# Binary adders and subtractors

- Half adder, full adder, parallel adder
- Half subtractor , full subtractor, parallel subtractor
- Subtraction using complements, parallel adder/subtractor
- Carry Look ahead adder, Decimal adder

# Binary Addition

Arithmatic Operation

Example : Addition of two 4-bit Numbers

- carry
- 1 1 0 — intermediate Carry
- 1 0 1 0 + — Operand 1 — 1010 — $10_{10}$
- LSB
- 0 1 1 1 — Operand2 — 0111 — $7_{10}$
- 1 0 0 0 1 — Sum

Final carry

$A + B$

Bit-wise addition

Half-adder

A, B → two, one bit → Sum, Carry out

A, B, $C_{in}$ → three, one-bit → Sum, Carry out

full-adder

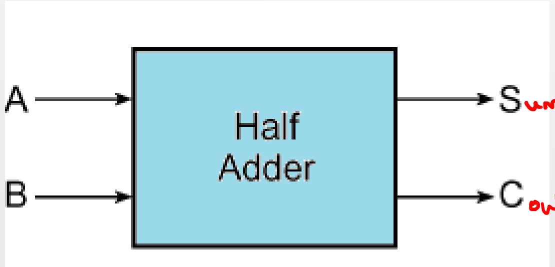
$17_{10} \simeq 11_{16} \simeq 10001_{(2)}$

$2^4\ 2^3\ 2^2\ 2^1\ 2^0$

# Half adder(HA)

- Adds 2, 1-bit numbers A and B , generated two outputs sum(S) and carry (C).

I/P
Two one-bits




| | I/P A | B (LSB) | O/P Sum | $C_{out}$ |
|---|---|---|---|---|
| $m_0$ $\bar{A}\bar{B}$ | 0 | 0 | 0 | 0 |
| $m_1$ $\bar{A}B$ | 0 | 1 | 1 | 0 |
| $m_2$ $A\bar{B}$ | 1 | 0 | 1 | 0 |
| $m_3$ $AB$ | 1 | 1 | 0 | 1 |

MSB
4 combination

Expression for sum and carry :

| A | 0 | 0 | 1 | 1 |
|---|---|---|---|---|
| +B | 0 | 1 | 0 | 1 |
| | 00 | 01 | 01 | 10 |

$Sum = \sum_m 1, 2 = \prod_M 0, 3$

$C_{out} = \sum_m 3 = \prod_M 0, 1, 2$

SOP POS

Sum

| A \ B | $\bar{B}$ 0 | B 1 |
|---|---|---|
| $\bar{A}$ 0 | 0 | 1 |
| A 1 | 1 | 0 |

$\boxed{S = \bar{A}B + A\bar{B}}$

$m_1 \quad m_2$

$\boxed{S = A \oplus B}$

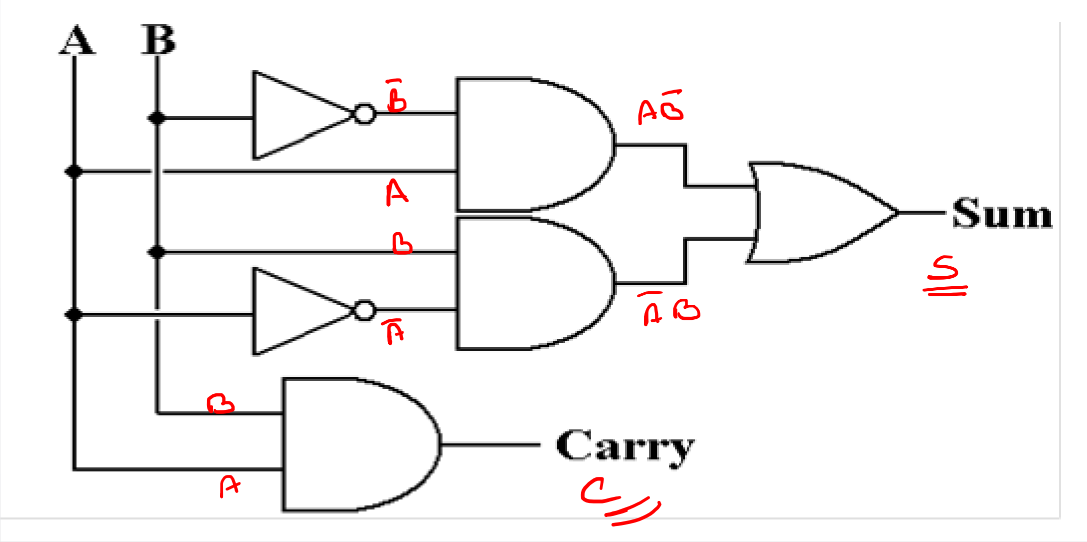
$C_{out}$

| A \ B | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

$\boxed{C_{out} = AB}$

# HA circuit

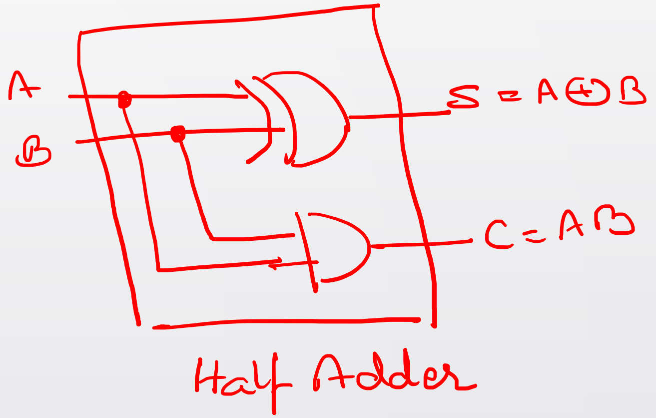
Using basic logic gates




$$S = \bar{A}B + A\bar{B}$$

$$C = AB$$

Using XOR and AND gate



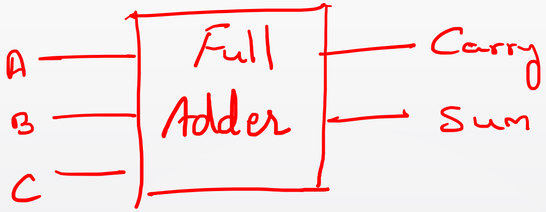
# Full adder

## Truth Table

0+0+1 →

$\begin{array}{r} 00 \\ 1 \\ \hline 01 \end{array}$

| A (MSB) B C (LSB) | | Carry | Sum |
|---|---|---|---|
| 0 0 0 | $m_0$ | 0 | 0 |
| 0 0 1 | $m_1$ | 0 | 1 |
| 0 1 0 | $m_2$ | 0 | 1 |
| 0 1 1 | $m_3$ | 1 | 0 |
| 1 0 0 | $m_4$ | 0 | 1 |
| 1 0 1 | $m_5$ | 1 | 0 |
| 1 1 0 | $m_6$ | 1 | 0 |
| 1 1 1 | $m_7$ | 1 | 1 |




$$Carry = \sum_m 3, 5, 6, 7 = \prod_M 0, 1, 2, 4$$

$$Sum = \sum_m 1, 2, 4, 7 = \prod_M 0, 3, 5, 6.$$

Sum

| A \ BC | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |

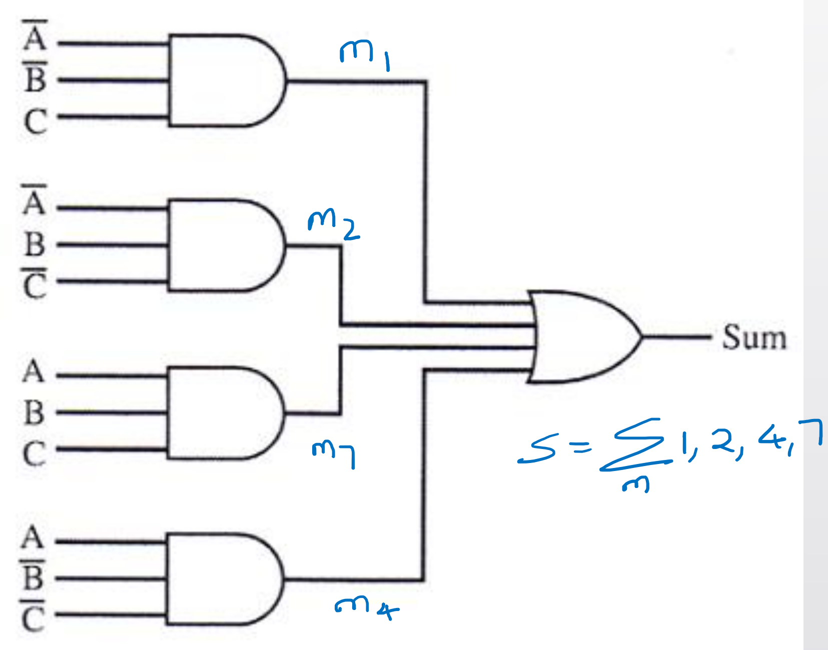
Carry

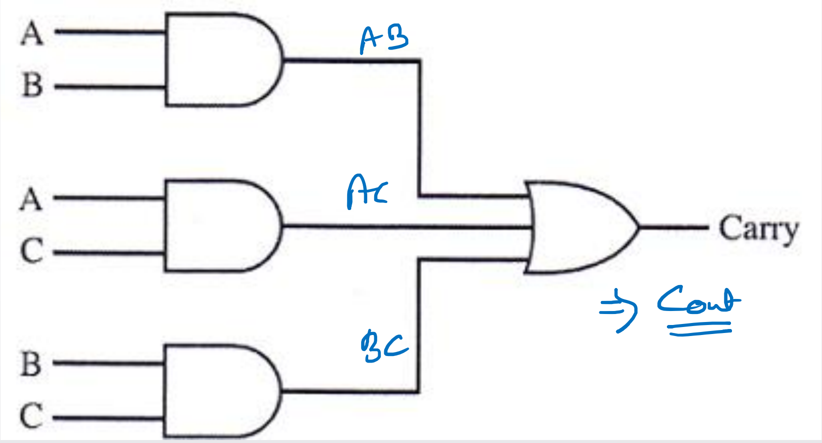
| A \ BC | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 [0] | 0 [1] | 1 [3] | 0 [2] |
| 1 | 0 [4] | 1 [5] | 1 [7] | 1 [6] |

$$sum = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$
$$= \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}\bar{C} + BC)$$
$$= \bar{A}(B \oplus C) + A(\overline{B \oplus C})$$
$$= A \oplus B \oplus C$$

$$Carry = BC + AC + AB$$

# FA circuit using basic logic gates



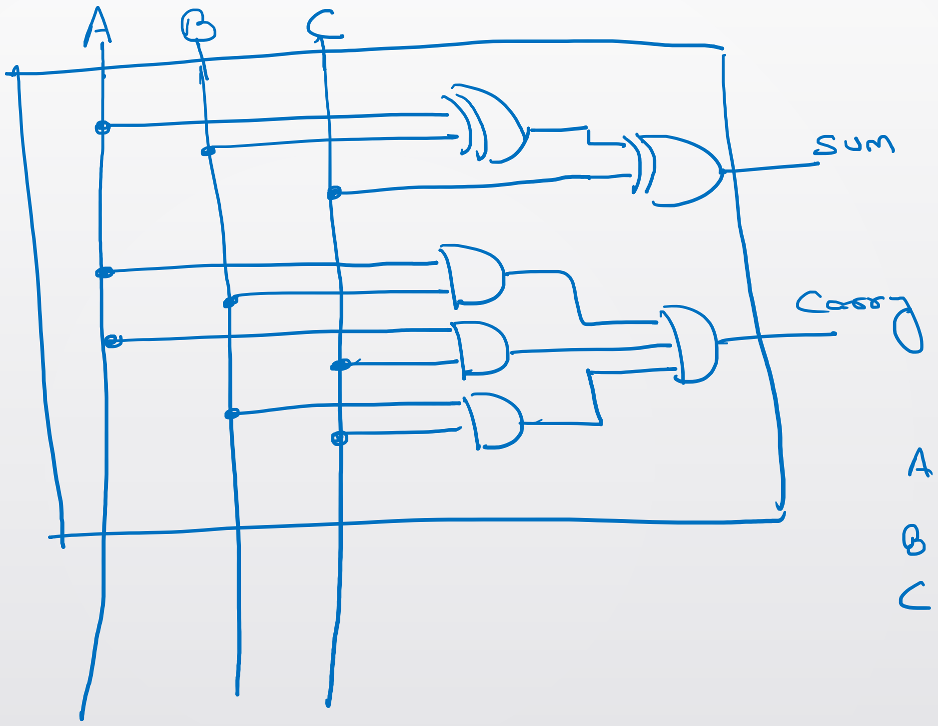



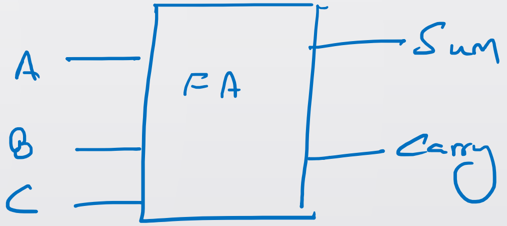
# Full adder circuit using XOR operations

FA with XOR

Sum = $A \oplus B \oplus C$

Carry = $AB + AC + BC$



# FA using 2 HA s and one external gate

**Half Adder**

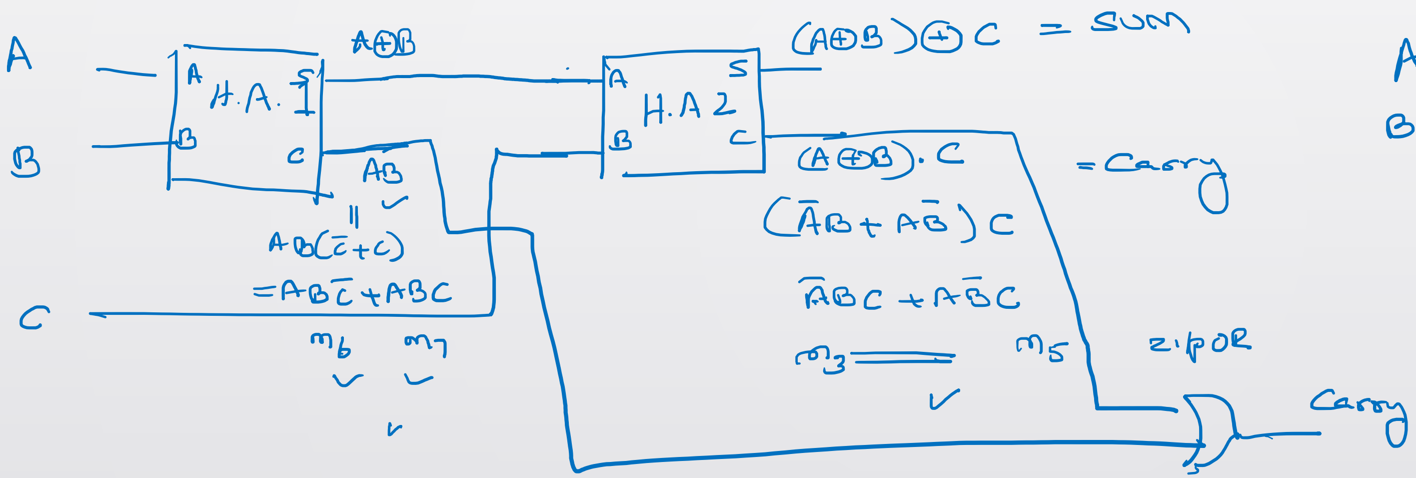
$S = A \oplus B$

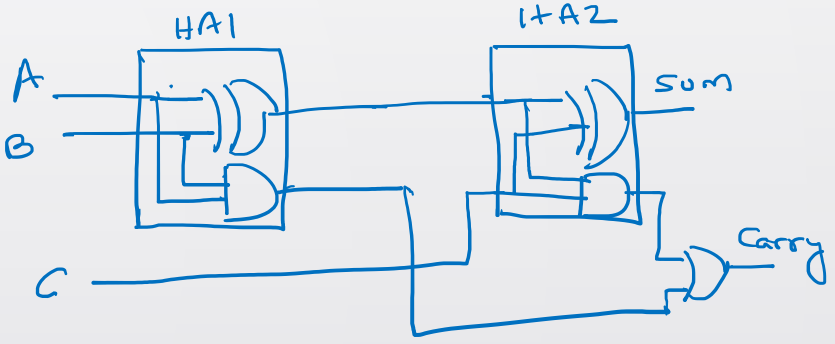
$C = A \cdot B$

**Full Adder**

$$\text{SUM} = \underbrace{(A \oplus B)}_{\text{S of HA1}} \oplus C$$

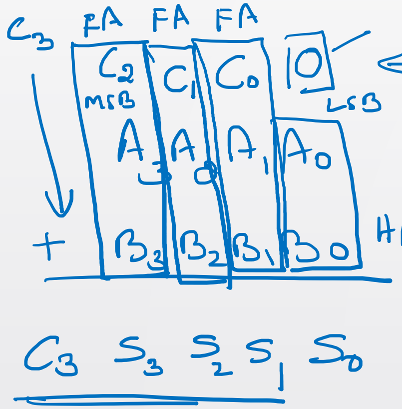
$$\text{Carry} = AB + AC + BC = \sum_m 3, 5, 6, 7 = (\underset{m_3}{\bar{A}BC} \overset{\text{OR}}{+} \underset{m_5}{A\bar{B}C}) \overset{\text{OR}}{+} (\underset{m_6}{AB\bar{C}} \overset{\text{OR}}{+} \underset{m_7}{ABC})$$



# 4-bit Parallel adder using FA blocks

- Consider addition of 2, 4-bit numbers: ($A_3$ $A_2$ $A_1$ $A_0$) and ($B_3$ $B_2$ $B_1$ $B_0$)



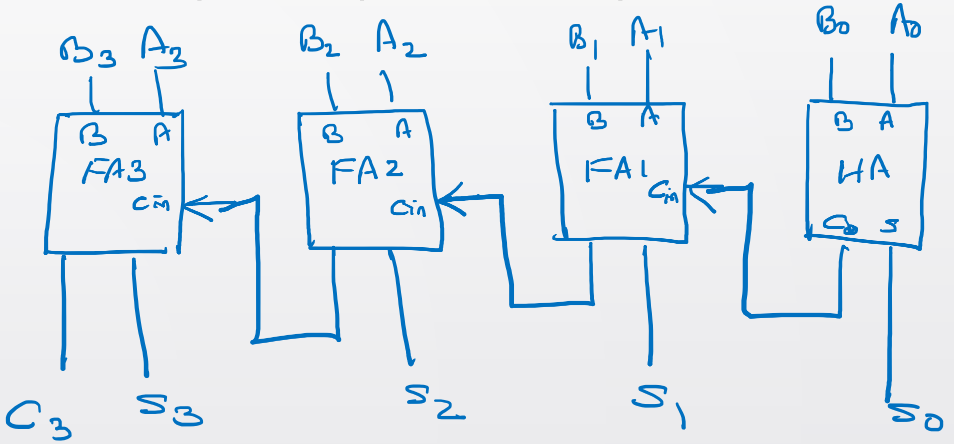

Carry Propagated

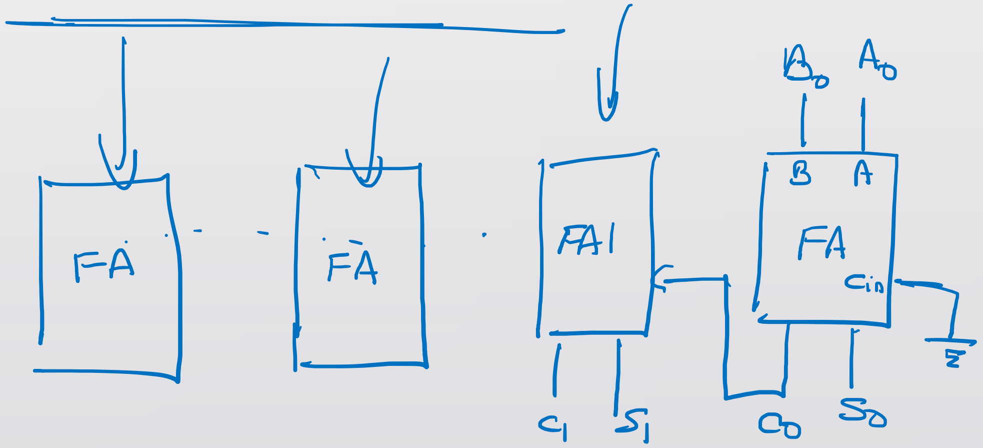
OR A (4-bit Number)

B (4-bit Number)

Addition → of two 4-bit Numbers

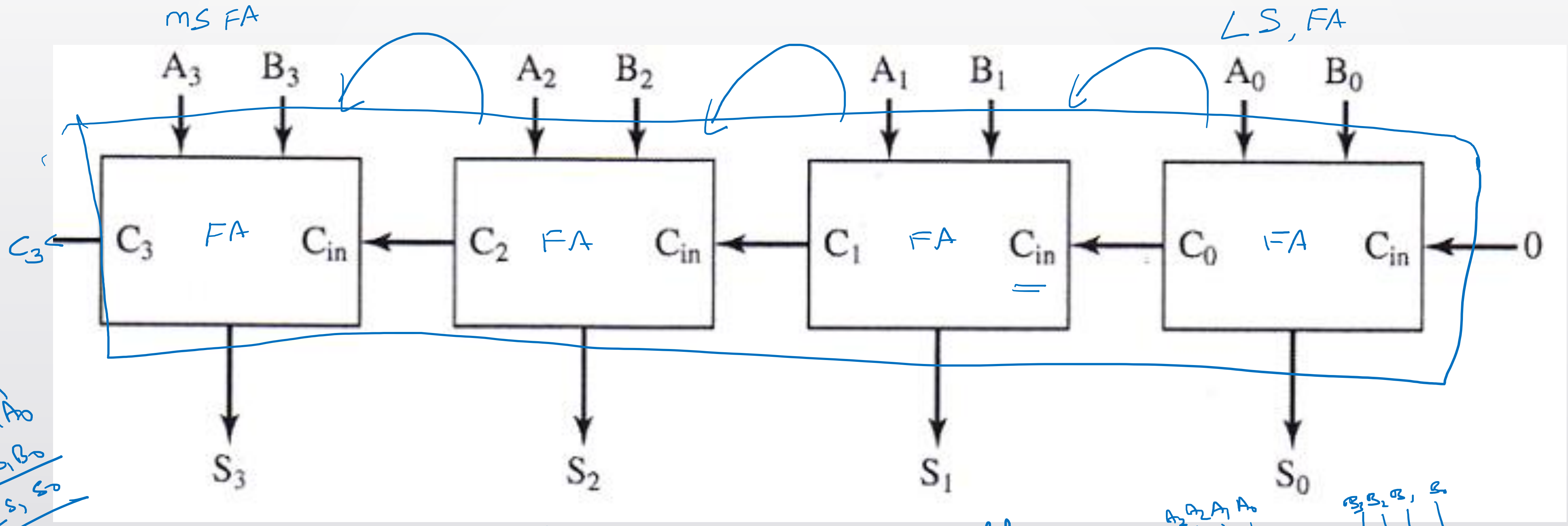
| Carry | SUM |
|---|---|
| 1-bit | 4-bit |
| Result 5 bit | |



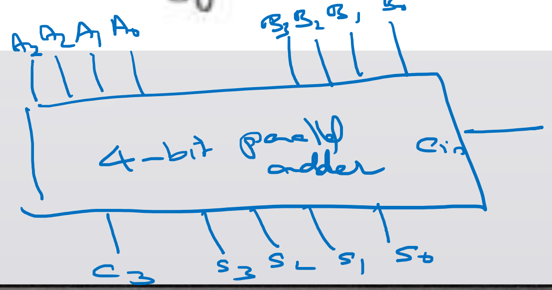
# 4-bit parallel adder

Also called as Carry Propagation Adder (CPA)




$C_2 C_1 C_0$
$A_3 A_2 A_1 A_0$
$B_3 B_2 B_1 B_0$
$C_3 S_3 S_2 S_1 S_0$

Each Gate : Response time ∝ Gate propagation delays

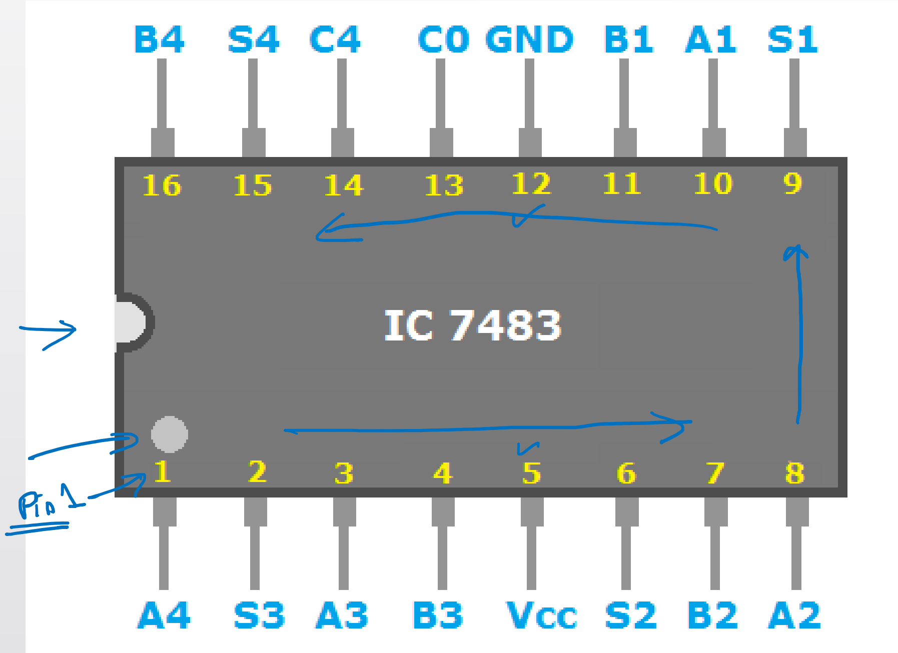


# 7483 IC : 4-BIT PARALLEL ADDER

Top notch

Groove

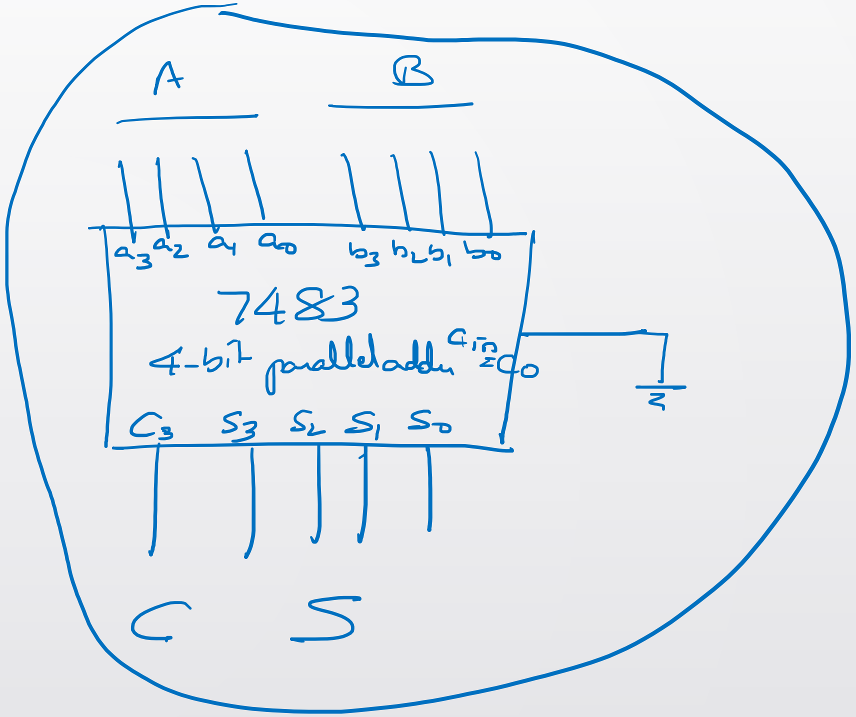
Pin 1




IC 7483 Pin Diagram

16 pin DIP

Activate



# Half subtractor

- Write the truth table and circuit for half subtractor

$$\begin{array}{r} A \\ -B \\ \hline A-B \end{array}$$

bit wise subtraction

A → 1-bit 0 or 1

B → 1-bit 0 or 1

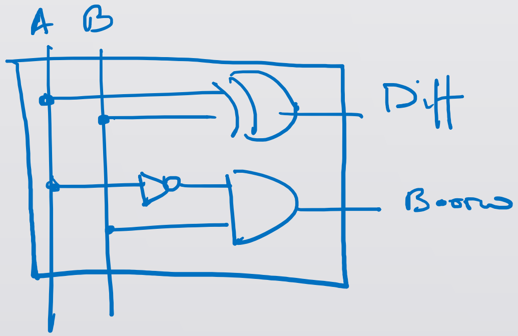
$$\begin{array}{r} 0 \\ -0 \\ \hline 00 \end{array} \qquad \begin{array}{r} {}^{1}0 \\ -1 \\ \hline 11 \end{array} \qquad \begin{array}{r} 1 \\ -0 \\ \hline 01 \end{array} \qquad \begin{array}{r} 1 \\ -1 \\ \hline 00 \end{array}$$

.... $2^2\ 2^1\ 2^0$    2

| | A | B | Difference | Borrow |
|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 0 |
| $m_1$ | 0 | 1 | 1 | 1 |
| $m_2$ | 1 | 0 | 1 | 0 |
| $m_3$ | 1 | 1 | 0 | 0 |

Difference $:= \sum_m 1, 2 = \prod_M 0, 3$

Borrow $= \sum_m 1 = \prod_M 0, 2, 3$




$-ve$

Difference

| A \ B | 0 | 1 |
|---|---|---|
| $\bar{A}$ 0 | 0 | 1 |
| A 1 | 1 | 0 |

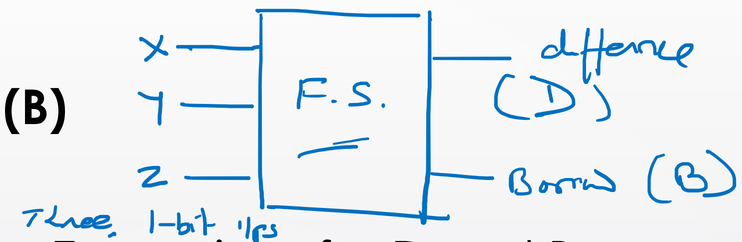
$\boxed{\text{Difference} = \bar{A}B + A\bar{B} = A \oplus B}$

| A \ B | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 0 | 0 |

$\boxed{\text{Borrow} = \bar{A}B}$

# Full subtractor

**DIFFERENCE (D) = X-Y-Z, Borrow (B)**




| X | Y | Z | D | B |
|---|---|---|---|---|
| 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 |

Expressions for D and B:

$$D = \sum_m 1, 2, 4, 7 = \prod_M 0, 3, 5, 6$$

$$\boxed{B = \sum_m 1, 2, 3, 7} = \prod_M 0, 4, 5, 6$$

D

| x \ yz | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |

$\bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$

$$D = x \oplus y \oplus z$$

B

| x \ yz | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 |

$$B = \bar{x}z + \bar{x}y + yz$$

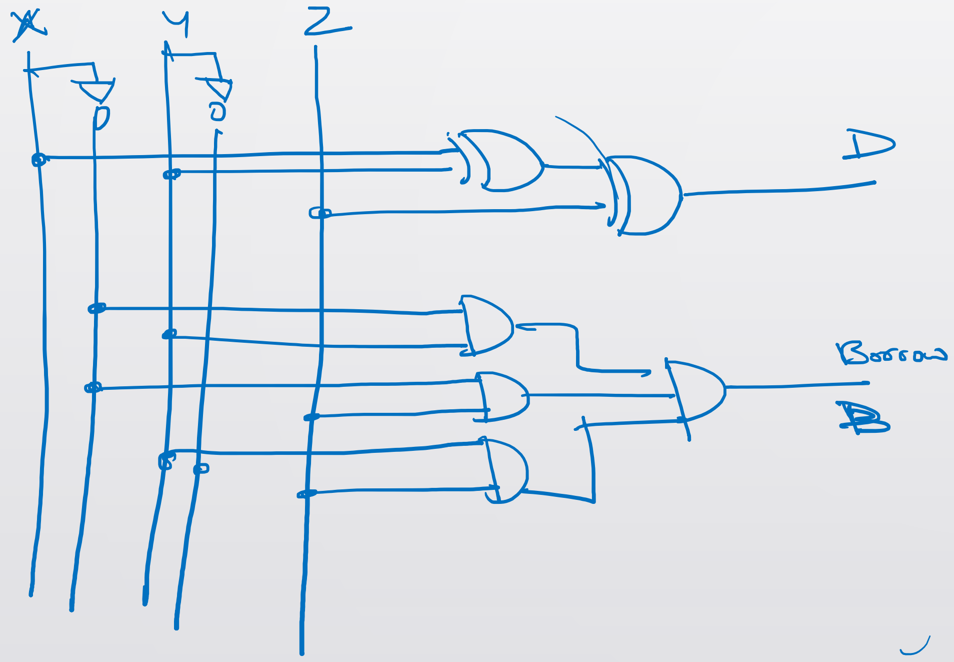
# FS circuit

- Draw the circuit for FS using
- (i) basic logic gates only
- (ii) XOR and basic logic gates

$D = x \oplus y \oplus z = \sum 1, 2, 4, 7 = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$

$B = \bar{x}y + \bar{x}z + yz$

PRACTICE

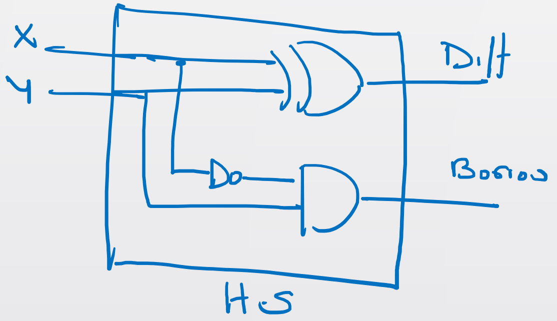


# Full subtractor using 2 HS s and one external gate

HS

Diff = $x \oplus y$

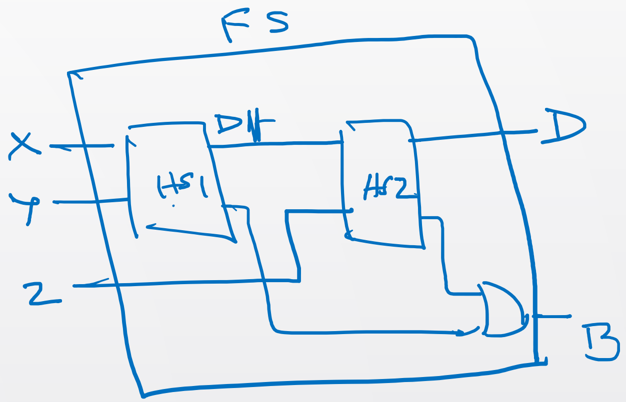
Borrow = $\bar{x}y$

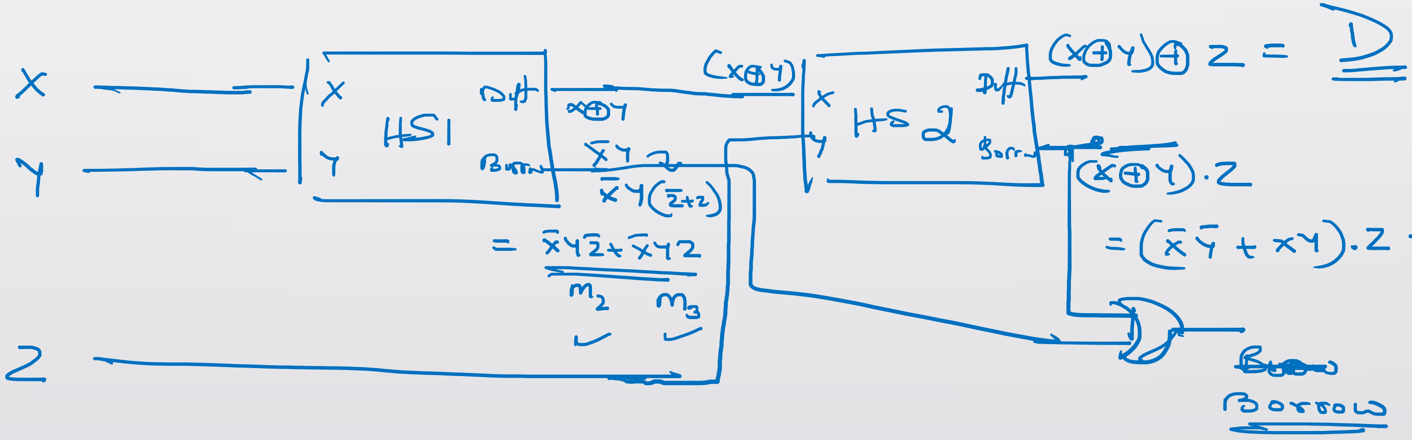



F.S

$$D = x \oplus y \oplus z = \overbrace{\underbrace{(x \oplus y)}_{\text{HS1 Diff term}} \oplus z}^{\text{HS2 Diff term}}$$

$$B = \sum_m 1, 2, 3, 7 = (m_1) + (m_2) + (m_3) + (m_7)$$




$x \oplus y$

$(x \oplus y)$

$(x \oplus y) \oplus z = D$

$\bar{x}y$

$\bar{x}y(\bar{z} + z)$

$= \underbrace{\bar{x}y\bar{z}}_{m_2} + \underbrace{\bar{x}yz}_{m_3}$

$\overline{(x \oplus y)} \cdot z$

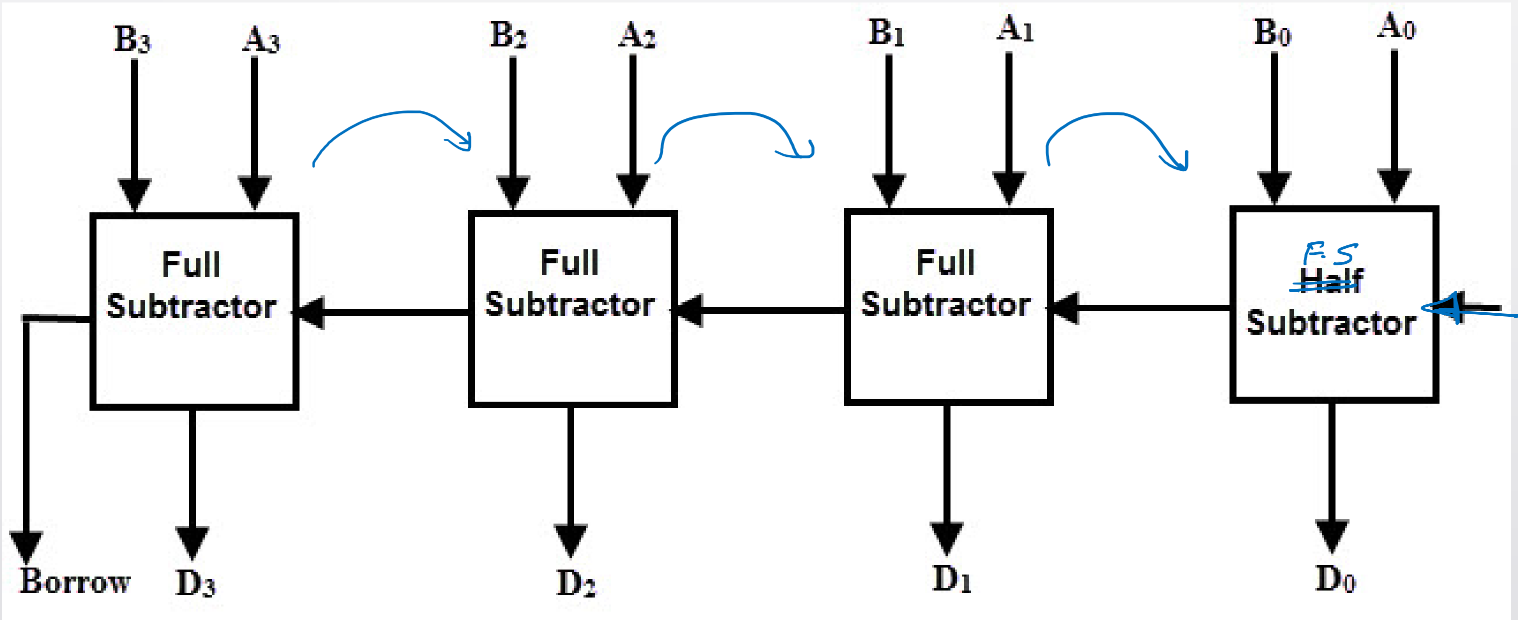
$= (\bar{x}\bar{y} + xy) \cdot z = \underbrace{\bar{x}\bar{y}z}_{m_1} + \underbrace{xyz}_{m_7}$

~~Borrow~~ Borrow

# 4-bit parallel subtractor using FS blocks

Consider subtraction of 2, 4-bit numbers: (A3 A2  A1 A0 ) and (B3 B2 B1 B0)

$A_3 A_2 A_1 A_0$
$-B_3 B_2 B_1 B_0$




246
683
−563

10000
2999
−8001

1000 →
1111

246
− 683
3

246
316
563
317

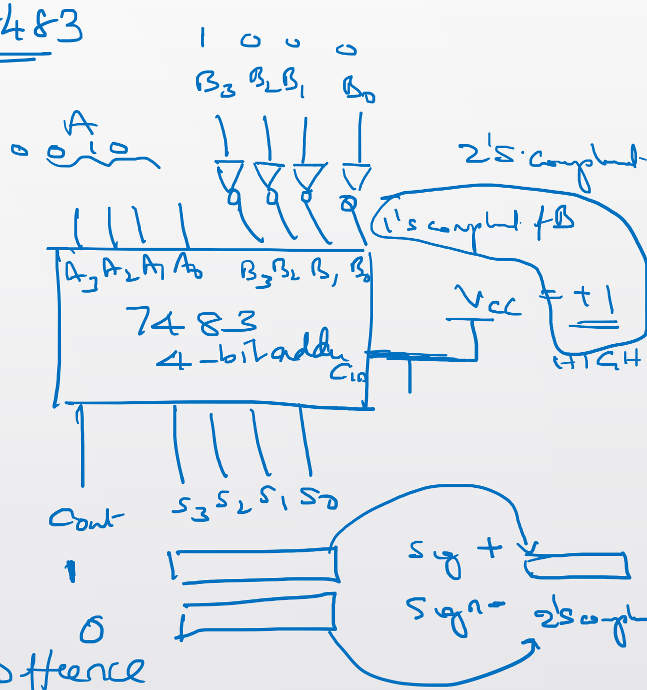
999
4

683
317
1000

−ve
563
437

# Subtraction using complements

use 4 bit parallel adders or use 7483




## ❑ Using 2's complement method

❑ Using 1's complement method

Let us use 2's complement addition

Example

$$\begin{array}{r} 8 \\ -\ 2 \\ \hline +\ 6 \end{array} \rightarrow \begin{array}{r} 1000 \\ -\ 0010 \\ \hline \end{array} \rightarrow \begin{array}{r} 1000 \\ 1101 \\ +1 \\ \hline 1\ 0110 \end{array}$$

2's complement: 1101 +1 (1's complement +1)

+ve sign if Drop carry

0110 → 6 Difference

retain only sum terms → Difference

Example

$$\begin{array}{r} 2 \\ -\ 8 \\ \hline -\ 6 \end{array} \rightarrow \begin{array}{r} 0010 \\ 1000 \end{array}$$

Retain → 0010

2's comp → 0111 +1 (1's comp +1)

$$\begin{array}{r} 0111 \\ 0010 \\ +1 \\ \hline 0\ 1010 \end{array}$$

if Cout = 0 sign −ve

2's complement

$$1010 \rightarrow \begin{array}{r} 0101 \\ 1 \\ \hline 0110 \end{array}$$

# Subtraction using complements

- Using 2's complement method

- **Using 1's complement method**

8 → 1000 → 1000

−2 → −0010 →(1's comp) + 1101

1 0101 (carry, SUM) → Carry +ve sign

also use carry to add to sum

0101 + 1 = 0110

SUM; add Carry to Least bit

2 → 0010

−8 → 1000

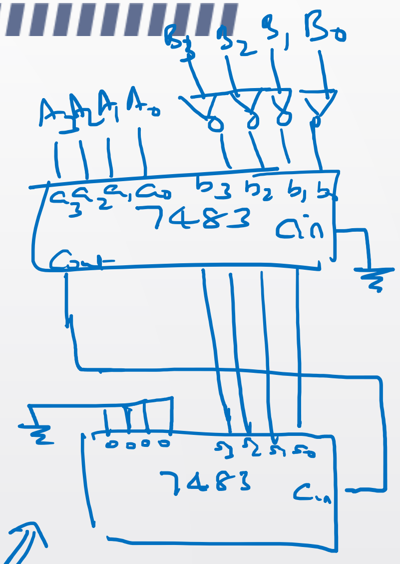
2   0010 → 0010

−8   − 1000 →(1's comp, Add) + 0111

0 1001

Cout or Carry is 0 ↓ −ve Number → Ans is in 1's complement form

1001 → 0110 = −6




2nd level

if Cout = 1

Cout = 0 ?

**Questions?**