CSN-221 - CAM - Computer Arithmetic

Name: Shashank Aital
Enr. No: 19114076
Batch 04
Branch: CSE

Q.1. $n=4$, $M=(-7) = (1001)_2$; $Q=(-5) = (1011)_2$

$-M=(0111)_2$ $\quad -M=(0101)$

Iterative procedure -

$7\times5 \Rightarrow$

```
      0 1 1 1
    × 0 1 0 1
    ---------
      0 1 1 1
  +  0 0 0 0
  + 0 1 1 1
  + 0 0 0 0
  -------------
  0 1 0 0 0 1 1   (= 35)
```

∴ Both signs are same, don't negate

$\Rightarrow (-7)\times(-5) = (35) = (0100011)_2$

Booth's Algorithm:

n	A	Q	q_{-1}	Comment
4	0000	1011	0	Initialize
4	0111	1011	0	10 ⇒ A - M
3	0011	1101	1	Right shift ; n → n-1
2	0001	1110	1	11 ⇒ Right shift ; n → n-1
2	1010	1110	1	01 ⇒ A + M
1	1101	0111	0	Right shift ; n → n-1
	0100	0111	0	10 ⇒ A - M
0	0010	0011	1	Right shift ; n → n-1
				Terminate

Ans = AQ = $(00100011)_2 = (35)_{10}$

Q.2. Given: Multiplicand, (M) = 0101 1010 1110 1110

Multiplier, (Q) = 0111 0111 1011 1101

Append a trailing '0' to multiplier (q_{-1})

Booth's code (comparisons) are,

01 → Add
11 → X
11 → X
10 → Sub ⇒ 4 Additions
01 → Add ⇒ 4 subtractions
11 → X
11 → X
11 → X
10 → Sub
01 → Add
11 → X
11 → X
11 → X
10 → Sub
01 → Add
10 → Sub

Q.3. Booth's algorithm adds/subtracts only when $q_0 q_{-1}$ are 01 or 10.

⇒ Worst case performance occurs when 01 or 10 occur frequently i.e 1 & 0 alternate.

⇒ $\boxed{\text{Option(A): 101010 ... 1010}}$ gives worst case performance

---

Q.4. (I) Ripple Carry Adder –

Consists of 3 Full Adders & 1 Half Adder.

In 1 Full adder ⇒ 7 AND Gates
2 OR Gates

In 1 Half adder ⇒ 2 AND Gates
1 OR Gate

$\Rightarrow$ Total ANDs = $7 \times 3 + 2 = \underline{\underline{23}}$

$\Rightarrow$ Total ORs = $2 \times 3 + 1 = \underline{\underline{7}}$

~~(II) Carry Look Ahead Adder-~~

→ ~~We use generator & propagator functions.~~

→ ~~For C_i,~~

~~no. of AND Gates = 6~~

~~no. of OR gates = 3~~

→ ~~For P_i,~~

~~no. of AND gates = 8~~

~~no. of OR gates = 4~~

→ ~~For G_i,~~

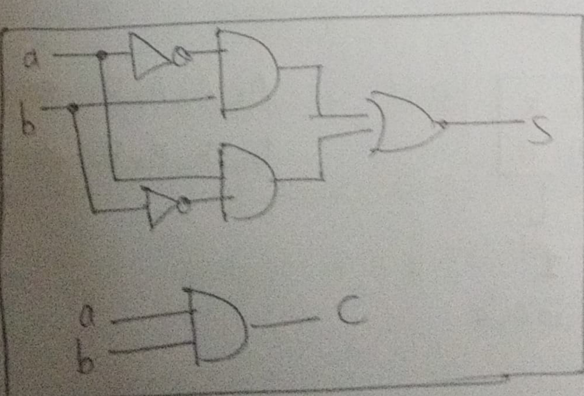
~~no. of AND gates = 4~~

$\Rightarrow$ ~~For S_i, no. of AND gates = $8+8+6+4 = \underline{\underline{26}}$.~~

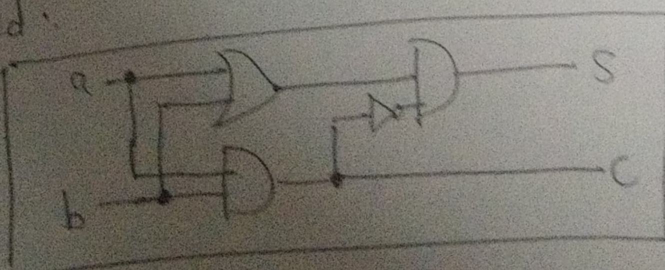
~~no. of OR gates = $4+4+3 = \underline{\underline{11}}$.~~

Circuits used in Q4:

Half Adder:



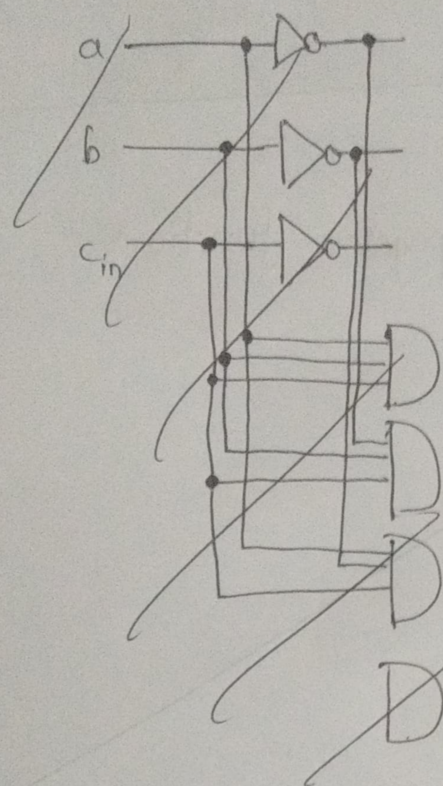
Optimized:

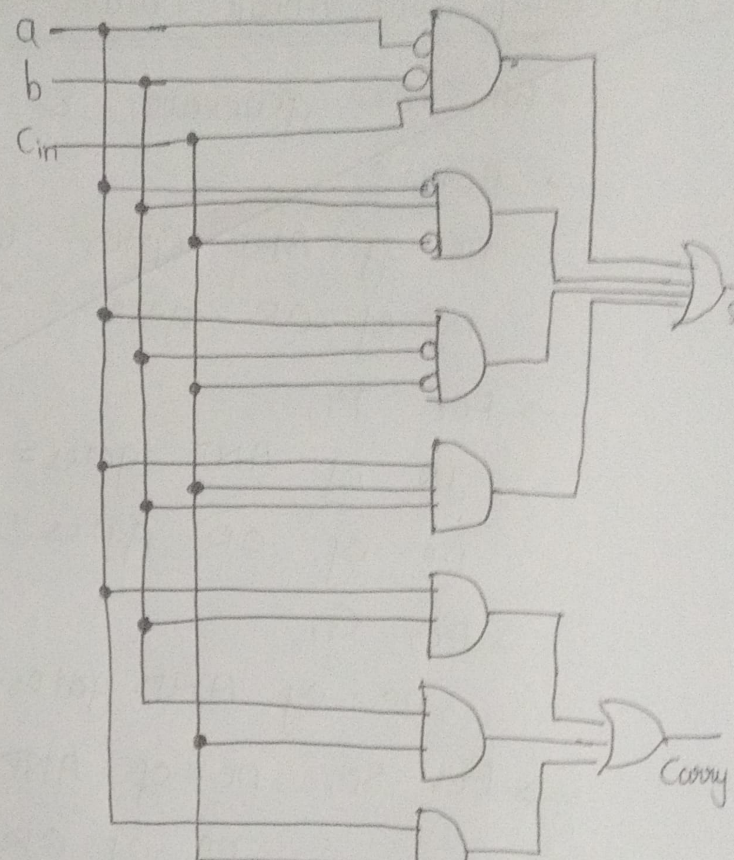


Full Adder:

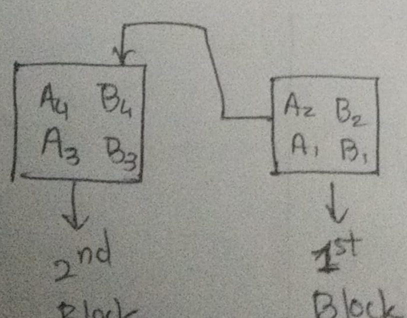
$$S = A \oplus B \oplus C$$

$$Carry = AB + BC + CA$$





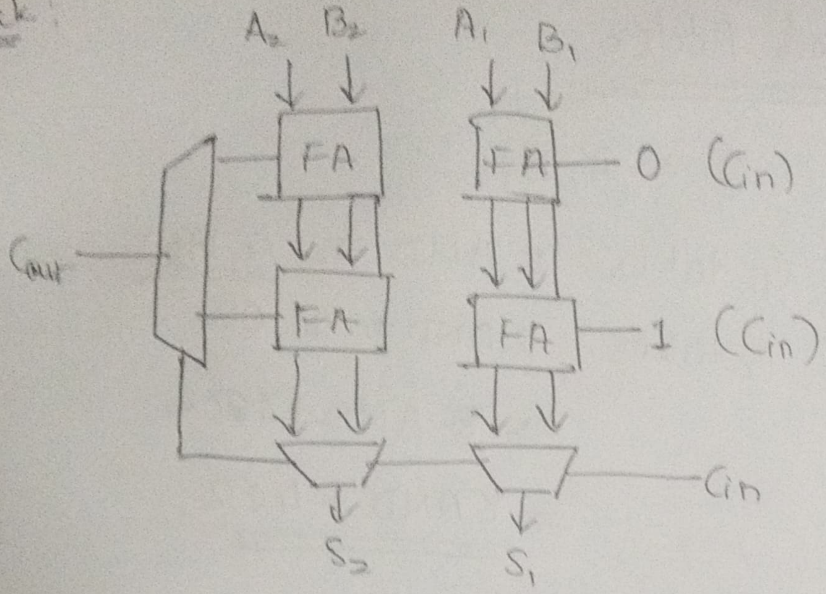
(II) <u>Carry</u> <u>Select</u> <u>Adder</u>-



$A_4 A_3 A_2 A_1$ ← 1st number

$B_4 B_3 B_2 B_1$ ← 2nd number

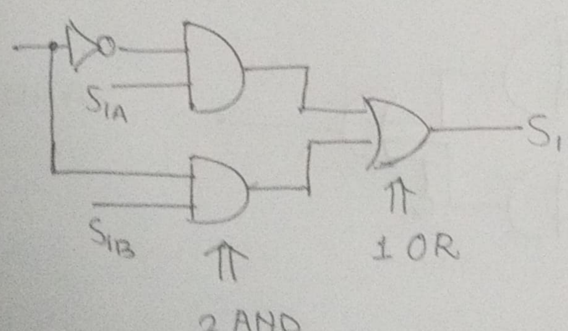
Block:



1 FA ⇒ 7 AND + 2 OR gates

⇒ 4 FA's ⇒ 28 AND + 8 OR gates

For MUX:



3 selector ⇒ (2 AND + 1 OR) × 3

⇒ 6 AND + 3 OR

→ 1 block ⇒ (6 + 28) AND + (3 + 8) OR

= 34 AND + 11 OR

(III) Carry Look Ahead Adder –

$P_i = A_i \oplus B_i$ $\qquad$ $G_i = A_i B_i$

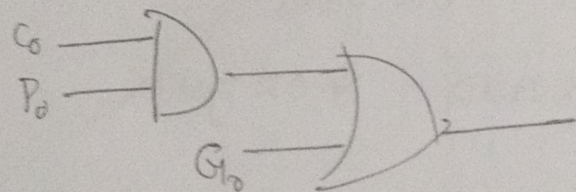
For every bit G_i takes 1 AND ⇒ 4 AND

For every bit, P_i takes 2AND + 1 OR

~~⇒ 8 AND + 4 OR~~

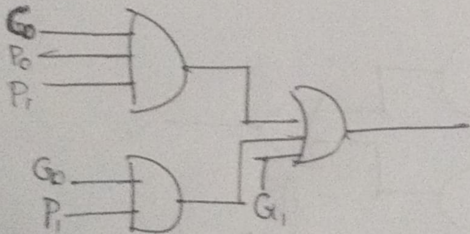
= 8AND + 4OR

C_1 ⇒



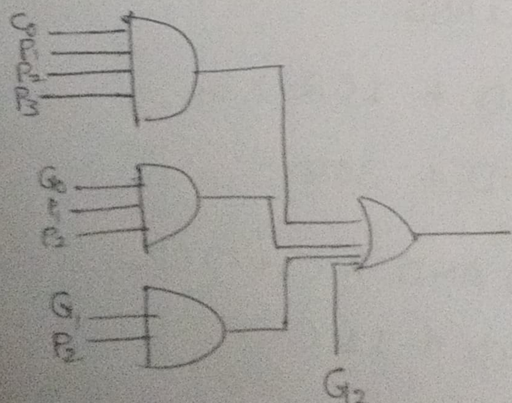
1AND + 1OR

C_2 ⇒



2AND + 1OR

C_3 ⇒



3AND + 1OR

Similarly, for C_4 ⇒ 4AND+1OR

Now, $S = C_{in}[A \oplus B] \Rightarrow$ 8 AND & 4 OR Gates

$\Rightarrow$ Total AND gates : $\overset{4}{\cancel{4+}}8+(1+2+3+4)+8 = \cancel{27\ \text{gates}}$ 30 gates

$\Rightarrow$ Total OR gates : $4+4+4 = 12$ Gates.

Q. 5. We add/subtract in Booth's Algorithm when we encounter $q_0 q_{-1} = 10$ or 01.

Now, if there are n bits $\Rightarrow$ these sums can be for n ~~B~~ times

$\Rightarrow$ Complexity $= O(n \log n)$ if we use carry look ahead adders.

$\Rightarrow$ Now, iterative process is slower than Booth's algorithm for numbers in 2's compliment form as it involves converting a number between its 2's compliment form.
