GF2 Software - Second Interim Report

Jonty Page
Pembroke College, University of Cambridge

May 30, 2019

1 Example Definition Files and Circuits

1.1 Example 1 - Full Adder Circuit
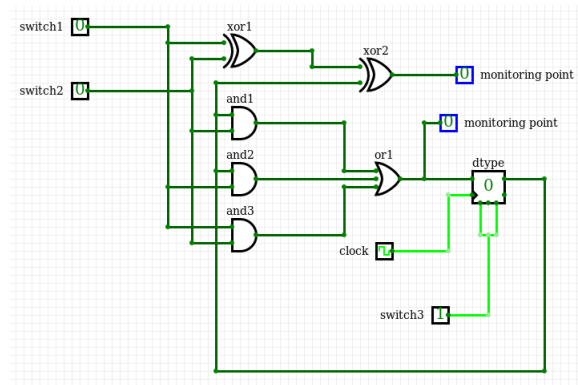
1.1.1 Full Adder Circuit Diagram

Figure 1: Full Adder Circuit Diagram

1.1.2 Full Adder Circuit Definition File

```
DEVICES {
switch1: SWITCH, initial 0;
switch2: SWITCH, initial 0;
switch3: SWITCH, initial 0;
xor1: XOR;
xor2: XOR;
and1: AND, inputs 2;
and2: AND, inputs 2;
and3: AND, inputs 2;
or1: OR, inputs 3;
clock: CLOCK, period 5;
dtype: DTYPE;
}

CONNECT {
switch1 = xor1.I1;
switch1 = and2.I2;
switch1 = and3.I1;
switch2 = and1.I2;
switch2 = xor1.I2;
switch2 = and3.I2;
clock = dtype.CLK;
```

```
xor1 = xor2.I1;
and1 = or1.I1;
and2 = or1.I2;
and3 = or1.I3;
or1 = dtype.DATA;
switch3 = dtype.SET;
switch3 = dtype.CLEAR;
dtype.Q = xor2.I2;
dtype.Q = and1.I1;
dtype.Q = and2.I1;
}

MONITOR{
xor2;
or1;
}
END
```

1.2   Example 2 - Shift Register

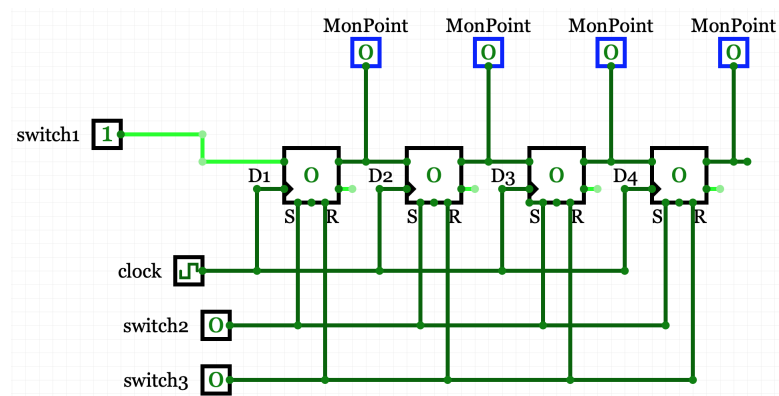### 1.2.1   Shift Register Circuit Diagram

Figure 2: Shift Register Circuit Diagram

1.2.2   Shift Register Circuit Definition File

```
DEVICES{
switch1: SWITCH, initial 1;
switch2: SWITCH, initial 0;
switch3: SWITCH, initial 0;
D1: DTYPE;
D2: DTYPE;
D3: DTYPE;
D4: DTYPE;
clock: CLOCK, period 5;
}

CONNECT{
switch1 = D1.DATA;
D1.Q = D2.DATA;
D2.Q = D3.DATA;
D3.Q = D4.DATA;
clock = D1.CLK;
clock = D2.CLK;
```

```
clock = D3.CLK;
clock = D4.CLK;
switch2 = D1.SET;
switch2 = D2.SET;
switch2 = D3.SET;
switch2 = D4.SET;
switch3 = D1.CLEAR;
switch3 = D2.CLEAR;
switch3 = D3.CLEAR;
switch3 = D4.CLEAR;
}

MONITOR{
D1.Q;
D2.Q;
D3.Q;
D4.Q;
}

END
```

1.3 Example 3 - Combinational Circuit

1.3.1 Combinational Circuit Diagram

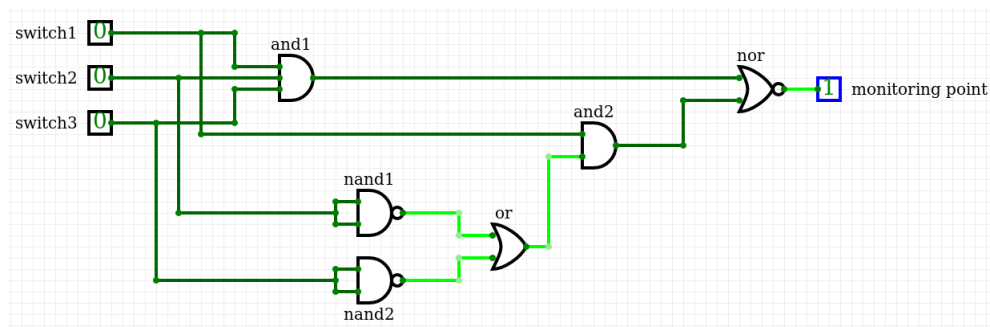Figure 3: Shift Register Circuit Diagram

1.3.2 Combinational Circuit Definition File

```
DEVICES{
switch1: SWITCH, initial 0;
switch2: SWITCH, initial 0;
switch3: SWITCH, initial 0;
and1: AND, inputs 3;
and2: AND, inputs 2;
nand1: NAND, inputs 2;
nand2: NAND, inputs 2;
or: OR, inputs 2;
nor: NOR, inputs 2;
}

CONNECT{
switch1 = and1.I1;
switch1 = and2.I1;
switch2 = and1.I2;
switch2 = nand1.I1;
```

```
switch2 = nand1.I2;
switch3 = and1.I3;
switch3 = nand2.I1;
switch3 = nand2.I2;
nand1 = or.I1;
nand2 = or.I2;
or = and2.I2;
and2 = nor.I2;
and1 = nor.I1;
}

MONITOR{
nor;
}

END
```

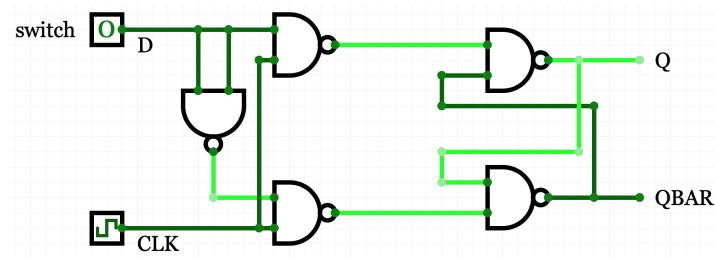## 1.4 Example 4 - dtype Circuit

1.4.1 dtype Circuit Diagram

Figure 4: dtype Circuit Diagram

1.4.2 dtype Circuit Definition File

```
DEVICES{
switch: SWITCH, initial 0;
clock: CLOCK, period 5;
nand1: NAND, inputs 2;
nand2: NAND, inputs 2;
nand3: NAND, inputs 2;
nand4: NAND, inputs 2;
nand5: NAND, inputs 2;
}

CONNECT{
clock = nand2.I2;
clock = nand1.I2;
switch = nand1.I1;
switch = nand5.I1;
switch = nand5.I2;
nand5 = nand2.I1;
nand1 = nand3.I1;
nand2 = nand4.I2;
nand3 = nand4.I1;
nand4 = nand3.I2;
}
```

```
MONITOR{
nand3;
nand4;
}
END
```

Logic Simulator
User Guide

To load graphical user interface (GUI), change directory into the folder *gf2_python* and run the *logsim.py* program using the command:

python logsim.py (def file location)

The location of a definition file is optional and the GUI will be preloaded with the file if the argument is given.

Performing Circuit Simulation:

(1) If not preloaded, then click browse button and select your desired '.txt' definition file.

(2) Add/Remove points in the circuit at which the trace will be recorded on the canvas by the simulator.

(3) Select the desired initial state of the switches in the circuit (this will not appear if no switches exist)

(4) Enter the number of cycles for which the circuit should be simulated and click the run button.

(5) The trace for each monitoring point will be displayed on the canvas. Pressing the continue button will continue simulating each trace for the specified clock cycles (max 100 per run/continue).
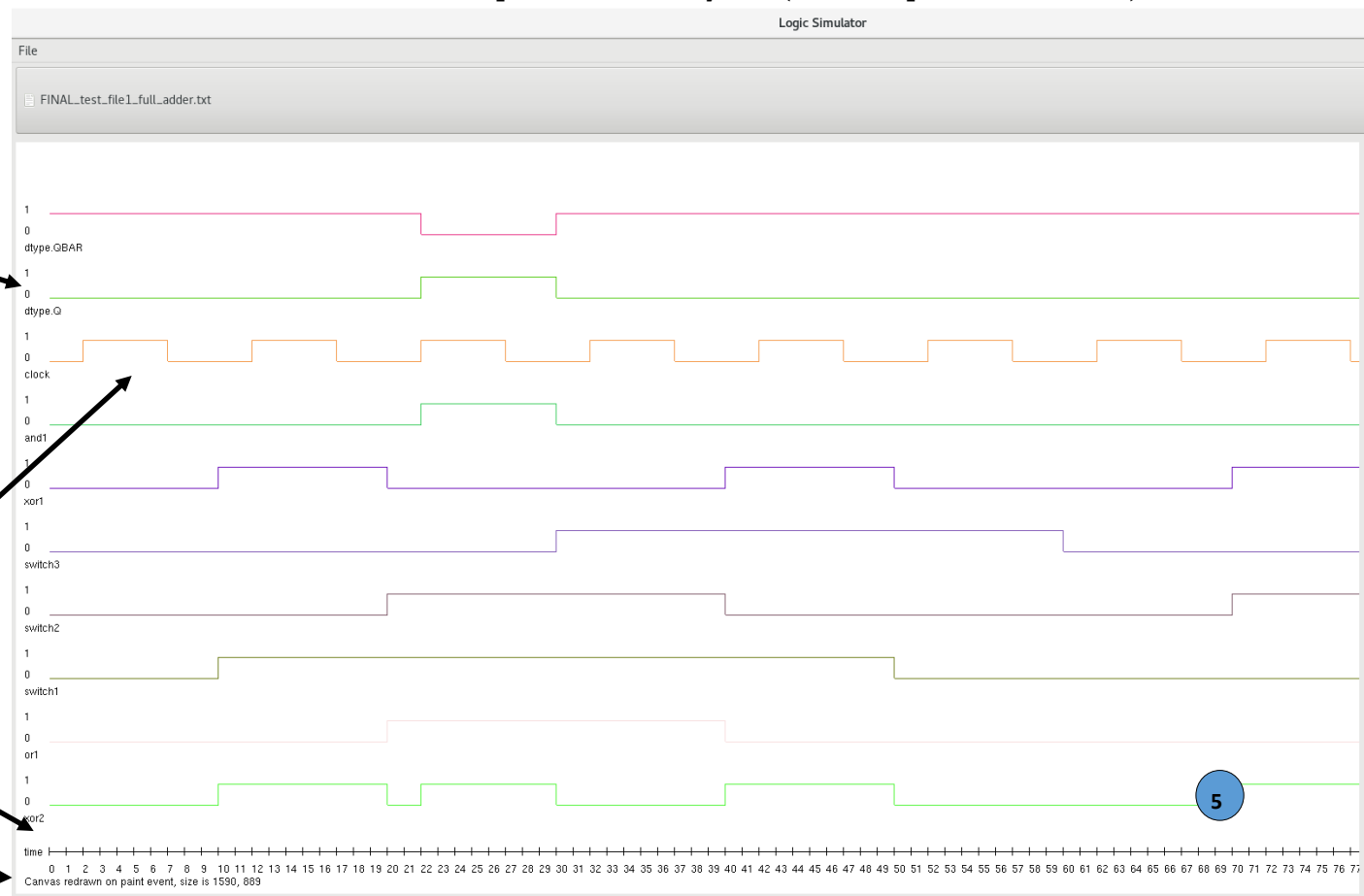
File Selector Dialog
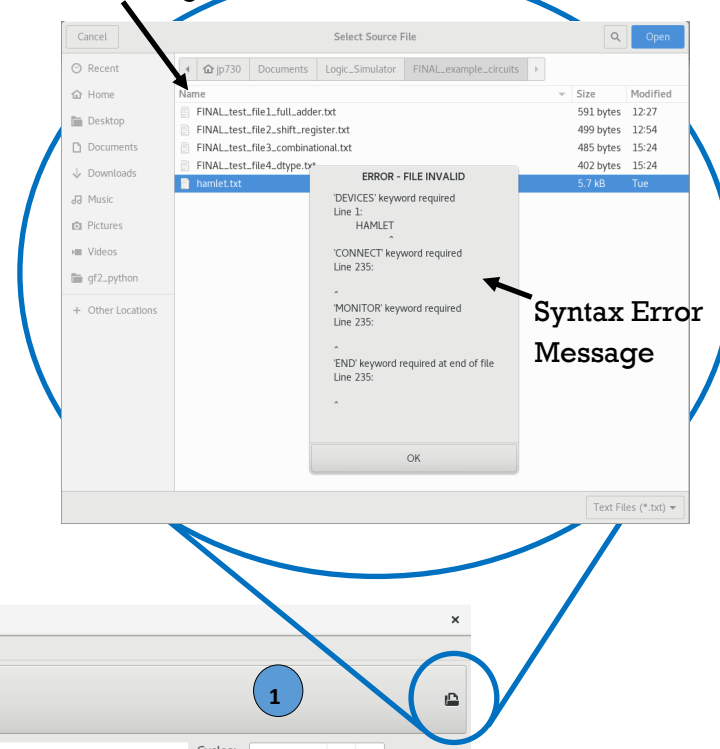
Syntax Error Message

Name of monitoring point and trace

N.B. Initial states of clocks and dtypes are randomised

Time axis

Notification Bar

Control Buttons and Cycles Selector

Current Monitor Points and Scrollbar

Current Switch States