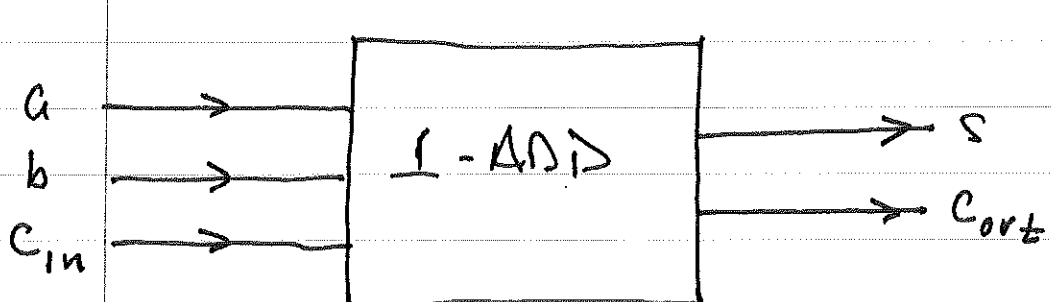
CMPS 10   11-9-10

Ex. n-Bit Full adder

for example n=8

```
 1 0 1 1 0 1 1 0 0
   1 0 0 1 0 1 1 0
   1 0 1 1 0 0 1 1
 -----------------
 1 0 1 0 0 1 0 0 1
```

First Build 1-Bit ADDer



inputs			outputs	
a	b	c	C_{out}	S
0	0	0	0	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	1	1

Circuit Design Algorithm:

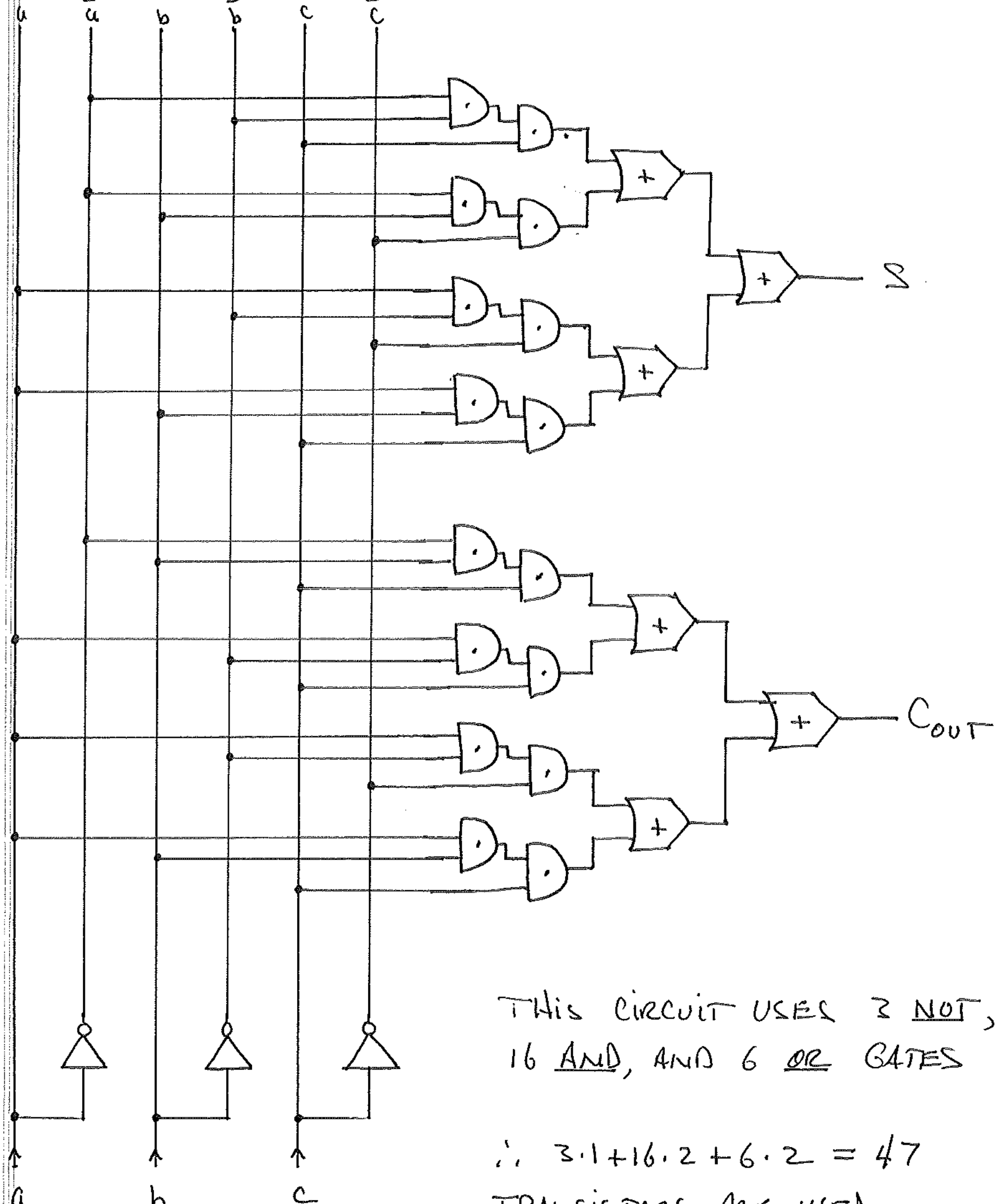
for each output column
- for each row containing a 1 in that col
  - form "product" corres. to inputs
- form "sum" of these "products

Design Circuit correspond. to logical exp.

$$C_{out} = \bar{a}\cdot b\cdot c + a\cdot \bar{b}\cdot c + a\cdot b\cdot \bar{c} + a\cdot b\cdot c$$

$$S = \bar{a}\cdot \bar{b}\cdot c + \bar{a}\cdot b\cdot \bar{c} + a\cdot \bar{b}\cdot \bar{c} + a\cdot b\cdot c$$

WE NEED A SYSTEMATIC WAY TO DRAW CIRCUITS:

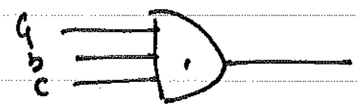


THIS CIRCUIT USES 3 NOT, 16 AND, AND 6 OR GATES

$\therefore\ 3\cdot 1 + 16\cdot 2 + 6\cdot 2 = 47$ TRANSISTORS ARE USED.

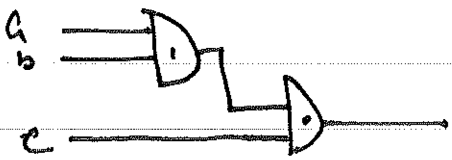
circuits $\longleftrightarrow$ Parenthesized logical expressions

$a \cdot b \cdot c$



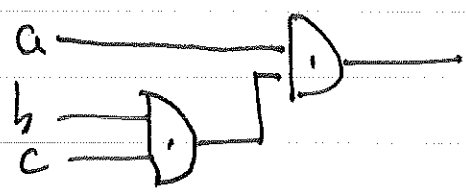
Dont use This

$(a \cdot b) \cdot c$



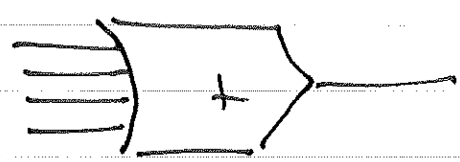
use this

$a \cdot (b \cdot c)$



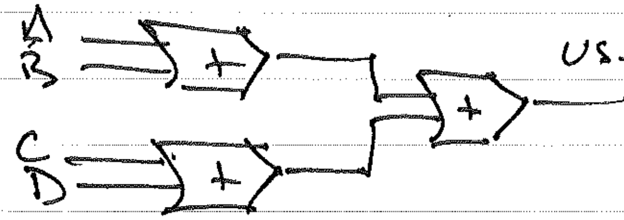
or this

$A + B + C + D$



Dont use this

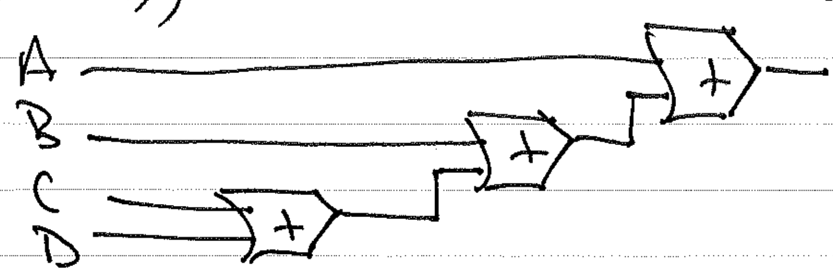
$(A + B) + (C + D)$



use this

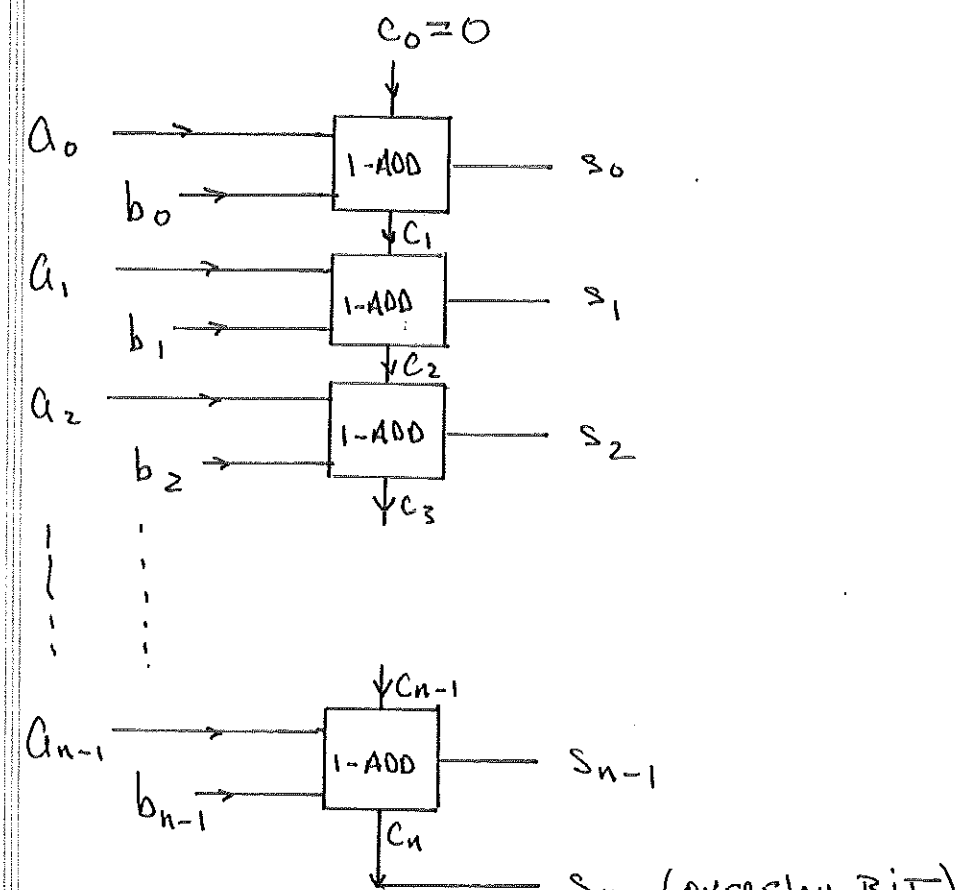
$A + (B + (C + D))$

or this



TO CONSTRUCT THE FULL n-BIT ADDER WE COMBINE n COPIES OF THE ABOVE CIRCUIT.

$$\underbrace{[a_{n-1} \cdots a_0]_2 + [b_{n-1} \cdots b_0]_2}_{\text{INPUTS}} = \underbrace{[s_n s_{n-1} \cdots s_0]_2}_{\text{OUTPUT}}$$

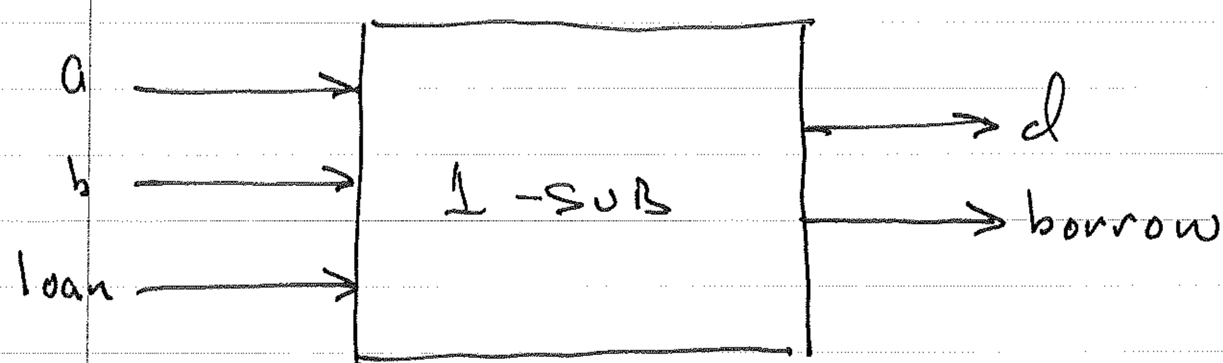


THE n-BIT FULL ADDER USES $47n$ TRANSISTORS. FOR EXAMPLE, A 32-BIT ADDER USES $47 \cdot 32 = 1504$ TRANSISTORS

Subtraction:

Ex

$$
\begin{array}{cccc}
0 & {}^{1}0 & & \\
\cancel{1} & \cancel{1} & {}^{1}0 & 1 \\
0 & 1 & 1 & 1 \\
\hline
0 & 1 & 1 & 0
\end{array}
$$




```
if   a-b-loan < 0
        borrow = 1
else
        borrow = 0
d = 2·borrow + a-b-loan
```

inputs			output	
a	b	loan	d	borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$d = \bar{a}\bar{b}l + \bar{a}b\bar{l} + a\bar{b}\bar{l} + abl$$

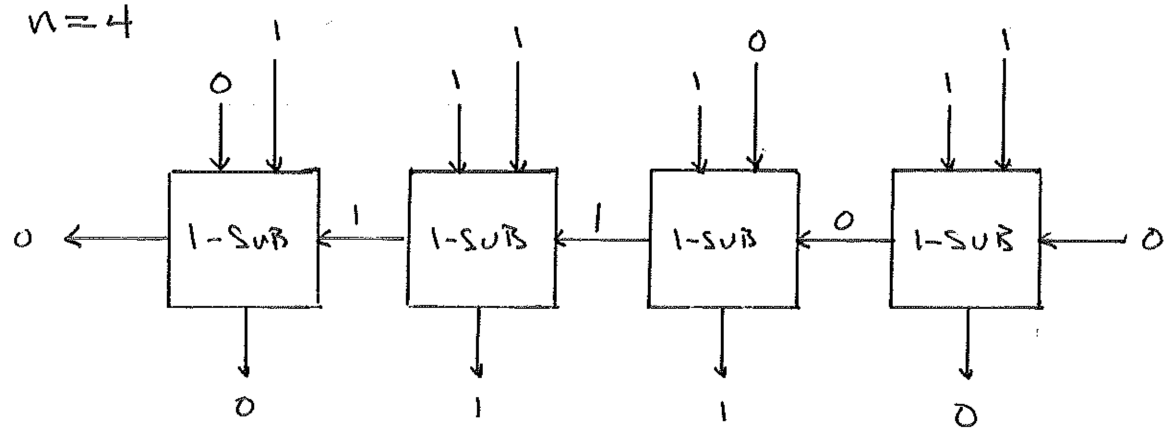
borrow = exercise !!

<u>Exercise</u>: Draw The Circuit for 1-SUB.

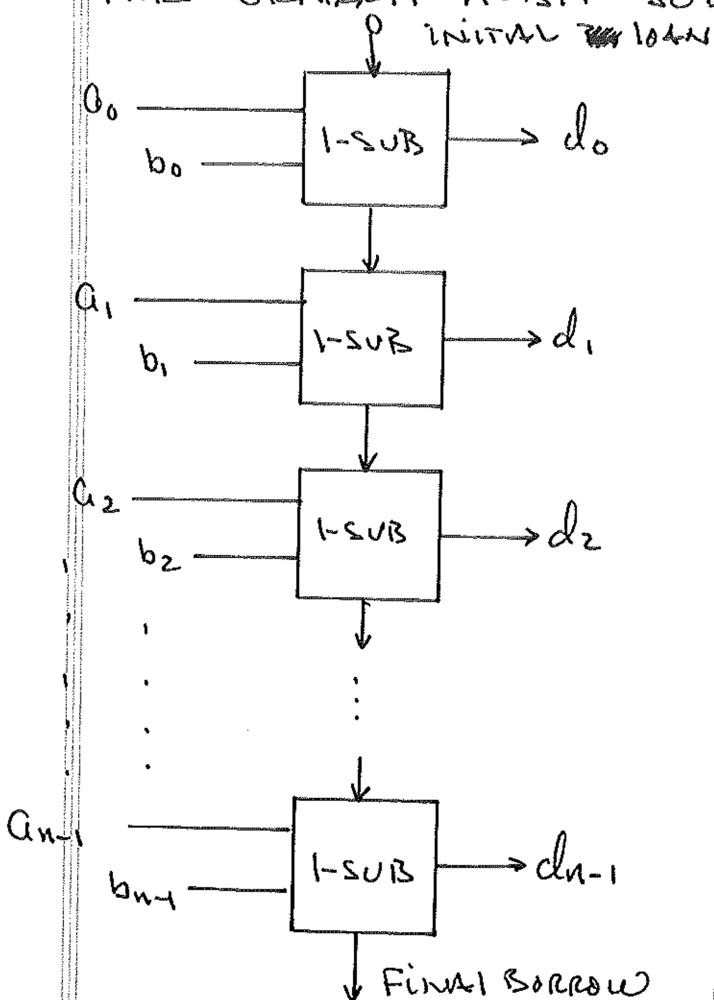
Once 1-SUB is built, we can build an n-bit subtractor for any n.

USING 1-SUB WE CAN BUILD AN n-BIT SUBTRACTOR

EX $n=4$



THE GENERAL n-BIT SUBTRACTOR looks like



$$[d_0 \cdots d_{n-1}] = [a_0 \cdots a_{n-1}] - [b_0 \cdots b_{n-1}]$$

$$\text{COMPARISON BIT} = \begin{cases} 1 & \text{IF } [a_0 \cdots a_{n-1}] < [b_0 \cdots b_{n-1}] \\ 0 & \text{OTHERWISE} \end{cases}$$