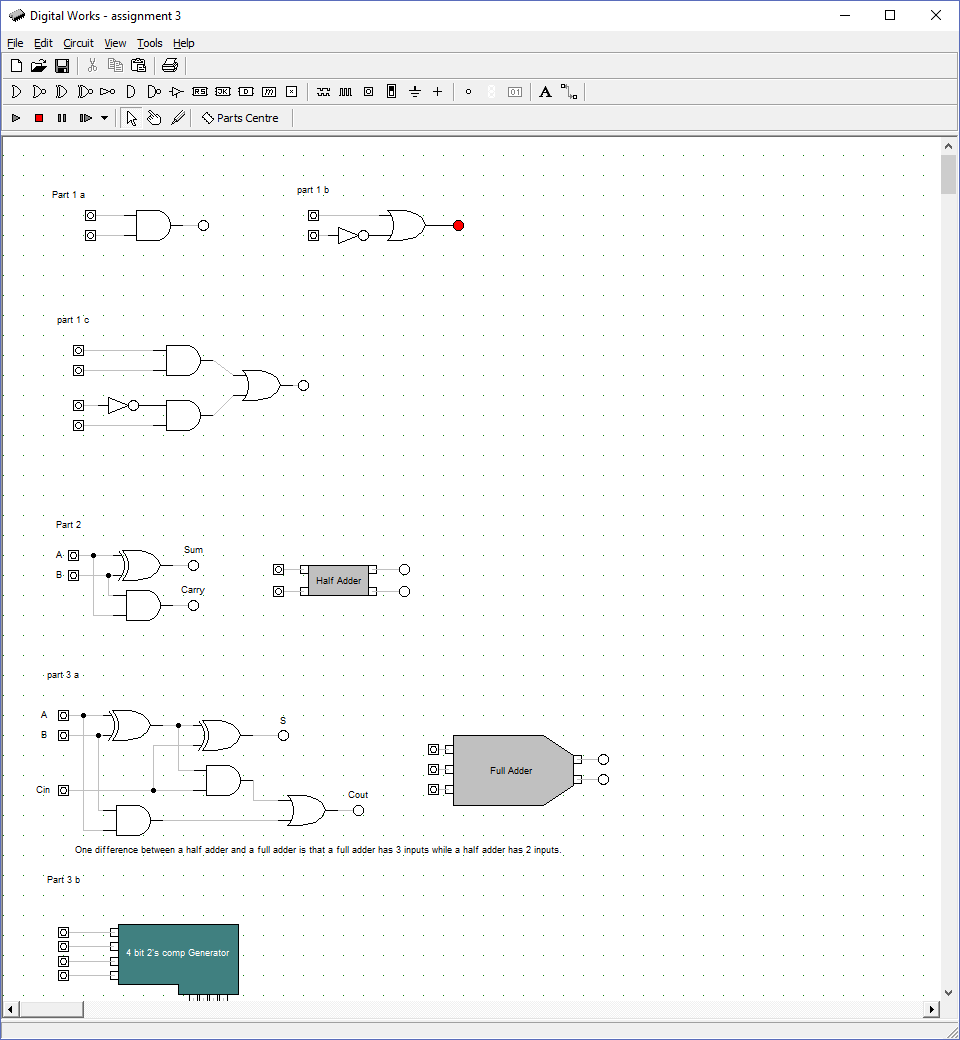
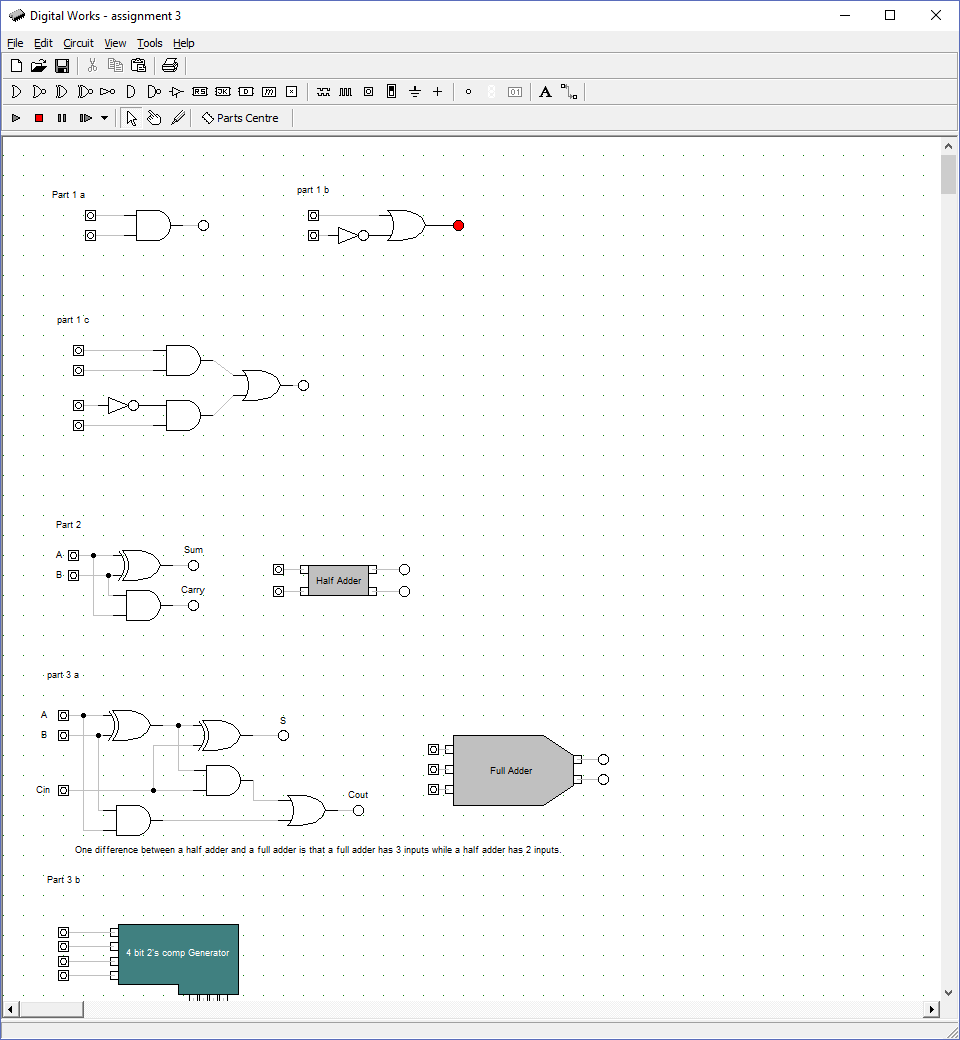
**CI209 – System Architecture**

**Assignment 3**

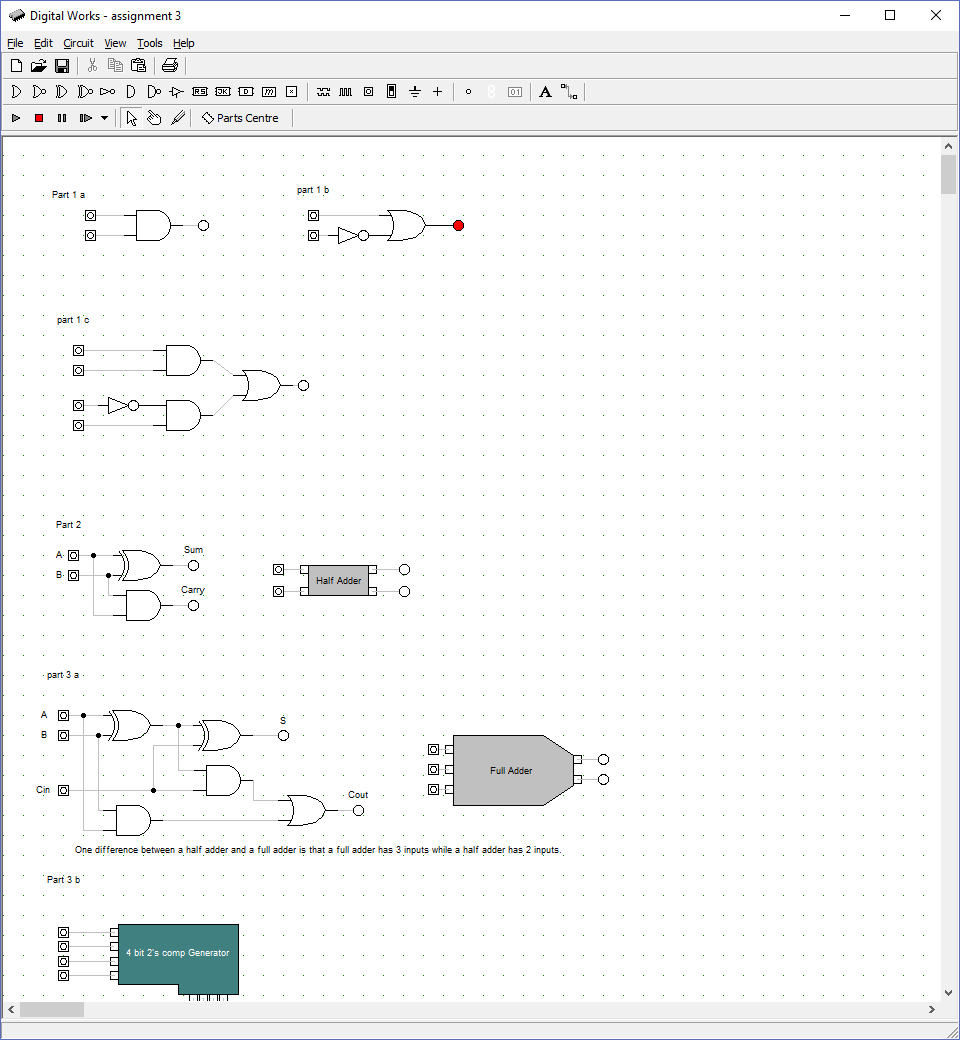
Here the LED only turns on when both switches are on. This is because I have used an AND gate. If Variable 1 AND Variable 2 are true, then the LED is on.

**1a)**



Here the LED turns on when either Variable 1 is true or Variable 2 is not true. This is because I have used an OR gate and a NOT gate, so Variable 1 OR NOT Variable 2.

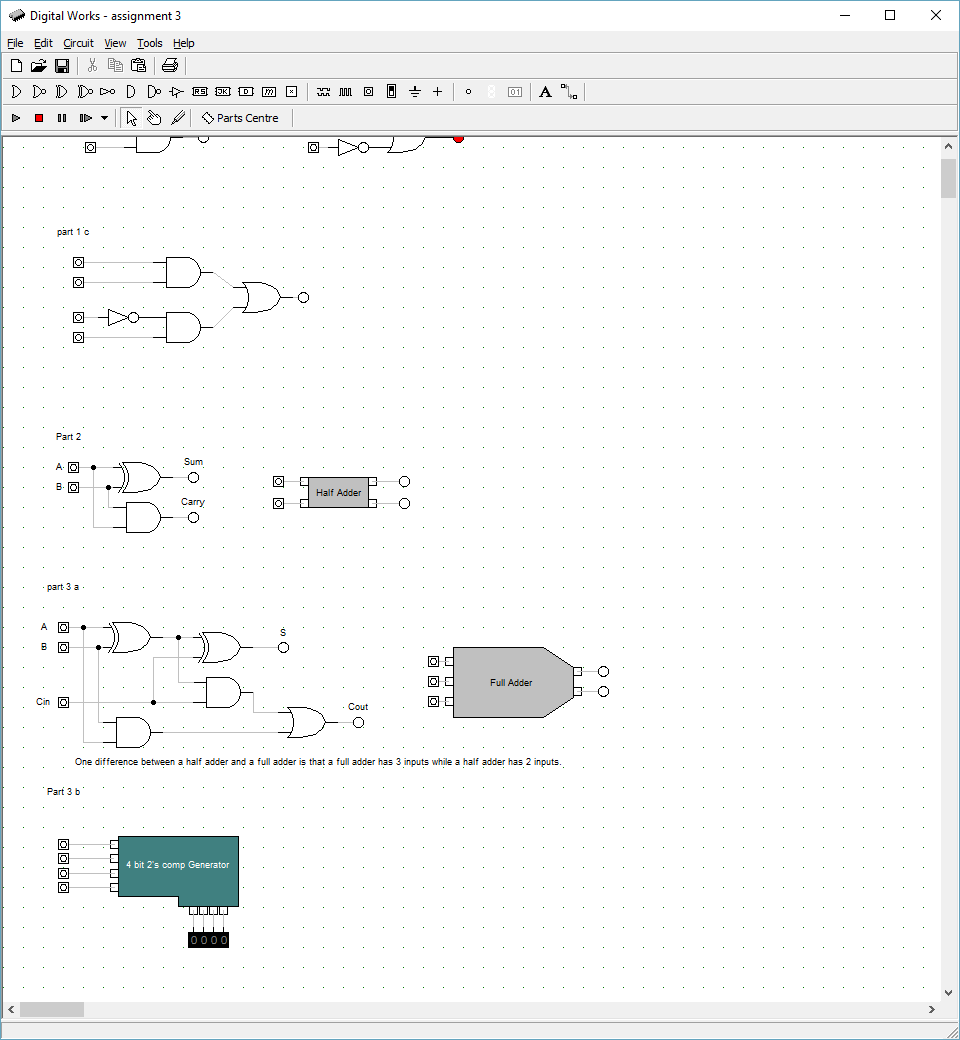
**b)**



Here there are 4 Variables. I have used a NOT gate and 2 AND gates leading to an OR gate. This means the LED is on if:

Variable 1 AND 2 are true OR Variable 3 is NOT true AND Variable 4 is true.

**c)**

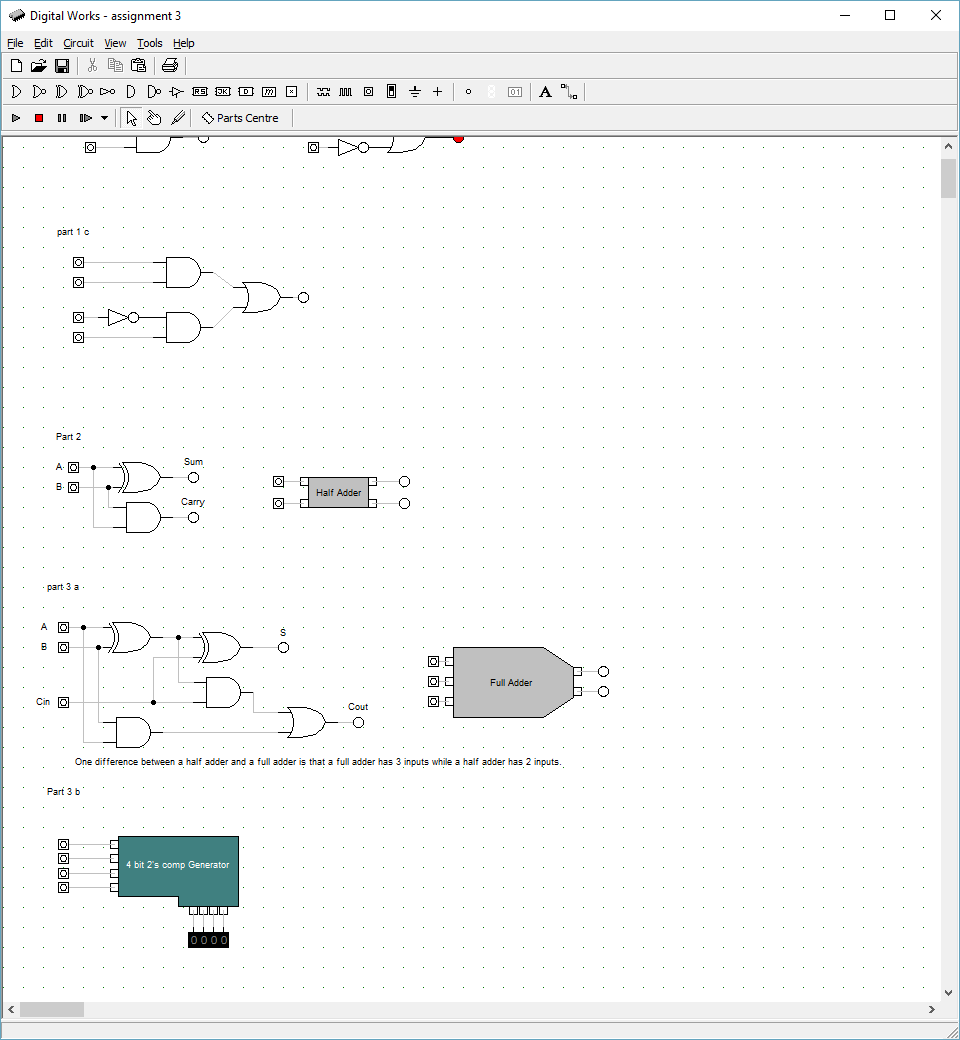


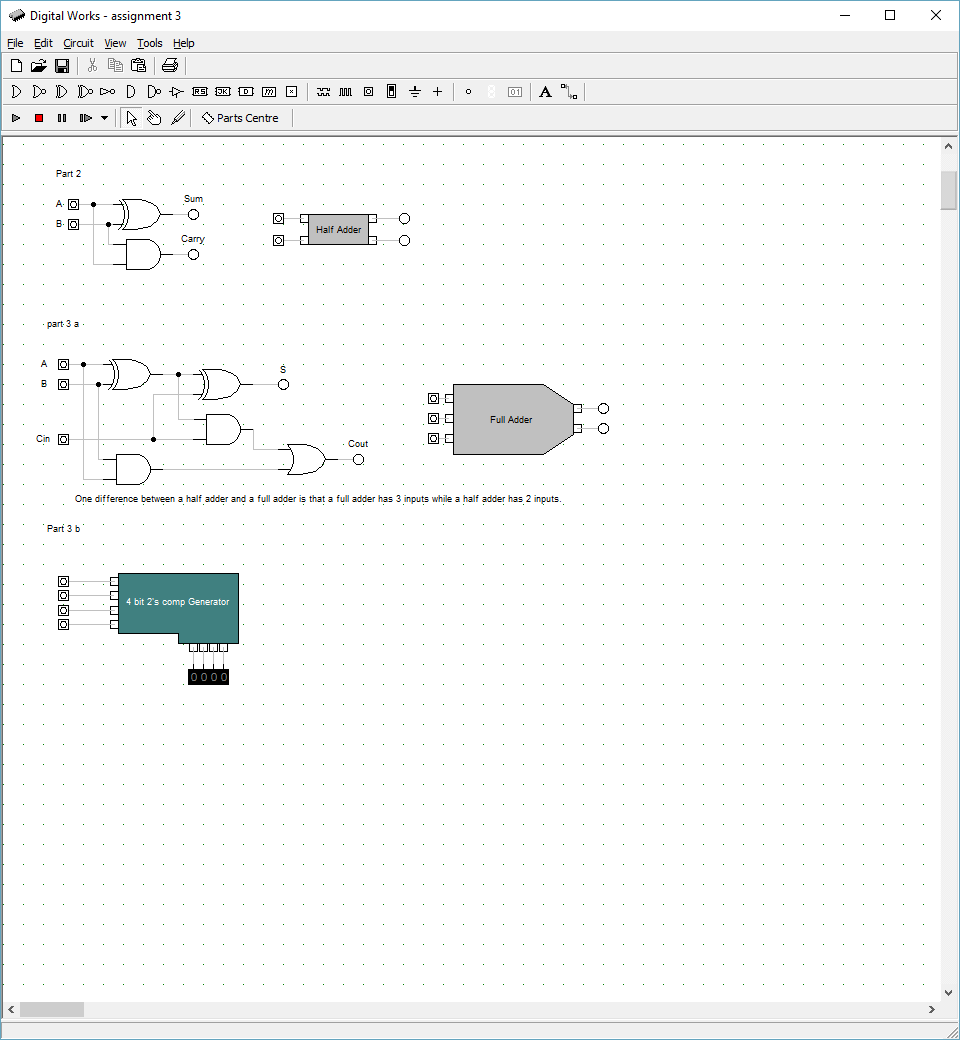
This is a half adder, below I have created a macro for the half adder.

For the half adder I have used 2 switches leading to an XOR gate which means if either variable 1 OR variable 2 is true then the LED SUM will be on. But if they are both true then the LED SUM will be off.

The switches are also connected to an AND gate so if both switches are on the LED Carry will be on.

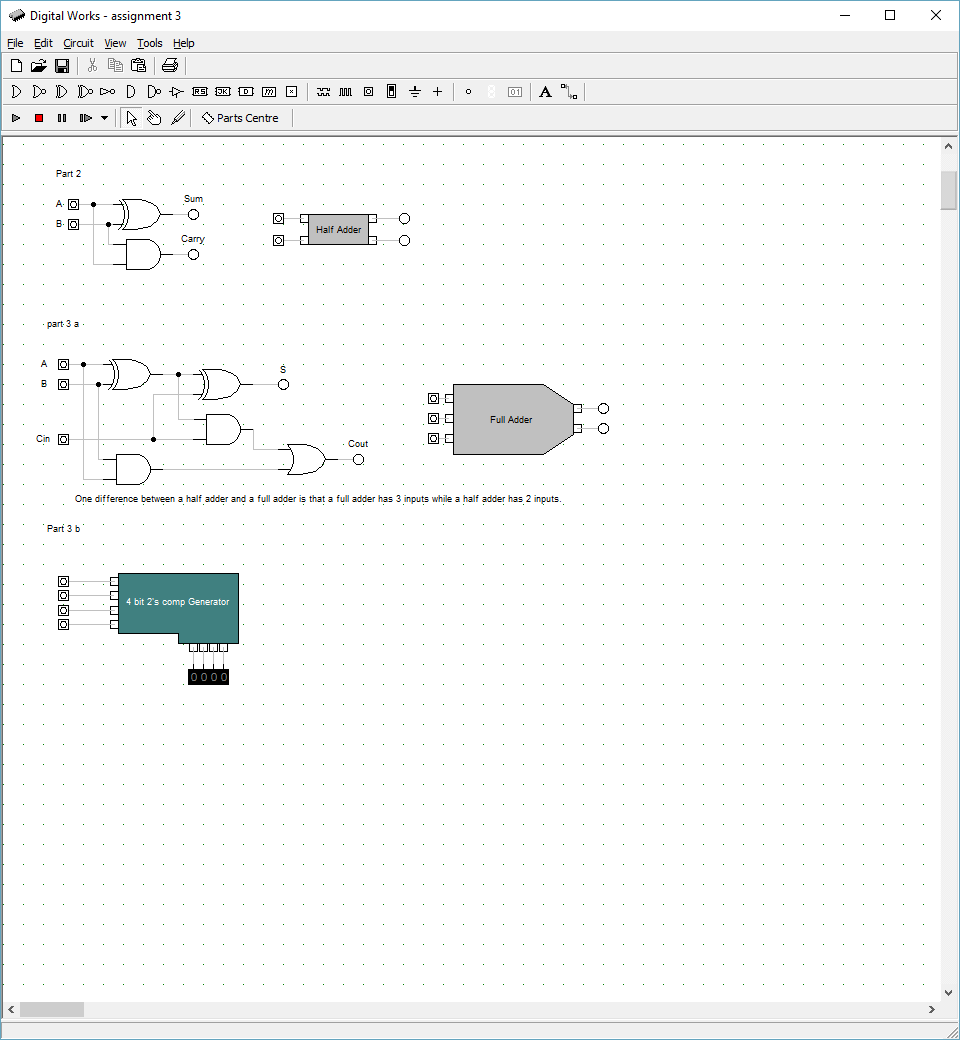
**2)**



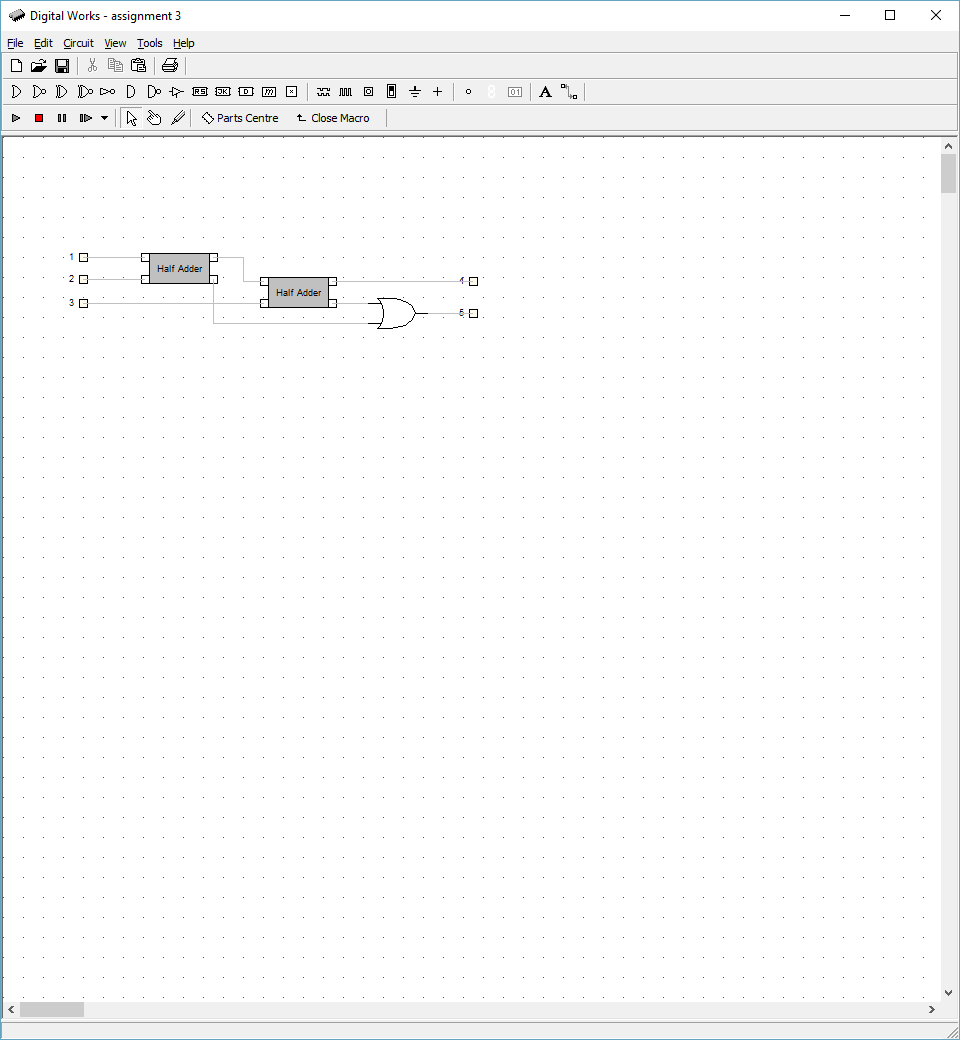


This is a full adder using two half adders, but no macros.

**3a)**



Here I have created a macro for the full adder using two half adder macros. The contents of the full adder macro can be seen below.



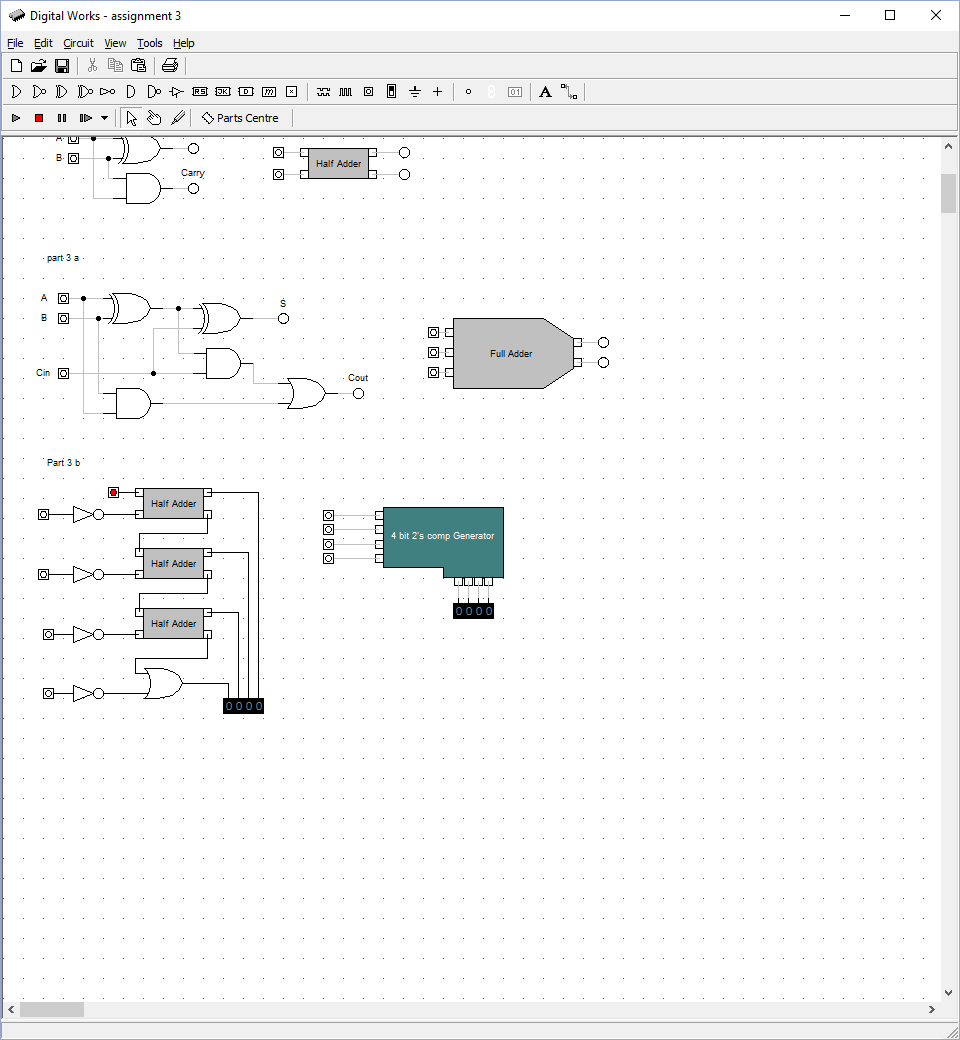
Here I have used 2 half adders to make a full adder, it also has 3 switches. 2 switches go the first half adder, the sum of this goes to the next half adder with the third switch. The sum of this is output as S. The other outputs from the half adders are carried as Cout.

One difference between a half adder and a full adder is that a full adder has 3 inputs and 2 outputs while a half adder has 2 inputs and 2 outputs.

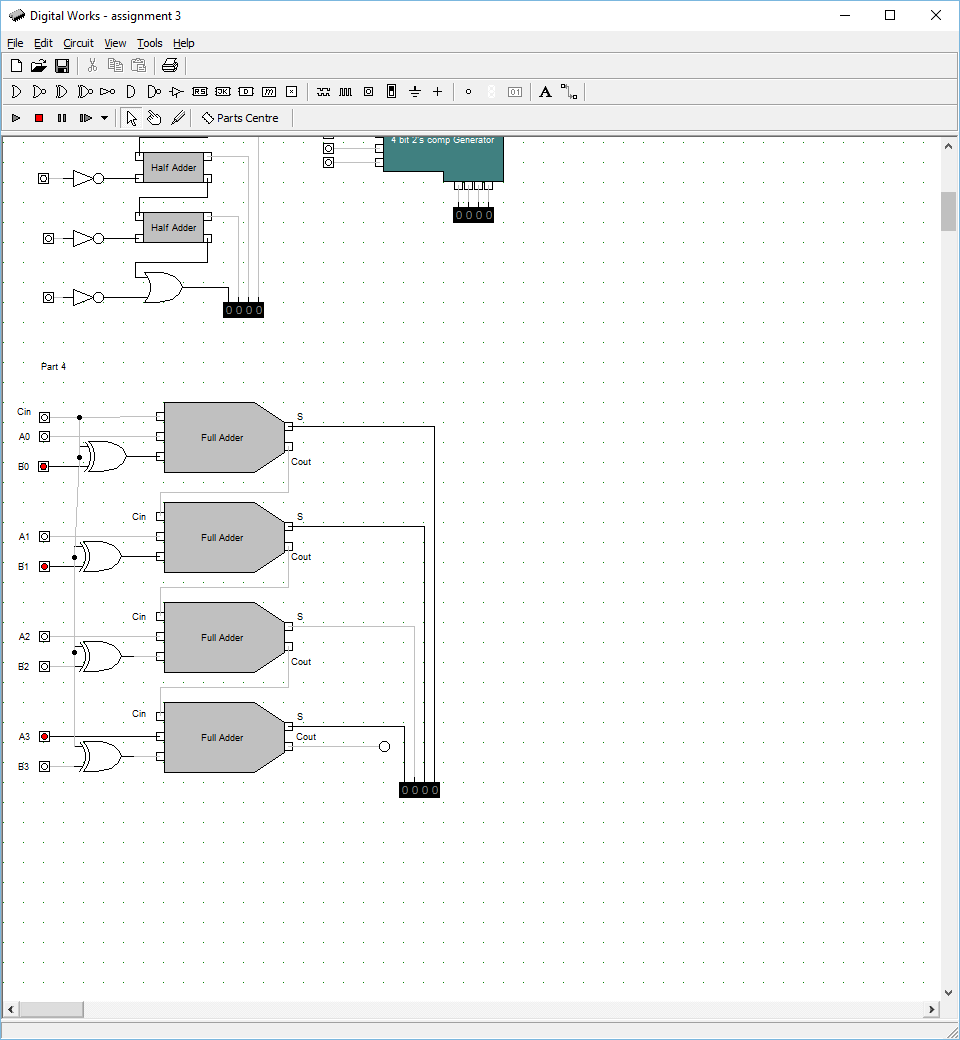
The full adder has inputs A, B and CIN. CIN is an input carry, the half adder has inputs A and B.

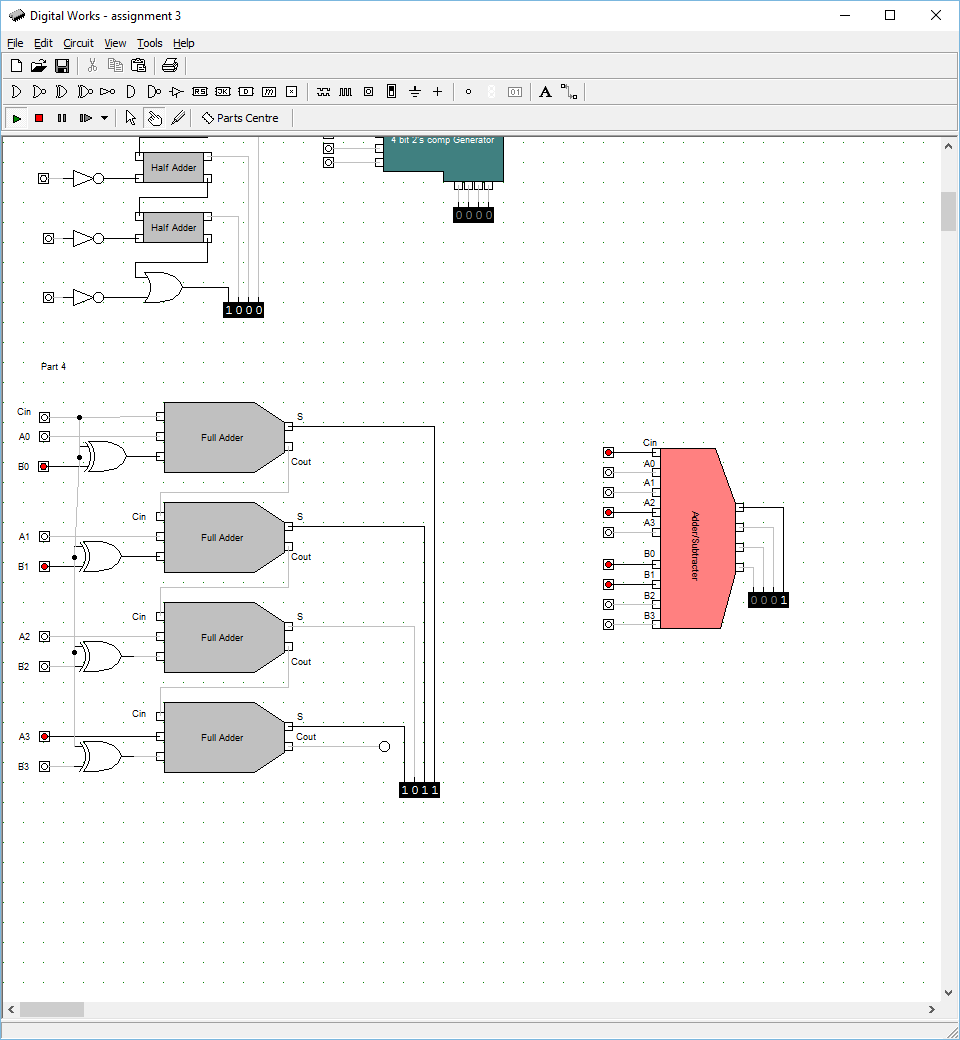
The full adder has outputs S and Cout, S is the output, Cout is the output carry, while the half adder has outputs Sum and Carry.

In the full adder, the first half adder, adds A and B to get a sum, the second half adder then adds this sum to Cin to get S and if the half adders have a carry this will output as Cout.



**3b)**



**4)**

Here I have made a circuit that can add or subtract two 4 bit binary numbers.

To the right I have also created a macro for this circuit in which the two 4 bit binary numbers are clearer as the 4 bits are together rather than every other bit. For example in the circuit above the inputs are: Cin, A0 ,B0 ,A1 ,B1 …and so on, but in the macro I have made the inputs are: Cin, A0, A1, A2, A3, B0, B1, B2, B3. This makes the 2 binary numbers easier to read.

If Cin = 1 then the circuit will subtract B from A, while if Cin = 0 it will add A to B.

Here I have used 4 inputs for A, 4 inputs for B and an input for Carry in. The Cin and B inputs are attached to XOR gates This is essentially the same as doing two’s complement and inverts B, while Cin Adds one.

**5)**