L1

CMPS 10 11-3-08

TRUTH TABLE FOR 1-ADD

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

Ex.

$$\begin{array}{rr} 1 & 1 \\ & 0 \\ & 1 \\ \hline & 0 \end{array}$$

Ex.

$$\begin{array}{rr} 0 & 0 \\ & 0 \\ & 0 \\ \hline & 0 \end{array}$$

Ex.

$$\begin{array}{rr} 1 & 1 \\ & 1 \\ & 1 \\ \hline & 1 \end{array}$$

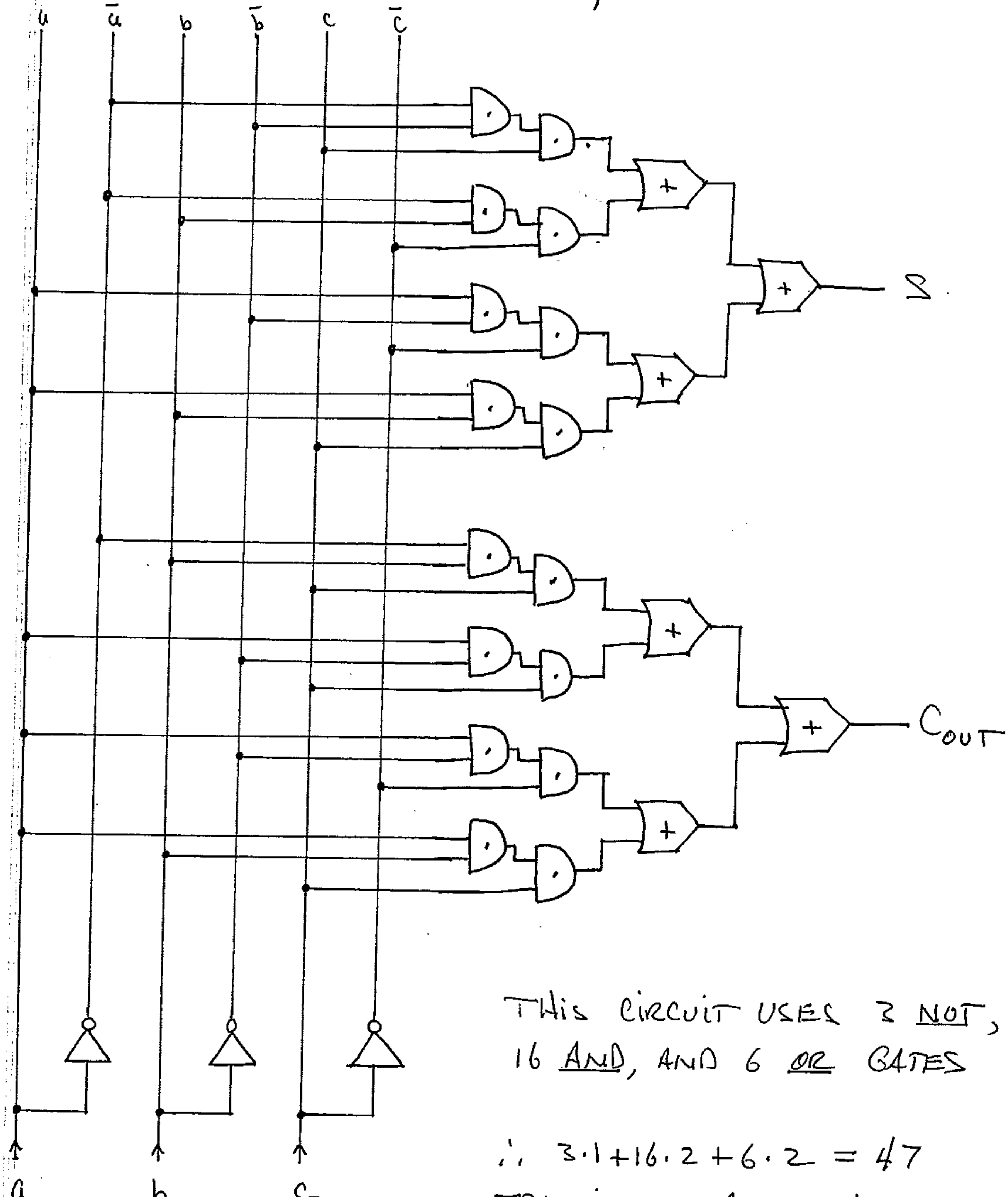
CIRCUIT DESIGN ALGORITHM:

1.) for each OUTPUT column
2.) for each row CONTAINING A 1 in THAT COLUMN
3.) form "PRODUCT" CORRESPONDING TO THOSE INPUTS
4.) form "SUM" OF THESE "PRODUCTS"

$$C_{OUT} = \left[(\bar{a}\cdot b\cdot c) + (a\cdot \bar{b}\cdot c)\right] + \left[(a\cdot b\cdot \bar{c}) + (a\cdot b\cdot c)\right]$$

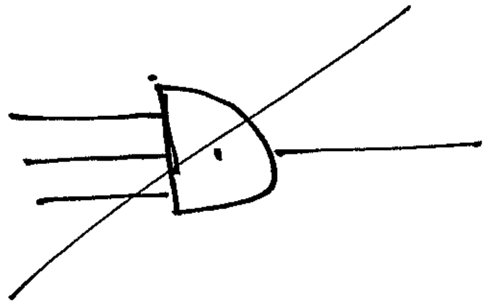
$$S = \left[(\bar{a}\cdot \bar{b}\cdot c) + (\bar{a}\cdot b\cdot \bar{c})\right] + \left[(a\cdot \bar{b}\cdot \bar{c}) + (a\cdot b\cdot c)\right]$$

WE NEED A SYSTEMATIC WAY TO DRAW CIRCUITS:

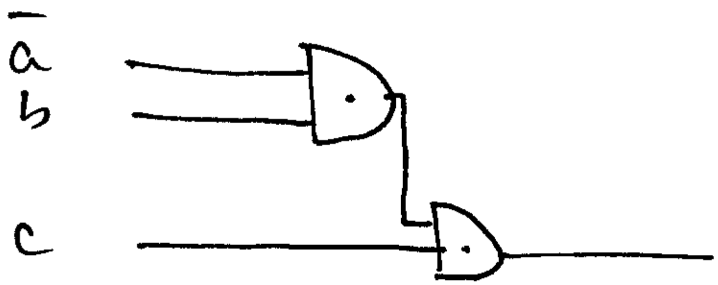


THIS CIRCUIT USES 3 NOT, 16 AND, AND 6 OR GATES

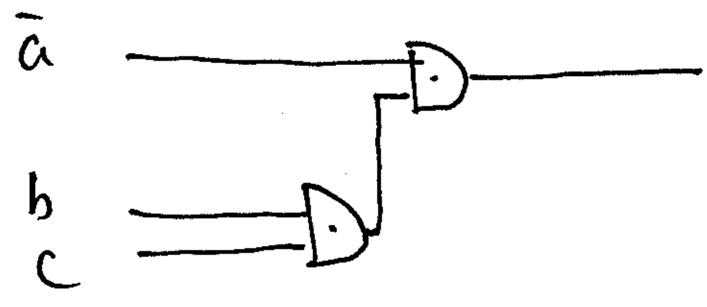
$\therefore\ 3\cdot 1 + 16\cdot 2 + 6\cdot 2 = 47$ TRANSISTORS ARE USED.



NO 3-INPUT
AND GATES

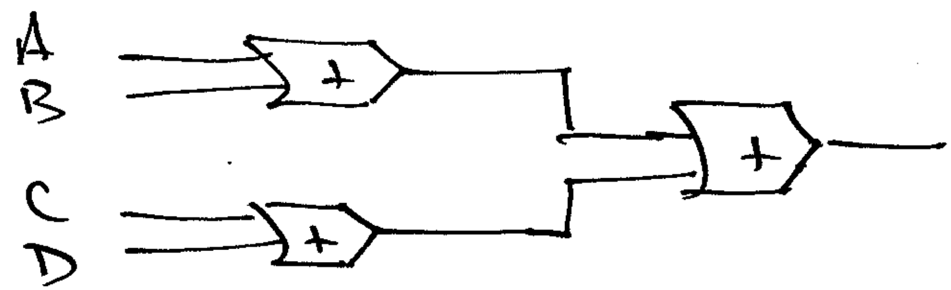


$(\bar{a} \cdot b) \cdot c$

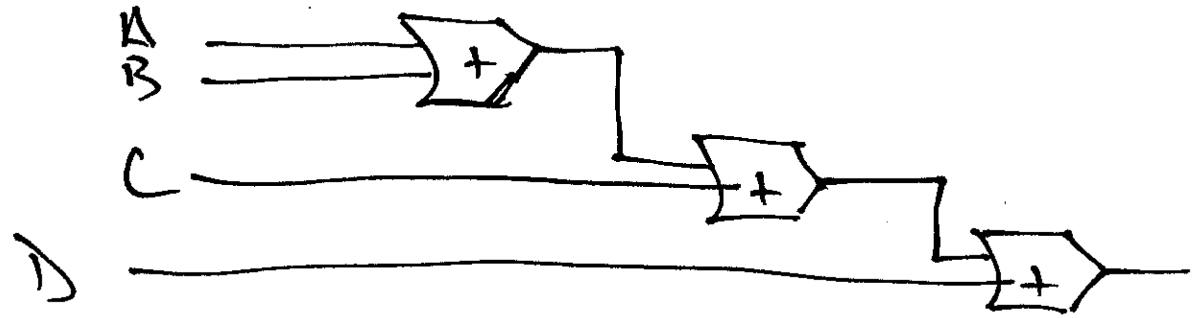


$\bar{a} \cdot (b \cdot c)$

L4



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



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

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

ASSOCIATIVE LAWS

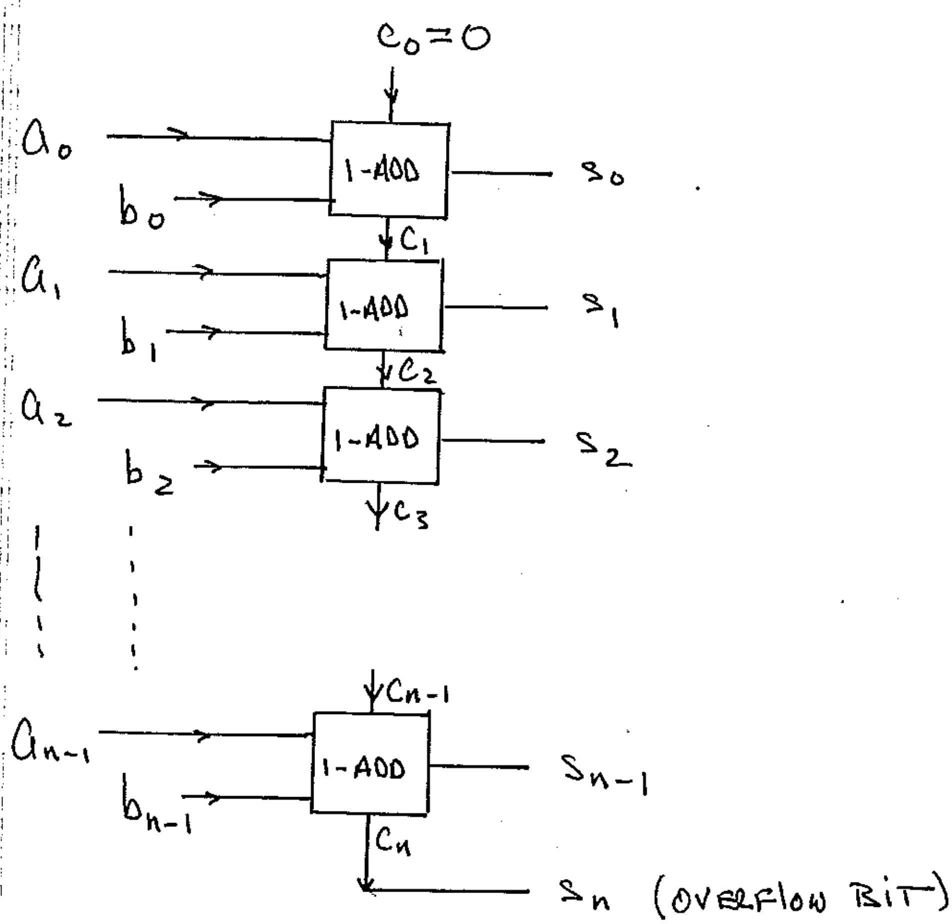
$$A \cdot (B \cdot C) \equiv (A \cdot B) \cdot C$$

$$A + (B + C) \equiv (A + B) + C$$

A	B	C	$B \cdot C$	$A \cdot (B \cdot C)$	$A \cdot B$	$(A \cdot B) \cdot C$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	1	0
1	1	1	1	1	1	1

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

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



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

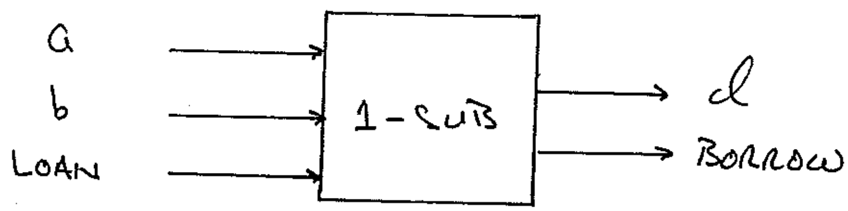
WHAT ABOUT SUBTRACTION ?

EX

$$\begin{array}{r} \overset{0}{\cancel{1}}\,\overset{10}{\cancel{1}}\,\overset{1}{0}\,1 = 13 \\ \underline{0\,1\,1\,1} = 7 \\ 0\,1\,1\,0 = 6 \end{array}$$

EXCERCISE

DESIGN A 1-BIT SUBTRACTION CIRCUIT

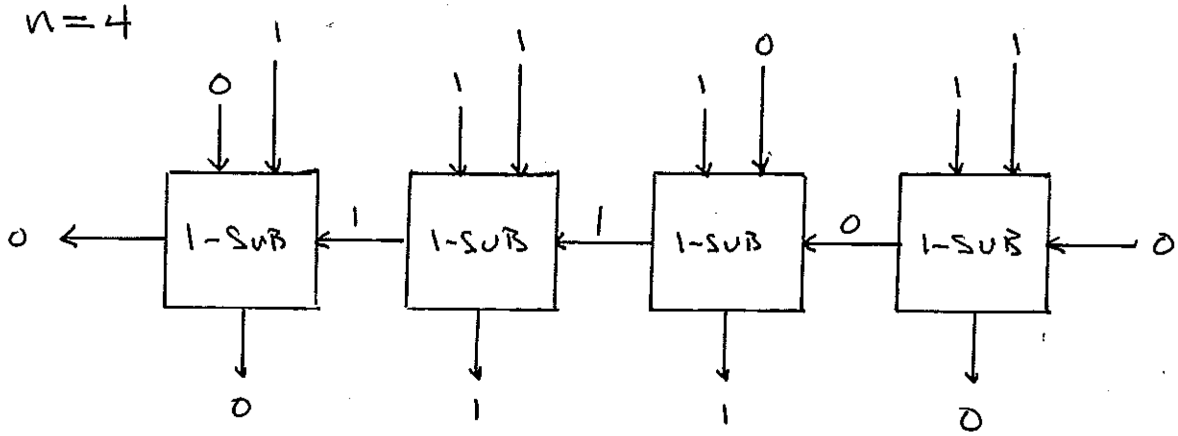


IF $a-b-\text{LOAN} < 0$ THEN BORROW = 1, OTHERWISE BORROW = 0. THEN $d = 2 \cdot \text{BORROW} + a - b - \text{LOAN}$

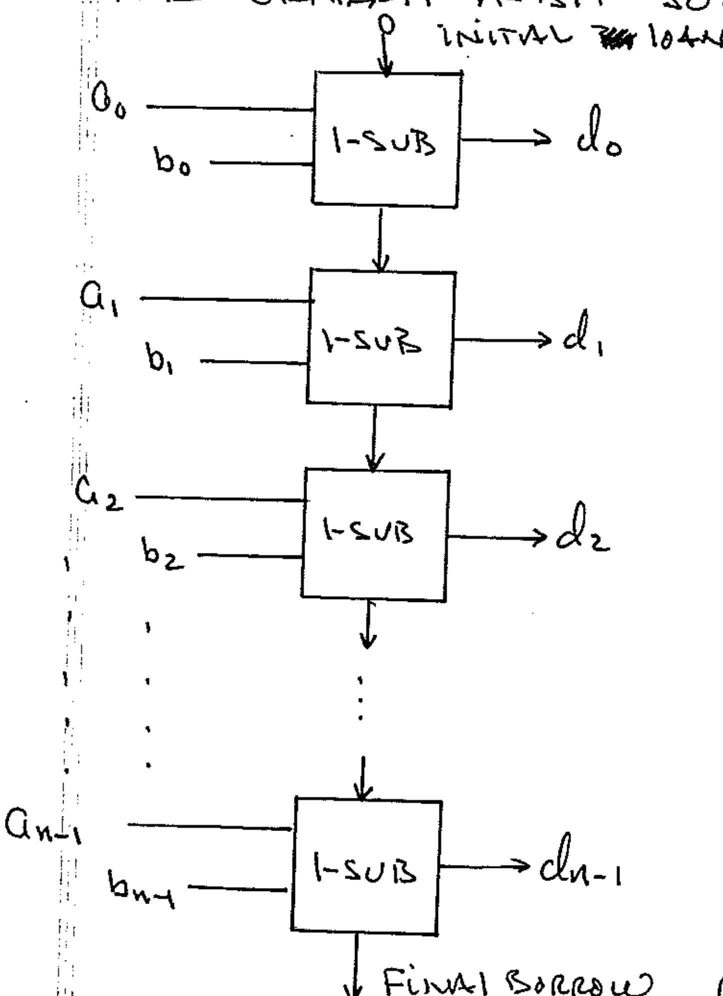
a	b	LOAN	d	BORROW
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

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

EX $n=4$



THE GENERAL n-BIT SUBTRACTOR looks like



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

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