Doug McLaren

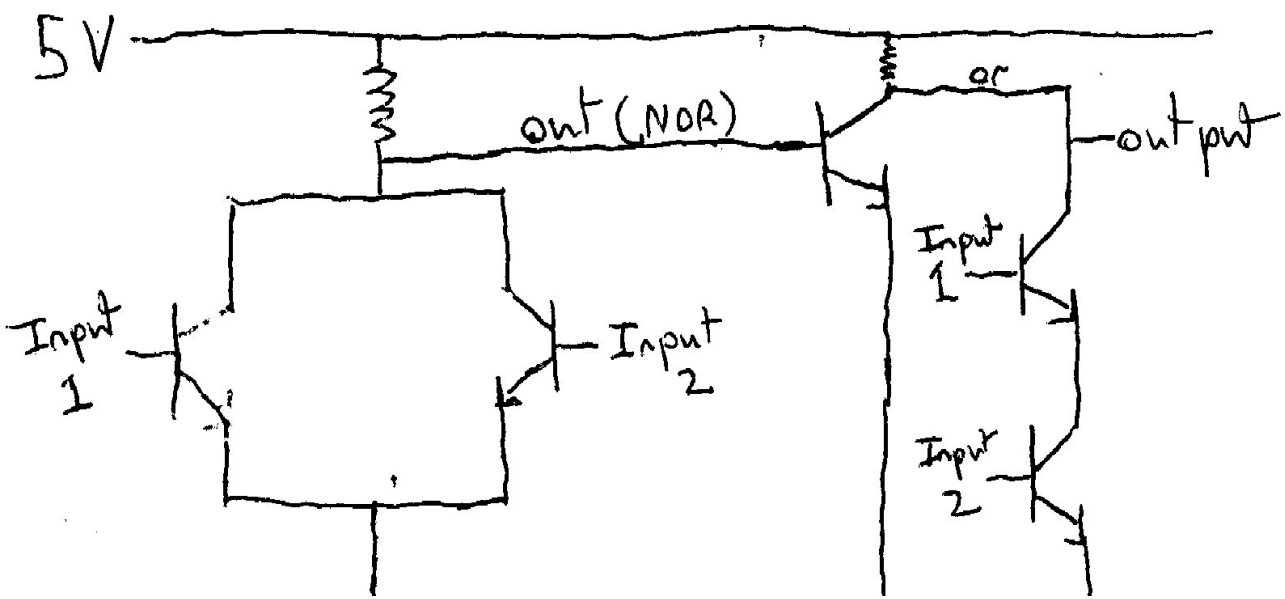
October 7, 2008

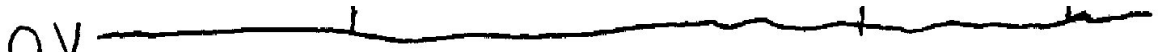
Outline:

0. Creating an XOR from 5 transistors and resistors?

XOR Stands for Exclusive Or Gate and its truth table is this-

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0



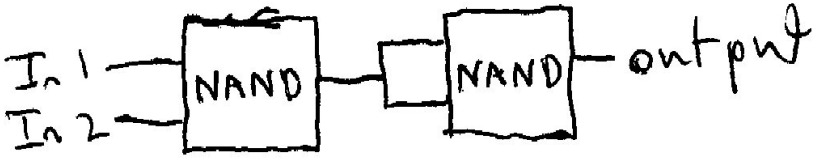


1. Creating AND, OR, NOT from NANDs.

And Gate -

Truth Table

In 1	In 2	out
0	0	0
1	0	0
0	1	0
1	1	1



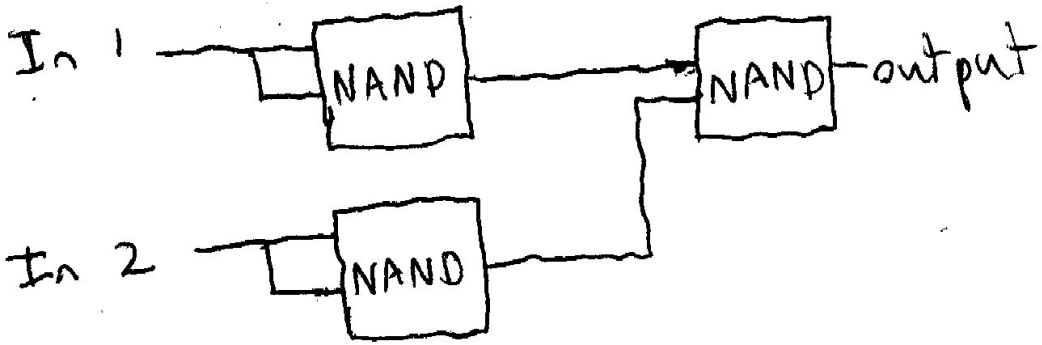
NAND Gate
Truth table

In 1	In 2	Out
0	0	1
1	0	1
0	1	1
1	1	0

Or Gate -

Truth Table

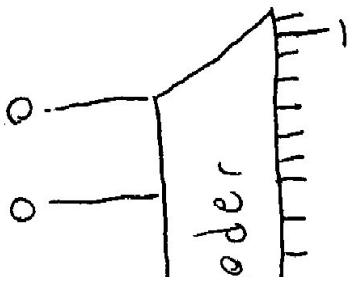
In 1	In 2	Out
0	0	0
1	0	1
0	1	1
1	1	1



2. Decoder. Computer instruction: operator (e.g. plus, -, *) and it has operands (what you do the operator on). Example from assembly language:
plus x1 x2
The result would be to add x1 and x2.
Assembly language translates to machine language and machine language only knows about binary – doesn't know what a letter is.
In machine language, we might have operators like +, -, * but we would represent them in binary. Let us say that the operator is represented by 4 bits so we can have 16 different operations. For example, 0110 is the encoding for + and 0111 is the encoding for -.
Inside the processor, the operator gets "decoded" to a signal on one out of 16 wires.
Counting in binary:
0000 – 0
0001 – 1
0010 – 2
0011 – 3
0100 - 4

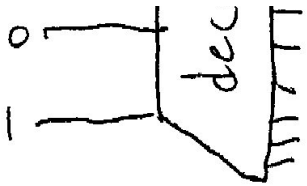
You have a controller and a controller has a certain number of values

4-to-16 decoder
bits
wires



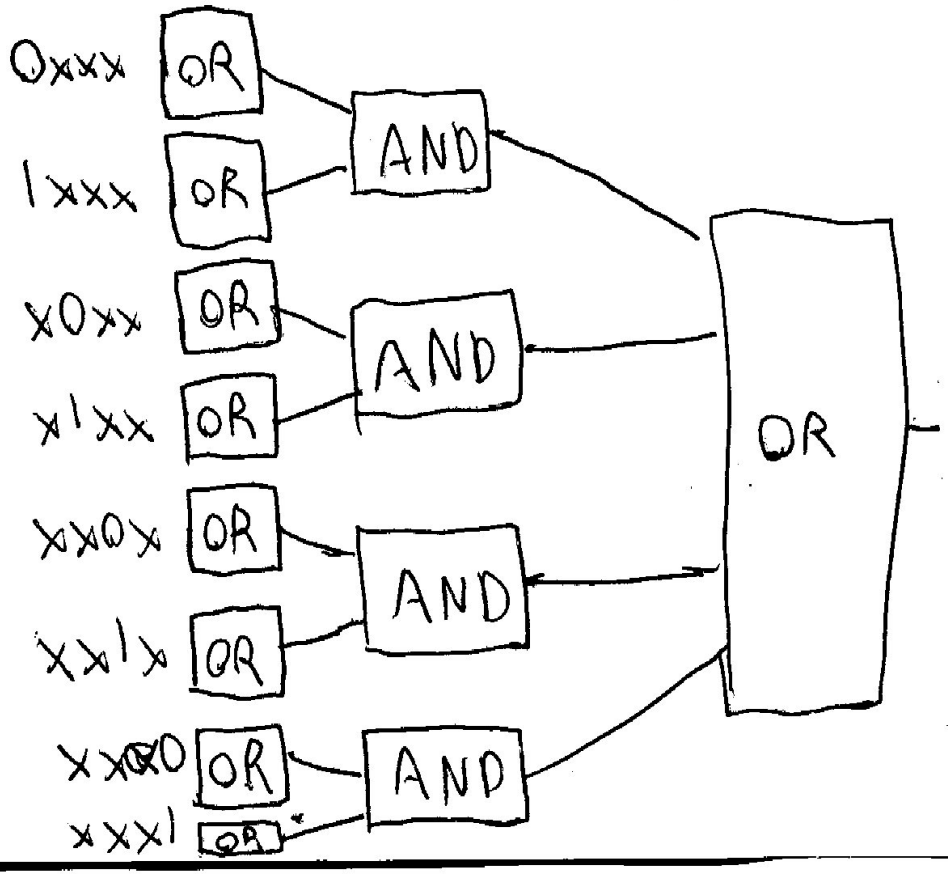
On this decoder we show the binary code 0001 getting put in

which would read out the value of 1



3. Dr. Ecco puzzle

Solution



0xxx corresponds to all 4 digit binary values with 0 being the first digit

Using this set of gates, we can see that if 2 circuits are going then one of the two sets of Or gates will run causing one of the And gates to run causing the final Or gate to run. Therefore if we have output of 1 from this set up 2 circuits must be running