UNIT III
COMBINATIONAL LOGIC CIRCUITS-1

Objective:

To familiarize the design of combinational logic circuits: adders and subtractors.

Syllabus:

Design of half adder, full adder, half subtractor, full subtractor, ripple adders and subtractors, ripple adder/subtractor using 1's and 2's complement method, serial adder.

Outcomes:

At the end of the unit, Students will be able to

- Design half adder and full adder.
- Design half subtractor and full subtractor.
- Design ripple adders and subtractors.
- Design serial adder.

Learning Material

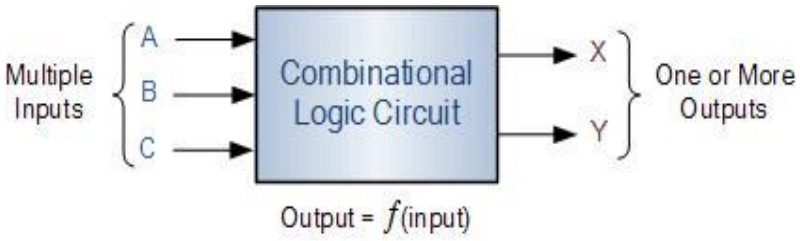
Digital circuits can be classified into two types:

1. Combinational digital circuits and
2. Sequential digital circuits.

COMBINATIONAL LOGIC CIRCUITS:

- Combination Logic Circuits are made up from basic gates (AND, OR, NOT) or universal gates (NAND, NOR) gates that are "combined" or connected together to produce more complicated switching circuits.
- These logic gates are the building blocks of combinational logic circuits.
- An example of a combinational circuit is a decoder, which converts the binary code data present at its input into a number of different output lines, one at a time producing an equivalent decimal code at its output.
- In these circuits "the outputs at any instant of time depends on the inputs present at that instant only." •
- For the design of Combinational digital circuits Basic gates (AND, OR, NOT) or universal gates (NAND, NOR) are used.

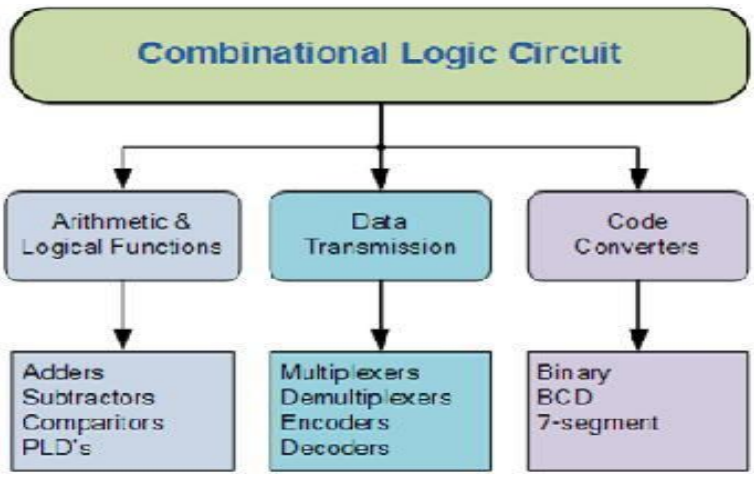
- Examples for combinational digital circuits are Half adder, Full adder, Half subtractor, Full subtractor, Code converter, Decoder, Multiplexer, Demultiplexer, Encoder, ROM, etc



SEQUENTIAL LOGIC CIRCUITS:

- Sequential logic differs from combinational logic in that the output of the logic device is dependent not only on the present inputs to the device, but also on past inputs;
- i.e., the output of a sequential logic device depends on its present internal state and the present inputs.
- This implies that a sequential logic device has some kind of memory of at least part of its "history" (i.e., its previous inputs).
- Examples for sequential digital circuits are Registers, Shift register, Counters etc.

Classification of Combinational Logic circuits



- **Representing Combinational Logic Functions:**

  ❖ The three main ways of specifying the function of a combinational logic circuit are:

  1. Boolean function– This forms the algebraic expression (Boolean Function) showing the operation of the logic circuit for each input variable either True or False that results in a logic "1" output.
  2. Truth Table – A truth table defines the function of a logic gate by providing a concise list that shows all the output states in tabular form for each possible combination of input variable that the gate could encounter.
  3. Logic Diagram – This is a graphical representation of a logic circuit that shows the wiring and connections of each individual logic gate, represented by a specific graphical symbol that implements the logic circuit.

- **To obtain the output Boolean functions from a logic diagram:**
  - Label all gate outputs that are a function of input variables.
  - Determine the Boolean functions for each gate output.
  - Repeat the process until the outputs of the circuit are obtained.
  - By repeated substitution, obtain the output Boolean function in terms of input variables.
- **To obtain the truth table from logic diagram:**
  - Determine the number of input variables in the circuit. For n inputs, form the 2^n possible input combinations and list the binary numbers from 0 to (2^n-1) in a table.
  - Label the outputs of selected gates with arbitrary symbols.
  - Obtain the truth table for the outputs of those gates which are a function of the input variables only.
  - Proceed to obtain the truth table for the outputs of these gates which are a function of previously defined values.
- **To design a logic diagram:**
  - Determine the required number of inputs and outputs and assign a symbol to each.
  - Derive the truth table that defines the required relationship between inputs and outputs.

- Obtain the simplified Boolean function for each output.
- Draw the logic diagram.

COMBINATIONAL LOGIC CIRCUIT DESIGN PROCEDURE:

Input: the specification of the problem.

Output: the logic circuit diagram (or Boolean functions).

Step 1: Determine the required number of inputs and outputs from the specification.

Step 2: Derive the truth table that defines the required relationship between inputs and outputs.

Step 3: Obtain the simplified Boolean function for each output as a function of the input variables.

Step 4: Draw the logic diagram and verify the correctness of the design.

COMBINATIONAL LOGIC CIRCUIT ANALYSIS PROCEDURE

Analysis: Determine the function that the circuit implements.

- Often start with a given logic

diagram. The analysis can be performed by:

- Finding Boolean functions
- Finding truth table !
- Without feedback paths or memory elements.

Second step: obtain the output Boolean functions or the truth table.

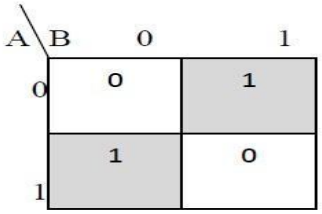
DESIGN OF HALF ADDER

- Simplest combinational logic circuit.
- Half adder is a combinational logic circuit with two inputs and two outputs.
- The half adder circuit is designed to add two single bit binary numbers A and B.
- It is the basic building block for addition of two single bit numbers.
- This circuit consists of two gates, an XOR gate that produces a logic 1 output whenever A is 1 and B is 0, or when B is 1 and A is 0. The AND gate produces a logic 1 at the carry output when both A and B are 1.
- This circuit has two outputs **carry and sum.**
- As the half adder has only two inputs it cannot add in a carry bit from a previous column, so it is not practical for anything other than 1-bit additions.

Truth table for Half adder

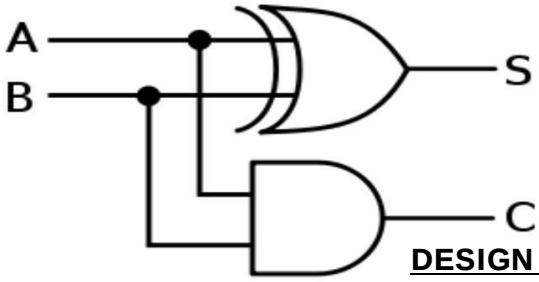
INPUTS		OUTPUTS	
A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

The simplified Boolean functions are obtained as follows:
These can be obtained from the truth table.



$S=A_1B+AB_1$
C= AB

Circuit:



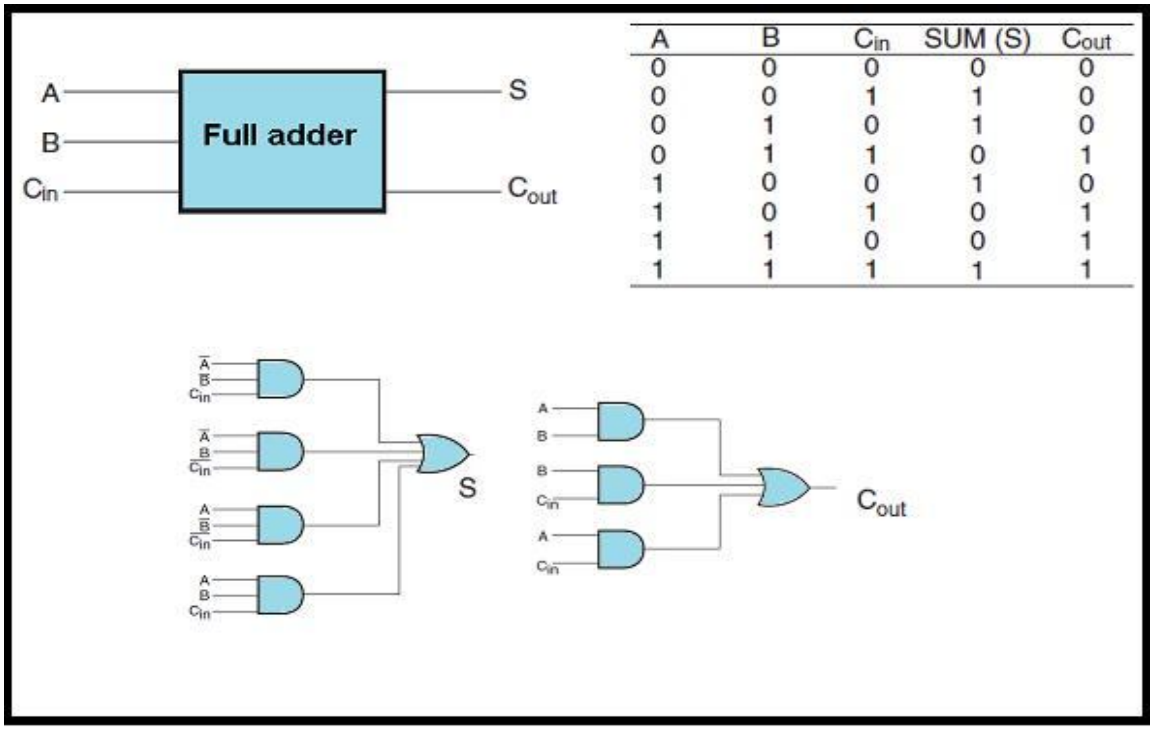
Block Diagram:

DESIGN OF FULL ADDER

- Full adder is developed to overcome the drawback of Half Adder circuit.
- It is a combinational circuit that performs the arithmetic sum of three bits.
- Two of the input variables, denoted by A and B represent the two significant bits to be added.
- The third input C_{in} represents the carry from the previous lower significant position.
- The full adder is a three input and two output combinational circuit.
- The two outputs are sum(S) and carry (C_0).
- Two outputs are necessary because the arithmetic sum of three digits ranges in value from 0 to 3 and the binary representation of 2 or 3 requires two bits.
- The binary variable S gives the value of the least significant bit of the sum. The binary variable C gives the output carry formed by adding the put carry and the bits of the words.
- The eight rows under the input variables designate all possible combinations of the three variables. The output variables are determined from the arithmetic sum of input bits.

- When all the input bits are 0, the output is 0. The S is equal to 1 when only one input is equal to 1 or when all the three inputs are equal to 1.
- The C output has a value of 1 if two or three inputs are equal to 1

Block diagram & Truth Table:



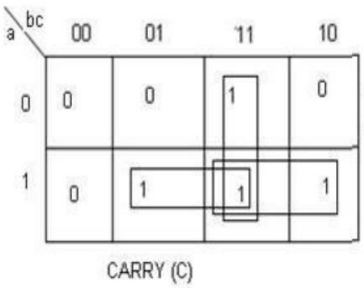
A	B	C_{in}	SUM (S)	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

The simplified Boolean functions are obtained as follows:

$S=A_1B_1C+A_1BC_1+AB_1C_1+ABC$

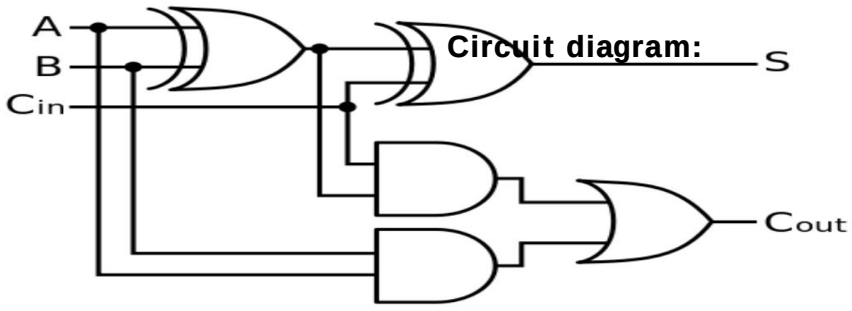
A \ BC_{IN}	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$C_0= AB+BC+AC$



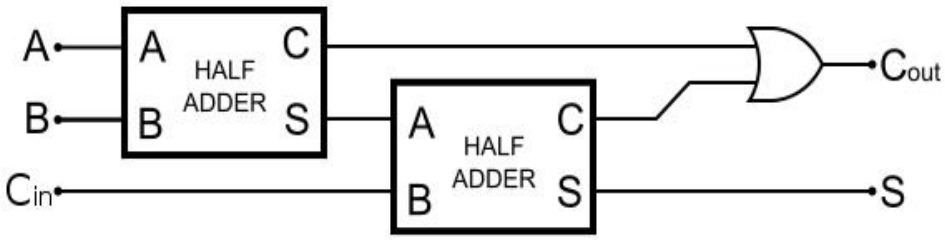
The equation we get for variable CARRY C is C = ab + ac + bc

Circuit diagram:



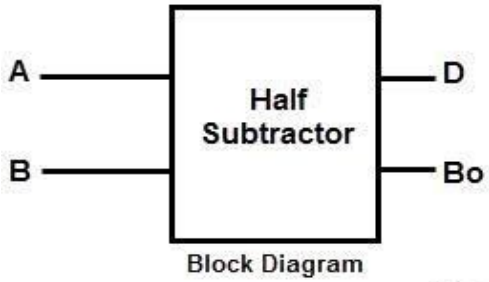
Implementation of full adder with two half adders:

- A full-adder can be constructed from two half-adders and an OR gate, as shown in Figure



HALF SUBTRACTOR

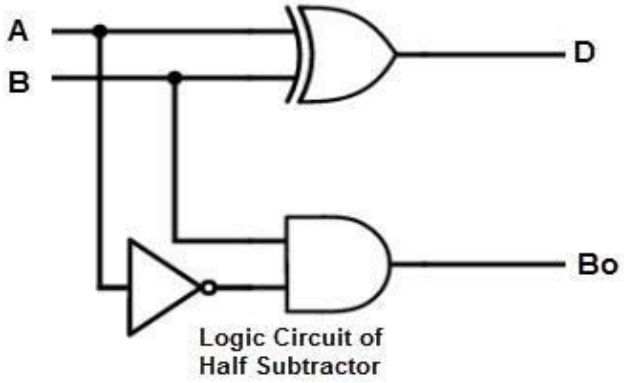
- Half subtractor is a combination circuit with two inputs and two outputs i.e. **Difference** and **Borrow**.
- It produces the difference between the two binary bits at the input and also produces an output borrow to indicate if a 1 has been borrowed.
- In the subtraction A − B, A is called as Minuend bit and B is called as Subtrahend bit.
- D is the difference in output and B_0 is the borrow output.



Block Diagram

A	B	D	B_o
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Truth Table



Logic Circuit of Half Subtractor

Difference $D= A_1B+AB_1=A \oplus B$

Borrow $B_0=A_1B$

Limitations of half subtractor:

- In multi-bit subtraction, we have to subtract bit along with the borrow of the previous digit subtraction. Such subtraction requires subtraction between three bits, which is not possible with half-subtractor.

FULL SUBTRACTOR

- The disadvantage of a half subtractor is rectified by full subtractor.
- The full subtractor is a combinational circuit with three inputs A, B, C and two outputs Difference(D) and Borrow(C_1). A is the 'minuend', B is 'subtrahend', C is the 'borrow' produced by the previous stage, D is the difference output and C_1 is the borrow output.

Full Subtractor-Truth Table				
Input			Output	
A	B	C	Difference	Borrow
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

From the Truth Table The Difference and Borrow will be written as

Difference=$A_1B_1C+A_1BC_1+AB_1C_1+ABC$ If we reduce it, we get:

Difference=A $\oplus$B $\oplus$C

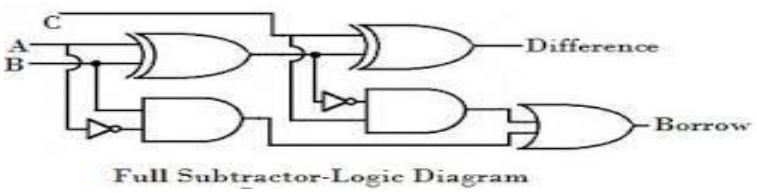
Borrow=$A_1B_1C+A_1BC_1+A_1BC+ABC$

=$A_1B_1C+A_1BC_1+A_1BC+A_1BC+A_1BC+ABC$ = $A_1BC=A_1BC+A_1BC+A_1BC$

=$A_1C(B_1+B)+A_1B(C_1+C)+BC(A_1+A)$

Borrow=A_1C+A_1B+BC

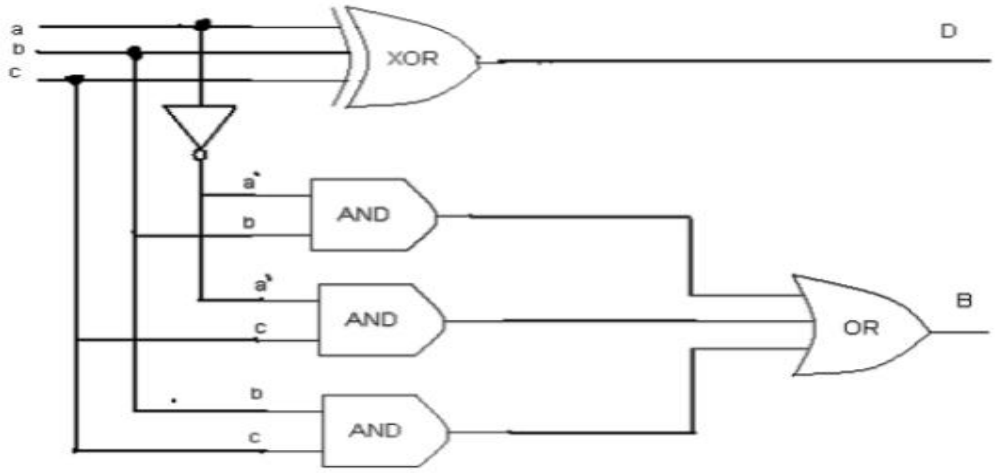
The logic diagram of Full Subtractor is :



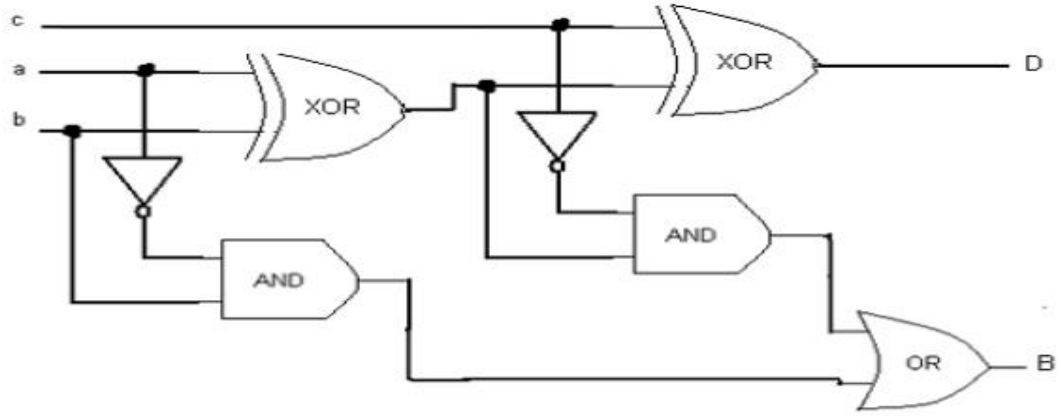
Full Subtractor-Logic Diagram

Implementation of full-subtractor:

- The simplified Boolean function for a full subtractor is implemented as follows



Implementation of full subtractor from two half subtractors:

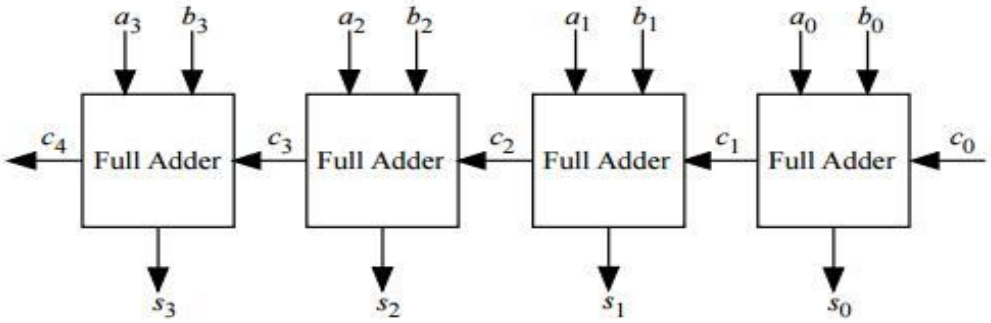


RIPPLE CARRY (PARALLEL) ADDER

- A Ripple carry adder is a digital circuit that produces the arithmetic sum of two binary numbers
- It can be constructed with full adders connected in cascaded, with the carry output from each full adder connected to the carry input of the next full adder in the chain.
- Addition of n-bit numbers requires a chain of n-full adders or a chain of one half adder and (n-1) full adders.

- In the below figure, the input carry to the least significant position is fixed at 0.

- It is an interconnection of four full adders (FA) circuits to provide four bit binary ripple adder.
- The augend bits of A and addend bits of B are designated by subscript numbers from right to left, with subscript 0 denoting the least significant bit.
- The carries are connected in a chain through the full adders. The input carry to the adder is C_0, and it ripples through the full adders to the output carry C_4.
- The S output generates the required sum bits.
- The bits are added with full adders from the least significant position to form the sum bit and the carry bit.
- The input carry C_0 is 0. The value C_{i+1} in a given significant position is the output carry of the full adder.
- This value is transferred into the input carry of the full adder that adds the bit next higher significant position towards the left.
- The sum bits are thus generated from the rightmost position and are available as soon as the previous carry is generated.
- All the carries must be generated for the correct sum bits to appear at the outputs.



Assume you want to add two operands A and B where

A= A3 A2 A1 A0

B= B3 B2 B1 B0

For example: A= 1 0 1 1

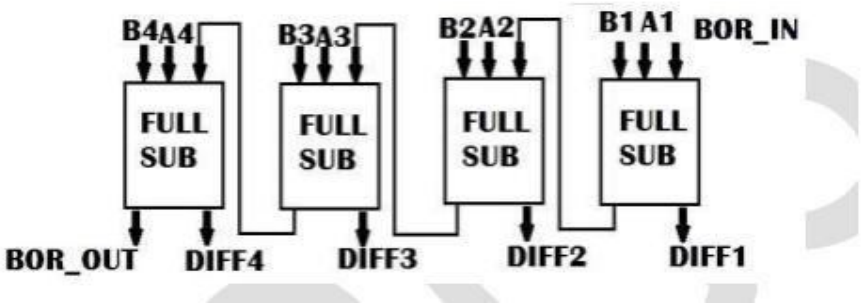
+ B= 1 1 0 1

A+B= 11 0 0 0 = C_{out} S3 S2 S1 S0

From the example above it can be seen that we are adding 3 bits at a time sequentially until all bits are added. A full adder is a combinational circuit that performs the arithmetic sum of three input bits: augends Ai, addend Bi and carry in C_{in} from the previous adder. Its results contain the sum Si and the carry out, C_{out} to the next stage.

RIPPLE BORROW (PARALLEL) SUBTRACTOR

- It is possible to create logical circuit using multiple full subtractor to subtract N(4)bit numbers.
- Each full subtractor inputs a borrow_in (borrow input) which is the borrow_out (borrow output) of previous subtractor.
- This kind of subtractor is ripple borrow subtractor since each borrow bit ripples to the next full subtractor.

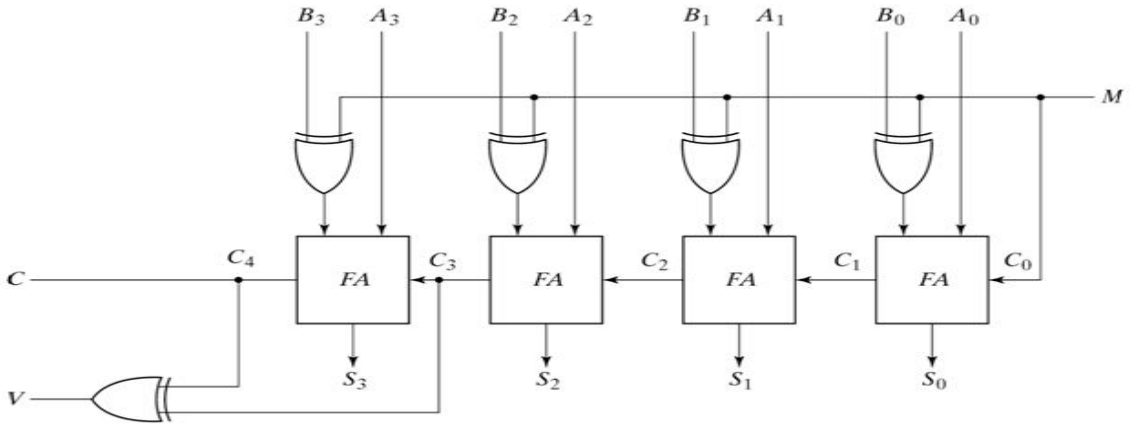


RIPPLE BORROW SUBTRACTOR

RIPPLE (PARALLEL) ADDER/ SUBTRACTOR

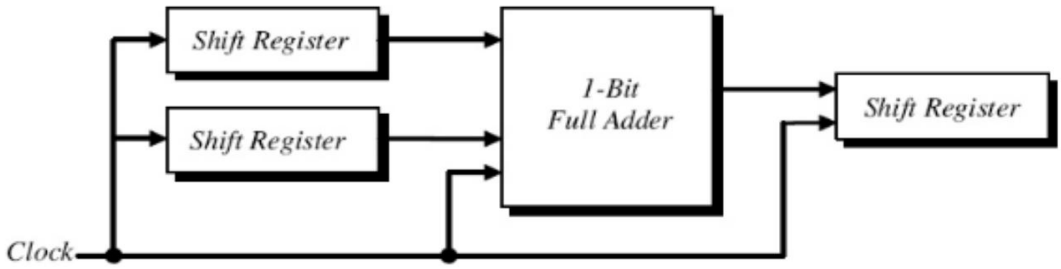
- The subtraction of unsigned binary numbers can be done by means of complements. The subtraction of A-B can be done by taking the 2's complement of B and adding it to A.
- The circuit for subtracting A-B consists of an adder with inverters placed between each data input B and the corresponding input of the full adder.
- The input carry C_0 must be equal to 1 when the subtraction is performed.

- The operation thus performed becomes A, plus the 1's complement of B, plus 1. This is equal to A plus the 2's complement f B.
- The addition and subtraction operations can be combined into one circuit with one common binary adder by including an exclusive OR gate with each full adder.
- The mode input M controls the operation. When M=0, the circuit is an adder and when M=1, the circuit becomes a subtractor.
- Each exclusive-OR gate receives input M and one of the inputs of B. When M=0, we have B(Ex-OR) 0=B.
- The full adders receive the value of B, the input carry is 0, the circuit performs A plus B.
- When M=1, we have B(Ex-OR) 1=B_1 and C_0=1.
- The B inputs are all complemented and a 1 is added through the input carry.
- The circuit performs the operation A plus the 2's complement of B.
- When two numbers with n digits are added and the sum is n+1 digits, we say that an overflow occurred.
- An overflow condition can be detected by observing the carry into the sign bit position and carry out of the sign bit position.
- If these two carries are not equal an overflow has occurred.
- An overflow is detected from the end carry out of the most significant position, when two unsigned numbers are added.
- An overflow cannot occur after an addition if one number is positive and other is negative.



SERIAL ADDER

- The **serial** binary **adder** or bit-**serial adder** is a digital circuit that performs binary addition bit by bit. The **serial** full **adder** has three single-bit inputs for the numbers to be added and the carry in. There are two single-bit outputs for the sum and carry out.
- Sequential serial adders are economically efficient and simple to build.
- A serial adder consists of a 1-bit full-adder and several shift registers.
- In serial adders, pairs of bits are added simultaneously during each clock cycle.
- Two right-shift registers are used to hold the numbers (*A* and *B*) to be added, while one left-shift register is used to hold the sum (*S*).
- Parallel adder perform the adding two bit operation very fast but the disadvantage of this adder is its require large number of gate. One the other hand in serial adder the bit addition is bit-by-bit. Serial adder require simple circuitry as compare to parallel adder , so causes of simple circuitry this give low speed and perform bit-by-bit operation.
- This is the one which would accept bit by bit input of the n-bit numbers and there is a bit by bit output of the n-bit Sum. In this adder we would be required one full adder and a memory element.
- Here lesser hardware is required. The circuit for serial addition is as follow:

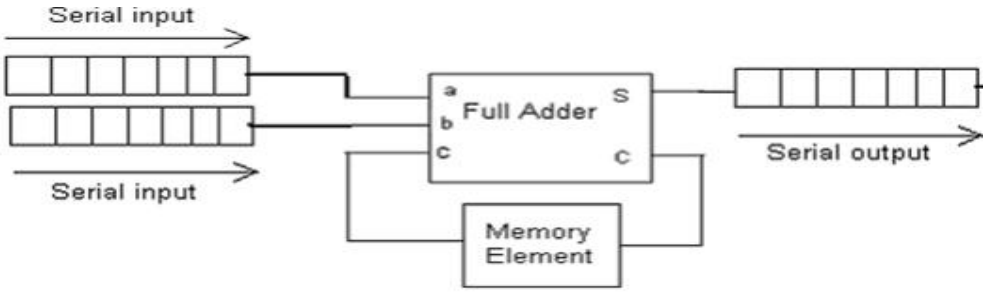


A

A	B	S	s_i	c_{i+1}
1011	0011	0000	0	1
0101	0001	1000	1	1
0010	0000	1100	1	0
0001	0000	1110	1	0

Finite-state machine adder performs the addition operation on the values stored in the input shift registers and stores the sum in a separate shift register during several clock cycles.

- During each clock cycle, two input bits a_i and b_i are shifted from the two input right-shift registers into the 1-bit full-adder, which adds the two bits and evaluates the sum bit s_i and the carryout bit c_{i+1}.
- The sum bit s_i, is shifted out to the left-shift register and the carryout bit c_{i+1} is stored in the state memory of the serial adder for the next two bits.
- The time sequence of the operation of a 4-bit serial adder is illustrated in the above figure.



Difference between Serial and parallel adders:

Basically, the difference between them is that:-

Serial Adder:

a) It is basically slow

b) It uses shift registers

c) It needs one full adder circuit.

d) It is sequential circuit.

e) Time needs for addition depends on number of bits.

Parallel adder:

1) It is faster than serial adder

2) It uses registers with parallel load capacity

3) No. of full adder circuit is actually equal to no. of bits in binary adder.

4) It is considered as a combination circuit

5) In this case, time does not depend on the number of bits.

UNIT-III
Assignment-Cum-Tutorial Questions

SECTION-A

Objective Questions

1. Combinational circuit has: [ ]

   a) memory b) no memory c) flip-flops d) counters

2. Simplified expression of half adder carry is [ ]

   a) c=xy+x b) c=y+x c) c=xy+y d) c=xy

3. Full adder performs sum of [ ]

   a) 2 bits b) 3 bits c) 4 bits d) 5 bits

4. Which are the fundamental inputs assigned or configured in the full adder circuit ? [ ]

   a. Addend, Augend & Sum b. Augend, Sum & Input Carry

   c. Addend, Augend & Input Carry d. Addend, Sum & Input Carry

5. Full subtractor is a---------- [ ]

   A. combinational circuit B. Sequential circuit

   C. combinational sequence D. series

6. Most significant bit of arithmetic addition is called [ ]

   a) Overflow b) carry c) output d) zero bit

7. Simplified expression of full adder carry is [ ]

   a). c=xy+xz+yz b). c=xy+xz c). c=xy+yz d). c=x+y+z

8. Two bit subtraction is done by [ ]

   a) demux b) mux c) full subtractor d) half subtractor

9. A half adder circuit does not have_____ [ ]

   a) carry in b) carry out c) Two inputs d) all the above

10. The output of combinational circuit depends only on present input. [T/F]

11. One way to make a four-bit adder perform subtraction is by: [ ]

a) Inverting the output. b) Inverting the carry-in.

c) Inverting the B inputs. d) Grounding the B inputs.

12. The number of full and half adders required to perform 16-bit addition is: [ ]

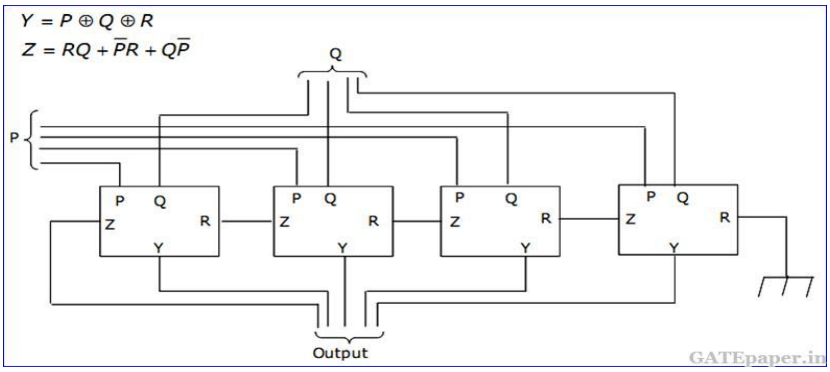
a) 8 half adders, 8 full adders b) 1 half adder, 15 full adders

c) 16 half adders, 0 full adders d) 4 half adders, 12 full adders

13. Borrow in two bit (x,y) subtraction is 0, as long as [ ]

a) y>x b) x=y **c) x>=y** d) y>=x

14.The circuit shown in the figure has 4 boxes each described by inputs P,Q,R and outputs Y, Z with the following relation. The circuit acts as a [ ]



A. 4 bit adder giving P + Q B. 4 bit subtractor giving P – Q

C. 4 bit subtractor giving Q – P D. 4 bit adder giving P + Q + R

15. How does an arithmetic operation take place in binary adders? [ ]

A. By addition of two bits corresponding to 2_n digit

B. By addition of resultant to carry from 2_{n-1} digit

C. both a & b D. none of the above

16. In a half-subtractor circuit with X and Y as inputs, the Borrow (M) and

Difference (N = X – Y) are given by [ ]

(A) $M = X \oplus Y,\ N = XY$
(B) $M = XY,\ N = X \oplus Y$
(C) $M = \bar{X}Y,\ N = X \oplus Y$
(D) $M = X\bar{Y},\ N = \overline{X \oplus Y}$

17.Combinational circuits are described by [ ]

A. Boolean functions B. algebraic functions

C. geometric functions D. linear equations

18.Which logic gate is used as a two-bit adder? [ ]

a) OR b) AND c) NAND d) NOR

19. In four bit adder-subtractor circuit, overflow occurs when---------- [ ]

A) Two input numbers are positive C) Two input numbers are positive

B) One number is positive and another is negative D) all of the above

SECTION-B

SUBJECTIVE QUESTIONS

1. Explain design of half adder.
2. Explain design of full adder.
3. Explain the design of half subtractor.
4. Explain the design of full subtractor.
5. Explain the design and working of ripple adder, ripple subtractor and ripple adder/ subtractor.
6. Explain serial adder.
7. Design a combinational circuit with three inputs x,y and z and three outputs: A,B and C. When the binary input is 0,1,2 and 3, the binary output is one greater than the input. When the binary input is 4,5,6 and 7, the binary output is two less than the input.

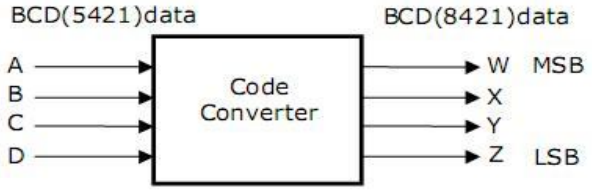
8. Design a combinational circuit that converts a four bit Gray code into a four bit binary code. Implement the circuit with exclusive OR gates.

9. A 'code converter' is to be designed to convert from the BCD (5421) code to normal BCD (8421) code. The input BCD combinations for each digit are given below. A block diagram of the converter is shown in figure.

10. Draw K- map for outputs W, X, Y and Z

11. Obtain minimized expression for the outputs W, X, Y and Z.

Decimal	BCD(5421)			
	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0



12. Design a combinational circuit with three inputs and one output.
    (i). the output is 1 when the binary value of the inputs is less than 3. The output is 0 otherwise.
    (ii). the output is 1 when the binary value of the input is an even number.

13. Design a circuit that has a 3-bit binary input and a single output (Z) specified as follows:
    - Z = 0, when the input is less than 5_{10}
    - Z = 1, otherwise
14. Design a circuit that has a 3-bit binary input and a single output that
    - output 1 if it is a prime number. eg 2_{10}, 3_{10}, 5_{10}, 7_{10}
    - otherwise output 0.
15. Given two input bits A and B, produce three outputs X, Y, and Z so that
    - X is 1 only when only when A > B,
    - Y is 1 only when A < B, and
    - Z is 1 only when A = B
16. Design a circuit with 4 inputs that has outputs with a binary value equal to the number of inputs that are HIGH.

SECTION-C

QUESTIONS AT THE LEVEL OF GATE

1. Using only four-bit adders, construct an eight-bit adder. Each four-bit adder has two four-bit inputs and one five-bit output. Your eight-bit adder should have two eight-bit inputs and a one eight-bit output (don't worry about the ninth output bit).
2. A bank wants to install an alarm system with 3 movement sensors. To prevent false alarms produced by single sensor activation, the alarm will be triggered only when at least two sensors activate simultaneously.

****