Name: Conor Loughran

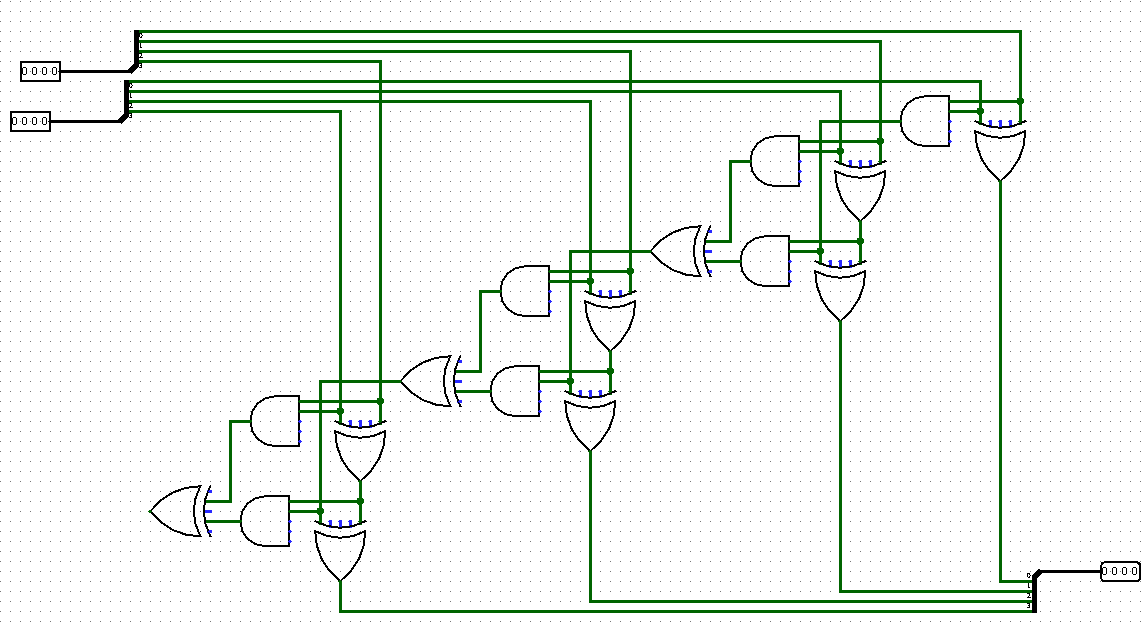
Student Number: B00795499

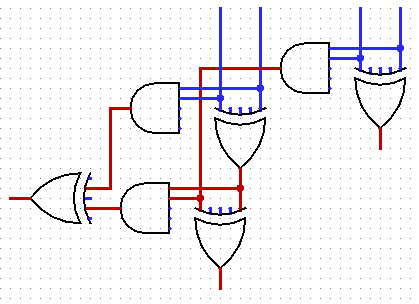
Module: COM181

contents

# Part 1

## Addition of two 4-bit numbers





A1

B1

A2

B2

Sum

Carry

Sum

Carry

The circuit used to add two separate 4-bit numbers consists of three full adders and a single half adder. The half adder begins the circuit as there is no previous carry number to deal with. To the left is a cut out section of the half adder and the first full adder. To begin we have the first bit of number “A” compared against the first bit of number “B” using an XOR gate that will only return true if a 1 and 0 is present. Any other combination won’t allow passage of current. If B1 and A1 are both 0 or both 1 then it will be sent through the AND gate and used to compare against the sum of B2 and A2. This process is repeated until it reaches the end of the circuit.

Above is the complete circuit of the addition function. The user selects a 4-bit number from the two selectors at the top which split the signal into four separate wires which allow for them to be compared and the output to be calculated. The output of each XOR gate is collected and fed back into a wire splitter to display the output number.

XOR1

XOR2

### Algebraic Formula

XOR1 =2.B2+.A2

Carry = A.B

Sum = ()+(.Carry)

Sum = +

### Truth Table

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A1 | B1 | A2 | B2 | .B2 | .A2 | .B2+.A2  *XOR1* | A1.B1  *Carry* | .XOR1 | .Carry | Sum |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |