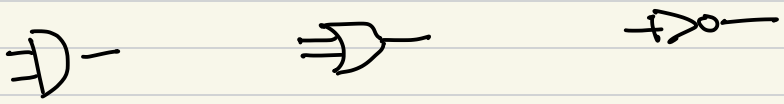
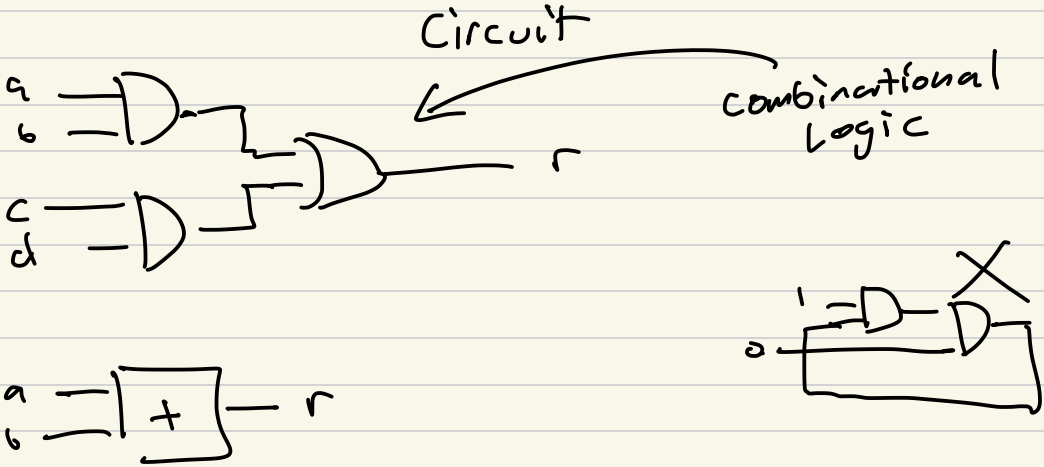
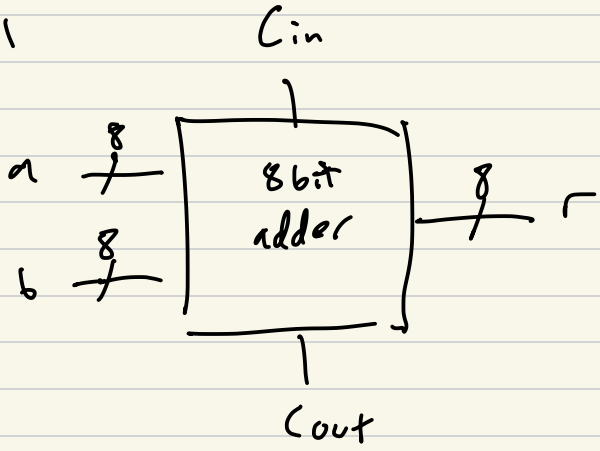
CS 315-01 Combinational Logic

AND OR NOT





Goal



sum-of-products

Function:

Input: Take a 3 bit number (3 bits)
$n_2 n_1 n_0$

Output: Two 1 bit outputs: even, odd

Goal: determine if the number of "1" bits is even or odd

010 → odd
110 → even

→ sum-of-products
1) truth table
2) identify rows with output '1'
3) build equation

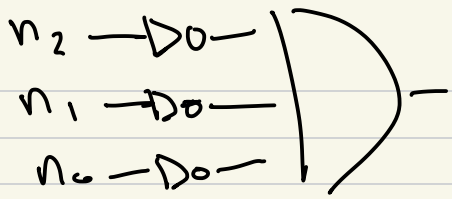
n_2	n_1	n_0	even	odd
0	0	0	1	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

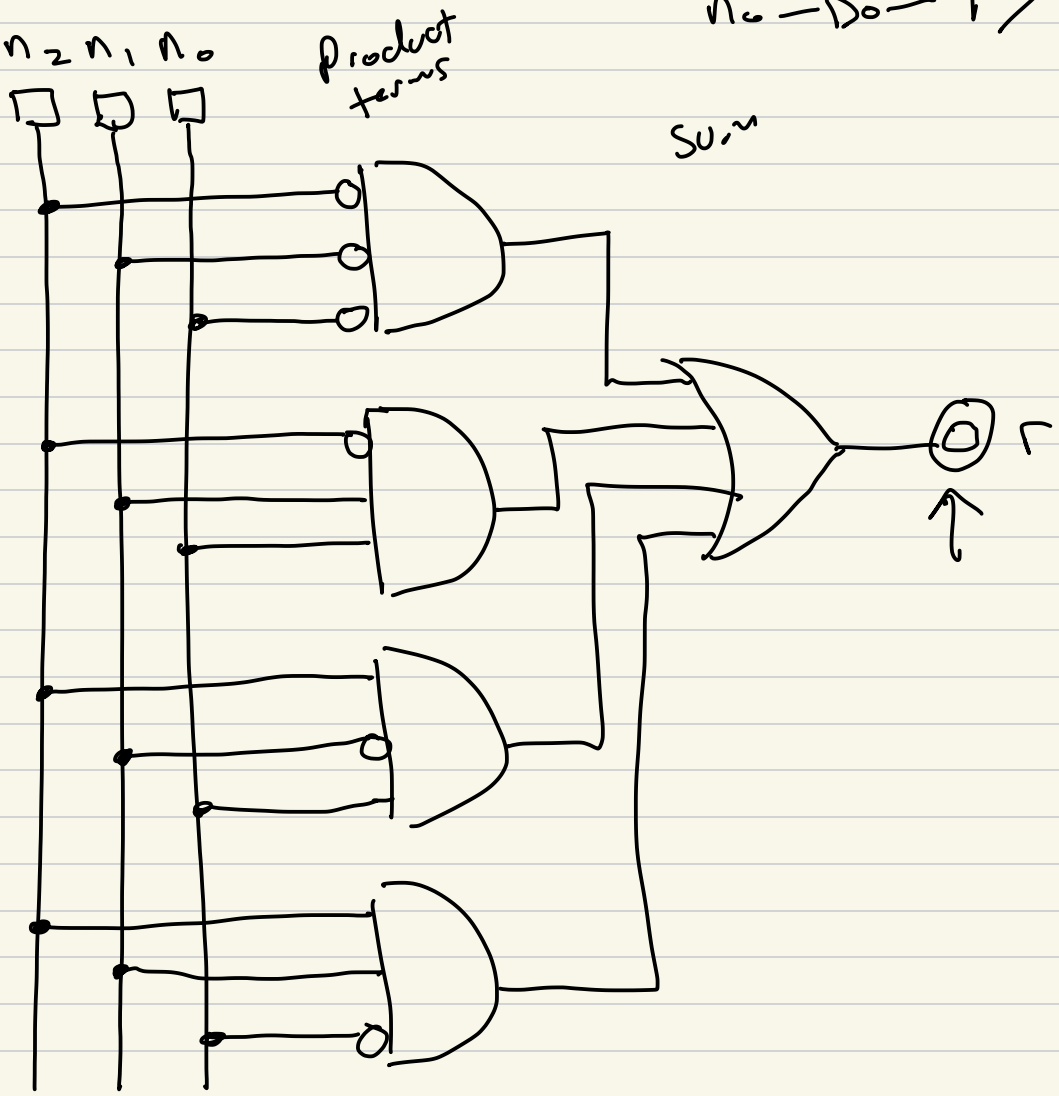
product

$$even = (\bar{n_2} \cdot \bar{n_1} \cdot \bar{n_0}) + (\bar{n_2} \cdot n_1 \cdot n_0) + (n_2 \cdot \bar{n_1} \cdot n_0) + (n_2 \cdot n_1 \cdot \bar{n_0})$$

sum

$odd = \overline{even}$





Lab 08 Part 2

max2 find the max 2-bit value

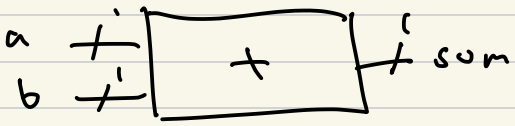
$a = 01$ $b = 10$

$max2(a, b) = b$

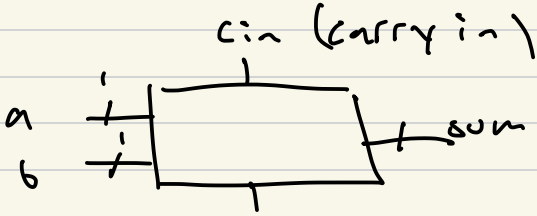
$(01, 10) = \boxed{10}$ ←

Goal: Implement an 8-bit adder

1-bit adder

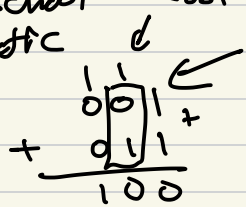


1-bit half adder



1-bit full adder

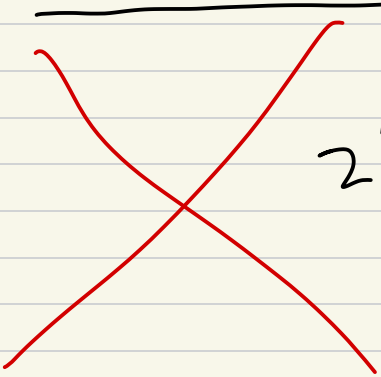
grade school arithmetic



a	b	cin	sum	cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

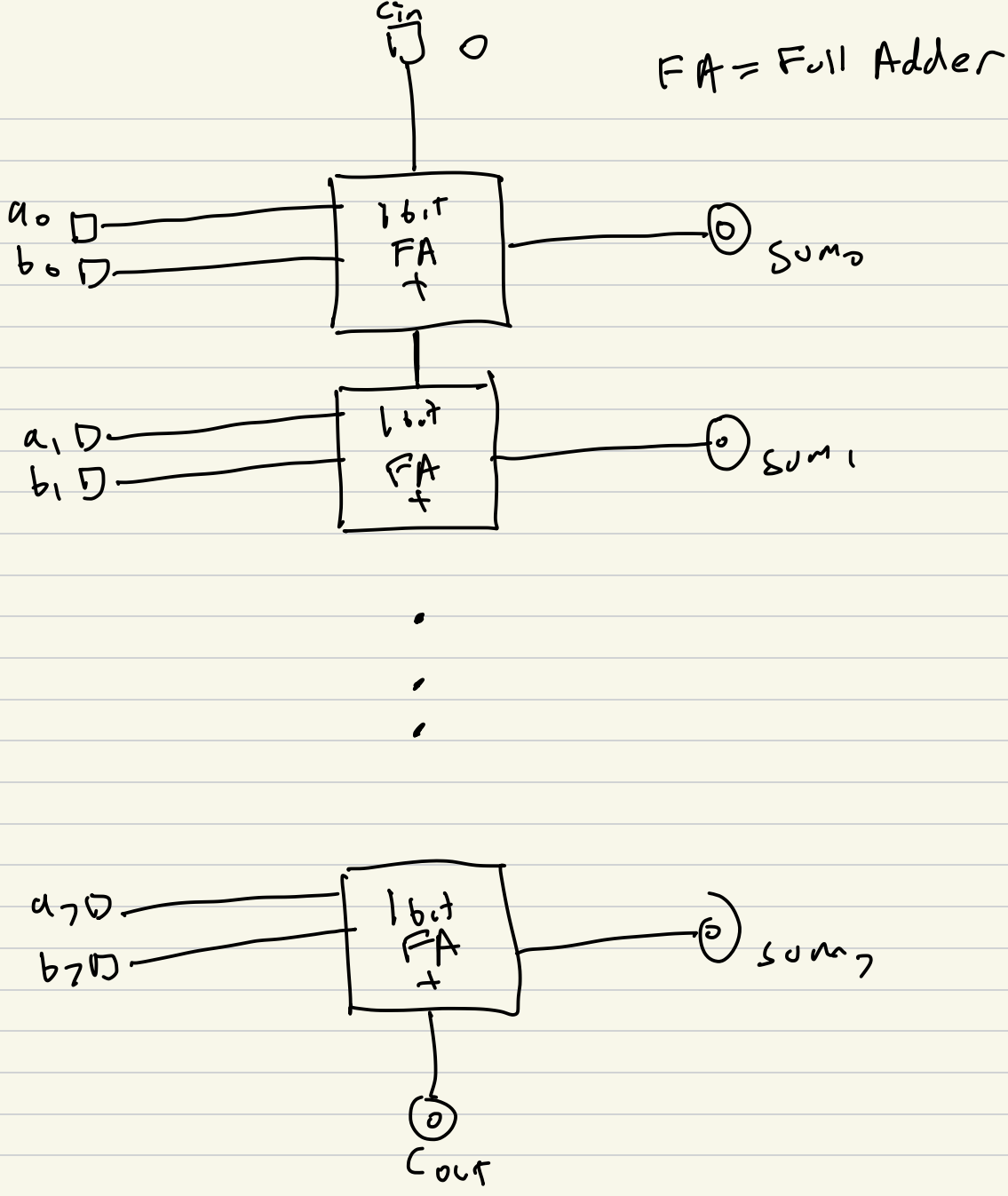
Could . . .

c_{in} a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	c_{out}	sum_7 -- sum_0



2^{17} rows

Combine 8, 1-bit full adders to build an 8 bit ripple-carry adder


Cin
0
FA = Full Adder
a0
b0
1 bit FA +
SUM0
a1
b1
1 bit FA +
SUM1
a7
b7
1 bit FA +
SUM7
Cout

8 bit Full adder