Appendix A

1 Example Files and Circuits

1.1 Example 1: Full Adder

1.1.1 Circuit

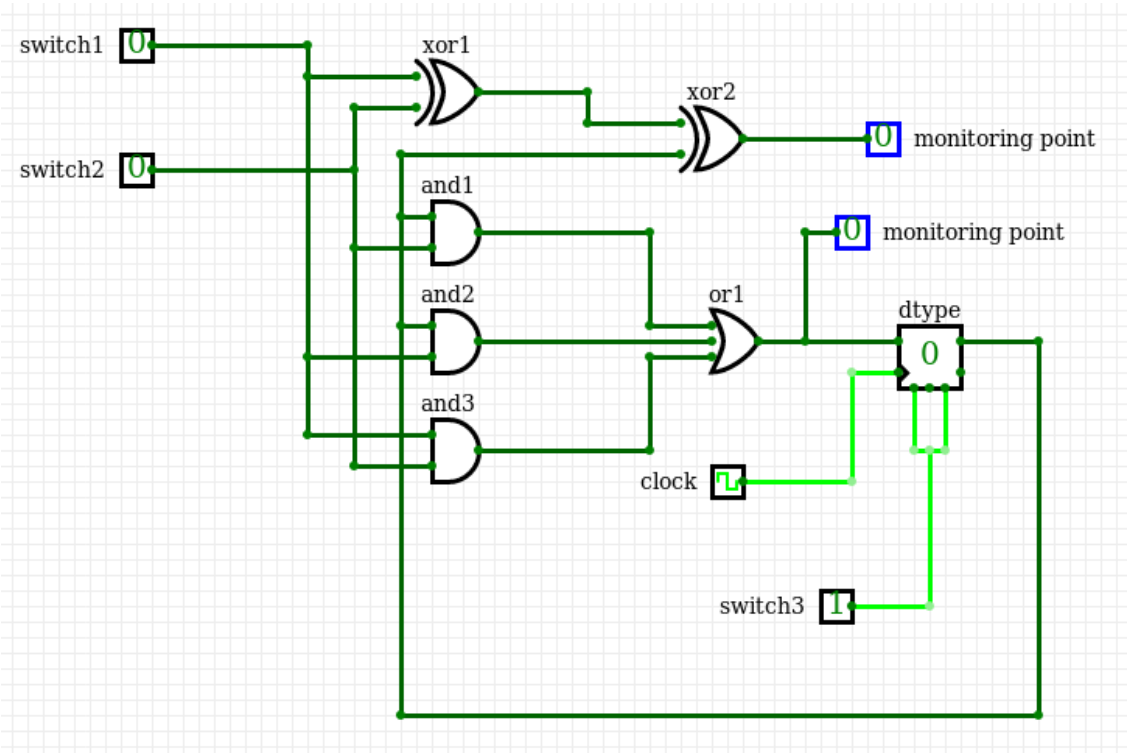

Figure 1: Full Adder Circuit

1.1.2 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
```

1.1.3 Test Result

See Figure 2.

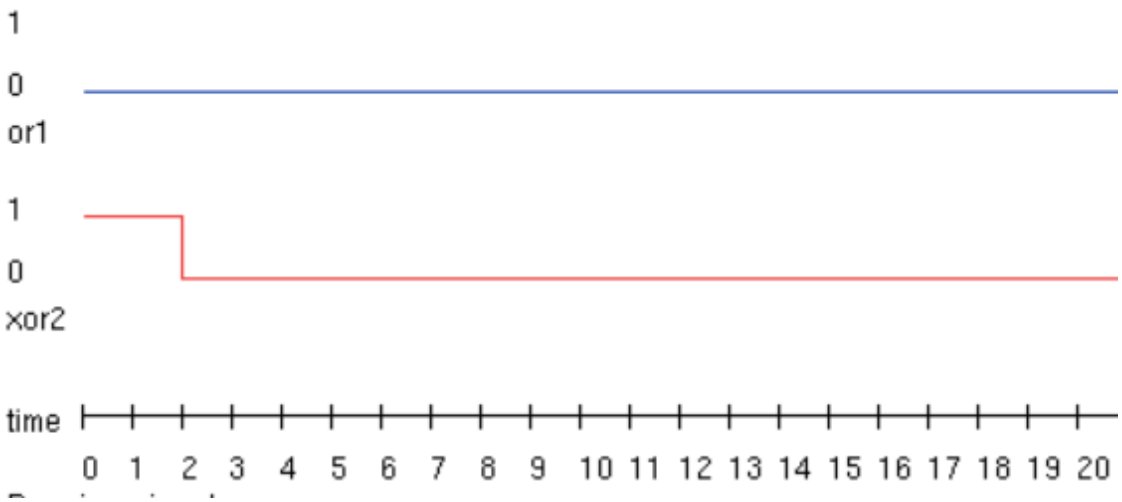

Figure 2: GUI test result of running test file 1

1.2 Example 2: 4-Bit Shift Register with Signal Generator

1.2.1 Circuit

1.2.2 Definition File

```
DEVICES{
sigg: SIGGEN, sequence 1, 0, 0, 1;
switch1: SWITCH, initial 0;
switch2: SWITCH, initial 0;
D1: DTYPE;
D2: DTYPE;
D3: DTYPE;
D4: DTYPE;
```

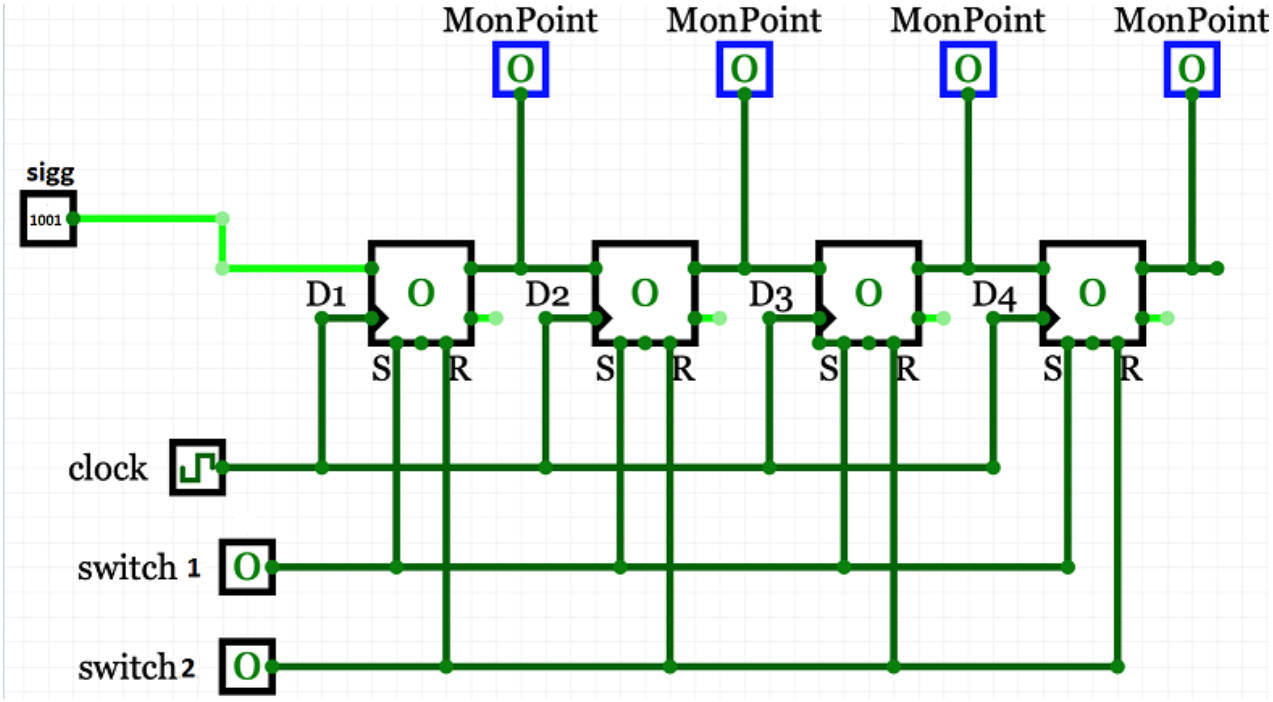

Figure 3: 4-Bit Shift Register

```
clock: CLOCK, period 5;
}

CONNECT{
sigg = 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;
switch1 = D1.SET;
switch1 = D2.SET;
switch1 = D3.SET;
switch1 = D4.SET;
switch2 = D1.CLEAR;
switch2 = D2.CLEAR;
switch2 = D3.CLEAR;
switch2 = D4.CLEAR;
}

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

END
```

1.2.3 Test Result

See Figure 4.

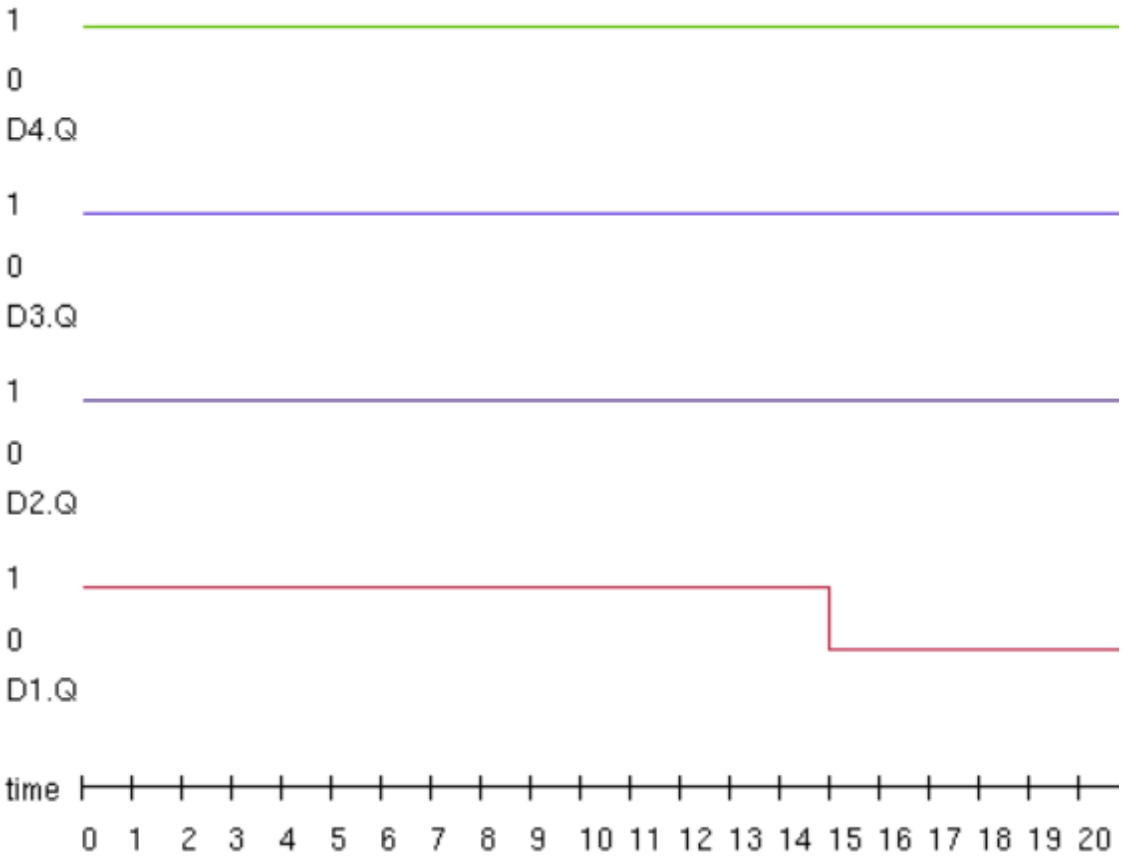

Figure 4: GUI test result of running test file 2

1.3 Example 3: Combinational

1.3.1 Circuit

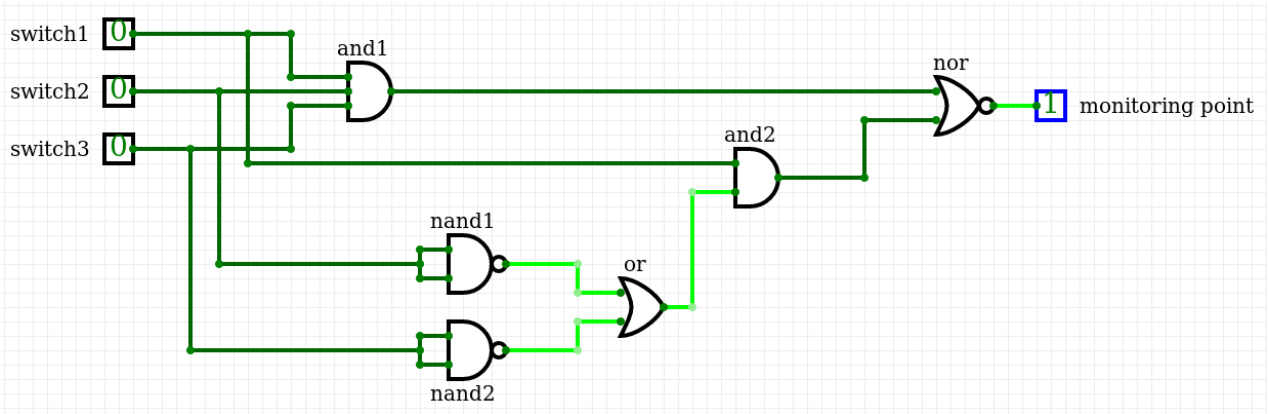

Figure 5: Combinational Circuit

1.3.2 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
```

1.3.3 Test Result

See Figure 6.

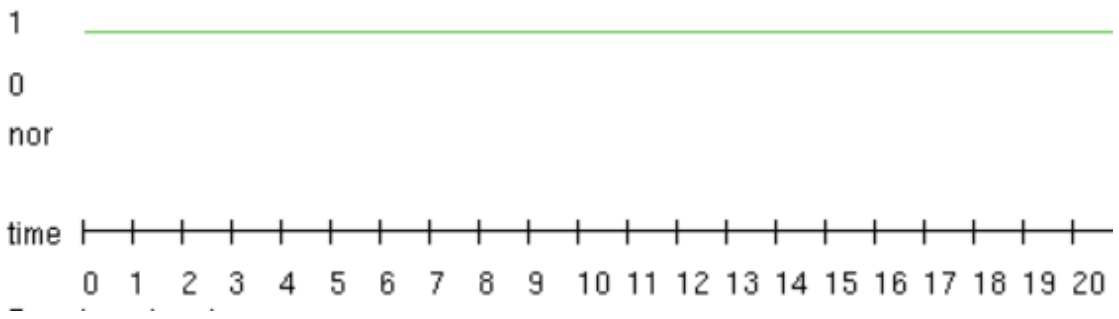

Figure 6: GUI test result of running test file 3

1.4 Example 4: D-Type Construction

1.4.1 Circuit

1.4.2 Definition File

```
DEVICES{
switch: SWITCH, initial 0;
```

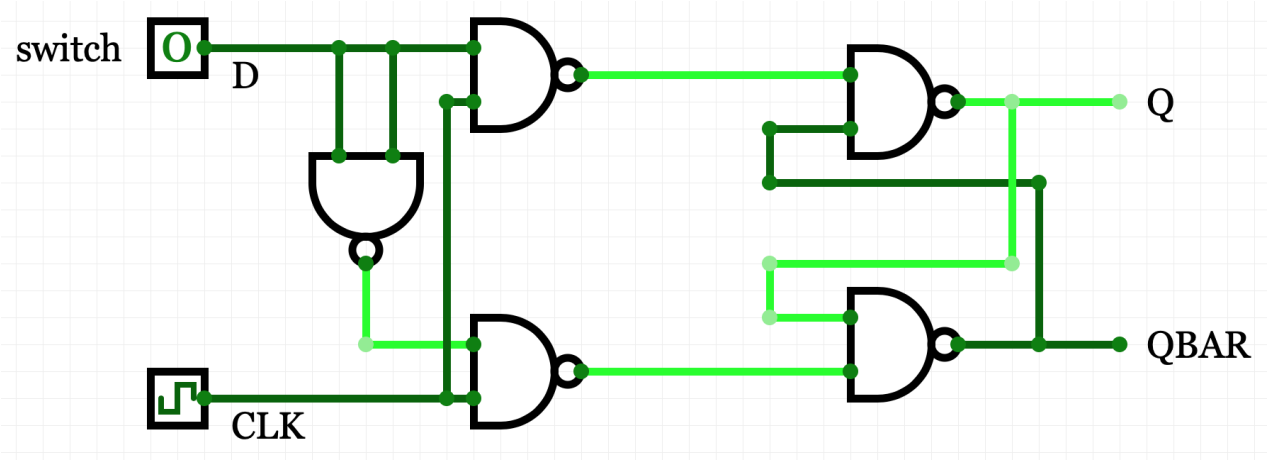

Figure 7: Construction of D-Type Flip Flop

```
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
```

1.4.3 Test Result

See Figure 8.

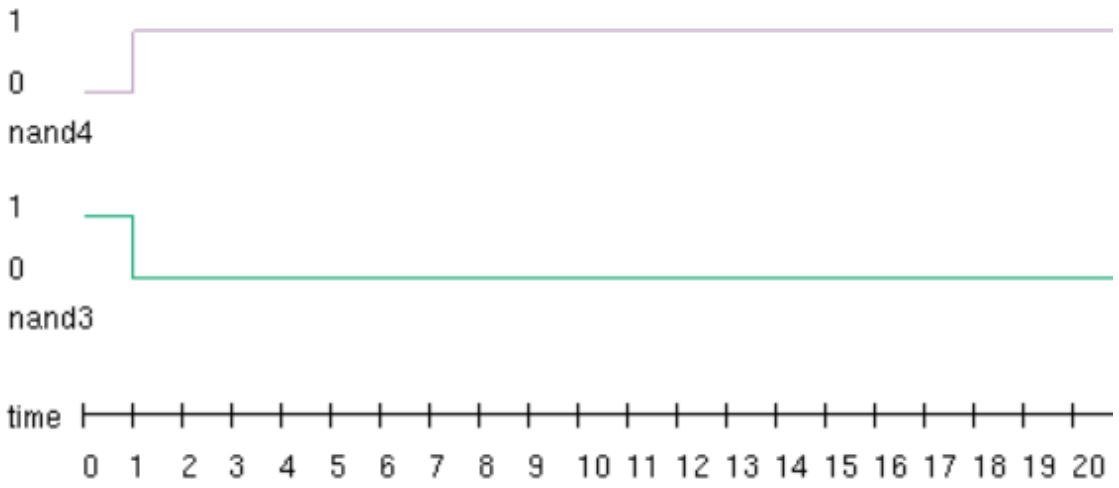

Figure 8: GUI test result of running test file 4

1.5 Example 5: Invalid Circuit (Hamlet Text)

This example demonstrates the response of the system to an input that does not intend to use the system appropriately.

1.5.1 Definition File

```
        HAMLET


        DRAMATIS PERSONAE


CLAUDIUS        king of Denmark. (KING CLAUDIUS:)

HAMLET  son to the late, and nephew to the present king.

POLONIUS        lord chamberlain. (LORD POLONIUS:)

HORATIO friend to Hamlet.

LAERTES son to Polonius.

LUCIANUS        nephew to the king.
```

1.5.2 Test Result

See Figure 9.

(a) Graphical User Interface

(b) Command Line Interface

Figure 9: System response to an erroneous file