Digital Arithmetic Operation and Circuits :-

Binary Addition:-

There are only four cases can occur in adding the two binary digits (bits) in any position. They are :-

0+0 =0
1+0 =1
1+1 =0 = 1 +carry of 1 into next position
1+1+1 =1 = 1 +carry of 1 into next position

Example:-

$$\begin{array}{cccccc} & & 0 & 1 & 1 & (3) \\ + & & 1 & 1 & 0 & (6) \\ \hline & 1 & 0 & 0 & 1 & (9) \end{array} \qquad \begin{array}{ccccccc} & & 1 & 0 & 0 & 1 & (9) \\ + & & 1 & 1 & 1 & 1 & (15) \\ \hline & 1 & 1 & 0 & 0 & 0 & (24) \end{array}$$

$$\begin{array}{cccccccc} & & 1 & 1. & 0 & 1 & 1 & (3.375) \\ + & & 1 & 0. & 1 & 1 & 0 & (2.750) \\ \hline & 1 & 1 & 0. & 0 & 0 & 1 & (6.125) \end{array}$$

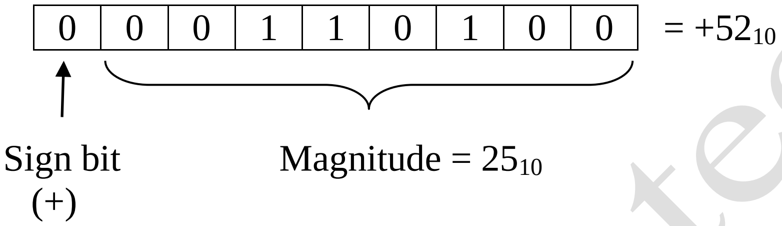
H.W :- add the following pairs of binary numbers

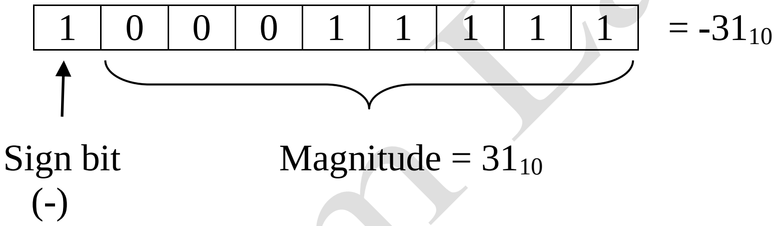
$$\begin{array}{cccccc} & 1 & 0 & 1 & 1 & 0 \\ + & 0 & 0 & 1 & 1 & 1 \\ \hline \end{array} \qquad \begin{array}{ccccccc} & 0 & 1 & 1. & 1 & 0 & 1 \\ + & 0 & 1 & 0. & 0 & 1 & 0 \\ \hline \end{array}$$

$$\begin{array}{ccccccccc} & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline \end{array}$$

Representation Of Signed Numbers :-

In digital computers, the binary numbers are represented by a set of binary storage device (usually flip-flops). Each device represents one bit. For example, a 8-bit FF register could store binary numbers ranging from 0000 0000 to 1111 1111 (**0** to **255**). This represents the magnitude of the number. Since most digital computers calculators handle negative as well as positive numbers, Some means of representing the sign of the number (+ or -). This is usually done by adding another bit as shown below.



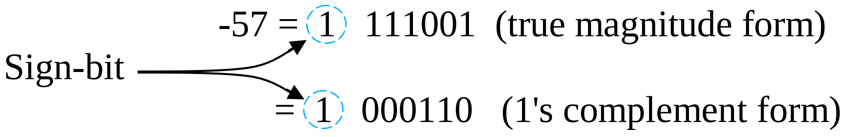


Although this true-magnitude system is straight forward and easy to understand, it is not as useful as two other systems. They are:-

1) 1's complement form.
2) 2's complement form.

1'st complement form :-

The 1's complement form of any binary number is obtained by changing each bit to it's complement for example :



Examples:-

-14 = 10001
-326 = 1010111001
-7 = 1000

2's complement form :-

The 2's complement form of a binary number is formed simply by taking the 1's complement of the number and adding 1 to the least-significant bit position. The procedure is illustrated below.

1	1	1	0	0	1	(decimal 57)
0	0	0	1	1	0	(1's complement)
				+	1	(add 1 to LSB)
0	0	0	1	1	1	(2's complement)

The three systems of representing signed numbers are summarized below:-

True Magnitude System	1's complement system	2's complement system
+57 = 0 111001 -57 = 1 111001	0 000110 1 000110	0 000111 1 000111

Notes:-

1) In al three systems the representation of positive numbers is the same.
2) While all three systems have been used in the past, most modern computers use the 2's complement system.

H.W

a) Represent each of the following signed decimal numbers as a signed binary decimal number in the 2's complement system. Use a total of five bits including the sign bit :

1) +13 2) -9 3) +3 4) -2 5) -8.

b) Each of the following numbers is a signed binary number in the 2's complement system. Determine the decimal value in each case:

1) 01100 2) 11010 3)10001.

Addition in the 2's-Complement system :-

Case1:- Two Positive numbers

The addition of two positive numbers is straight forward. Consider the addition of (+9) and (+4).

(+9)		0	1	0	0	1	(augend)
+(+4)	+	0	0	1	0	0	(addend)
		0	1	1	0	1	(sum= +13)
Sign bit		↑					

Case2:- Positive Number and Smaller Negative Number

Consider the addition of +9 and -4. Remember that the -4 will be in its 2's-complement form.

(+9)		0	1	0	0	1	(augend)
+(-4)	+	1	1	1	0	0	(addend)
	1	0	0	1	0	1	(sum= +5)
discarded	↑						

Case3:- Positive Number and Negative Number

Consider the addition of -9 and +4

(-9)		1	0	1	1	1
+(+4)	+	0	0	1	0	0
		1	1	0	1	1
Negative sign bit		↑				

Case4:- Two Negative Numbers

(-9)		1	0	1	1	1	
+(-4)	+	1	1	1	0	0	
	1	1	0	0	1	1	(sum= -13)
discard	↑						

Case5:- Equal and Opposite Numbers

(-9)		1	0	1	1	1
+(+9)	+	0	1	0	0	1
	1	0	0	0	0	0
discard	↑					

When subtracting one binary number (the subtrahend) from another number (the minuend), the procedure is as follows:

1) Take the 2's of the subtrahend.
2) Add the 2's complement of the subtrahend to the minuend.
3) The two number must have the number of bits.

Example:-

1)

+9		0	1	0	0	1
-(-4)	+	0	0	1	0	0
+13		0	1	1	0	1

2)

-9		1	0	1	1	1
-(+4)	+	1	1	1	0	0
-13	1	1	0	0	1	1
discard	↑					

3)

+8		0	1	0	0	0
-(-4)	+	0	0	1	0	0
+12	0	0	1	1	0	0

4)

-7		1	1	0	0	1
-(-3)	+	0	0	0	1	1
-4		1	1	1	0	0

Subtraction in the 1's-Complement:-

9 1001
-6 0110 to 1's 1001
=3 to add with 9

1001
1001
10010
(+) → 1
0011

Last digit summation with least digit

12 1100
-7 1000 7 in 1's com
5 10100
(+) → 1
0101

Multiplication of Binary Numbers :-

The multiplication of binary numbers is done in the same manner as multiplication of decimal numbers.

Example:-

			1	0	0	1	(Multiplicand)
			1	0	1	1	(Multiplier)
			1	0	0	1	
		1	0	0	1		
	0	0	0	0			
1	0	0	1				
1	1	0	0	0	1	1	

In practical case the procedure of multiplication is as follows :-

$$
\begin{array}{ccccccc l}
 & & & 1 & 0 & 0 & 1 & \text{First partial product} \\
 & & 1 & 0 & 0 & 1 & & 2^{nd}\text{ partial product} \\
\cline{3-7}
 & & 1 & 1 & 0 & 1 & 1 & \text{Sum of first two partial products} \\
 & 0 & 0 & 0 & 0 & & & \text{Third partial product} \\
\cline{2-7}
 & 0 & 1 & 1 & 0 & 1 & 1 & \text{Sum of first three partial products} \\
1 & 0 & 0 & 1 & & & & \text{Fourth partial product} \\
\cline{1-7}
1 & 1 & 0 & 0 & 0 & 1 & 1 & \text{Final result}
\end{array}
$$

Multiplication in the 2's complement system :-

For a signed numbers, the multiplication process is same as above with the following points:

1) If the two numbers are positive, the multiplication process is straight forward and the result is positive.
2) If the two numbers are negative, they will be in 2's complement form. Change two numbers to positive form and the result will be positive.
3) If one of the two numbers is negative and it will be in 2's complement form, change it to positive and take the 2's complement of the result.

Example: - Multiply (0111) and (1110).

$$
\begin{array}{ccccccc}
 & & & 0 & 1 & 1 & 1 \\
 & & & 1 & 1 & 1 & 0 \\
\cline{4-7}
 & & & 0 & 0 & 0 & 0 \\
 & & 0 & 1 & 1 & 1 & \\
\cline{3-7}
 & & 0 & 1 & 1 & 1 & 0 \\
 & 0 & 1 & 1 & 1 & & \\
\cline{2-7}
 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & & & \\
\cline{1-7}
1 & 1 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

Example:-

+9				0	1	0	0	1		
x (-4)				1	0	1	0	0		
-36					0	0	0	0		
				0	0	0	0			
			1	0	0	1				
		0	0	0	0					
	1		1	0	0	1	0	0	$\xrightarrow{2's}$	1 01100

Binary Division :-

The binary division is the same as decimal division in long division form.

Example:-

1001 (dividend)
÷ 011 (divisor)

			0	0	1	1
0	1	1	1	0	0	1
			0	1	1	
			0	0	1	1
			0	0	1	1
			0	0	0	0

In most modern digital machines the division process is performed by repeat subtraction process.

1	0	0	1	
1	1	0	1	①
0	1	1	0	
1	1	0	1	②

$$\begin{array}{ccccc} \cline{2-5} \cancel{1} & 0 & 0 & 1 & 1 \\ & 1 & 1 & 0 & 1 \\ \cline{2-5} \cancel{1} & 0 & 0 & 0 & 0 \end{array}$$ ③

Binary Subtraction :-

0 – 0 = 0
1 – 0 = 1
$10_2 - 1 = 10_2$ 10_2 minus 1 equals 1

Example:-

$$\begin{array}{cccc} & 0 & 1 & 1 \\ - & 0 & 0 & 1 \\ \hline & 0 & 1 & 0 \end{array} \quad \begin{array}{c} 3 \\ -1 \\ \hline 2 \end{array} \qquad \begin{array}{cccc} & 0 & 1 & 1 \\ - & 0 & 1 & 0 \\ \hline & 0 & 0 & 1 \end{array} \quad \begin{array}{c} 3 \\ -2 \\ \hline 1 \end{array}$$

$$\begin{array}{cccc} & 1 & 0 & 1 \\ - & 0 & 1 & 1 \\ \hline & 0 & 1 & 0 \end{array} \quad \begin{array}{c} 5 \\ -3 \\ \hline 2 \end{array}$$

BCD Addition

Sum Equals Nine Or Less :-

The addition is carried out as in normal binary addition.

Example:-

$$\begin{array}{cc} 4 & 3 \\ 3 & 4 \\ \hline 7 & 7 \end{array} \qquad\qquad \begin{array}{cccccccc} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{array}$$

Sum Greater than Nine :-

$$\begin{array}{cc} 5 & 9 \\ 3 & 8 \\ \hline 9 & 7 \end{array} \qquad\qquad \begin{array}{cccccccc} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ & & & & 0 & 1 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}$$

Hexadecimal Arithmetic :-

1- Addition:-

For Hex numbers addition the following procedure is suggested

- Add the two hex digits in decimal.
- If the sum15 or less, it can be directly expressed as a hex digit.
- If the sum is greater than or equal to 16, subtract 16 and carry a 1 to the next digit position.

Example:-

$$\begin{array}{ccc} & 5 & 8 \\ + & 2 & 4 \\ \hline & 7 & C \end{array} \qquad\qquad \begin{array}{ccc} & 5 & 8 \\ + & 4 & B \\ \hline & A & 3 \end{array} \qquad\qquad \begin{array}{cccc} & 3 & A & F \\ + & 2 & 3 & C \\ \hline & 5 & E & B \end{array}$$

2- Subtraction:-

Subtraction of hex numbers is performed in the same way as binary numbers. There are two ways to find 2's complement of hex numbers.

a) Convert the number to binary form and take the 2's complement then convert it back to hex form.

Example:-

Find 2's complement of (73A)

7	3	A
0 1 1 1	0 0 1 1	1 0 1 0
1 0 0 0	1 1 0 0	0 1 0 1
		1
1 0 0 0	1 1 0 0	0 1 1 0
8	C	6

b) Subtract each hex digit from F, then add 1.

Example :-

F	F	F
-7	-3	-A
8	C	5
		1
8	C	6

9'S AND 10's COMPLEMENT

9'S Complement:

The 9'S complement of decimal numbers is found by subtraction of each of the decimal is as follows:

Digit	9'S complement
0	9
1	8
2	7
3	6
4	5
5	4
6	3
7	2
8	1
9	0

Find the 9'S complement of each decimal numbers: 28, 56, and 115

$$\begin{array}{rrr} & 9 & 9 \\ - & 2 & 8 \\ \hline & 7 & 1 \end{array} \qquad \begin{array}{rrr} & 9 & 9 \\ - & 5 & 6 \\ \hline & 4 & 3 \end{array} \qquad \begin{array}{rrrr} & 9 & 9 & 9 \\ - & 1 & 1 & 5 \\ \hline & 8 & 8 & 4 \end{array}$$

9'S complement

9'S Complement subtraction:

$$\begin{array}{rr} & 8\\ - & 3\\ \hline & 5\end{array} \qquad \begin{array}{rrl} & 8 & \\ + & 6 & \\ \hline \textcircled{1} & 4 & \\ + & 1 & \text{Add carry to result}\\ \hline & 5 & \end{array}$$

$$\begin{array}{rrr} & 1 & 3\\ & - & 7\\ \hline & & 6\end{array} \qquad \begin{array}{rrrl} & 1 & 3 & \\ + & 9 & 2 & \\ \hline \textcircled{1} & 0 & 5 & \\ & + & 1 & \text{Add carry to result}\\ \hline & & 6 & \end{array}$$

10'S Complement:

The 10'S complement of decimal number is equal to the 9'S complement plus 1.

Complement the following decimal number to their 10'S complement from: 8, 428

Solution:-

$$\begin{array}{rrl} & 9 & \\ - & 8 & \\ \hline & 1 & \text{9'S complement of 8 add 1}\\ + & 1 & \\ \hline & 2 & \text{10'S complement of 8}\end{array}$$

$$\begin{array}{rrrrl} & 9 & 9 & 9 & \\ - & 4 & 2 & 8 & \\ \hline & 5 & 7 & 1 & \text{9'S complement of 428}\\ + & & & 1 & \\ \hline & 5 & 7 & 2 & \text{10'S complement of 428}\end{array}$$

10'S complement subtraction:

Regular Subtraction — 10'S complement subtraction

$$\begin{array}{rr} & 8\\ - & 3\\ \hline & 5\end{array} \qquad \begin{array}{rrl} & 8 & \\ + & 7 & \text{10'S complement subtraction}\\ \hline \textcircled{1} & 5 & \text{Drop carry}\end{array}$$

Arithmetic circuits design :-

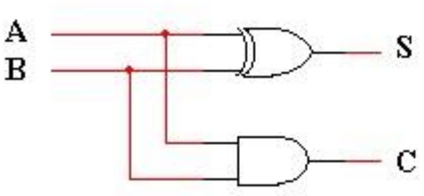
1) Half-Adder

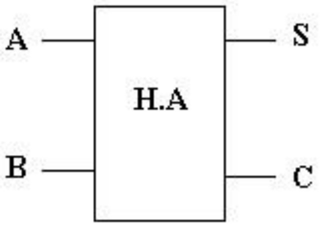
The half-adder circuit has two inputs and produce sum(S) and carry (C) as shown below.

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = A \oplus B$$

$$C = A.\,B$$





2) Full-Adder

The full-adder circuit has three inputs (two inputs represent numbers and other input represent carry from previous stage).

A	B	C_{in}	S	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

S

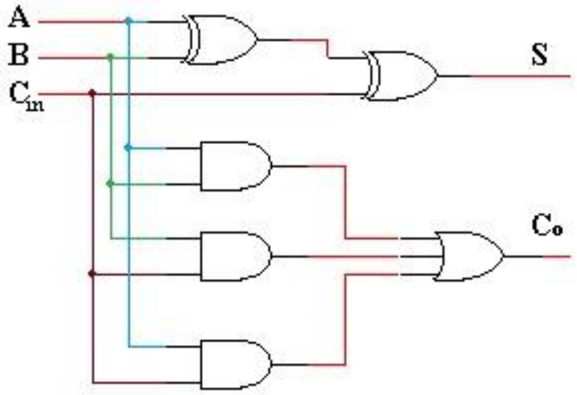
	$\overline{C_{in}}$	C_{in}
$\bar{A}\bar{B}$	0	1
$\bar{A}B$	1	0
AB	0	1
$A\bar{B}$	1	0

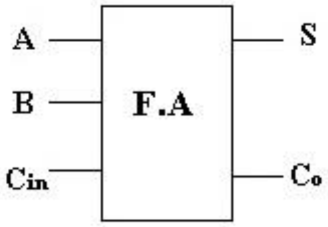
$$S = \bar{A}B\overline{C_{in}} + A\bar{B}\overline{C_{in}} + \bar{A}\bar{B}C_{in} + ABC_{in}$$
$$S = \overline{C_{in}}(\bar{A}B + A\bar{B}) + C_{in}(\bar{A}\bar{B} + AB)$$
$$S = C_{in} \oplus (A \oplus B)$$

C_o

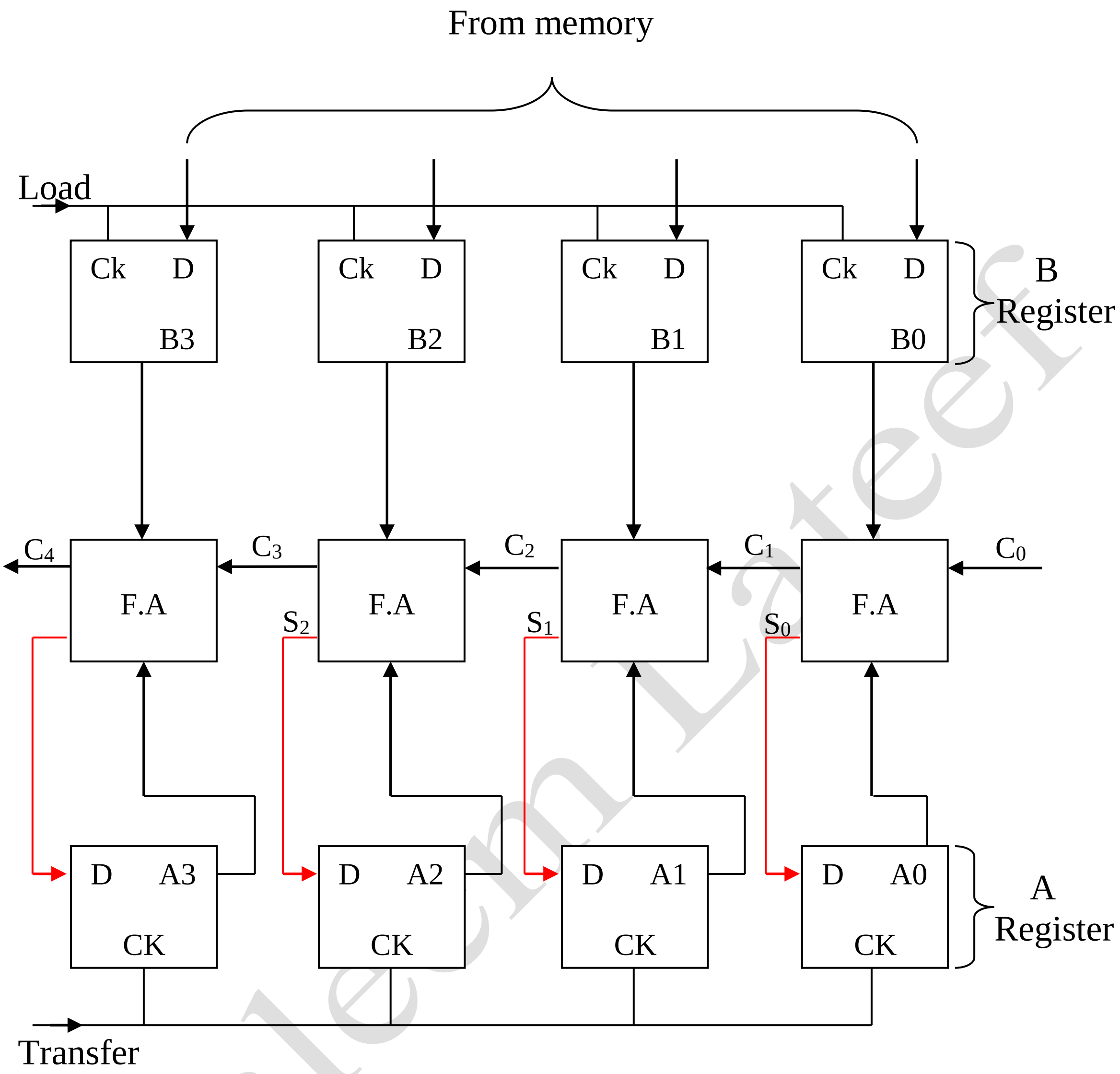
	$\overline{C_{in}}$	C_{in}
$\bar{A}\bar{B}$	0	0
$\bar{A}B$	0	1
AB	1	1
$A\bar{B}$	0	1

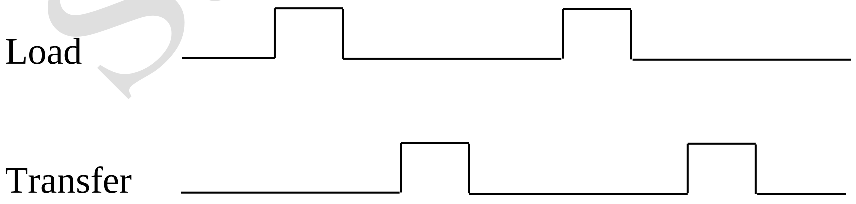
$$C_o = AB + BC_{in} + AC_{in}$$





Complete 4-bit parallel adder :-





Circuit Principle :-

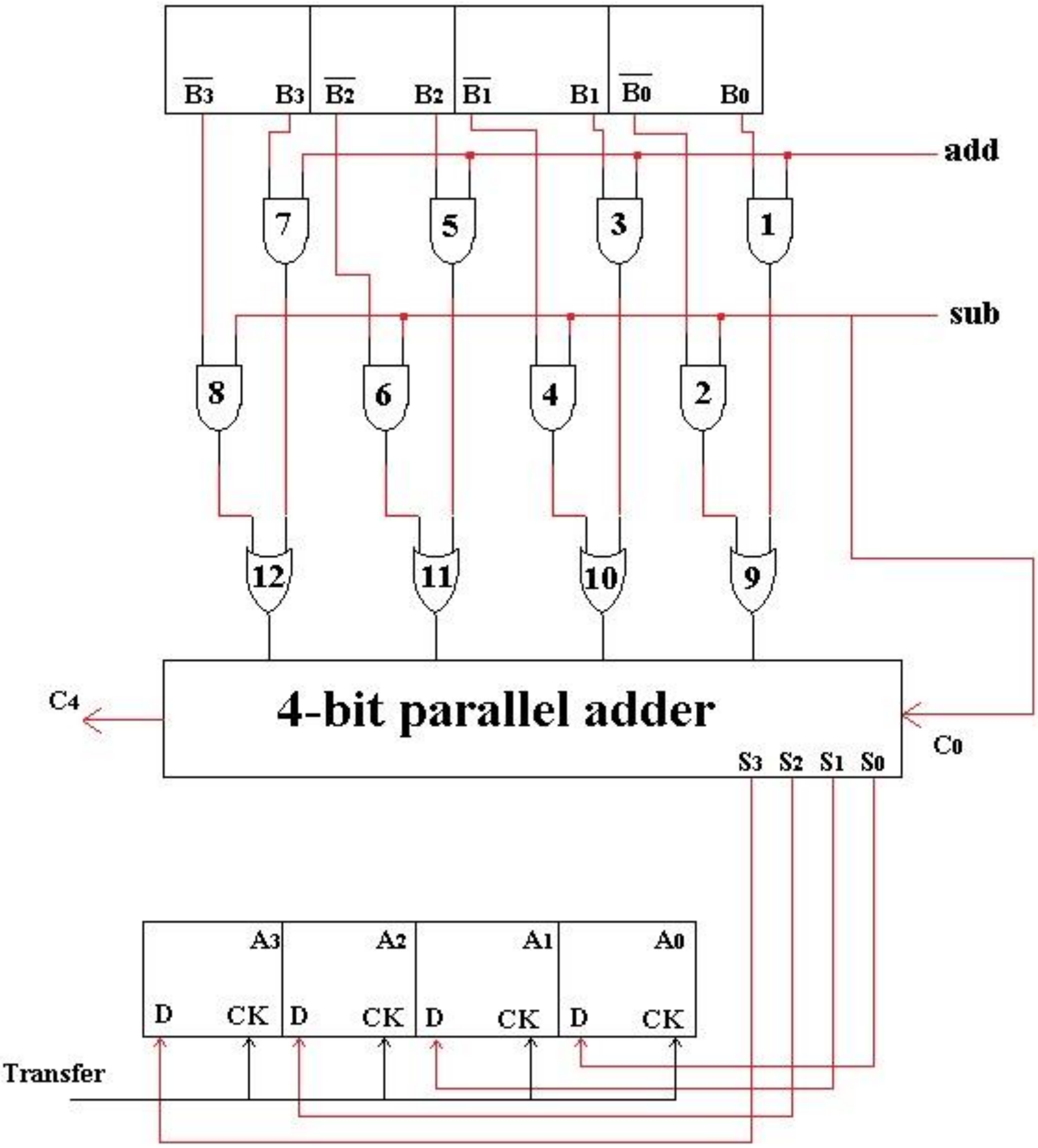
1) The contents of the A register (i.e, the binary number stored in A3-A0 is added to the contents of the B register by the four Fas, and the sum is produced at outputs (S3-S0), C4 is the carry out of the fourth FA.
2) The sum (S3-S0) is stored in the A register on the PGT of the Transfer pulse.

Sequence of operations :-

We will now describe the process by which the circuit will add the binary numbers 1001 and 0101. Assume C_0=0 and A=0000.

1- A=0000
2- M → B the B=1001 on the PGT of load.
3- The full-adders produce a sum of 1001 [S=1001].
4- The S → A [A=1001] on the PGT of transfer.
5- M → B the B=0101 on the PGT of load.
6- The full-adders produce a sum of 1110 [S=1110].
7- The S →A [A=1110] on the PGT of transfer.

2's complement circuit (Adder-Subtractor)



The operation of this circuit is described as follows :-

1- Assume that add=1 and sub=0. AND gates 2,4,6 and 8 outputs will be 0. AND gates 1,3,5,7 allowing their outputs to pass the B_0,B_1,B_2 and B_3 levels, respectively.
2- The B_0-B_3 levels pass through the OR gates into the 4-bit parallel adder to be added to the A_0-A_3 bits. The sum appears at the S_0-S_3 outputs.
3- Note that sub=0 causes C_0=0 into the adder.
4- Now assume that add=0 and sub=1, AND gates 1,3,5 and 7 will pass $\overline{B_0}, \overline{B_1}, \overline{B_2}$ and $\overline{B_3}$ to their outputs respectively.
5- The $\overline{B_0} - \overline{B_3}$ levels pass through the OR gates into the adder to be added to the A_0-A_3 bits. Note also that C_0 is now 1. Thus, the B-register number has essentially been 2's-complement.
6- The difference appears at the S_0-S_3 outputs.

BCD Adder :-

The BCD addition rules is discussed previously and reviewed here.

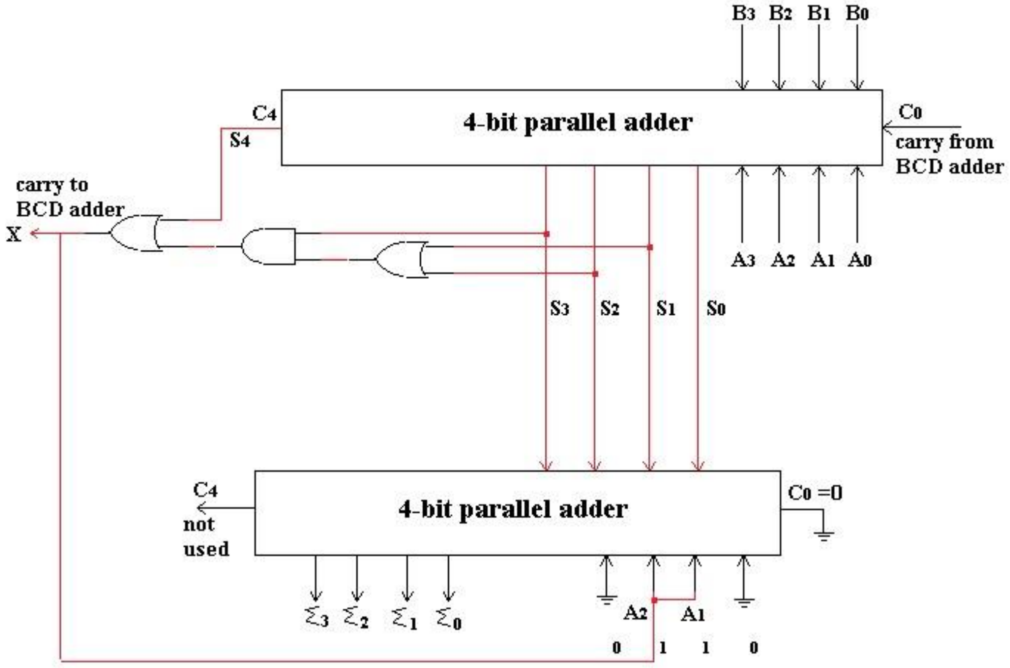
1) If the result is less or equal 9(1001) leave it alone.
2) If the result is greater than 9(1001) add 0110 to the result.

Assume we have the result of five bits S_0 S_1 S_2 S_3 S_4. The cases need to be corrected are :-

1) Whenever S_4 =1
2) Whenever S_3 =1 and either S_2 or S_1 or both are 1.

S4	S3	S2	S1	S0	
0	1	0	1	0	(10)
0	1	0	1	1	(11)
0	1	1	0	0	(12)
0	1	1	0	1	(13)
0	1	1	1	0	(14)
0	1	1	1	1	(15)
1	0	0	0	0	(16)
1	0	0	0	1	(17)
1	0	0	1	0	(18)

$$X=S_4+S_3(S_1+S_2)$$



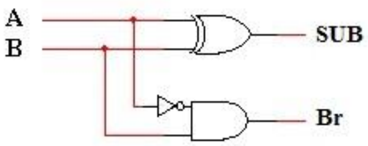
Half Subtractor :-

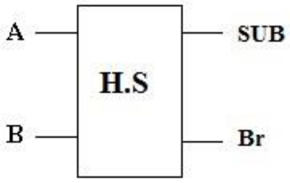
In a similar fashion subtraction can be performed using binary numbers. The truth table for a single bit or half_subtractor with inputs A and B given below along with it's circuit diagram where outputs are SUB and Br where SUB is the subtraction result and Br is the borrow from subtraction process.

A	B	SUB	Br
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$SUB = A \oplus B$$

$$Br = \overline{A} . B$$





It's noticed that the SUB output is similar to the half adder output, the only difference is that between carry and borrow between the two logical circuits.