I2;
  }
  device B {
     S1 to B.I1;
     S3 to B.I2;
  }
  device C {
     S2 to C.I1;
     B to C.I2;
  }
  device D {
     A to D.I1;
     B to D.I2;
  }
  device E {
     D to E.I1;
     S2 to E.I2;
     C to E.I3;
  }
  device F {
      CKL1 to F.CLK;
      D to F.SET;
      C to F.CLEAR;
      S2 to F.DATA;
  }
  device G {
     E to G.I1;
     F.Q to G.I2;
  }
  device H {
     C to H.I1;
     F.QBAR to H.I2;
  }
}

MONITOR {
  B, F.Q, CKL1;
}
```



Worked Example 2 – SR Flip-Flop

```
DEVICES {
   A, B are NAND gates;
   S1, S2 are SWITCH;
   A, B have 2 inputs;
}

CONNECTIONS {
   device A {
      S1 to A.I1;
      B to A.I2;
   }
   device B {
     A to B.I1;
     S2 to B.I2;
    }n
}
MONITOR {
   A, B;
}
```



Worked Example 3 – 4 Bit (1 Nibble) Ripple Counter

```
DEVICES {
  A, B, C, D are DTYPE;
  RST, ST are SWITCH;
  CKL1 is CLOCK;
  CKL1 has cycle 1;
}

CONNECTIONS {
  device A {
     A.QBAR to A.DATA;
     RST to A.CLEAR;
     ST to A.SET;
     CKL1 to A.CLK;
  }
  device B {
     B.QBAR to B.DATA;
     RST to B.CLEAR;
     ST to B.SET;
     A.Q to B.CLK;
  }
  device C {
     C.QBAR to C.DATA;
     RST to C.CLEAR;
     ST to C.SET;
     B.Q to C.CLK;
  }
  device D {
     D.QBAR to D.DATA;
     RST to D.CLEAR;
     ST to D.SET;
     C.Q to D.CLK;   }
}

MONITOR {
  A.Q, B.Q, C.Q, D.Q, CKL1;
}
```