Practical Sheet 9

Logic Circuits

Aims

Section		Aim
1	Boolean Logic and Truth Tables	Use the XLogicCircuit simulator to build and test simple logic circuits. Test them against a truth table.
2	Build an Adder Circuit	Build an adder circuit for 4 bit numbers to make clear how logic circuits are used to make computers.

Related topics

Topic 10.1 Files
Topic 10.2 Designing for File I/O

Initial Practical Steps

You should be able to run the XLogicCircuits too, which can be downloaded from course web site (or from the XLogicCircuits home page). It is a Java program: to run it either double-click or type

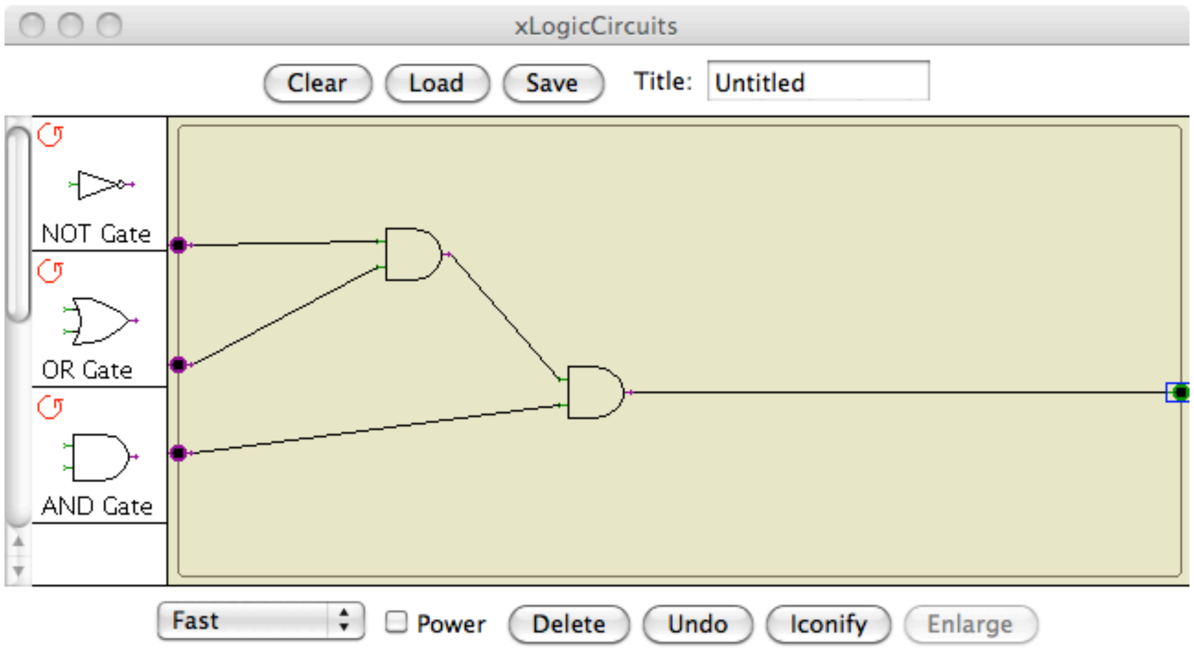
```
      java –jar  XLogicCircuits.jar
```

at a command prompt (in the directory containing XLogicCircuits.jar). If the command java –version does not respond then Java is not installed on your computer.

1 Boolean Logic and Truth Tables

1.1 Three Input AND Gate

Exercise 1.1: To practice using XLogicCircuits, build the three input 'and' gate shown below.



Exercise 1.2: Test the circuit and complete the truth table:

X	Y	Z	Output

1.2 NAND Gate

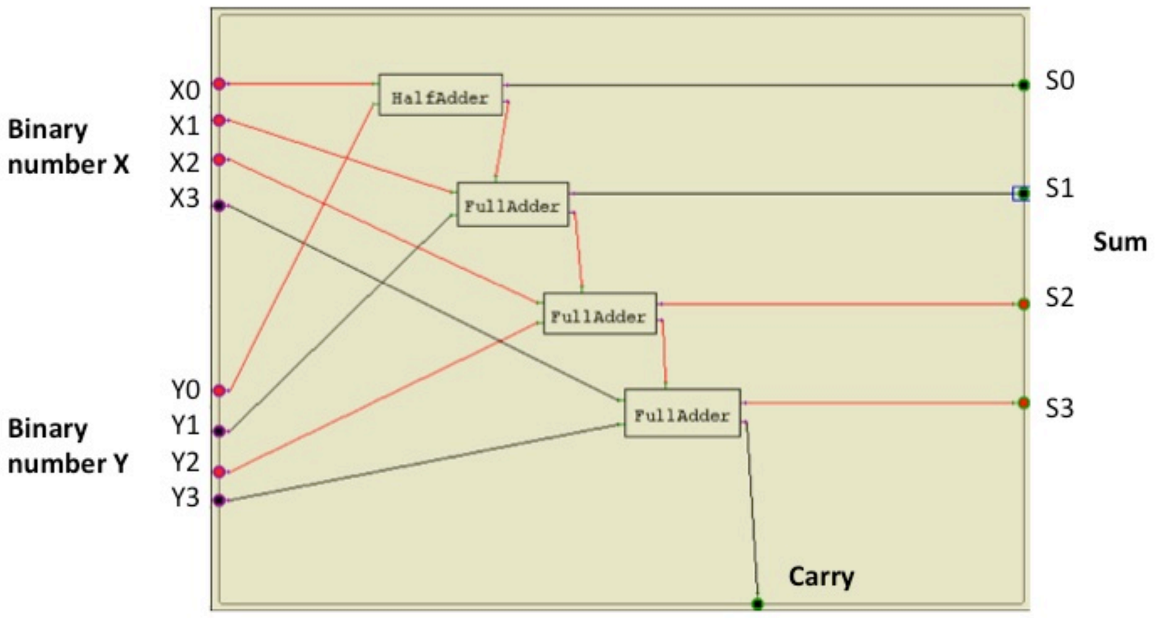
A NAND (so called as it is 'not and') gate has the formula: X 'nand' Y = not (X and Y).

Exercise 1.3: Build this gate using XLogicCircuits.

Interesting fact: you can build any circuit using just NAND gates. See http://en.wikipedia.org/wiki/NAND_logic for more details.

2 Build an Adder Circuit

We now build an adder for 4-bit binary numbers. This is the finished circuit (shown adding 0111 + 0101 = 1100). (Note that there is no way to label the inputs and outputs in XLogicCircuits – the labels are added.)



The rest of the sheet sets out steps to build this circuit. However, if you prefer, the can download the circuit from the web page and use it to do sums (see Exercise 2.8 below).

2.1 Step 1. Design and implement a half-adder

A half-adder is a logic circuit that takes as its inputs two bits X and Y to produce two outputs: their sum S and the carry-out bit C. Design a half-adder circuit in the following steps:

Exercise 2.1 Write down the truth table for the half-adder. Remember that this represents one column of a binary addition, with no carry in. Some entries are already completed:

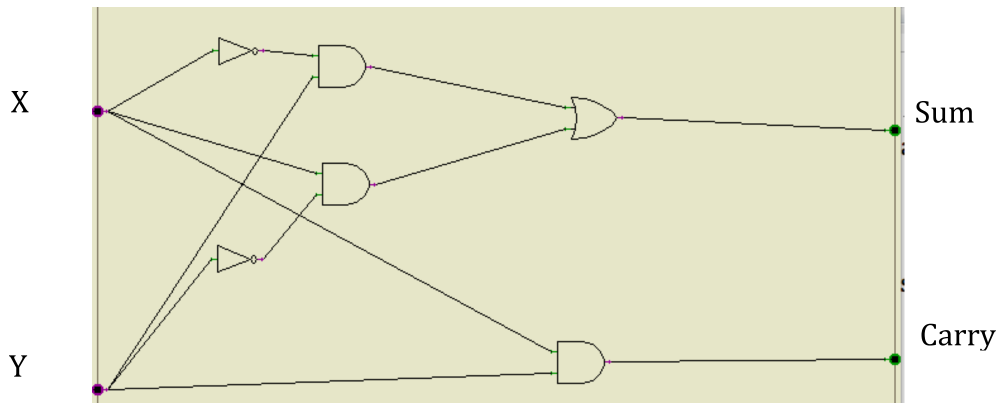
X	Y	Sum	Carry
0	0		
0	1	1	0
1	0		
1	1		1

Exercise 2.2 The Boolean expressions for the sum S and for the carry bit C are as follows. Check that these expressions correspond to your truth table.

Sum = (X and not Y) or (not X and Y)

Carry = X and Y

Exercise 2.3 Implement the half-adder circuit using XLogicCircuits, as shown below.



Exercise 2.4 Check that it operates as expected, by comparing it with the truth table. If it does, name it HalfAdder and 'iconify' it so you can use it later.

2.2 Task 2. Combine Two Half-Adders into A Full Adder

A full-adder represents a column of a binary addition, with:

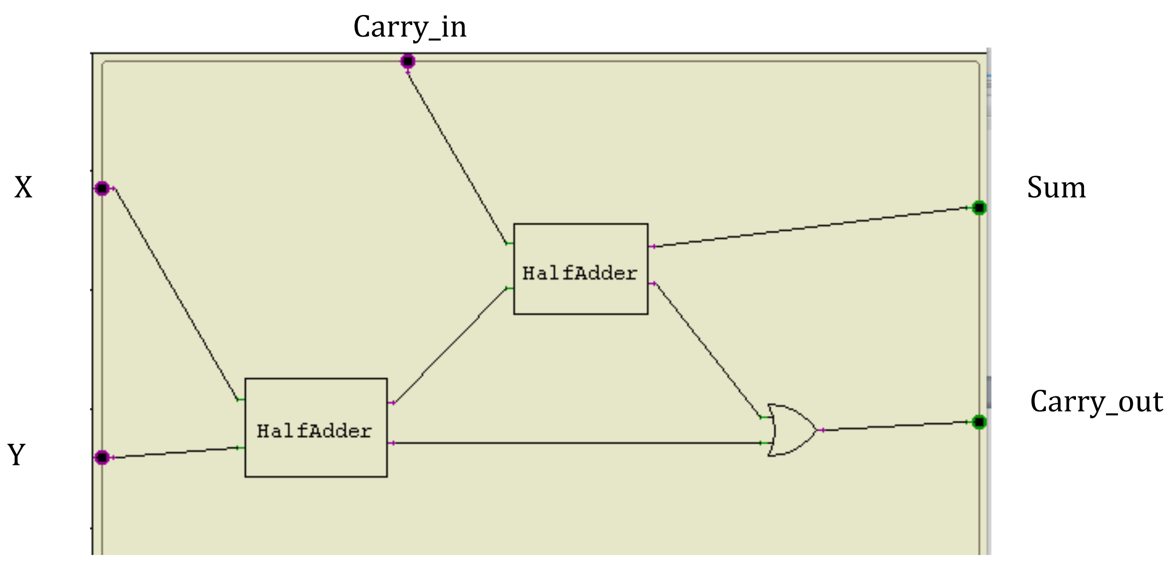
- A carry input from the previous column.
- Two digits X and Y.
- A sum.
- A carry to the next column.

A full adder can be built by using 2 half-adders:

- The first half adder adds the inputs X and Y

- The second half adder adds the carry input to sum of X + Y.
- The two carry outputs are combined using an OR gate.

Exercise 2.5 Create an XLogicCircuit implementation of the full-adder in this way using the half adder that you have already done. The circuit is shown below.



Exercise 2.6 Check that it works by trying some sums, or by writing the complete truth table.

2.3 Task 3. Design and implement a 4-bit adder

Exercise 2.7 Using the 1-bit adder circuits already implemented as a building block, implement a 4-bit 'ripple adder' by connecting the carry output of each full-adder to the carry out of the next one. The inputs to this adder are two 4-bit binary numbers and a carry; the outputs are a 4-bit binary number and a carry.

The complete circuit is shown at the start of this section. Note that, as there is no carry input for the least significant bit, the first stage of the addition uses a half adder.

Exercise 2.8 Use the adder to do some sums. Reflect on what the circuit shows about how computers work. What parts of the overall picture (how computers work) are still missing?

3 Summary

- The logic gates are AND, OR and NOT.
- Binary digits correspond to logic values and can be combined using AND, OR and NOT gates.
- An adder circuit can be build from gates. This shows how a computer is created from gates.