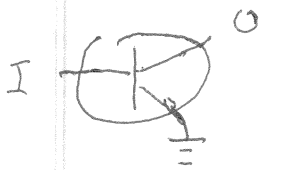
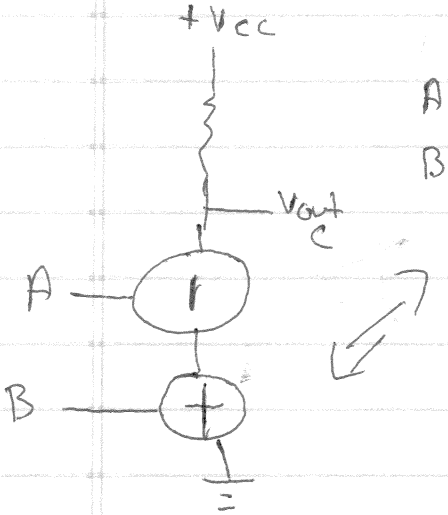
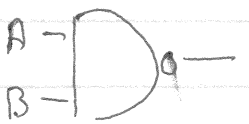
4-1



C	A	B
0	0	0
0	0	1
0	1	0
1	1	1

AND

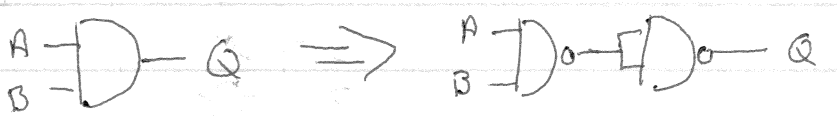




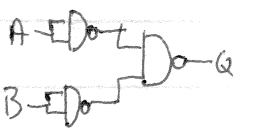
C	A	B
1	0	0
1	0	1
1	1	0
0	1	1

NAND

NOT	A
1	0
0	1



OR	A	B
0	0	0
1	0	1
1	1	0
1	1	1

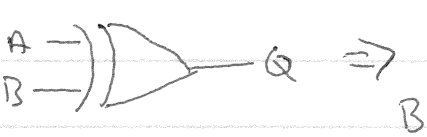


De morgan's Law :

!A AND !B = !(~~OR~~ A or B)

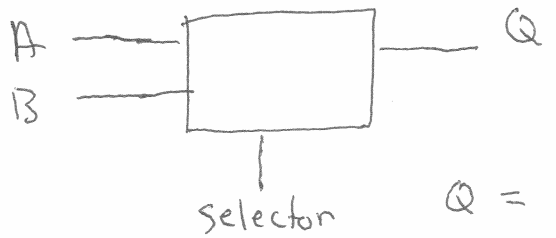
NOR = !OR = 

XOR	A	B
0	0	0
1	0	1
1	1	0
0	1	1



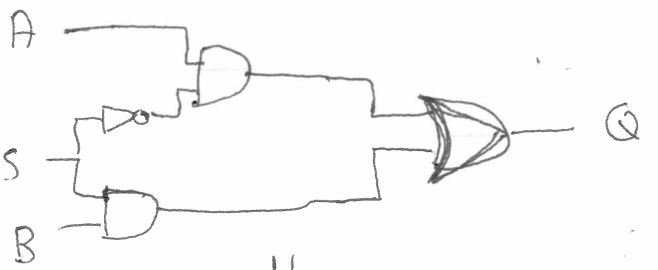
XOR AB = OR ($\bar{A}$ and B) (A and $\bar{B}$)

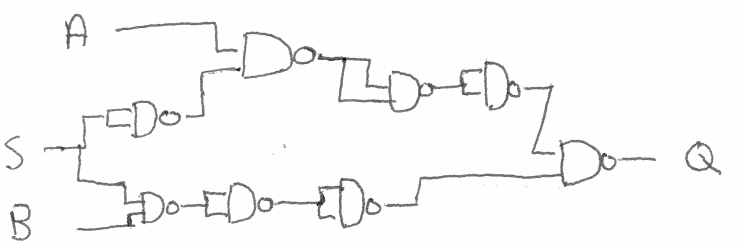
1-2 / Multiplexing

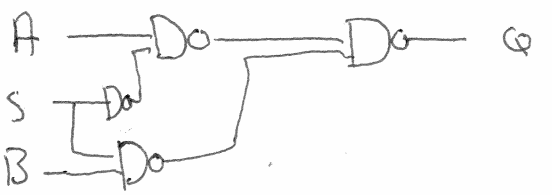


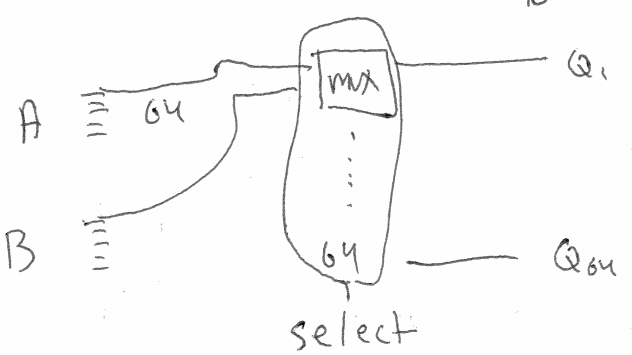
Q = A if S = 0
B if S = 1

A	B	S	Q
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

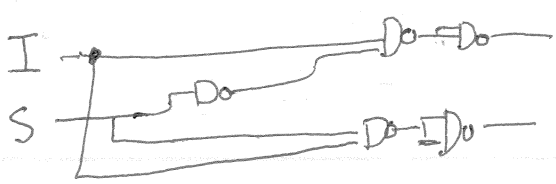








4-3/ ~~Decoder~~ Demux /



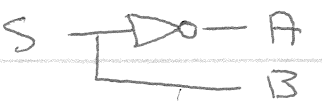
If I is always TRUE, Decoder

S	A	B
0	1	0
1	0	1

S	B	A
0	0	1
1	1	0

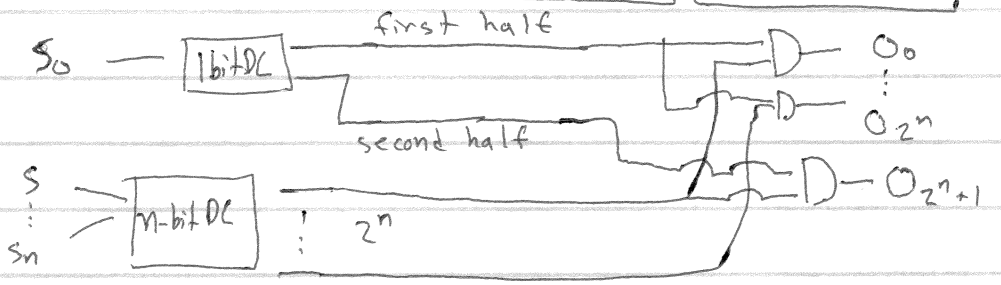
2-bit decoder ⇒ 4 output wires

0 0	0 0 0 1
0 1	0 0 1 0
1 0	0 1 0 0
1 1	1 0 0 0



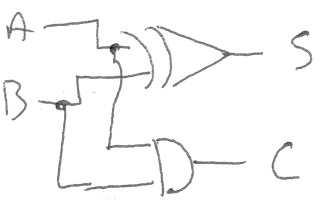
n+1 -bit decoder ⇒ 2^{n+1} output wires

S_0 S_1 ... S_n | O_0 ... O_{2^n} | O_{2^n+1} ... $O_{2^{n+2^n}}$



1-4/

A	B	C	S
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	0

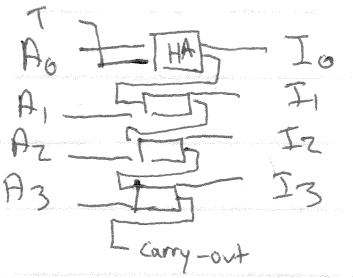


Half-Adder

2-bit increment

$HA(A_0, T) = (C_0, S_0)$

$HA(A_1, C_0) = (C_1, S_1)$



Next time:
Full Adder
ALU