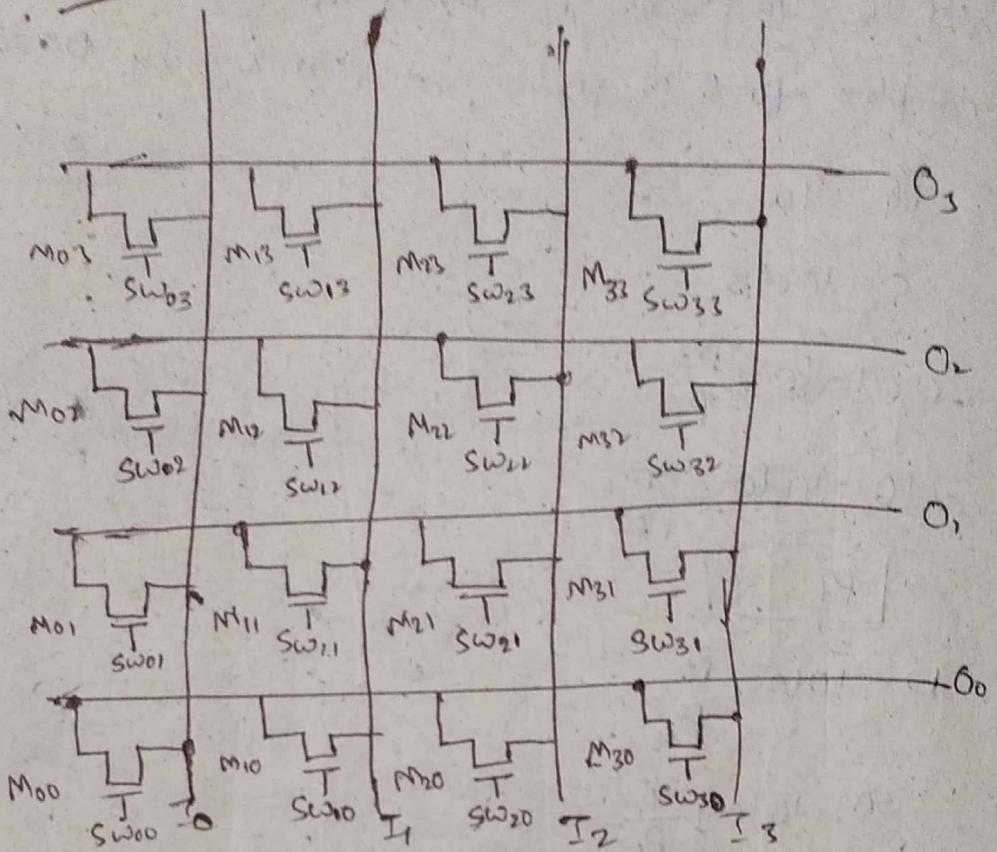
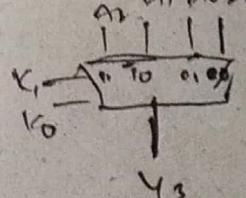


Barrel Shifter



Drawback :- requires 16 T & each is having a independent gate connection. (i.e. we have 16 different control input which makes circuit complex)

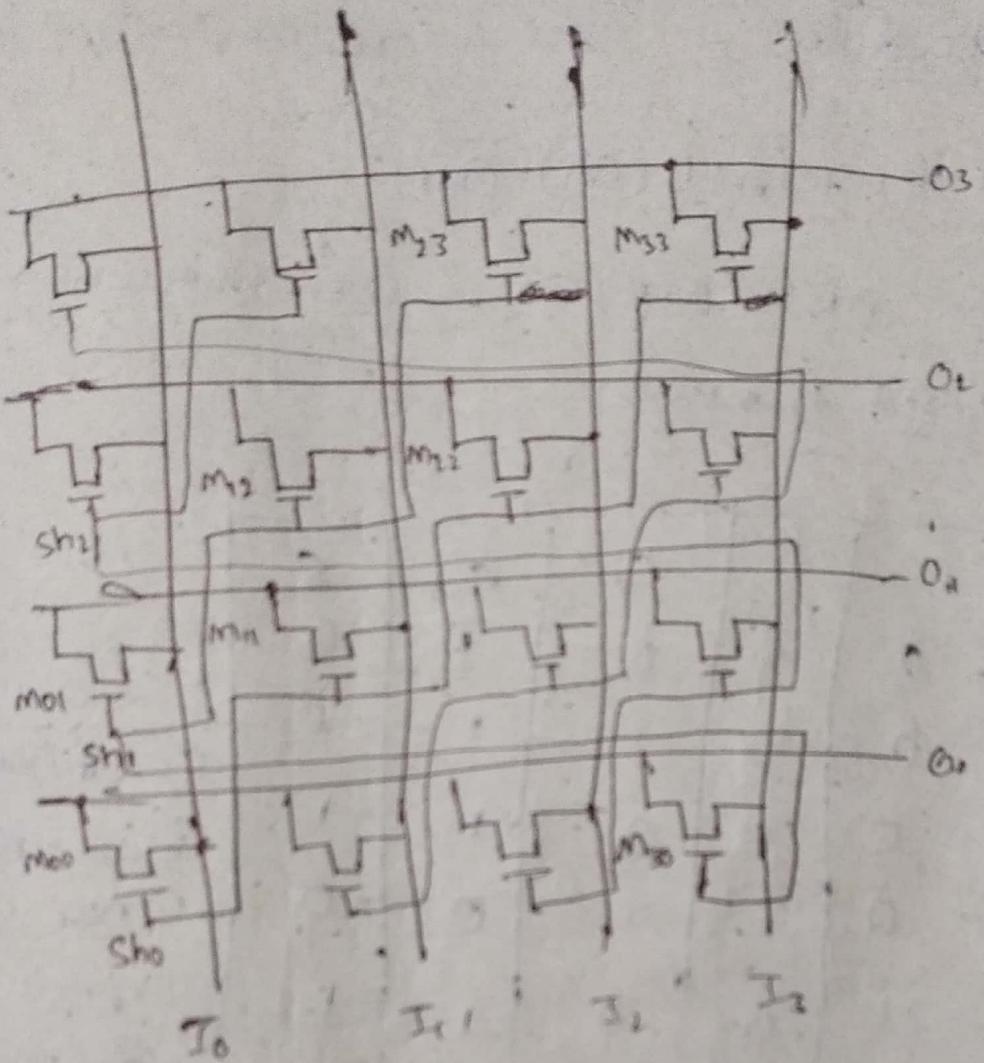
	A3	A2	A1	A0
00	A3	A2	A1	A0
01	A0	A3	A2	A1
10	A1	A0	A3	A2
11	A2	A1	A0	A3



3 0 1 2
2 3 0 1
1 2 3 0

O3 O2 O1 O0
A3 A2 A1 A0
A3 A2 A1 A0
A1 A0 A3 A2
A1 A0 A3 A2

inlog, min 911



- * 4 control inputs $Sh_0, Sh_1, Sh_2 \& Sh_3$
which must be mutually exclusive in the active state
- * NMOS pass transistors are used will degrade the output logic I
- * Cros may be used
- * Total no of transistors is N^2 e.g. 8 bit BUT area dominated by interconnect wiring

Single Bit adder
Full adder : using revised truth table

$$S = ABC + (A+B+Cin) \cdot Cout$$

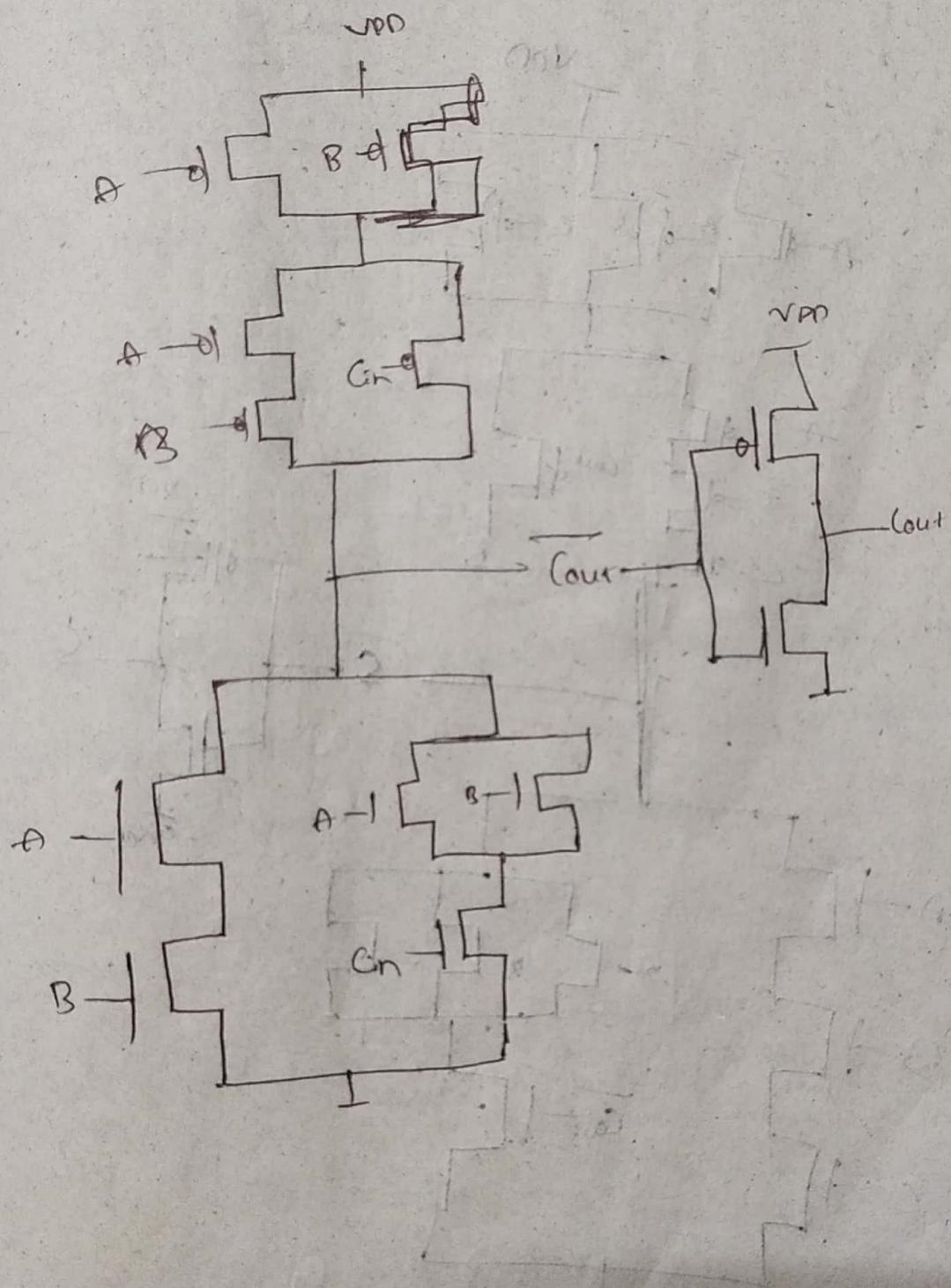
$$Cout = AB + (A+B)Cin \rightarrow \text{from truth table}$$

revised truth table

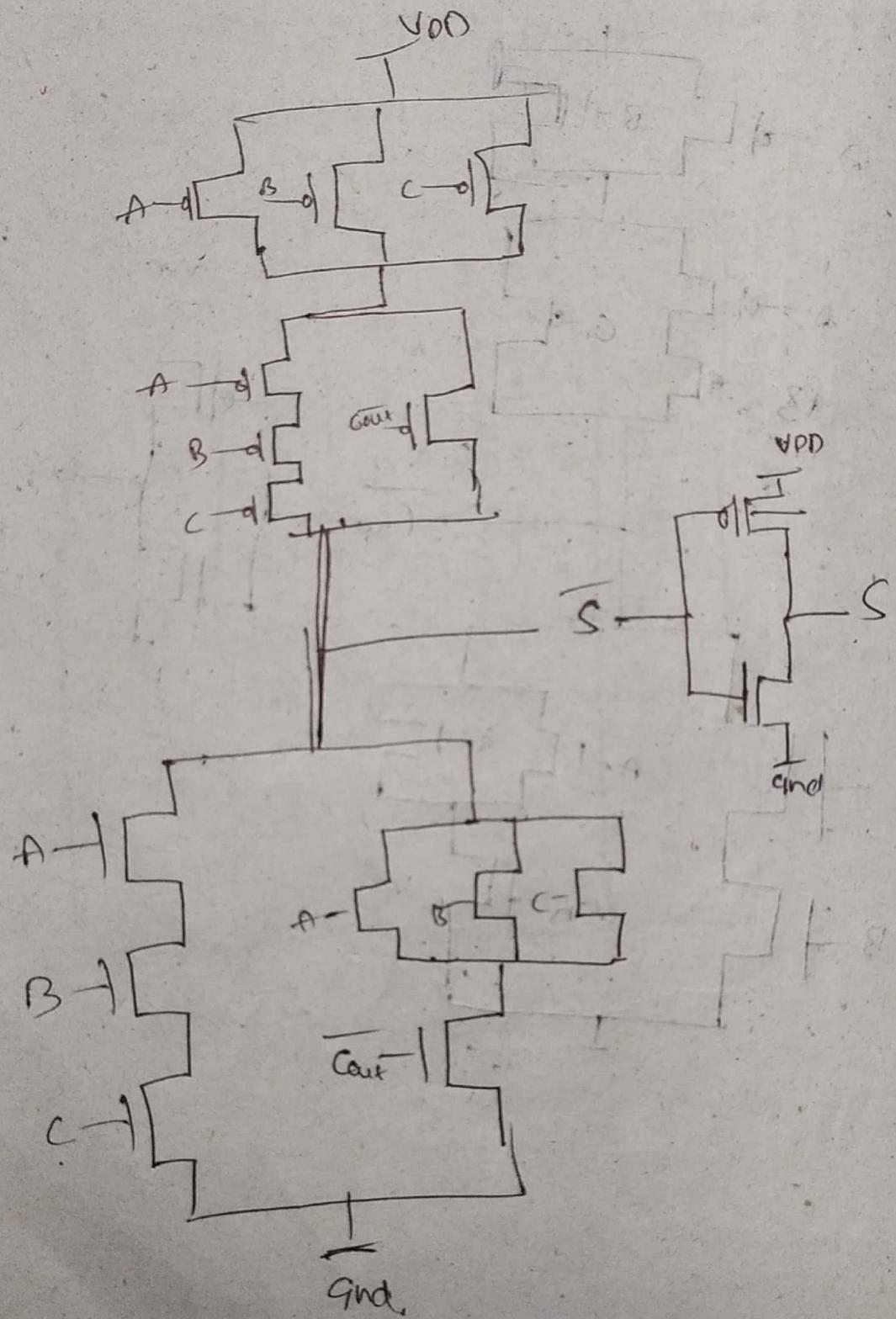
A	B	Cin	Cout	S
0	0	0	1	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	1

$S = \overline{Cin} \overline{Cout} +$
 $\overline{B} \overline{Cout} + A \overline{Cout}$
 $.ABC$
 $\rightarrow ABC +$
 $\overline{Cout}(A+B+C)$

$$C_{out} = \underline{AB} + (\bar{A} + \bar{B}) C_{in}$$



$$\underline{\text{Sum}} = \overline{ABC} + \overline{(A+B+C)} \text{ (out)$$



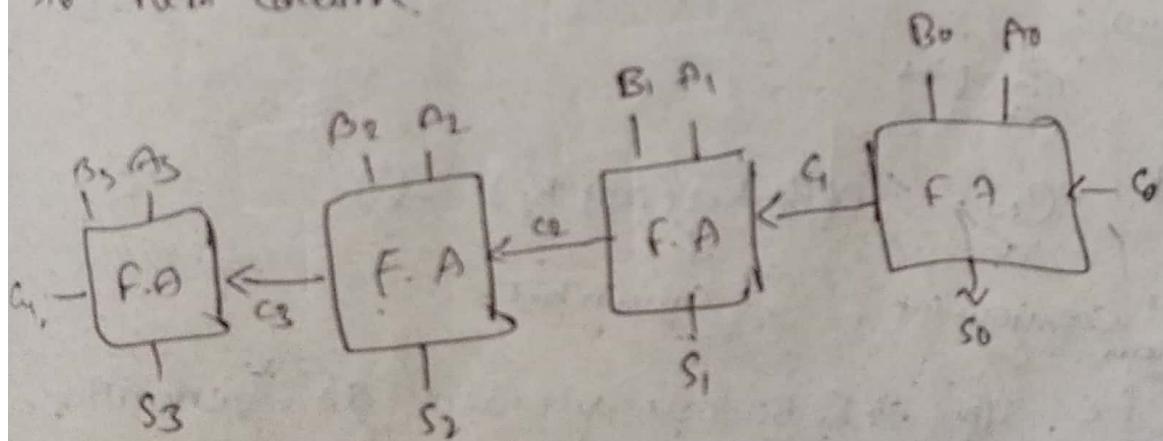
Binary word adder

$$\begin{array}{r}
 A_3 \ A_2 \ A_1 \ A_0 \\
 + B_3 \ B_2 \ B_1 \ B_0 \\
 \hline
 S_3 \ S_2 \ S_1 \ S_0
 \end{array}$$

Ripple Carry adder (RCA)

$$\begin{array}{r}
 C_3 \quad G \quad C_1 \quad G \\
 A_3 \quad A_2 \quad A_1 \quad A_0 \\
 + B_3 \quad B_2 \quad B_1 \quad B_0 \\
 \hline
 S_3 \quad S_2 \quad S_1 \quad S_0
 \end{array}$$

* An n-bit RCA requires n Full Adder with carryout bit C_{i+1} used as the carry-in bit to the next column.



Carry Look-Ahead Adder

- * CLA designed to overcome delay issue in RA adder.
 - eliminates the ripple (carrying) effect of the carry bit.
 - * Algorithm based calculating all carry terms at once.
 - rewrite $c_{i+1} = a_i b_i + g (a_i \oplus b_i)$
- $$c_{i+1} = g_i + p_i$$
- where $g_i = a_i b_i$
 $p_i = a_i \oplus b_i$
- $$\therefore S = p_i \oplus c_i$$

P_{loss}

- * no cascade, decreases time delay

Cons:

Complex

* $c_{i+1} = A_i B_i + g (A_i \oplus B_i) \quad \text{--- (1)}$



Carryout bit



Carryin bit

Stream:

i.e. if A_i & B_i are viewed as "generating"

Carry out Bit (c_{i+1}) So. $c_{i+1} = A_i B_i \Rightarrow$ "generate term".

2nd terms:

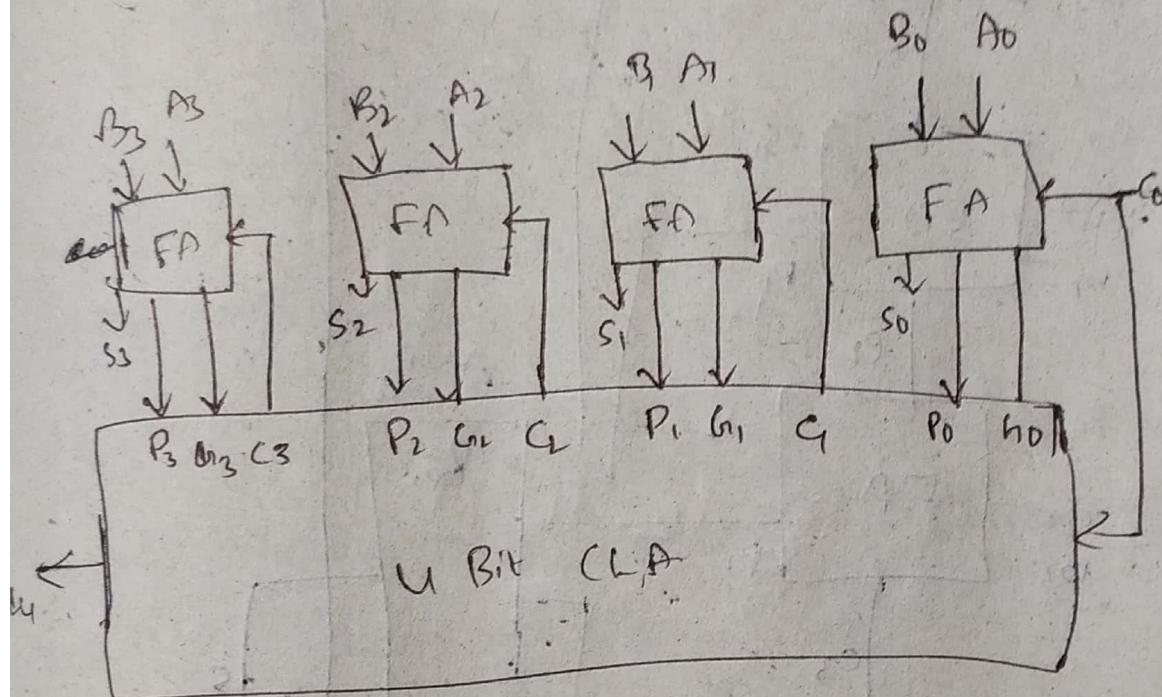
$p_i = A_i \oplus B_i$ "propagate term".

modified

$$C_i + 1 = C_{i-1} + P_i \cdot G_i$$

$$S = P_i \oplus C_i$$

$$(g) \quad C_i = C_{i-1} + P_i \cdot G_i \quad C_i = C_{i-1} + P_i \cdot C_{i-1}$$
$$= C_{i-1} + P_i (C_{i-1} + P_{i-1} \cdot G_{i-1})$$



High Speed address

~~Carry~~

$$\text{Generate } G_i = D_i \wedge B_i \rightarrow AB$$

$$P_i = A_i \oplus B_i$$

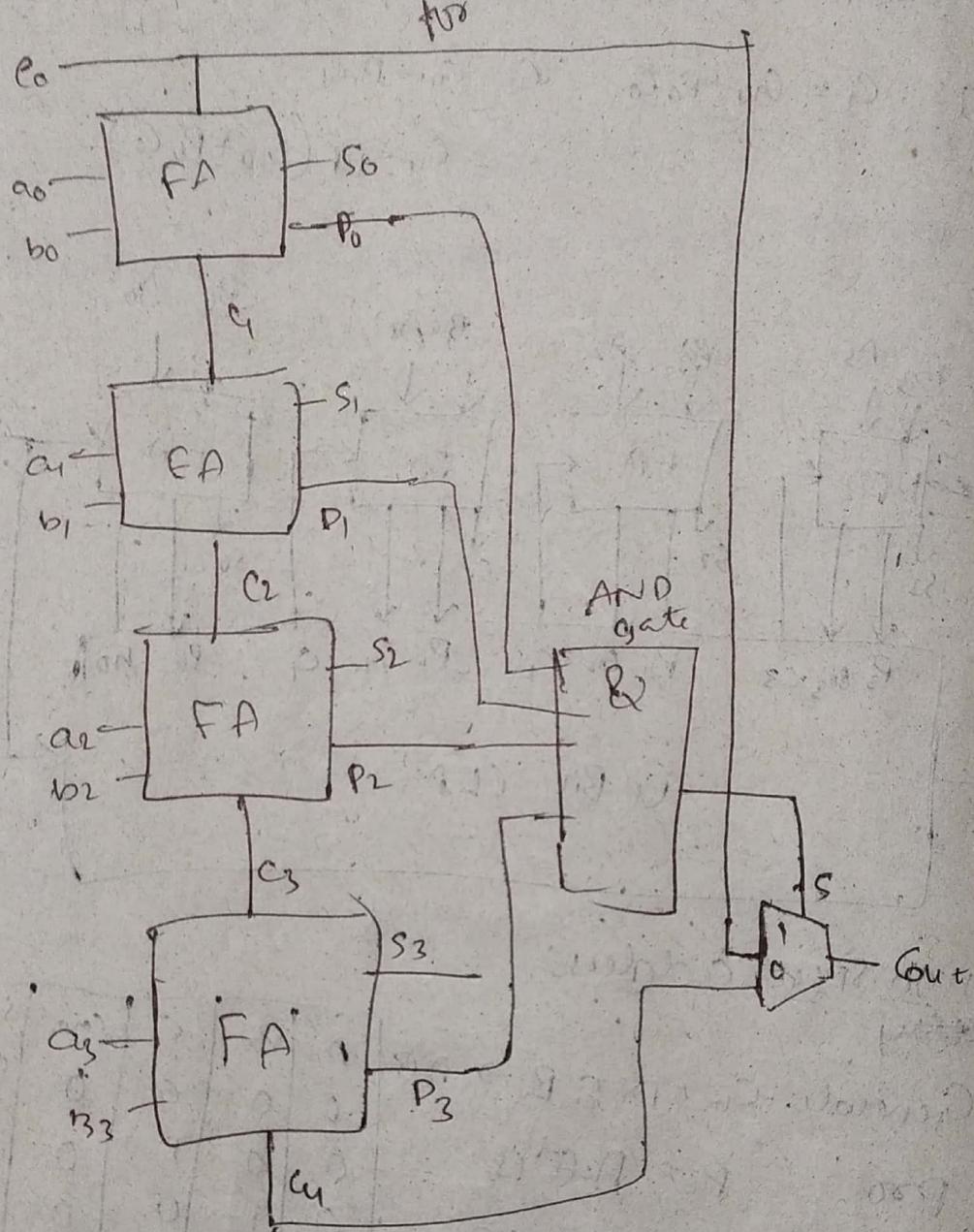
$$K_i = !P_i !B_i$$

$$C = AB + C_{in} (A \oplus B) ; C_i + 1 = G_i + P_i \cdot C_i$$

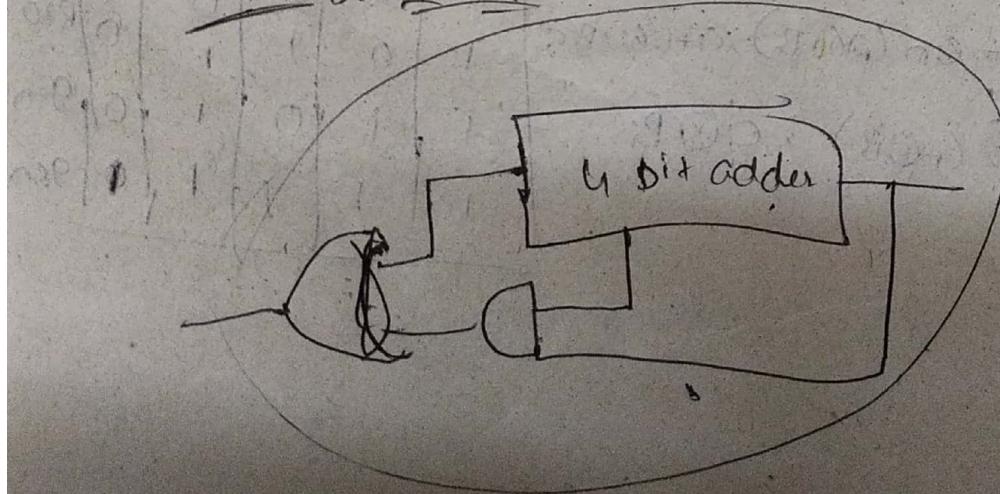
$$S = C_{in} \oplus (A \oplus B) ; C_i \oplus P_i$$

	A	B	C	Cout	S	Stat
0	0	0	0	0	0	del
0	0	0	1	0	1	del
0	1	0	0	0	1	pro
0	1	0	1	1	0	pro
1	0	0	0	0	0	pro
1	0	0	1	1	1	pro
1	1	1	0	1	0	gen
1	1	1	1	1	1	gen

Carry Skip / Carry Bypass adder



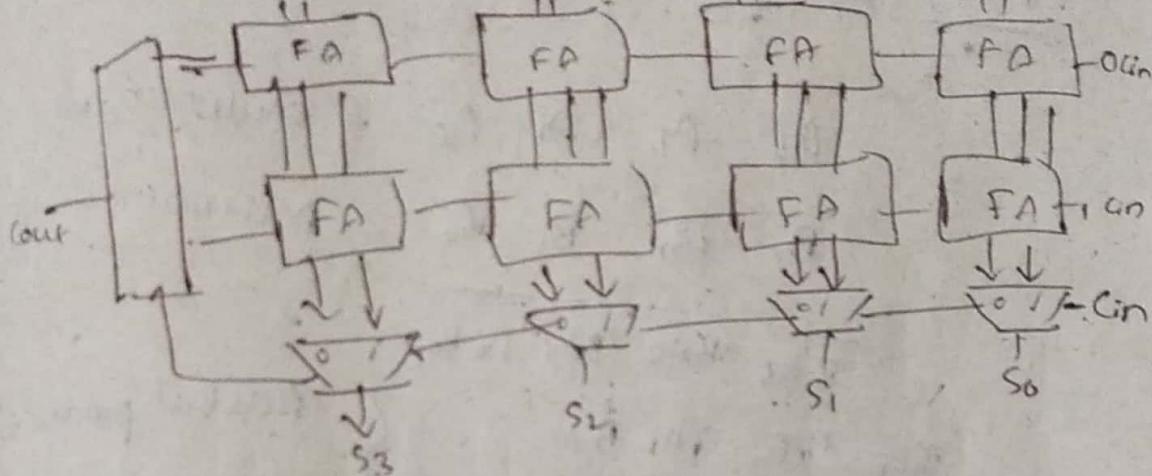
Carry skip logic



Carry Select adder

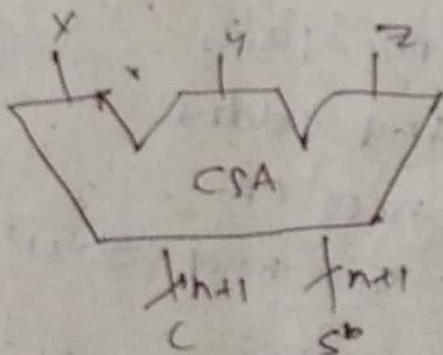
If Carry-in value is delayed. It performs correctly.

Later it takes the correct value using mux



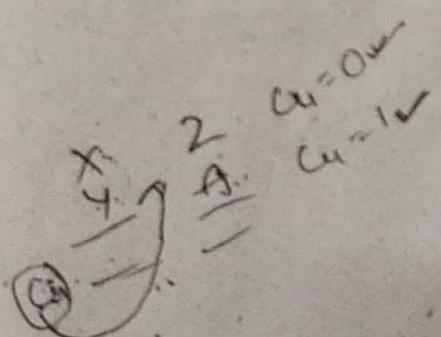
Carry Save adder-

Carry Save adder adds 3 nos such that
carries generated are not propagated rather
they are saved in carry vector



$$\begin{array}{r}
 x = 001011 \\
 y = 010101 \\
 z = 111101 \\
 \hline
 s = 0100011 \\
 c = 0111010
 \end{array}$$

$s + c = \underline{\underline{11011101}}$ → $x + y + z$



Multiplicand =

Array multiplier

$A_3 \ A_2 \ A_1 \ A_0$ — multiplicand

$B_3 \ B_2 \ B_1 \ B_0$ — multiplier

$B_0A_3 \ B_0A_2 \ B_0A_1 \ B_0A_0$
 $B_1A_3 \ B_1A_2 \ B_1A_1 \ B_1A_0$

$B_2A_3 \ B_2A_2 \ B_2A_1 \ B_2A_0$

$B_3A_3 \ B_3A_2 \ B_3A_1 \ B_3A_0$

} Partial products

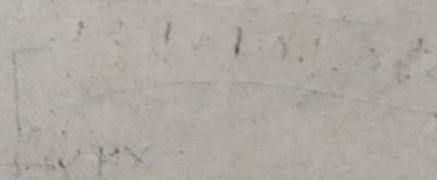
$P_7 \ P_6 \ P_5 \ P_4 \ P_3 \ P_2 \ P_1 \ P_0$

$m \times n$ AND gates

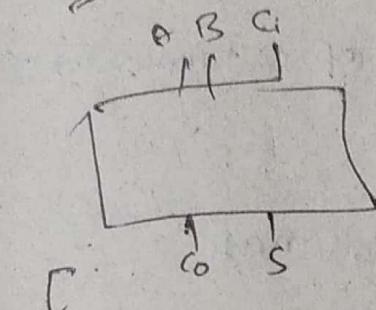
first address

No. of FA

Total no. of adders:



wallace tree multiplier



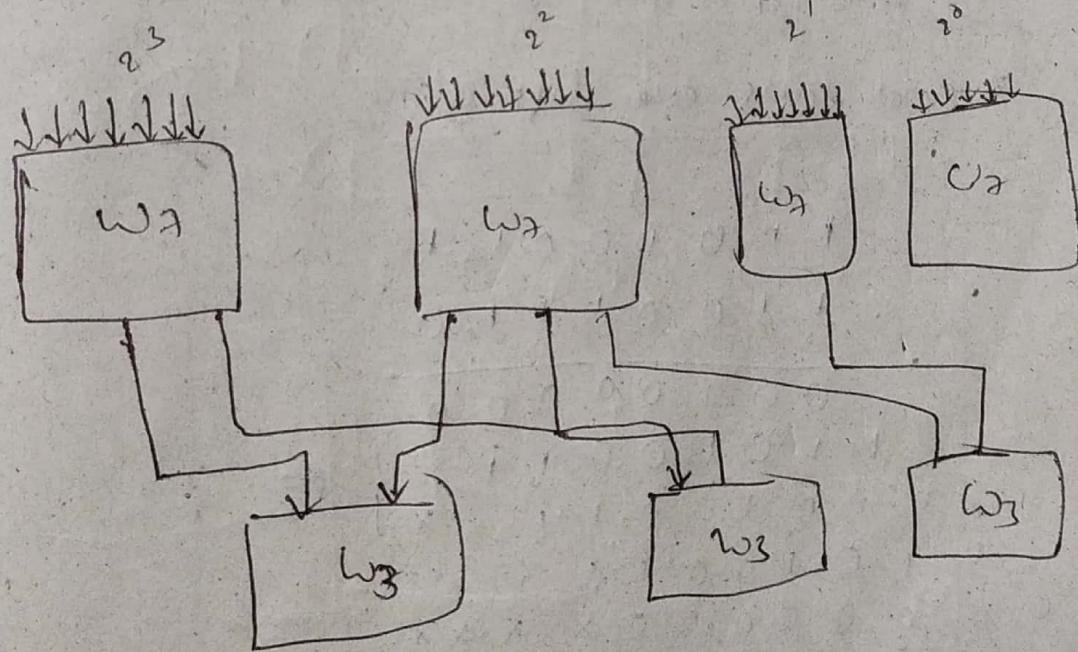
Carry save adder

log N O/P to N' inputs

Carry save adder

Carry propagate adder

A	B	C	$i's$	C	S
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	2	1	0
1	0	0	1	0	1
1	0	1	2	1	0
1	1	0	2	1	0
1	1	1	3	1	1



photo

- * works exactly like long-hand multiplication
- * ~~not~~ products are the result of a simple AND operation
- * Requires N^2 AND gates
- * Trick is in adding up the columns
- * To add up columns, add up three rows at a time
- * The result for each set of three rows is a set of two rows
- * Each resulting set of two rows has a row for the carryout
- * Odd rows are left alone

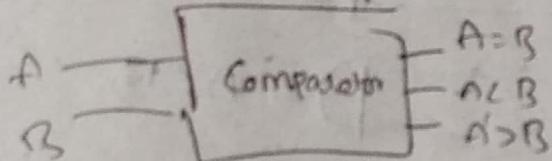
$$\begin{array}{r}
 1101001.1 \\
 11001+1.0 \\
 \hline
 0000000.0 \\
 1101000.1\,\,1\,\,x\,\,} \\
 1101001.1\,\,x\,\,x \\
 11010011\,\,x\,\,x\,\,x \\
 000000000\,\,x\,\,x\,\,x \\
 000000000\,\,x\,\,x\,\,x\,\,x\,\,x \\
 1010.011x\,\,x\,\,x\,\,x\,\,x\,\,x \\
 100
 \end{array}$$

$$\begin{array}{r}
 1011101.010 \\
 1000001000 \\
 \hline
 \text{Uu will} \quad \hline
 \end{array}$$

In final stage with two rows

Using ripple carry adder it adds.

Comparator:



$$A < B: A_1'B_1 + A_1'A_0B_0$$

$$A = B: (A_0 \oplus B_0) (\bar{A} \oplus \bar{B})$$

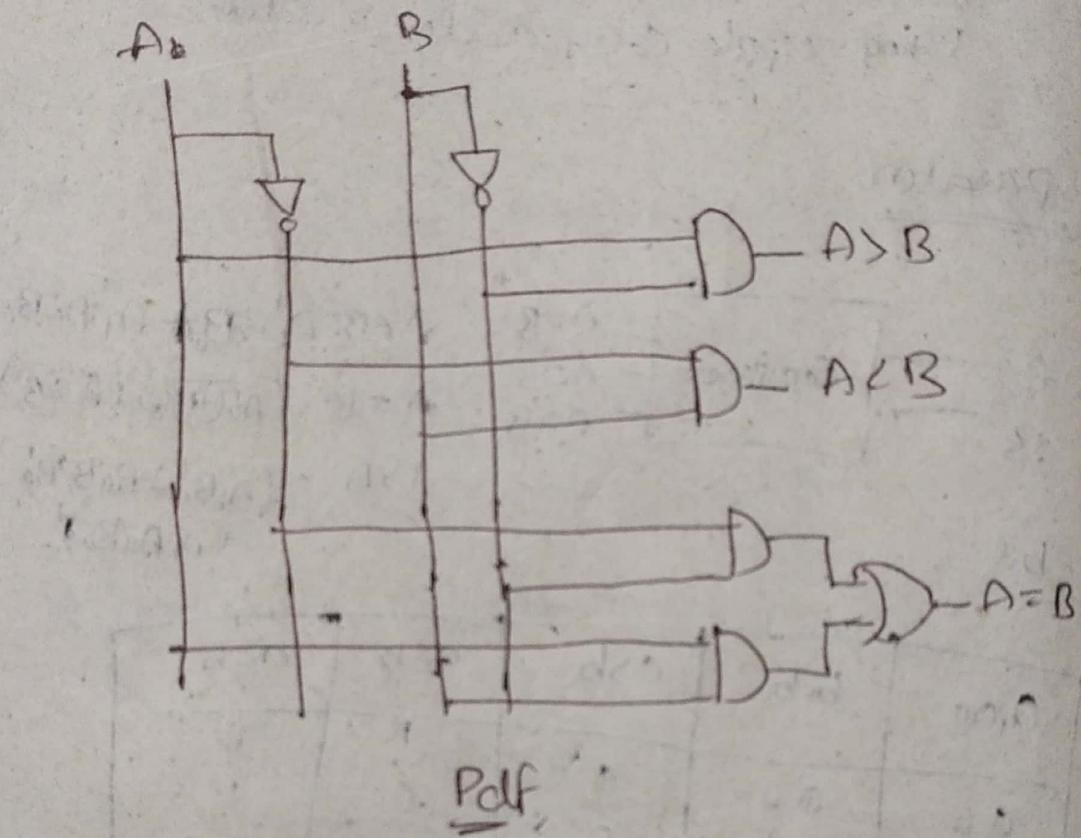
$$\begin{aligned}
 A > B = & (A_1B_1 + A_0B_1B_0' \\
 & + A_1A_0B_0)'
 \end{aligned}$$

2 bit

$A_{1,0}$	$B_{1,0}$	$A > B$	$A = B$	$A < B$
00	00	0	1	0
00	01	0	0	1
00	10	0	0	1
00	11	0	0	1
01	00	1	0	0
01	01	0	1	0
01	10	0	0	1
01	11	0	0	1
10	00	1	0	0
10	01	0	1	0
10	10	0	0	1
10	11	0	0	0
11	00	1	0	0
11	01	0	0	0
11	10	1	0	0
11	11	0	0	0

1 bit

A	B	$A \oplus B$	$A \otimes B$	$A > B$	
0	0	0	1	10	$A \otimes B = AB$
0	1	1	0	0	$A < B = A'B$
1	0	0	0	1	
1	1	0	1	0	$A = B : A'B' + AB$



All 1's detector using AND gate
All 0's detector using NOR gate.

ALU:

- * Used in CPU, microprocessors,
- * Arithmetic & logic operations

Arithmetic

Addition

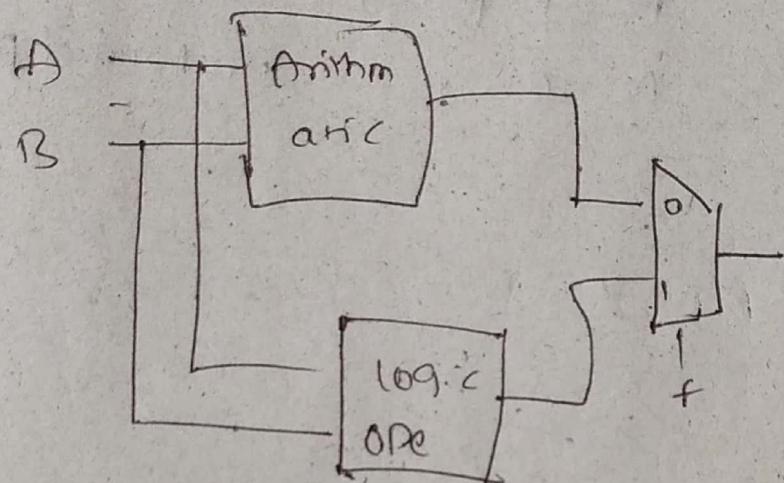
Subtraction

Multiplication

Division

Logic

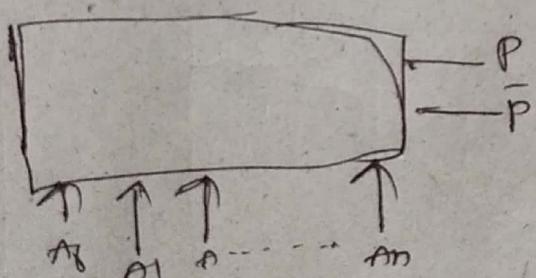
NOT, AND, OR, NAND,
NOR, XOR, XNOR.



Parity generator:

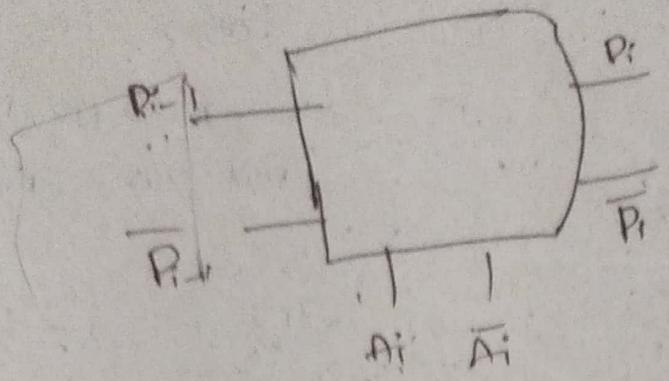
The circuit shows the parity of binary

of Parity no. of words.



$P=1$ even no. of 1's at inputs

$P=0$ odd



$A_i = 1$ parity is changed $P_i = \bar{P}_{i-1}$

$A_i = 0$, " unchanged $P_i = P_{i-1}$