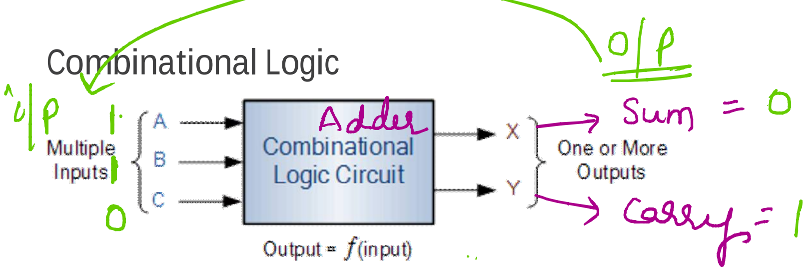
Combinational Logic Circuits

The outputs of **Combinational Logic Circuits** are only determined by the logical function of their current input state, logic “0” or logic “1”, at any given instant in time.
The result is that combinational logic circuits have no feedback, and any changes to the signals being applied to their inputs will immediately have an effect at the output. In other words, in a **Combinational Logic Circuit**, the output is dependant at all times on the combination of its inputs. Thus a combinational circuit is *memoryless*.

So if one of its inputs condition changes state, from 0-1 or 1-0, so too will the resulting output as by default combinational logic circuits have “no memory”, “timing” or “feedback loops” within their design.

Combinational Logic



1+0+0 = 1

Sum = 1

Carry = 0

1+1+0 = 10

Carry Sum

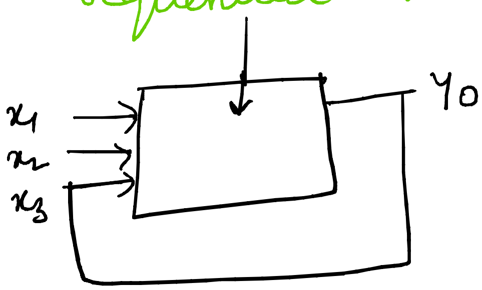
Combinational Logic Circuits are made up from basic logic NAND, NOR or NOT gates that are “combined” or connected together to produce more complicated switching circuits. These logic gates are the building blocks of combinational logic circuits.

>

Sequential ckt → o/p depends on

present i/p

past history of i/p



Combinational ckts (MSI)

Adder

Subtractor

Comparators → compare two i/p

multiplexer → many to → one

SSI → < 10

MSI — < 100

VLI — < 1000 gate

VVLI > 1000

A>B, A<B, A=B

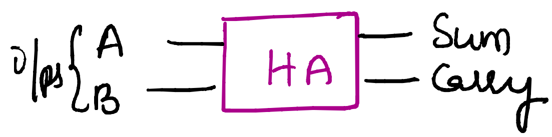
multiplexer → many to → one
Demultiplexer → one to many
Encoder
Decoders } → code conversion

Half adder

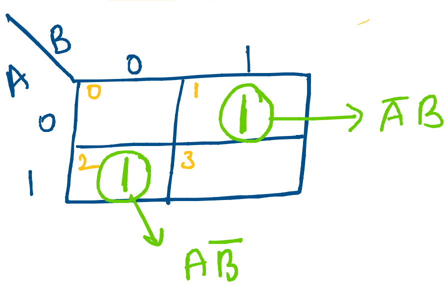
2 bit = 2^2 combinations

		Sum	Carry
0+0	=	0	0
0+1	=	1	0
1+0	=	1	0
1+1	=	0	1

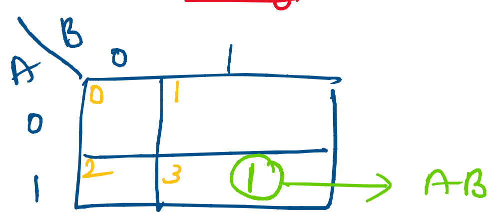
	i/ps		o/ps	
Decimal	A	B	Sum	Carry
0	0	0	0	0
1	0	1	1	0
2	1	0	1	0
3	1	1	0	1
			$A \oplus B$	AB



Sum



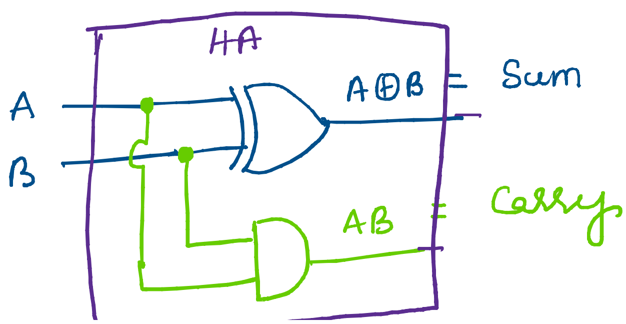
Carry

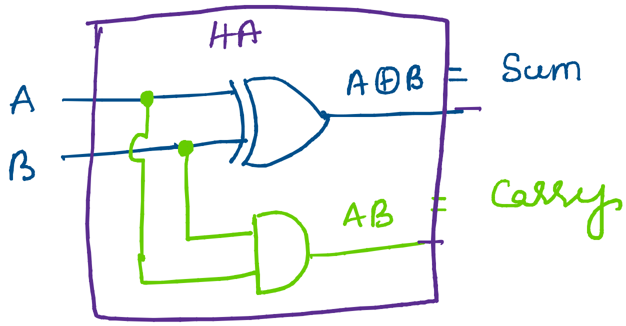


Sum = $\bar{A}B + A\bar{B} \Rightarrow A \oplus B$

Carry = AB

Half adder



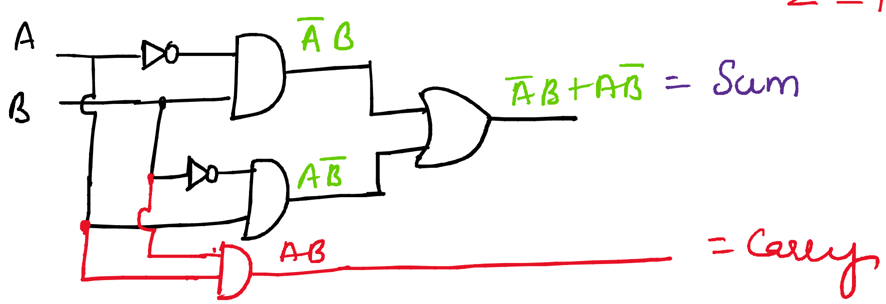


Half adder using basic gates

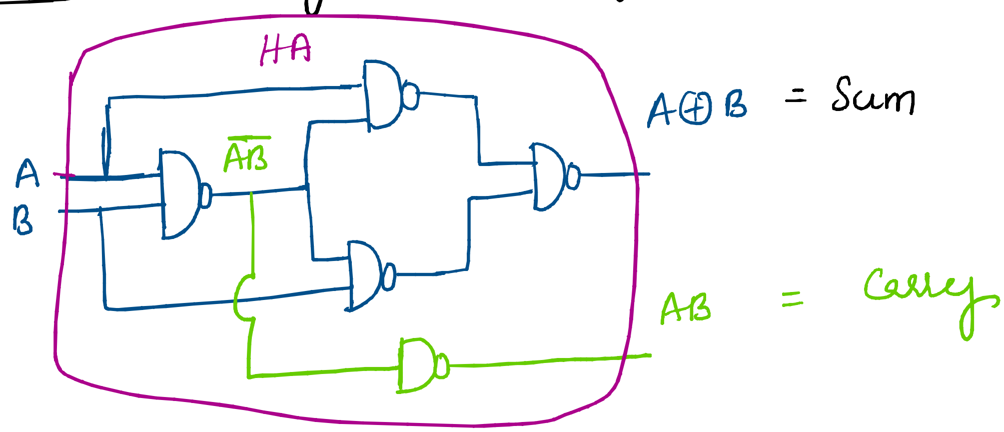
Sum = $\bar{A}B + A\bar{B}$

Carry = AB

3 → AND
1 – OR
2 – NOT



Half adder using NAND gates



HA → 5 NAND gates

full adder

3 bits = 2^3 combinations

	Inputs			Outputs	
	A	B	C	Sum	carry
0	0	0	0	0	0

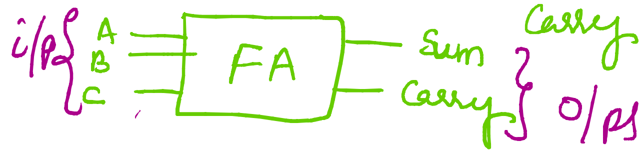
$1+1+1 = (3)_{10} = (11)_2$

	A	B	C	Sum	Carry
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

Sum $\rightarrow A \oplus B \oplus C$

$1+1+1 = (3)_{10} = (11)_2$

Carry Sum



Combinational ckt

$A \odot B = \overline{A \oplus B}$

for Sum

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

$\bar{A}\bar{B}C$, $\bar{A}B\bar{C}$, $A\bar{B}\bar{C}$, ABC

$$\text{Sum} = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

$$= C(\underbrace{\bar{A}\bar{B} + AB}_{A \odot B}) + \bar{C}(\bar{A}B + A\bar{B})$$

$$= \underbrace{C}_{B}(\underbrace{\overline{A \oplus B}}_{\bar{A}}) + \underbrace{\bar{C}}_{\bar{B}}(\underbrace{A \oplus B}_{A})$$

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

$$= \underbrace{A \oplus B}_{A} \oplus C$$

for Carry

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

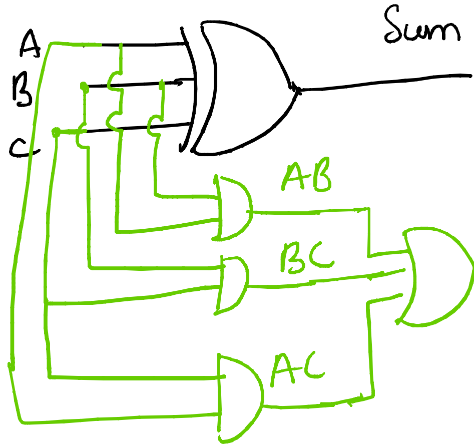
$BC \rightarrow$ 2 literals

AB

AC

$$\text{Carry} = AB + BC + AC$$

full adder

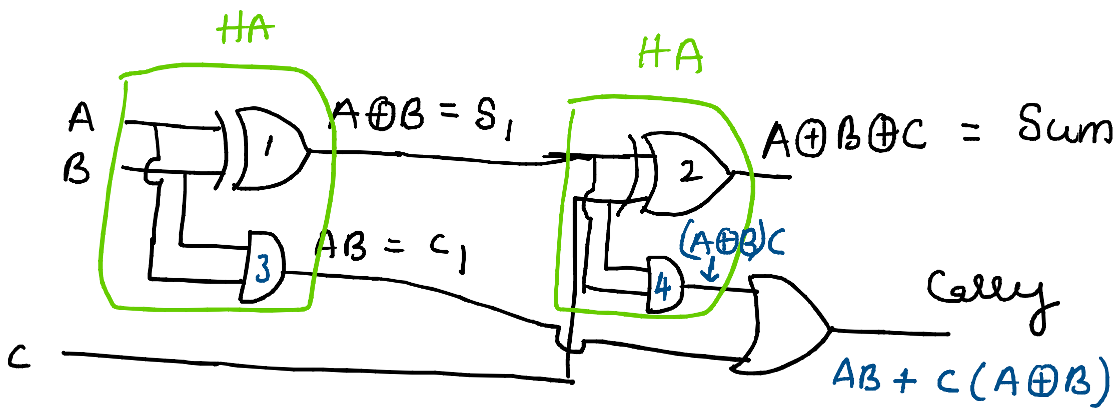


$\text{Sum} = A \oplus B \oplus C$

$= \bar{A}\bar{B}C + A\bar{B}\bar{C} + \bar{A}B\bar{C} + ABC$

$\text{Carry} = \underline{AB + BC + AC}$

full adder using two half adder



$\underline{\text{Carry}} = AB + C(A \oplus B)$

$= AB + C(\bar{A}B + A\bar{B})$

$= AB + \underbrace{\bar{A}BC + A\bar{B}C}$

$= AB(C+1) + \bar{A}BC + A\bar{B}C$

$= \underbrace{ABC} + AB + \bar{A}BC + \underbrace{A\bar{B}C}$

$= AC(B+\bar{B}) + AB + \bar{A}BC$

$= AC + AB + \bar{A}BC$

$AC + AB(1+C) + \bar{A}BC$

$= AC + AB + \underbrace{ABC + \bar{A}BC}$

$$AC + AB + BC(A + \bar{A})$$

$$= AC + AB + BC$$

A → 1 bit
B → 1 bit → HA → 1 bit → A = 1 bit, B = 1 bit

A B C
↓ ↓ ↓
1 1 1 → FA → 3 bit ⇒ A →, B →, C →

A = 1011 ⇒ A = 4 bits
B = 1100 ⇒ B = 4 bits

Half adder → A = 1 bit, B = 1 bit } + add → Sum, Carry

Full adder = A = 1 bit, B = 1 bit, C = 1 bit } + add → Sum, Carry

	MSB			LSB
A =	1	0	1	1
B =	1	1	0	0
Carry 1)	0	1	1	1

HA

	MSB	FA	FA	LSB
A =	1	1	1	1
B =	1	1	1	1

Carry, FA, HA

Carry Sum
1+1 = 1 0
1+1+1 = 1 1

$A_0 + B_0 = C_0 S_0$

C_2 C_1 C_0

A_3 A_2 A_1 A_0

A = 1 1 1 1

B_3 B_2 B_1 B_0

$A = \quad 1 \quad 1 \quad 1 \quad 1$

$B = \quad 1 \quad 1 \quad 1 \quad 1$

Carry → 1) 1 1 1 0

$A = \quad 1 \quad 1 \quad 1 \quad 1$

$\quad B_3 \; B_2 \; B_1 \; B_0$

$B = \quad 1 \quad 1 \quad 1 \quad 1$

Carry → 1) 1 1 1 0

C_3) $S_3 \; S_2 \; S_1 \; S_0$

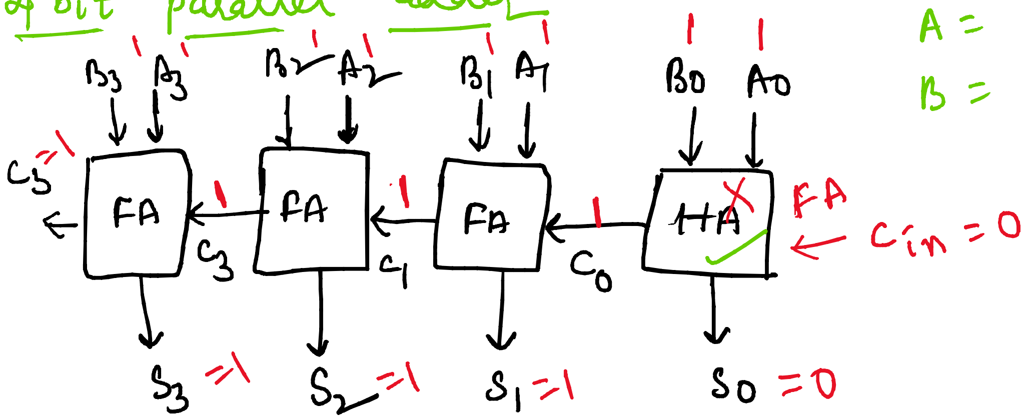
4 bit adder → needs one HA
Three FA

$A_1 + B_1 + C_0 = C_1 \, S_1$

$A_2 + B_2 + C_1 = C_2 \, S_2$

$A_3 + B_3 + C_2 = C_3 \, S_3$

4 bit parallel adder



$\quad A_2 \; A_1 \quad A_0$

$A = 1 \; 1 \; 1 \; 1$

$B = 1 \; 1 \; 1 \; 1$

Half Subtractor

A	B	Difference	Borrow
inputs		outputs	
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

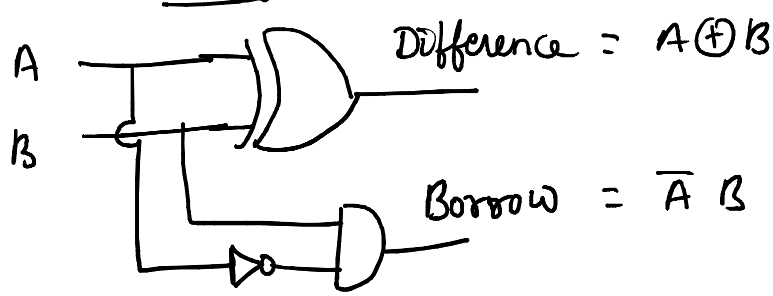
$\bar{A} B$

↓ XOR

Difference = $A \oplus B$

Borrow = $\bar{A} B$

Half Subtractor

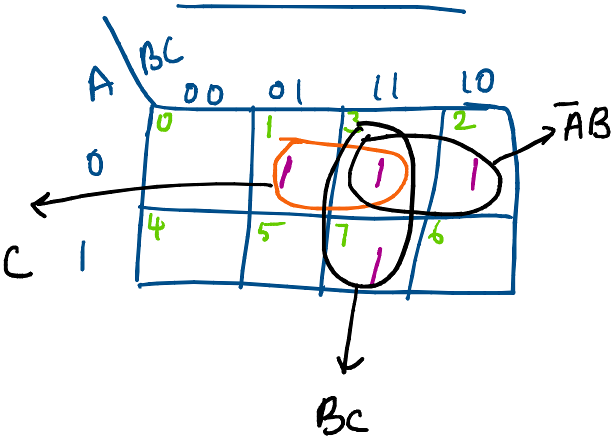


Full Subtractor using two Half Subtractors

	A	B	C	Diff	Borrow
0	0	0	0	0	0
1	0	0	1	1	1
2	0	1	0	1	1
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	0
6	1	1	0	0	0
7	1	1	1	1	1

Diff = $A \oplus B \oplus C$

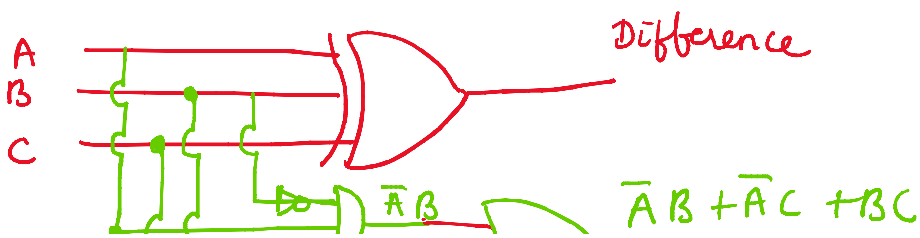
for Borrow

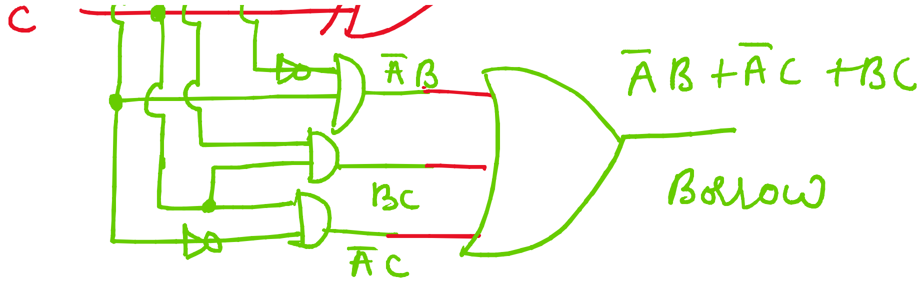


Full Subtractor

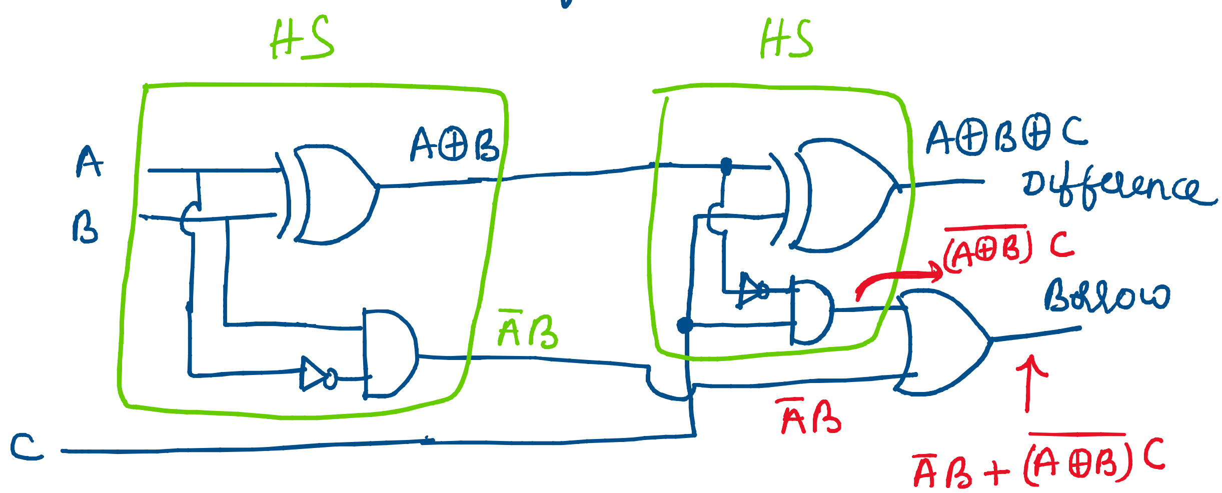
Difference = $A \oplus B \oplus C$

Borrow = $\bar{A}B + \bar{A}C + BC$





Full subtractor using two half subtractor



Borrow $= \bar{A}B + \overline{(A \oplus B)}\, C$

$\overline{A \oplus B} = A \odot B$

$A \odot B = \bar{A}\bar{B} + AB$

$= \bar{A}B + (\bar{A}\bar{B} + AB)\, C$

$= \bar{A}B + \bar{A}\bar{B}C + ABC$

$= \bar{A}B(1 + C) + \bar{A}\bar{B}C + ABC$

$= \bar{A}B + \bar{A}BC + \bar{A}\bar{B}C + ABC$

$\bar{A}B + \bar{A}C(B + \bar{B}) + ABC$

$\bar{A}B(1 + C) + \bar{A}C + ABC$

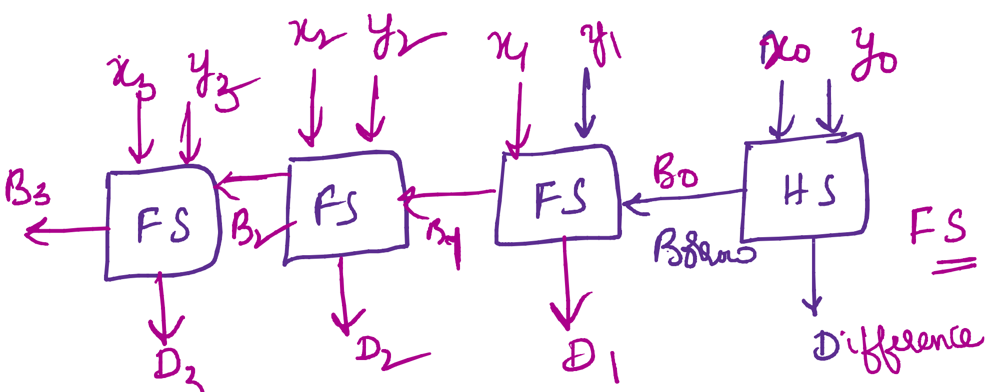
$\bar{A}B + \bar{A}BC + \bar{A}C + ABC$

$\bar{A}B + \quad BC(A + \bar{A}) + \bar{A}C$

$$\bar{A}B + \overline{1\; BC}\,(A+\bar{A}) + \bar{A}C$$

$$\bar{A}B + BC + \bar{A}C \checkmark$$

4 bit parallel subtractor



FS

Add	Sub
8	8
+7	−7

$$A = (8)_{10} = 1000$$
$$B = (7)_{10} \quad -\ 0111$$
$$0001 \checkmark$$

} Subtraction

B is in 2's ⇒ 1's = 1000 + 1 = 1001

$(7)_{10}$ in 2's Complement = 1001

A = 1000

2's Comple B = + 1001

} Subtraction using → 1 adder

2's Comple B = 1001

$$\begin{array}{r} 1001 \\ + \\ \hline \text{carry} \rightarrow 1)\ 0001 \end{array}$$

~~neglect~~

using addition procedure

1's Complement

1's Comp

$$\begin{array}{rl} A = & 1000 \\ B = & 1000 \\ \hline \text{Carry} \rightarrow & 1)0000 \\ & 1 \\ \hline & 0001 \end{array}$$

2 → adders

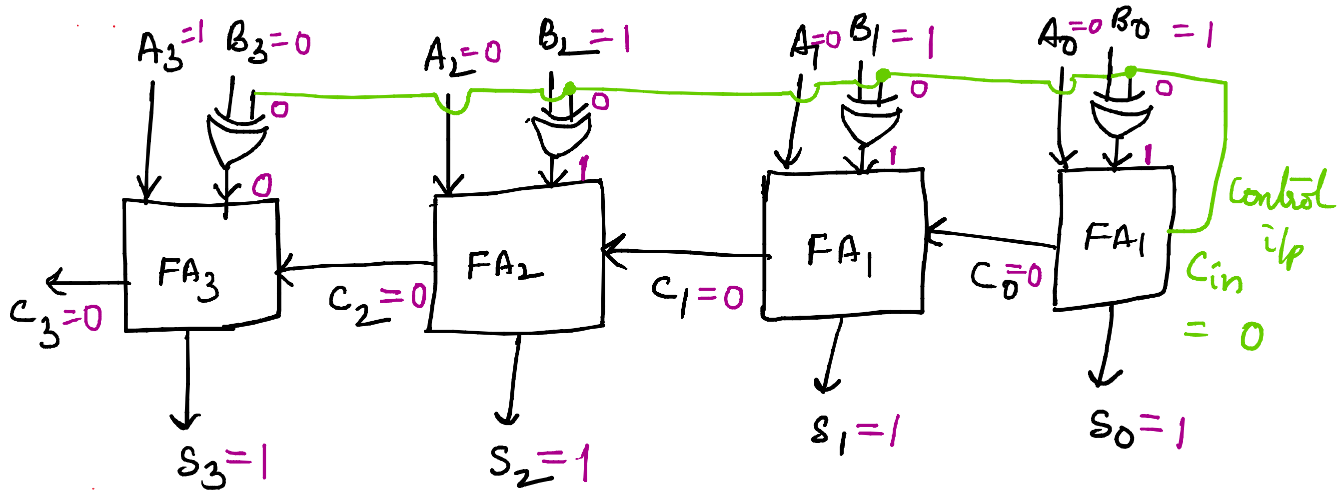
Addition + **Subtraction**

$$\begin{array}{rl} A = & 1000 \\ B = & 0111 \\ \hline & 1111 \end{array}$$

$$\begin{array}{rl} A = & 1000 \\ B = & 0111 \\ \hline & 0001 \end{array}$$

addition procedure 2's Complem

Adder - Subtractor (4 bit adder/subtractor)



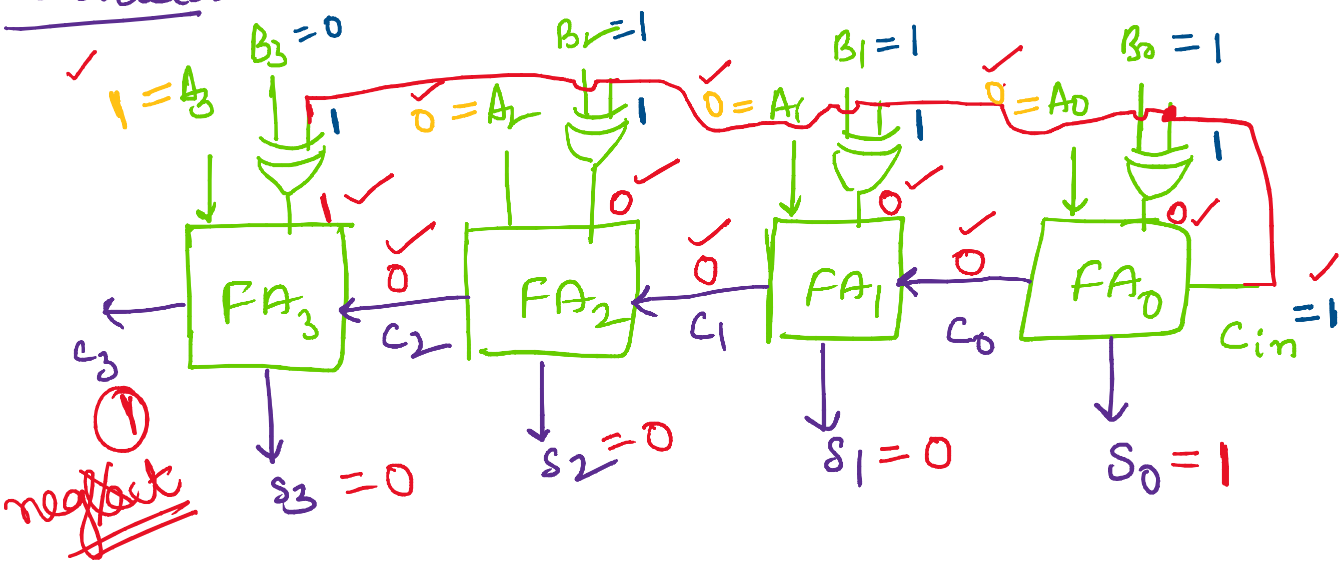
$A = 1000$ ($A_3\,A_2\,A_1\,A_0$)

$B = 0111 \rightarrow 1's \rightarrow 1000$

$2's \quad 0001$

$A = 1000$
$B = 0111$
$\overline{\quad 1111 \quad}$

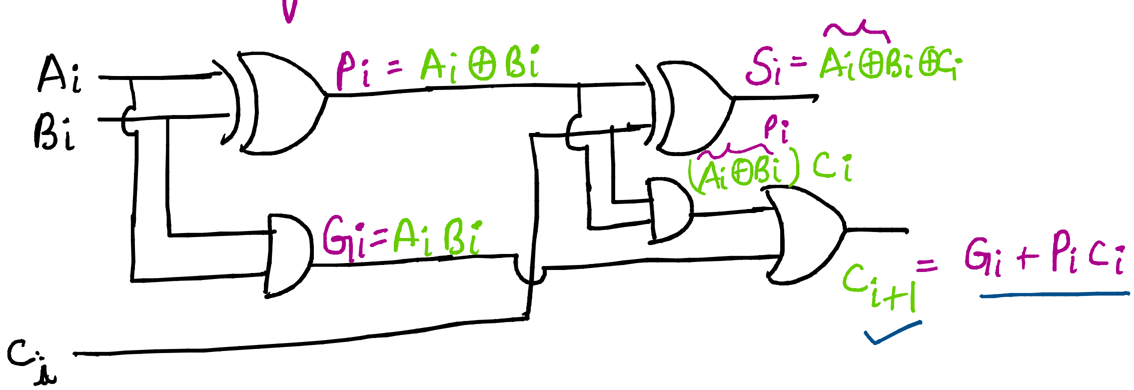
Subtractor



0 0 01

Carry look ahead generator

Full adder using two half adders



Carry generate $(G_i) = A_i B_i$

Carry propagate $(P_i) = A_i \oplus B_i$

O/p Carry $(C_{i+1}) = G_i + P_i C_i$

o/p carry $(C_{i+1}) = G_i + P_i C_i$

$i = 1 \Rightarrow C_2 = G_1 + P_1 C_1$

$i = 2 \Rightarrow C_3 = G_2 + P_2 C_2$

$= G_2 + P_2 (G_1 + P_1 C_1)$

$= \underline{G_2 + P_2 G_1 + P_2 P_1} C_1$

$i = 3 \Rightarrow C_4 = G_3 + P_3 C_3$

$= G_3 + P_3 (G_2 + P_2 G_1 + P_2 P_1 C_1)$

i/ps $\Rightarrow$ $\underline{P_1 \; C_1 \; G_1 \; P_2 \; G_2 \; P_3 \; G_3}$