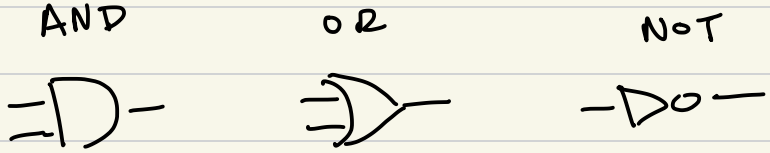
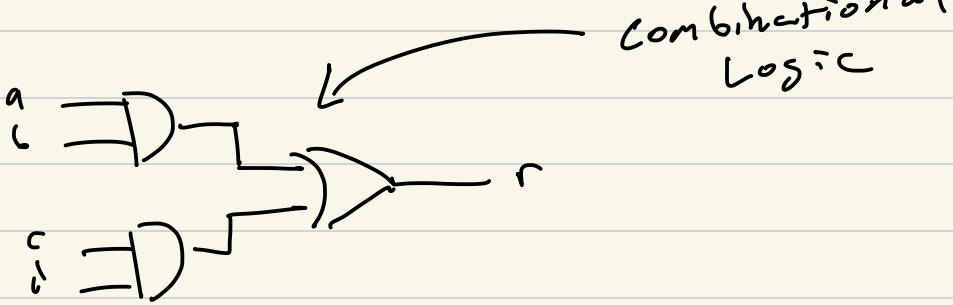
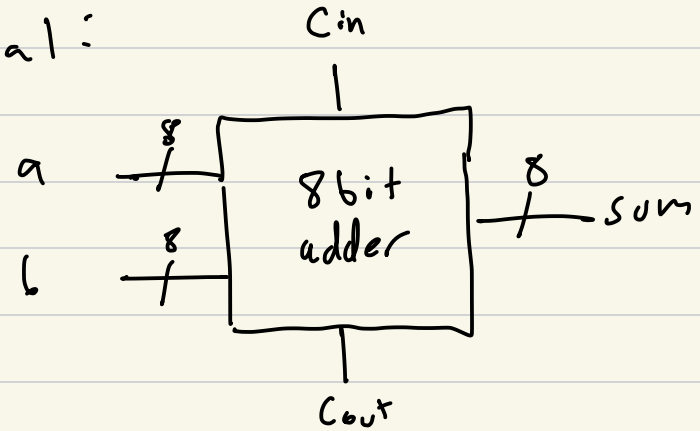
CS 315-01 Combinational Logic Adders



Sum-of-products



Goal:



Sum-of-products

A new function

3-bit number $n_2 n_1 n_0$ (6.75)

Two 1-bit outputs: even odd

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

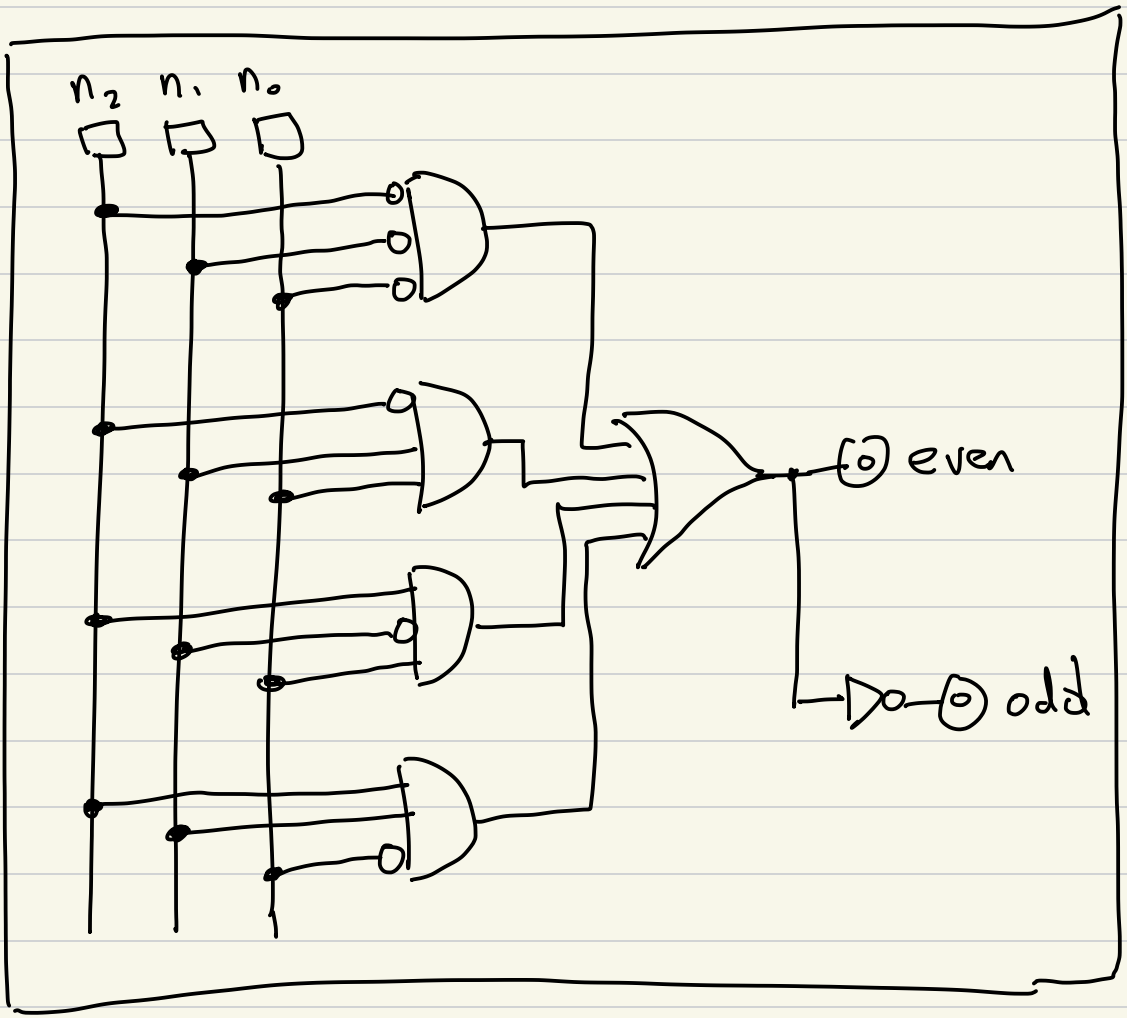
1	1	0	even
1	1	1	odd
1	0	0	odd

Sum-of-products
1) truth table
2) Look at rows with output of 1
3) form SOP equation

	n_2	n_1	n_0	even	odd
1	0	0	0	1	0
	0	0	1	0	1
	0	1	0	0	1
2	0	1	1	1	0
	1	0	0	0	1
3	1	0	1	1	0
4	1	1	0	1	0
	1	1	1	0	1

$$even = (\bar{n}_2 \cdot \bar{n}_1 \cdot \bar{n}_0) + (\bar{n}_2 \cdot n_1 \cdot n_0) + (n_2 \cdot \bar{n}_1 \cdot n_0) + (n_2 \cdot n_1 \cdot \bar{n}_0)$$

$$odd = \overline{even}$$



Digital Components

even-odd-bits

even-odd-multi

Lab 04 Part 2 Max 2

2 bits $a_1 a_0$ 2 bits $b_1 b_0$ 2 bits $r_1 r_0$

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

T_{max}

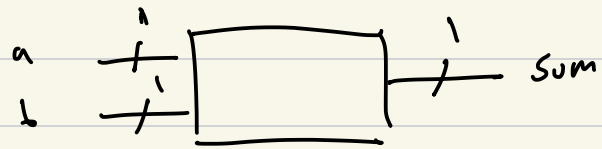
$r_1 =$ _____

$r_0 =$ _____

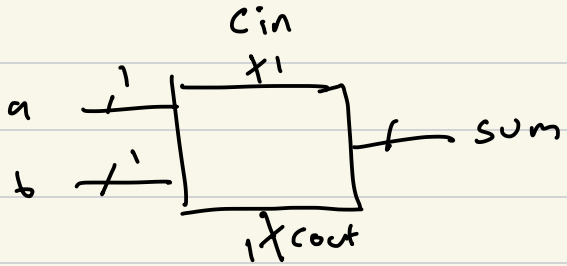
hint:

use not (invert) to reduce the number of product term

1 bit Adder

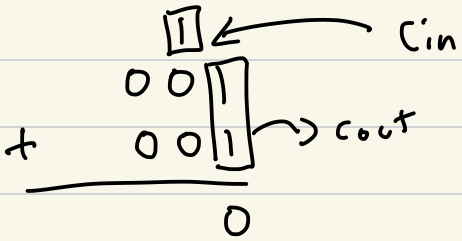


1 bit
half adder



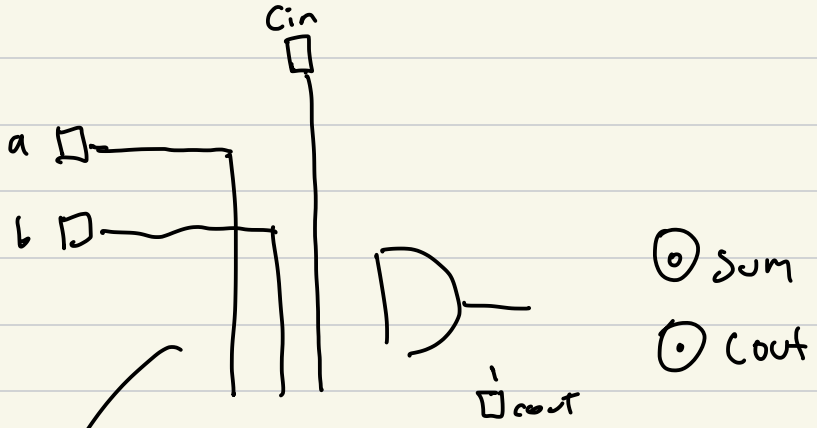
1 bit
full adder

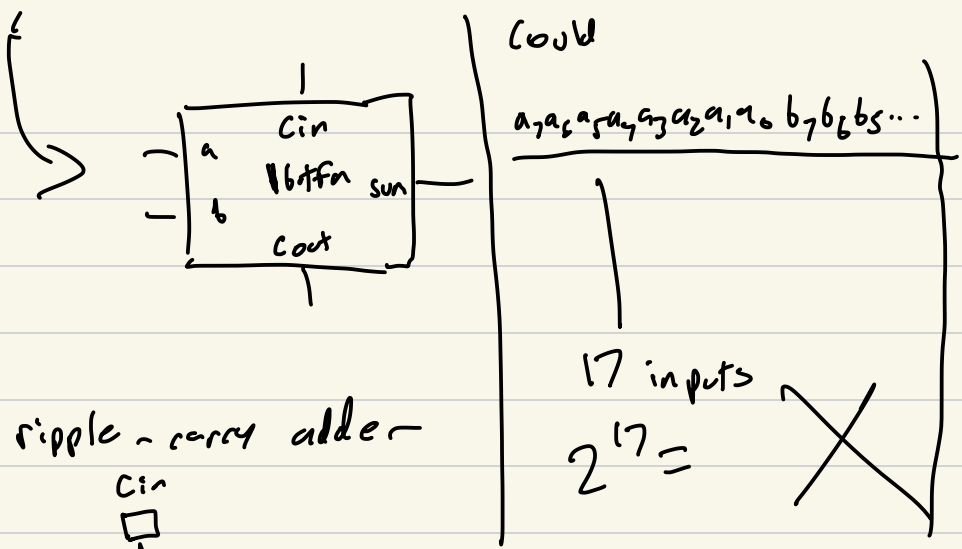
Build truth table (part 3)



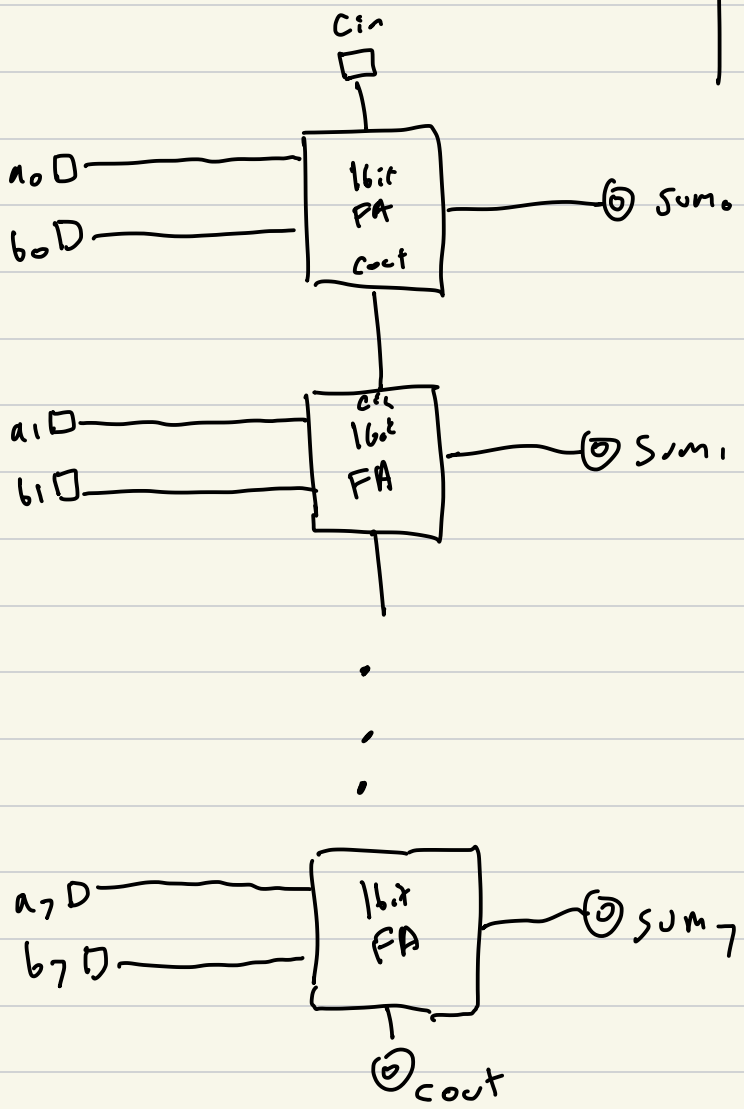
1bit full adder component

layout node

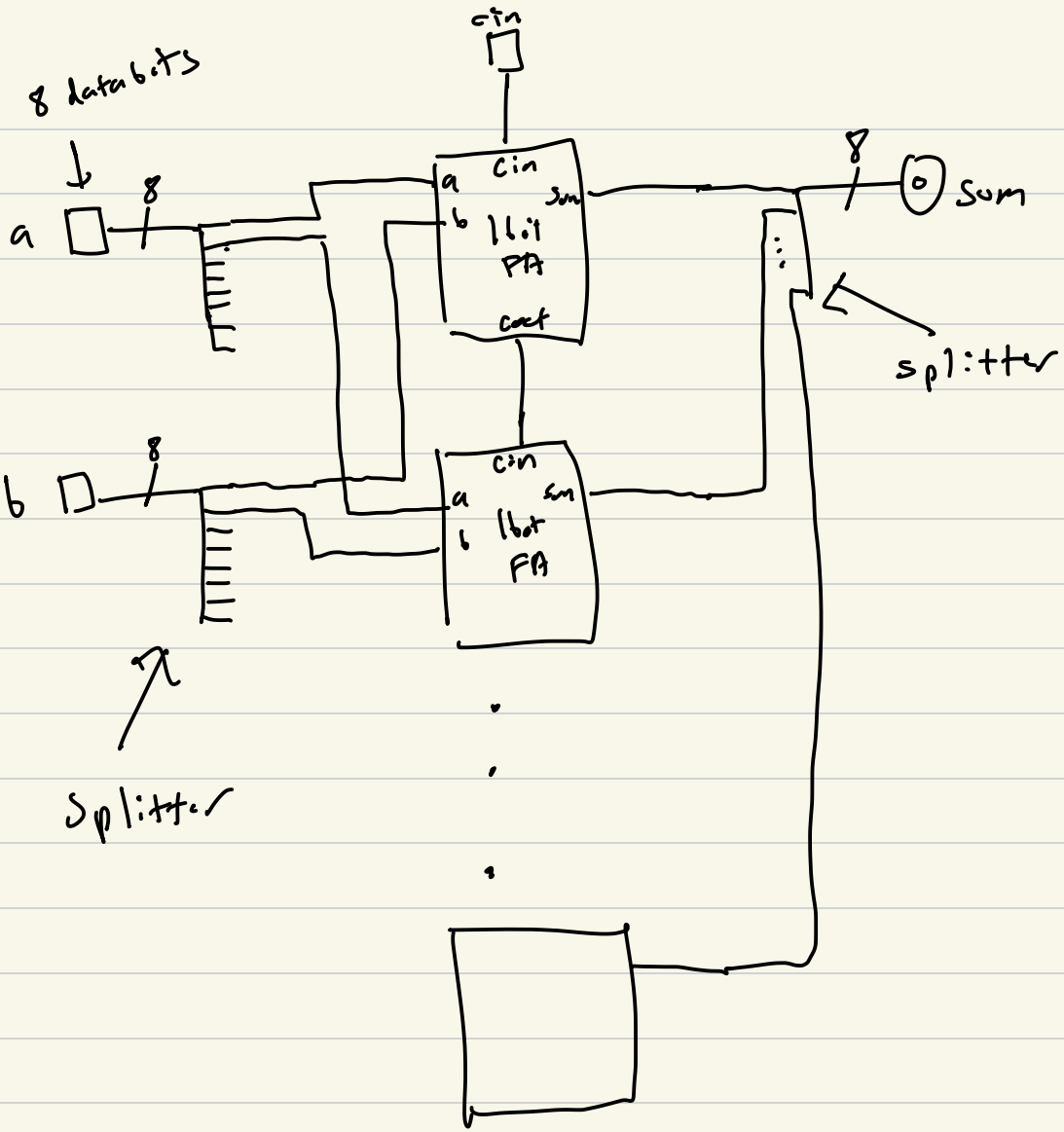




8 bit ripple-carry adder



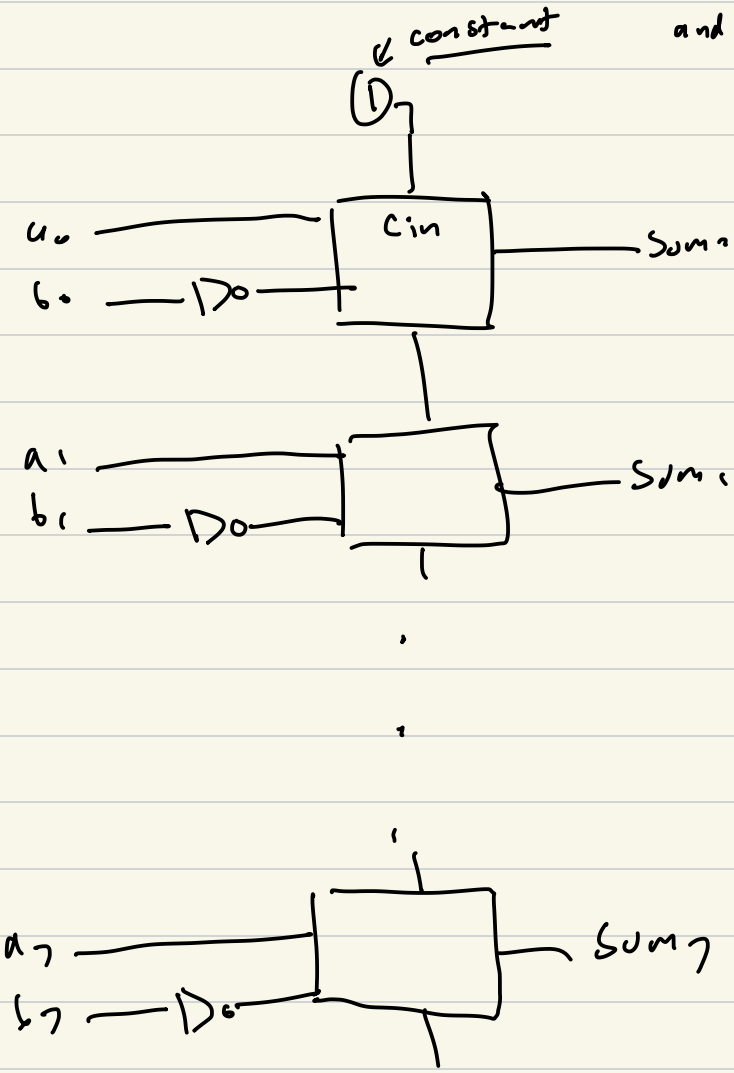
FA = Full Adder


8 data bits
cin
a
8
Cin
a
b
sum
1bit
FA
cout
8
0
sum
splitter
b
8
Cin
a
sum
b
1bit
FA
splitter

Subtraction

$A - B == A + (-B)$ for 2's comp

↑ invert and add 1



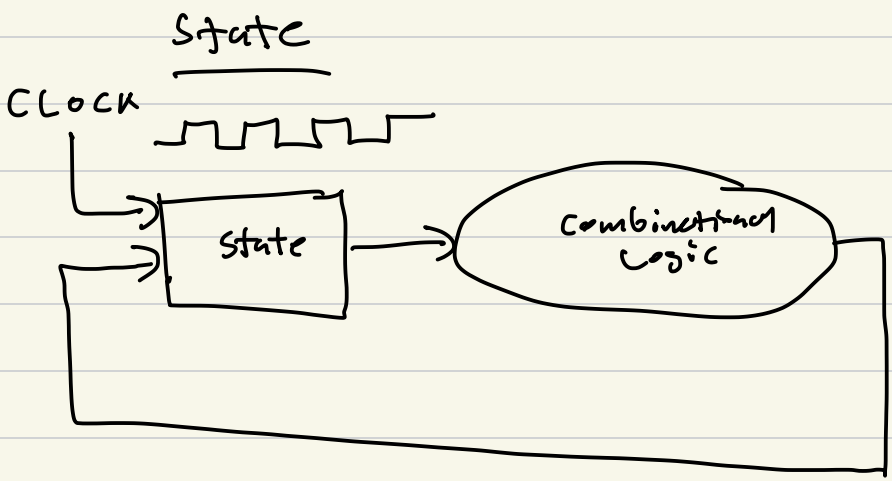
Combinational Logic

↳ components

Multiplexor

Decoder

Sequential Logic



processor