J.Pritchard U1661665

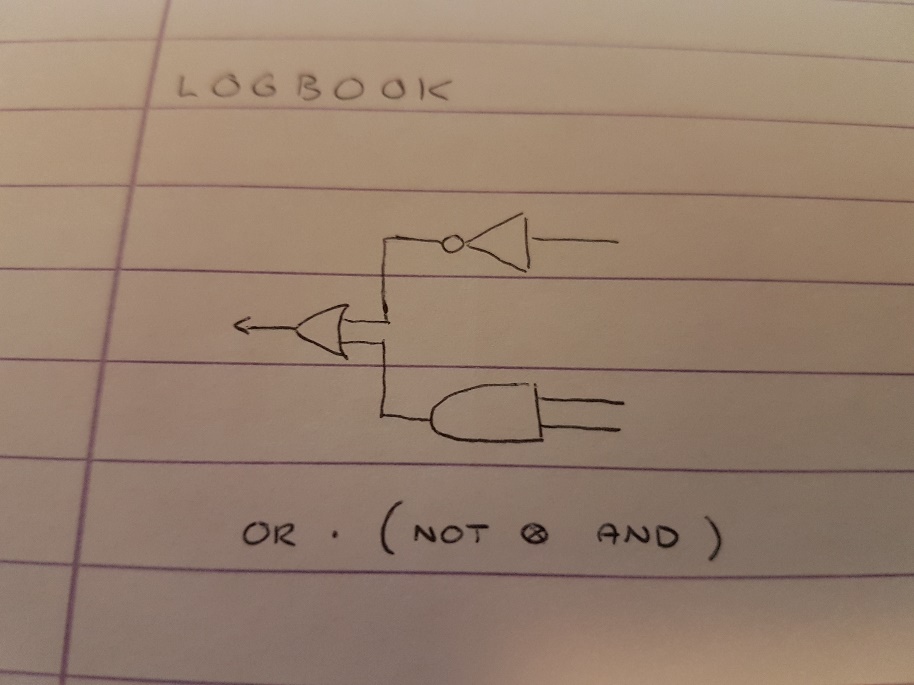
U1661665@unimail.hud.ac.uk

Logbook

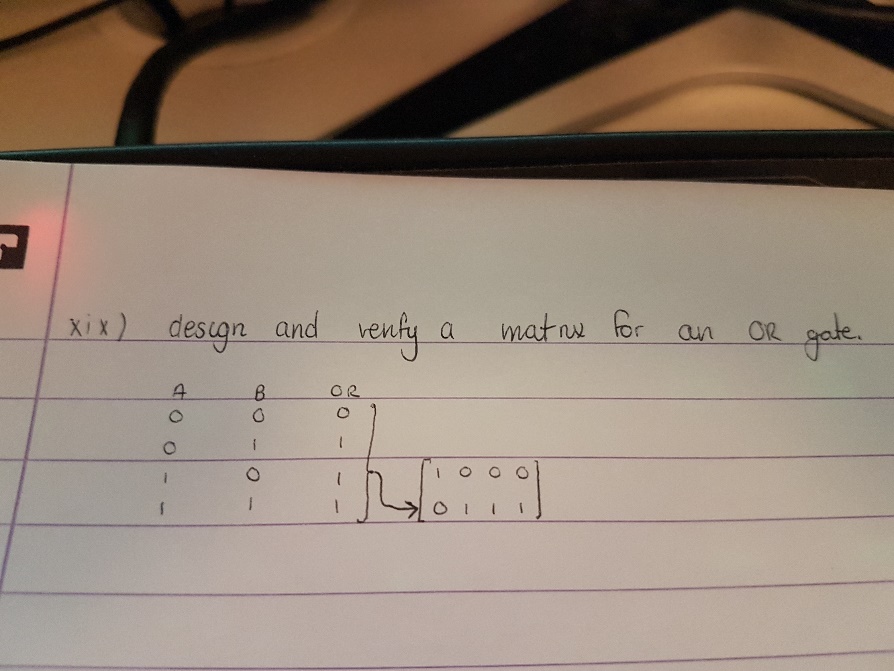
Algorithms – Processes and data

# Practical 13 (Week 17)

## (Logbook) What is the matrix for the following circuit?



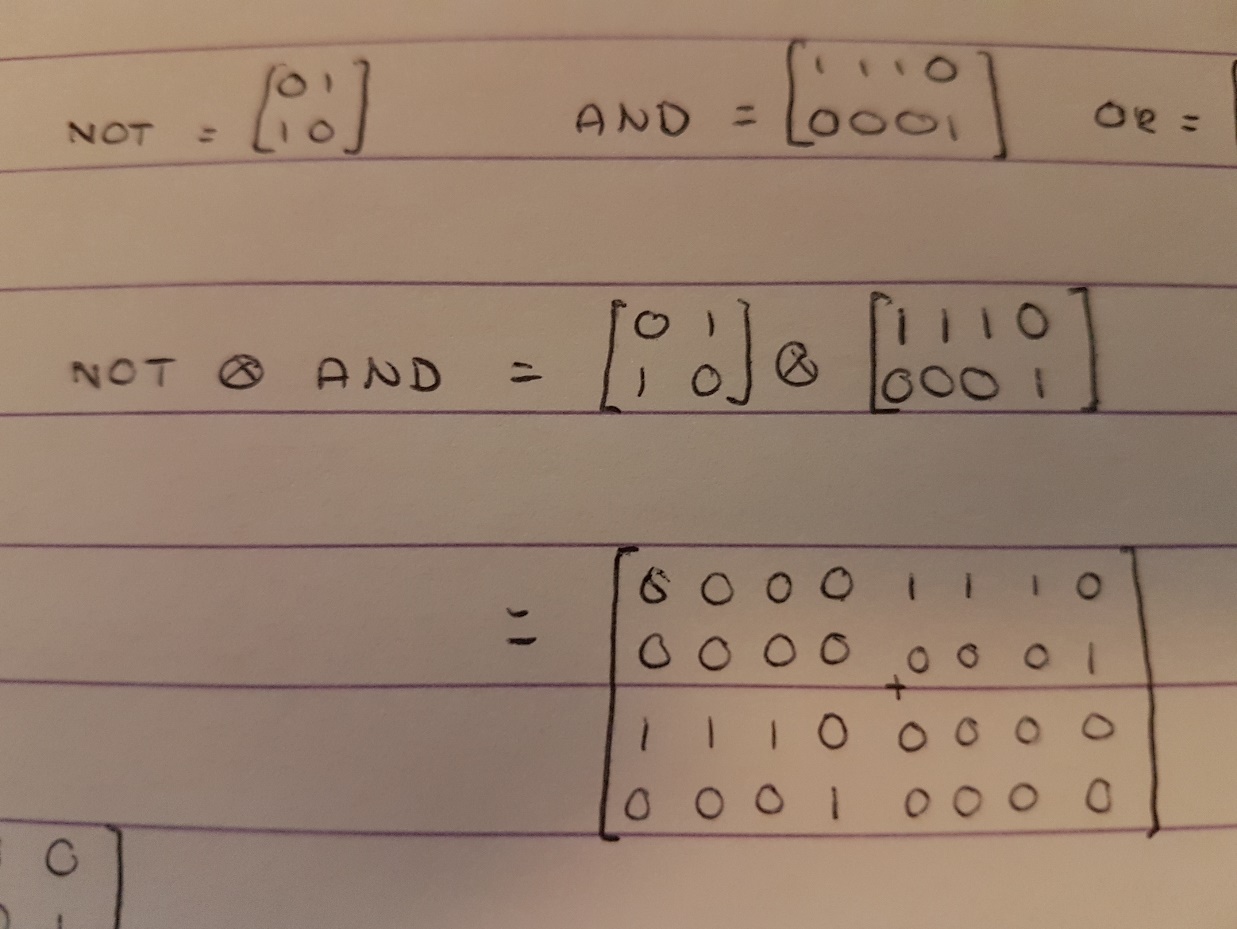
The Logbook is made of a parallel circuit of a NOT & AND, followed by a sequential OR. This is shown by the equation below the diagram.



I was asked previously to design a matrix for an OR gate.

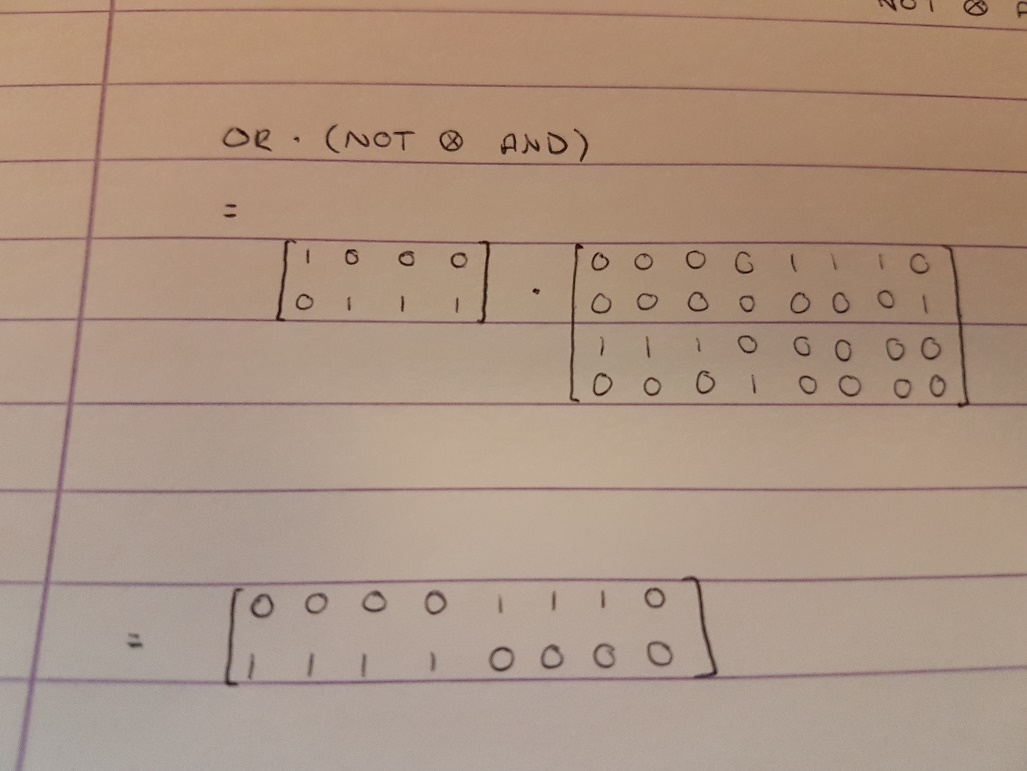
This consists of taking the truth table values for a regular OR gate, putting them top to bottom/left to right, then substituting the values for their individual bit sequence matrices. (1 = [0;1], 0 = [1;0]).

NOTE: the ordering of the 0 and 1 values in the truth table derivation is sensitive here. This must be done in the correct way (as shown) for the resulting matrix to be a correct representation of the gate.

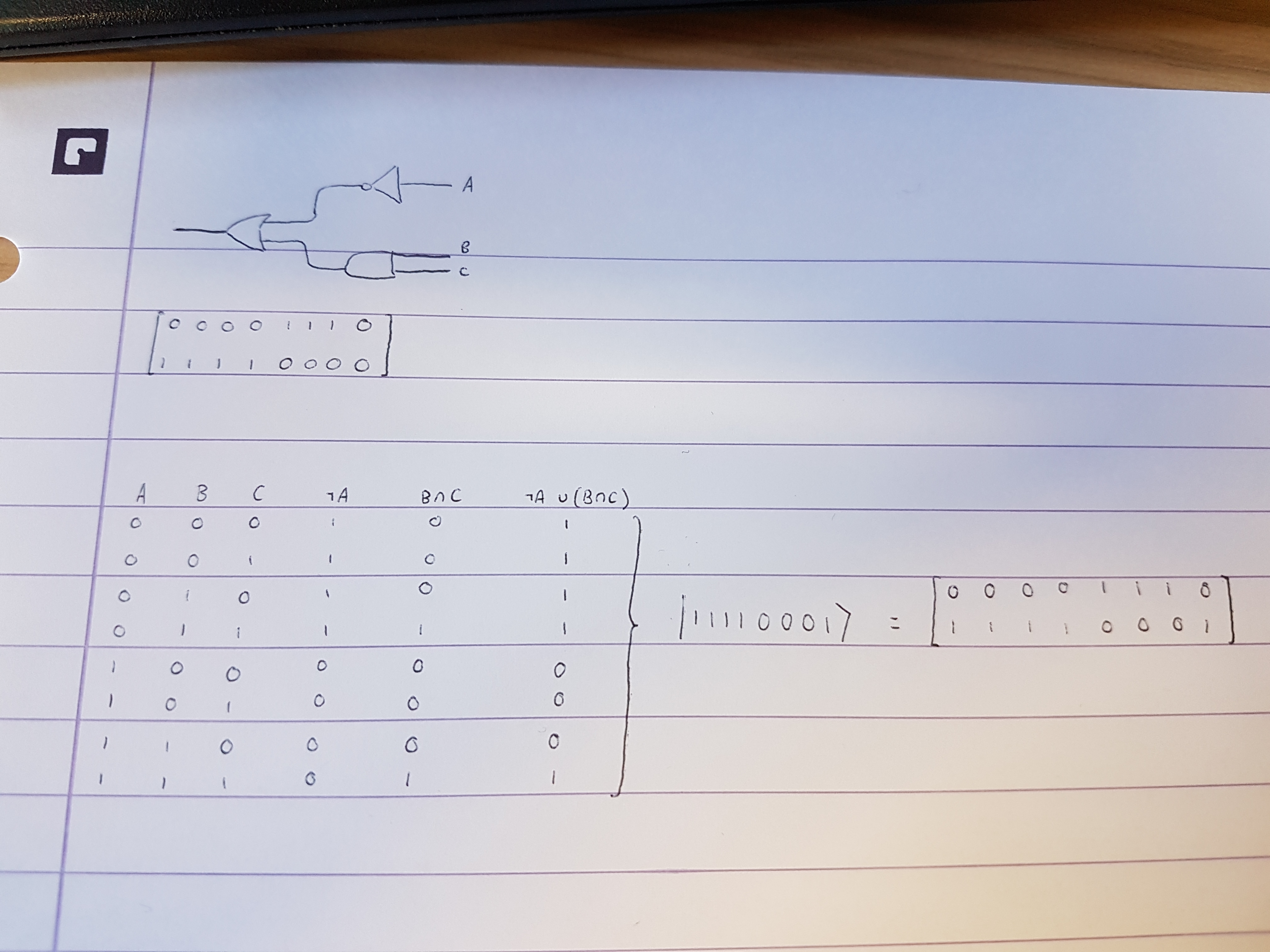


The first step to calculating the matrix for the entire circuit is calculating the parallel component first. This is done by calculating the tensor product of the two gates. Tensor product is calculating the scalar product of each element of matrix A with the entirety of matrix B.

This results in the matrix calculated in the picture.



The result of the previous calculation (NOT ⊗ AND) can then be applied to the sequential OR gate by simply multiplying the two matrices. This results in the complete gate for the circuit.

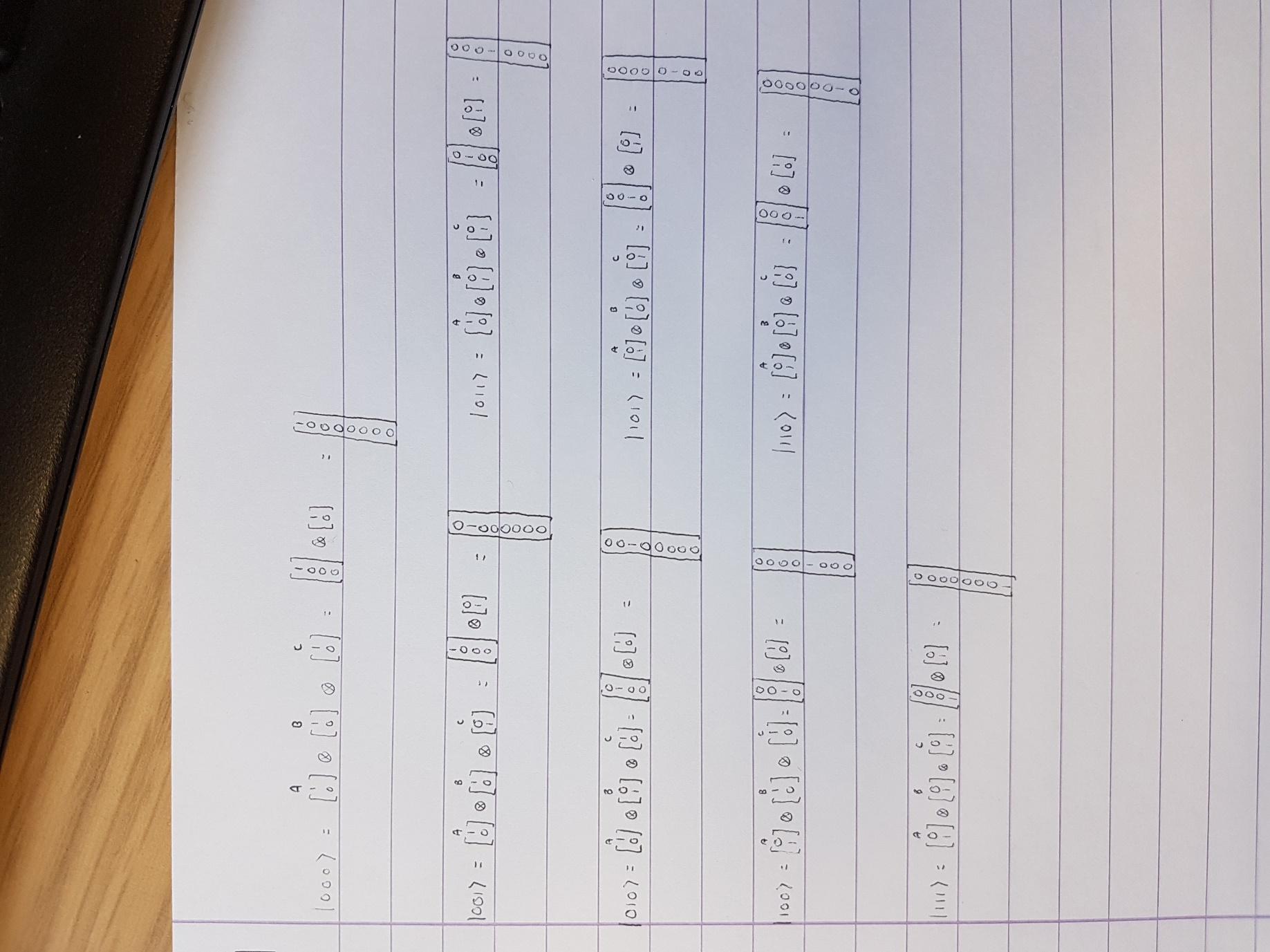


I can verify the Identity Matrix I created for the circuit by working it out with the truth table derivation. In the same way that I used them to derive the matrix for the OR gate, I can use the same ordering method to create a bit stream for the circuit, then turn this into the identity matrix for the circuit.

The two Matrices match, therefore the matrix I derived must be the identity matrix for the circuit.

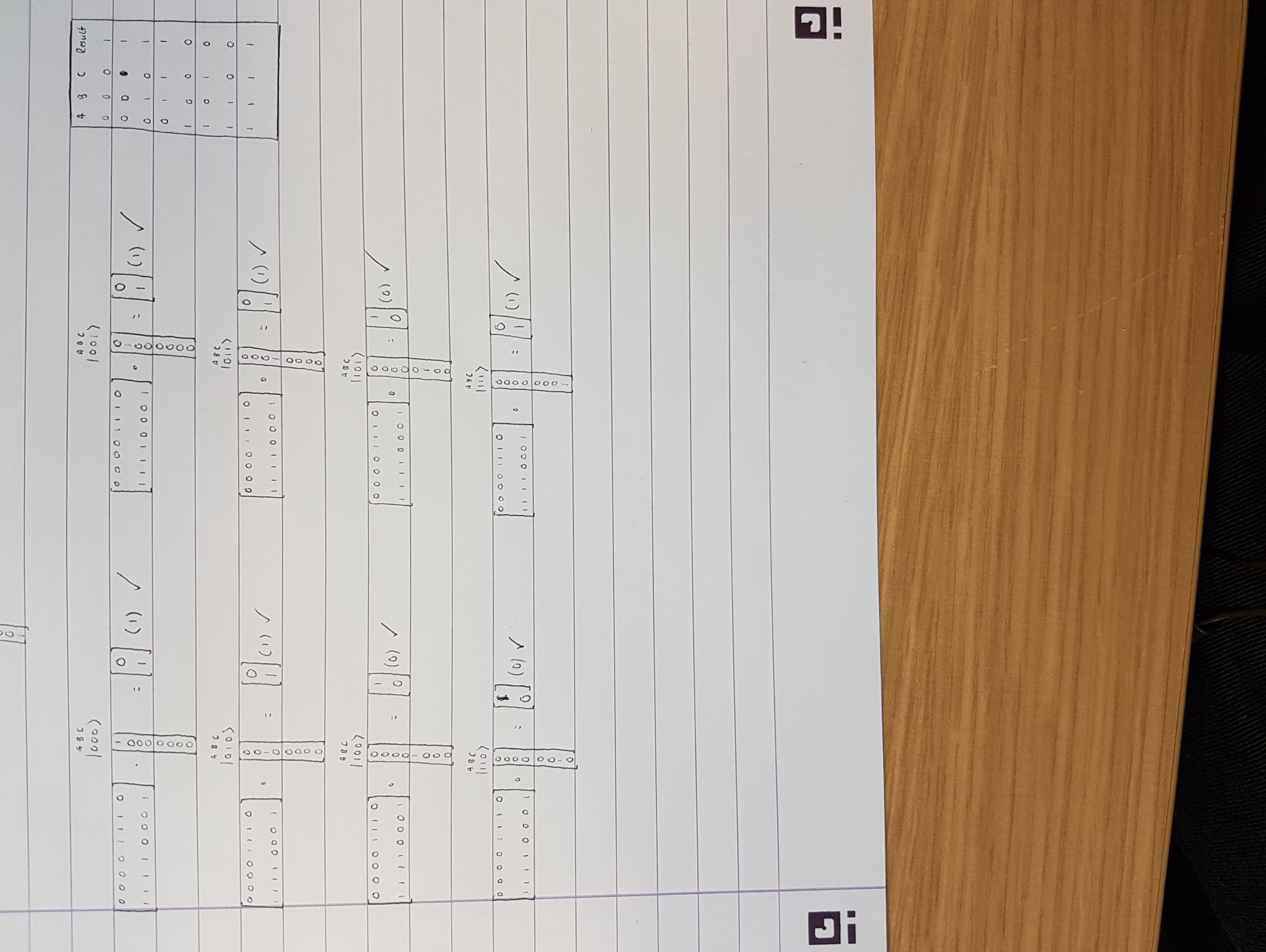
Additionally, there is some testing that can be done to the matrix.

The first step to this is calculating the bit stream matrices for each of the 8 (23) bit streams possible for the truth table/input to the circuit matrix.



Once these have been calculated they can be applied to the circuit matrix via matrix multiplication. This will produce a matrix for a bit of either 1 or 0.

This can be compared with the result for the according bit stream in the truth table. If it matches, then the circuit matrix is a match for that particular bit stream. The circuit matrix can be considered completely correct if every case is a match. The picture below shows that my constructed matrix is a perfect match.



## Self Evaluation.

For 3/5 marks this week I was asked to create a generally correct matrix model for the specified circuit. I have done this.

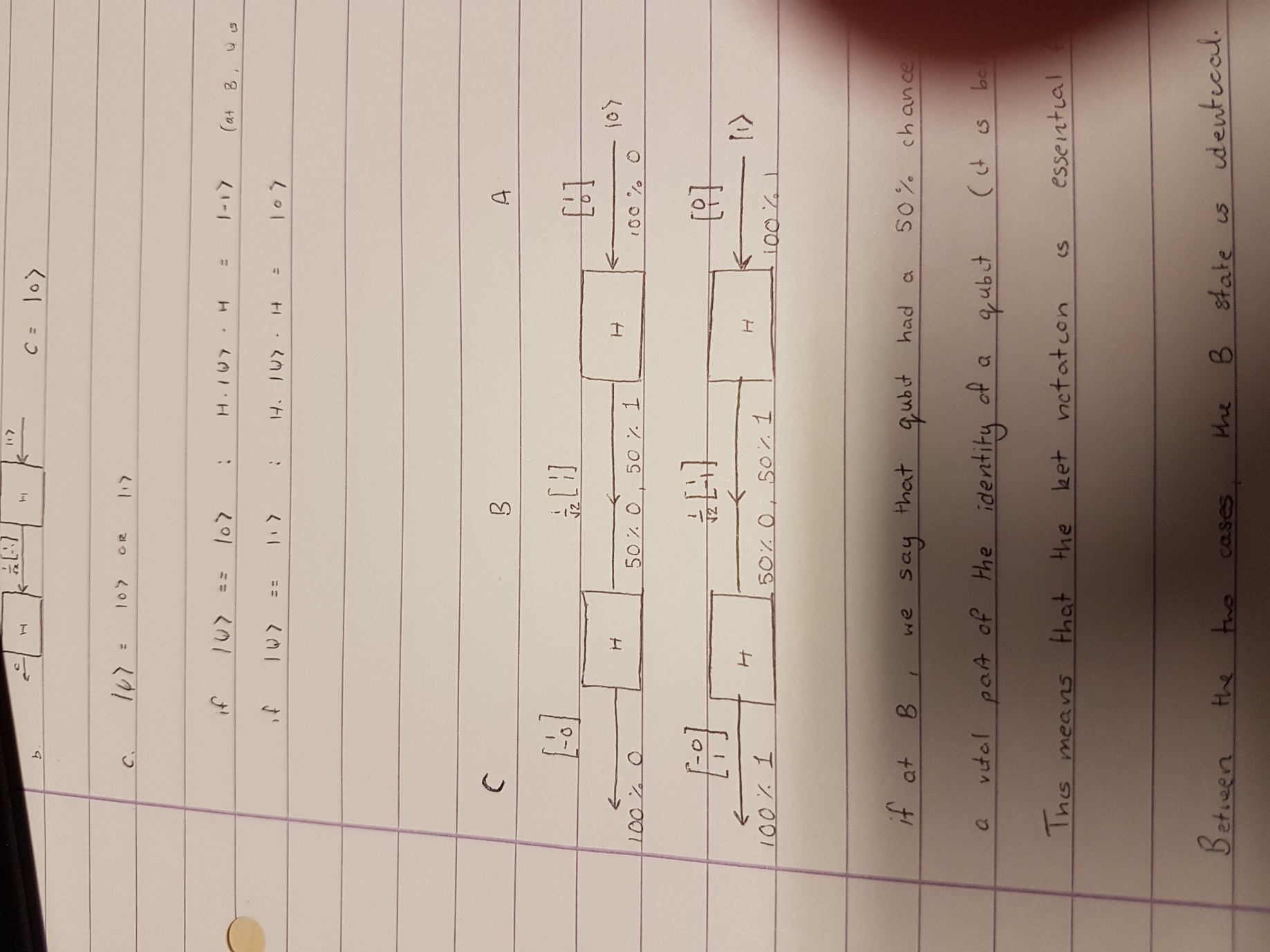
For (4/5)/5 marks I was asked to both explain the derivation and test the model. Throughout my creation I have explained each step of the construction of the matrix and provided verification and testing after the result was obtained.

For the reasons above, I believe my work this week is worth the full 5 marks available.

# Practical 14 (Week 20)

## (Logbook) Question 1.

If |**Ψ**> is a pure state (|**Ψ**> == |0> OR |**Ψ**> == |1>, what happens if |**Ψ**> is passed as an input to a circuit consisting of two Hadamard gates in sequence?



The above picture demonstrates the results of each possibility along with their probabilistic notations below the C, B and A areas.