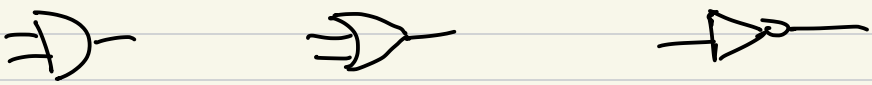
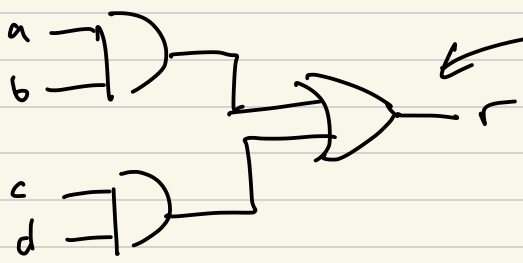
CS315-02 Combinational Logic

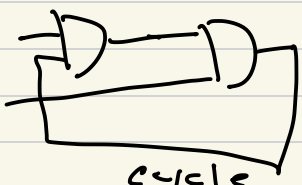
AND OR NOT

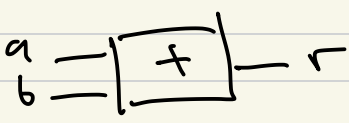


Circuit

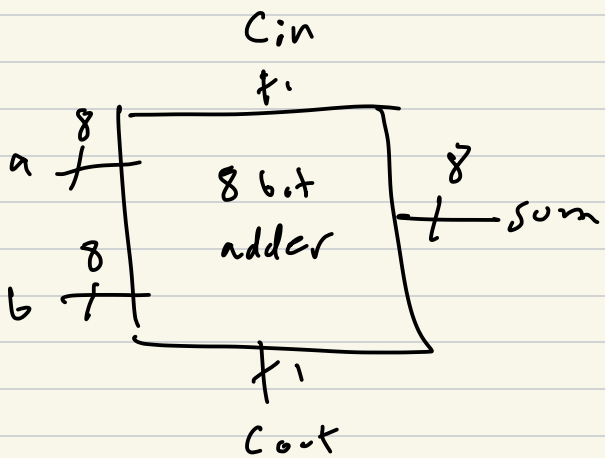
Combinational logic







Goal



sum_of_products

function:

input: a 3 bit number (3 bits)
$n_2\ n_1\ n_0$

output: two bits: even, odd

goal: determine if the number of "1" bits are even or odd

$n_2 n_1 n_0$

0 1 0 → odd
1 1 0 → even

sum-of-products

1) truth table
2) identify row with "1" output
3) build the SOP 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 terms

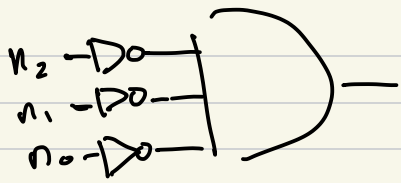
$$even = (\overline{n_2} \cdot \overline{n_1} \cdot \overline{n_0}) + (\overline{n_2} \cdot n_1 \cdot n_0)$$
$$+ (n_2 \cdot \overline{n_1} \cdot n_0) + (n_2 \cdot n_1 \cdot \overline{n_0})$$

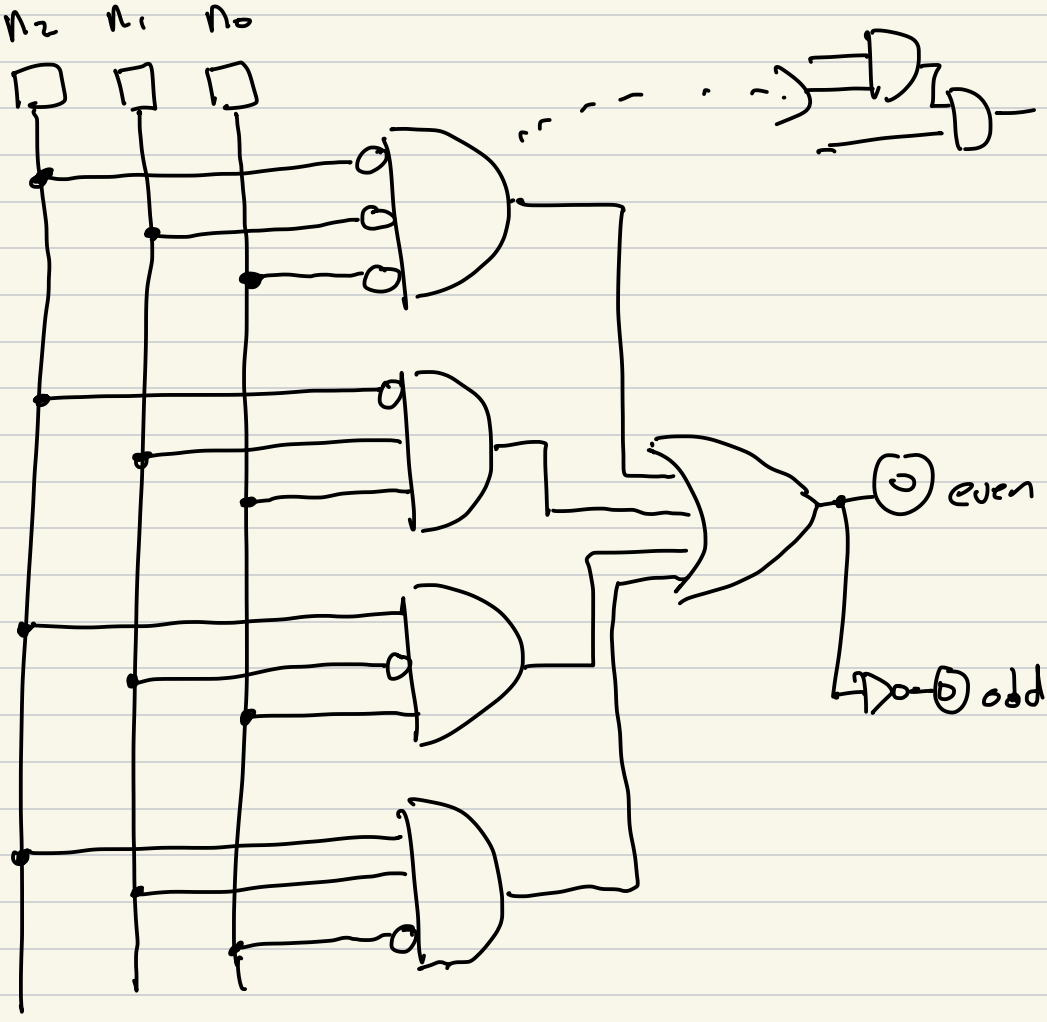
sum term

odd = $\overline{even}$

$(\bar{0} \cdot \bar{0} \cdot \bar{0})$

$(1 \cdot 1 \cdot 1) = 1$





Lab 08 Part 2

max2 finds the max value of two 2-bit numbers

a = 01
b = 10

max2(a, b) = b
max2(01, 10) = 10

$2^4 = 16$

a_1 a_0 b_1 b_0	r_1 r_0
⋮	⋮
0 1 1 0	1 0
⋮	⋮

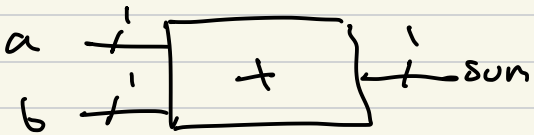
SoP

$r_1 =$ + +

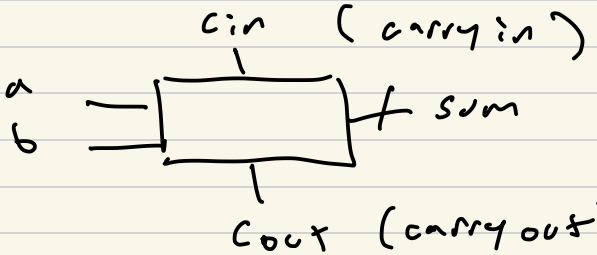
$r_0 =$ + +

Goal: Implement an 8-bit Adder

1-bit adder



1-bit half adder



1-bit full adder

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

grade school arithmetic

```
  1 1
  0 0 1
+ 0 1 1
  0 0
```

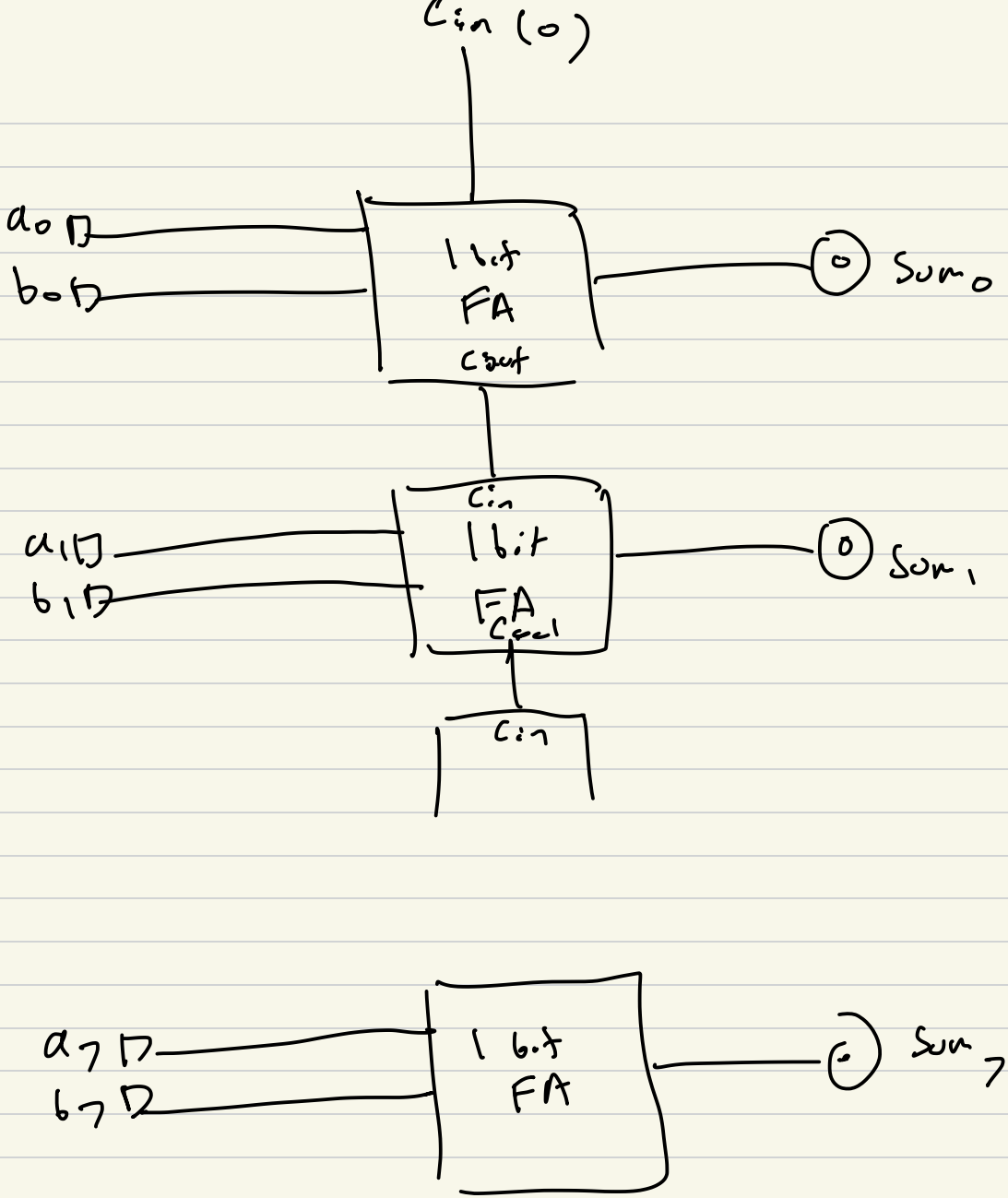
```
  1
  0 1 0 1
+ 0 1 1 0
```

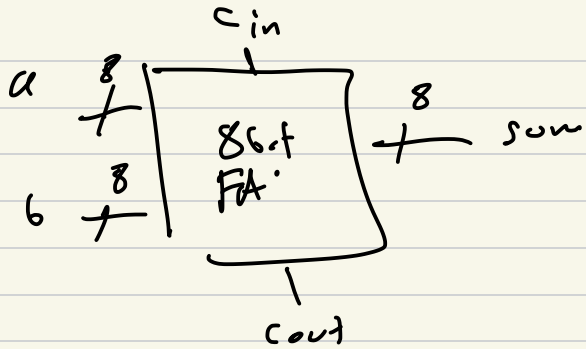
8 bit adder

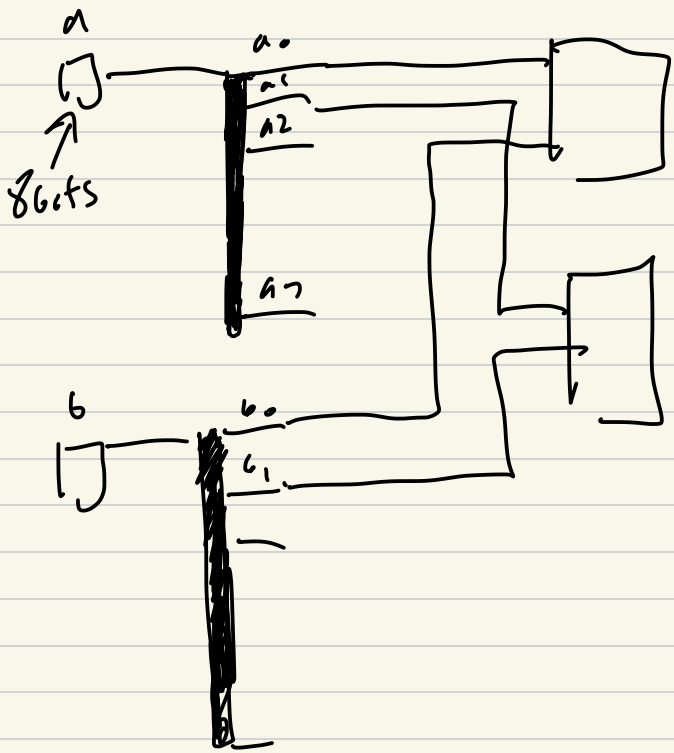
$a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0\ b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0\ C_{in}$ | $sum_7 - sum_0$ C_{out}

2^{17} rows

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


Cin (0)
a0
b0
1 bit FA
Cout
Sum0
Cin
a1
b1
1 bit FA
Cout
Sum1
Cin
a7
b7
1 bit FA
Sum7


Cin
a
8
8
8
sum
b
8
8bit
FA
cout


a
a0
a1
a2
a7
8bits
b
b0
b1