Karnaugh Maps

In an IB question you are typically given a scenario in which there are three possible inputs and you are told the outputs. For example:

A register contains 3 binary values X, Y and Z (X is the MSB and Z the LSB):

X	Y	Z

When the binary number in this register is the equivalent of 1,3,5,6 or 7, the output is 1:

Complete the truth table for this system:

X	Y	Z	Outcome
0	0	0	0
0	0	1	1

What is the Boolean expression that this truth table represents?

Simplify the expression

Student name:

You may use a Karnaugh Map

Z \ XY	00	01	11	10
0				
1				

Draw the corresponding circuit using only **and**, **or** and **not** gates.

A control system for a conveyor belt uses three sensors to detect the movement of the belt, the presence or absence of items on the belt and the press of a stop button by the operator. Each sensor X, Y and Z will send a 0 or a 1 to a 3-bit register with X as the MSB and Z as the LSB. Under the conditions corresponding to the equivalent decimal values of 1,2 5 and 6 a buzzer will sound.

Give the truth table for this system:

X	Y	Z	Buzzer

What is the expression?

Simplify the Boolean expression for this truth table:

Z \ XY	00	01	11	10
0				
1				

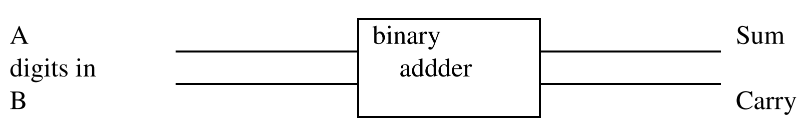
Construct the circuit:

Application of boolean logic to electronic circuit design.

Consider the addition of two binary numbers, A and B:

A	1	1	0	1
+ B	1	0	1	0
sum	0	1	1	0
carry 1.				

A black box binary adder might work like this:



Expressed as a truth table it looks like this:

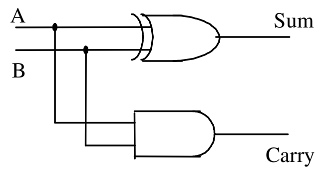
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

It is apparent that the Sum is A XOR B whereas the Carry is A AND B, expressed as boolean algebra:

$$\text{Sum} = A \oplus B$$

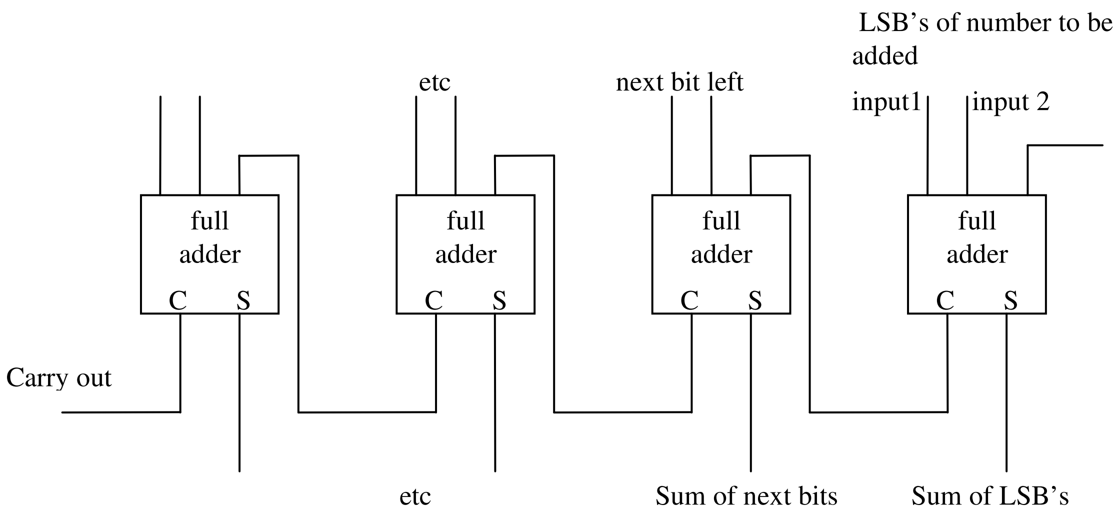
$$\text{Carry} = A \cdot B$$

It is possible to construct a circuit from AND, OR and NOT gates alone:



Full Adder

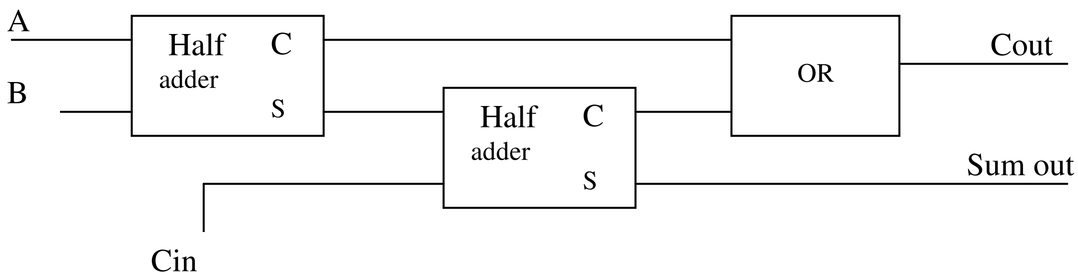
When adding binary numbers the carry must be added to the next column on the left, a full-adder is constructed from two half-adders:



A parallel adder for adding nibbles (4-bits or half a byte)

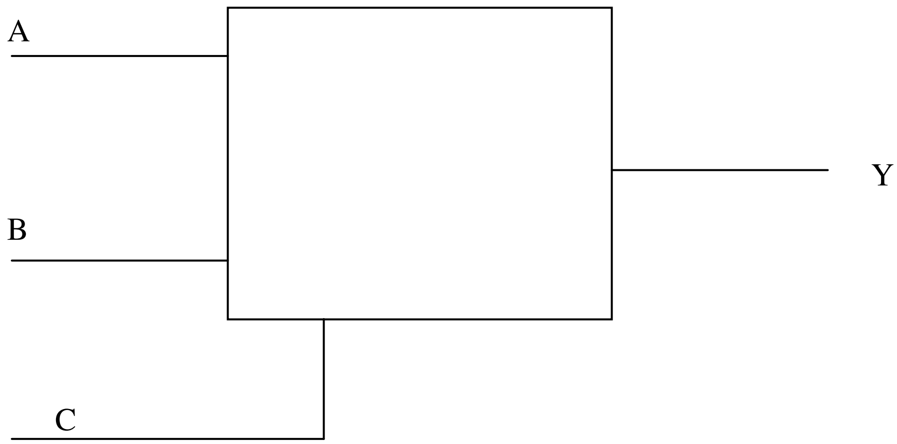
Student name:

A full adder is constructed from 2 half adders with the carry from each half adder going to an OR gate. Schematically:



What would a full-adder circuit look like? Draw it using XOR, OR and AND gates.

The following schematic circuit is a 2-to-1 line multiplexer, if C is 0 then A Y=A else Y=B:



Student name:

Draw the truth table and construct the circuit using appropriate gates.

Design a circuit that compares three inputs A, B and C and whose output Y is a one if all inputs are equal.

Design a circuit that compares the same three inputs as above but outputs a one if any two out of the three inputs are the same.

Construct the truth table for both circuits.

Laws of Boolean Algebra

1. Commutative laws

   A + B = B + A   A . B = B . A

2. Associative laws

   A + (B + C) = (A + B) + CA . (B . C) = (A . B) . C

3. Distributive laws

   A . (B + C) = (A . B) + (A . C)   A+ (B . C) = (A + B) . (A + C)

4. Tautology laws

   (a) A = A   A + A = A

   (b) A + NOT A = 1   A . NOT A = 0

5. Absorption laws

   A + (A . B) = A   A . (A + B) = A

6. Common sense laws

   (a) 0 . A = 0   1 + A = 1

   (b) 1 . A = A   0 + A = A

   (c) NOT 0 = 1   NOT 1 = 0

7. Double complement law

   NOT NOT A = A

8. De Morgan’s law

   (NOT A + NOT B) = NOT (A . B)   (NOT A . NOT B) = NOT (A + B)

   (NOT the individual terms. Change the sign. NOT the lot)