Chapter 6

LOGIC CIRCUITS AND GATES ON PRE A* - ALGEBRA

This Chapter describes the concept of *logic circuits and gates on Pre A*-algebra***.** We analyze the concept of logic on Pre A*-algebra**.** Pre A*-algebra is regular extension of Boolean logic to 3 truth-values, where 0 stands for false,1 stands for true but the third truth-value stands for an undefined truth-value. We give truth tables for 2 inputs,3 inputs on Pre A*-algebra**.** By well known definitions of logic gates,AND gate,OR gate,NOT gate,NAND gate,NOR gates in the Boolean algebra, we define logic gates,AND gate,OR gate,NOT gate,NAND gate,NOR gates in Pre A*-algebra**.**We use the operations $\wedge,\vee$ for AND gate,OR gate respectively where the complementation is used by NOT gate. We introduce the concept of logic circuits on Pre A*-algebra and we establish the logic circuits on Pre A*-algebra .We prove logic gates forms a Pre A*-algebra .In a similar way **,** we prove that logic circuits forms a Pre A*-algebra.

In this chapter, we analyze the concept of logic on Pre A*-Algebra**.** Pre A*-algebra is regular extension of Boolean logic to 3 truth-values, where 0 stands for false,1 stands for true but the third truth-value stands for an undefined truth-value. We give truth tables for 2 inputs,3 inputs on Pre A*-Algebra**.** By well known definitions of logic gates,AND gate,OR gate,NOT gate,NAND gate,NOR gates in the Boolean algebra, we define logic gates,AND gate,OR gate,NOT gate,NAND gate,NOR gates in Pre A*-Algebra**.**We use the operations $\wedge,\vee$ for AND gate,OR gate respectively where the complementation is used by NOT gate. We introduce the concept of logic circuits on Pre A*-Algebra and we establish the logic circuits on Pre A*-Algebra .We prove logic gates forms a Pre A*-Algebra .In a similar way **,** we prove that logic circuits forms a Pre A*-Algebra.

In the first section, we analyze the concept of logic on Pre A*-Algebra**.**By a well known definition of switching algebra and its applications we define logic in Pre A*-algebra

We will use digital logic in_ Pre A* – algebra for the elements 0,1,2. in_ Pre A* – algebra we can use logic signals 0,1,2.Suppose 0 for False,1 for True,2 for some specific signal.

For Pre A* – algebra to implement a system,we will choose one of the physical manifestations to represent each value.The inputs and outputs of a digital system represent real quantities,these are 0,1,2 each for different signals.

In the second section, we introduce logic gates in Pre A* - algebra By well known definitions of logic gates,AND gate,OR gate,NOT gate,NAND gate,NOR gates in the Boolean algebra, we define logic gates,AND gate,OR gate,NOT gate,NAND gate,NOR gates in Pre A*-Algebra**.**We use the operations $\wedge,\vee$ for AND gate,OR gate respectively where the complementation is used by NOT gate.

Inputs and outputs of the logic gates can occur in two levels termed as HIGH,LOW or TRUE,FALSEor ON,OFF or simply 1,0 where as for Pre A*-algebra,we can use 3 levels in which 1 for HIGH or TRUEor ON,2 for LOW or FALSE or OFF and 2 for any other indication.

In Pre A*-algebra the truth table for two inputs has 3^2 possible outputs and for three inputs has 3^3 possible outputs.

In Pre A*-algebra,OR gate is used for v operation.OR gate may takes output 2,even if one of its input is 2.

In Pre A*-algebra, AND gate is used for Λ operation, AND gate may takes output 2,even if one of its input is 2.

In Pre A*-algebra the output of the NOT gate assumes the logic 1 state,when its input is in logic 0 state and assumes the logic 0 state,when its input is in logic 1 state and assumes the logic 2 state,when its input is in logic 2 state.

We define NAND and NOR gates in Pre A* – algebra

The negation of AND gate is called as the NAND gate in Pre A* – algebra and the negation of OR gate is called as the NOR gate in Pre A* – algebra

In the third section, we introduce logic gates in Pre A* - algebra We give the definition of Logic circuits in Pre A*-algebra,

For Pre A*-algebra,we can assume three values 0,1,2 each is restricted to a particular state we can say that switch is open if x=0,and closed if x=1 and we choos some specific state if x=2.

Next we define AND – OR Circuits in Pre A* – algebra

We prove logic gates forms a Pre A* – algebra.

In a similar way we prove logic circuits forms a Pre A* – algebra

6.1.1 Logic on Pre A* – algebra:

We will use digital logic in Pre A* – algebra for the elements 0,1,2. in Pre A* – algebra we can use logic signals 0,1,2.Suppose 0 for False,1 for True,2 for some specific signal.

For Pre A* – algebra to implement a system,we will choose one of the physical manifestations to represent each value.The inputs and outputs of a digital system represent real quantities,these are 0,1,2 each for different signals.

6.1.2 Example:

A traffic controller:

There are two streets,and the light is green on each street for a fixed period of time.It then goes to yellow for another fixed period of time and finally to red.There are no inputs to this system other than the clock.There are six outputs,one for each colour in each direction.Traffic controllers may have many more outputs,if for example,there are left-turnsignals.Also there may be several inputs to indicate when there are vehicles waiting at a red signal or passing a green one.In Pre A* – algebra we can choose 0,1,2 for different signals in this example.

6.1.3 Switching algebra:

The Switching algebra is binary,that is ,all variables and constants take on one of two values 0,1.Quantities that are not naturally binary must then be coded into binary format.Physically,they may be represent a light off or on,a switch up or down,o low voltage or a high one,or a magnetic field in one direction or the other.In digital system,the input may be decimal digit or the output may be letter grade.

6.2 LOGIC GATES ON PRE A*-ALGEBRA

First we define the logic gates and then we investigate the logic circuits.

6.2.1 LOGIC GATES:-

Logic gates are the fundamental building blocks of digital systems.the name logic gate is derived from the ability of such a device to make decisions,in the sence that it produces one output level when some combinations of input levels are present,and a different output level when other combinations of input levels are present.

There are three basic logic gates that are OR Gate, AND gate ,NOT gate which are described below.The fact that computers areable to perform very complex logic operations,stems from the way these elementary gates are interconnected.The interconnection of gates to perform a variety of logical operations is called logic design.

6.2.2 Truth table: A table which lists all the possible combinations of input variables and the corresponding outputs is called a truth table.It shows how the logic circuits output responds to various combinations of logic levels at the inputs.In Pre A*-algebra the truth table for two inputs has 3^2 possible outputs and for three inputs has 3^3 possible outputs.

x	y	x ∨ y	x ∧ y
0	0	0	0
0	1	1	0
0	2	2	0
1	0	1	0
1	1	1	1
1	2	2	2
2	0	2	2
2	1	2	2
2	2	2	2

Table 1

The truth table of OR gate and AND gate for two inputs in Pre A*-algebra

6.2.3 Now we will give list of three inputs for Truth table in Pre A*-algebra

x	**y**	**z**
0	**0**	**0**
0	**0**	**1**
0	**0**	**2**
0	**1**	**0**
0	**1**	**1**
0	**1**	**2**
0	**2**	**0**
0	**2**	**1**
0	**2**	**2**

1 0 0

1 0 1

1 0 2

1 1 0

1 1 1

1 1 2

1 2 0

1 2 1

1 2 2

2 0 0

2 0 1

2 0 2

2 1 0

2 1 1

2 1 2

2 2 0

2 2 1

2 2 2

Table2 is the list of three inputs for Truth table in Pre A*-algebra

6.2.4 LOGIC GATES ON PRE A*-ALGEBRA:

Inputs and outputs of the logic gates can occur in two levels termed as HIGH,LOW or TRUE,FALSEor ON,OFF or simply 1,0 where as for Pre A*-algebra,we can use 3 levels in which 1 for HIGH or TRUEor ON,2 for LOW or FALSE or OFF and 2 for any other indication.

We adopt the connection that the lines entering the gate symbol from the left are input lines and the single line on the right is the output line.

6.2.5 (a) OR Gate :-

OR Gate may havetwo or more inputs but only one output.The output assumes the logic state 1,even if one of its input is logic 1 state.Its output assumes the logic state0 state,only when each one of its input is logic 0 state.OR gate may takes output 1,even if one of its input is 1.

In Pre A*-algebra,OR gate is used for v operation.OR gate may takes output 2,even if one of its input is 2.

Fig (i) shows an OR Gate with inputs x and y and output A = x v y where V is defined by the truth table in Fig. (ii).

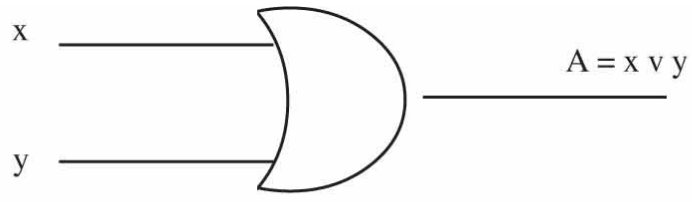

Fig (i)

x	y	x v y
1	1	1
1	0	1
1	2	2
0	1	1
0	0	0
0	2	2
2	1	2
2	0	2
2	2	2

Fig (ii)

Logic symbol and truth table for OR gate in Pre A*-algebra

Thus the output A = 0 only, when input x = 0 and y = 0, otherwise A = 1 or 2 gate may have mere than two inputs Fig (iii) shows an OR gate with four inputs p, q, r, s and output A = pvqvrvs

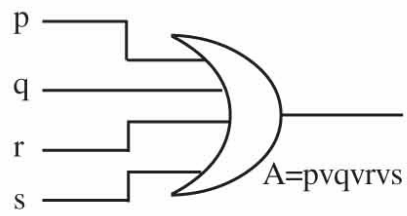

Fig (iii)

The output A = 0 if and only if all the inputs are 0.

Suppose for instance, the input data for the OR gate in fig (iii) are the following sequences:

P = 1 0 2 0 1 2

q = 0 0 1 0 0 1

r = 2 1 0 0 0 2

s = 0 1 2 0 1 0

pvqvrvs= 2 1 2 0 1 2

The OR gate only yields 0 when all input bits are 0. This occurs only in the 4^{th} position (reading from left to right).

Thus the output is the sequence A = 212012

6.2.6 AND Gate :

AND Gate may have two or more inputs but only one output.The output assumes the logic state 1,even if all of its input is logic 1 state.Its output assumes the logic state 0 state,only when one of its input is logic 0 state.AND gate may takes output 1,even if each one of its input is 1.

In Pre A*-algebra, AND gate is used for **Λ** operation, AND gate may takes output 2,even if one of its input is 2.

Fig (iv) shows an AND gate with inputs p and q and output A = p Λ q where multiplication is defined by pq, the truth table is shown in Fig (v).

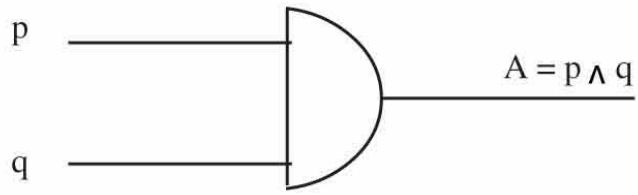

Fig (iv)

p	q	p ∧ q
1	1	1
1	0	0
1	2	2
0	1	0
0	0	0
0	2	2
2	1	2
2	0	2
2	2	2

Fig (v)

Logic symbol and truth table for AND gate on Pre A*-algebra

Thus the output A = 1 when input p = 1 and q = 1 otherwise A = 0 or 2

Such as AND gate may have more than two inputs.

Fig (vi) shows an AND gate with four inputs p, q, r, s and output A = pΛqΛrΛs.

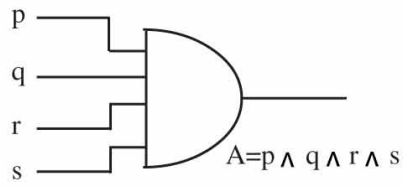

Fig (vi)

The output A = 1 if and only if all the inputs are 1

Suppose, for instance, the input data for the AND gate in Fig (vi) have the following sequences.

P = 1 1 1 0 2 1

q = 0 2 1 1 0 1

r = 2 0 1 2 2 1

s = 1 0 1 2 0 1

p∧q∧r∧ s = 2 2 1 2 2 1

The AND gate only yields 1 when all input bits are 1. This occurs in 3rd and 6th positions. Thus the output in the sequence 2 2 1 2 2 1

6.2.7 NOT gate :

NOT gate has only one input and only one output.It is a device whose output is always the complement of its input. That is In Pre A*-algebra NOT gate is used for ~ operation. In Pre A*-algebra the output of the NOT gate assumes the logic 1 state,when its input is in logic 0 state and assumes the logic 0 state,when its input is in logic 1 state and assumes the logic 2 state,when its input is in logic 2 state.

Fig (vii) shows a NOT gate, also called a complement, with input x and output $x^{\sim}$ where complement denoted by the ~, in defined by the truth table in Fig (viii).

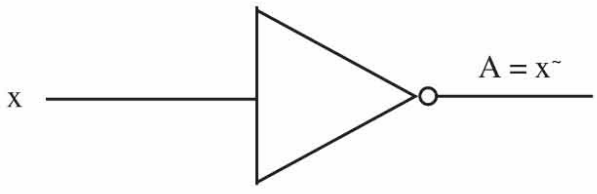

Fig (vii)

x	x~
0	1
1	0
2	2

Fig (viii)

Logic symbol and truth table for NOT gate on Pre A*-algebra

The value of the output A = $x^{\sim}$ is the complement of the input x.

i.e., $x^{\sim} = 1$ when $x = 0$

$x^{\sim} = 0$ when $x = 1$

$x^{\sim} = 2$ when $x = 2$

We emphasize that a NOT gate can have only one input, whereas the OR ad AND gates may have two or more inputs.

Suppose, for instance a NOT gate is asked to process the following three sequences.

A_1 = 0 2 0 0 0 1

A_2 = 1 0 2 0 0 1

A_3 = 2 0 1 0 0 1

The NOT gate changes 0 to 1, 1 to 0

Note that 2 does not changes. Thus

$A_1^{\sim}$ = 1 2 1 1 1 0

$A_2^{\sim}$ = 0 1 2 1 1 0

$A_3^{\sim}$ = 2 1 0 1 1 0

are the three corresponding outputs.

6.2.8 NAND and NOR gates on Pre A* – algebra:

The negation of AND gate is called as the NAND gate in Pre A* – algebra and the negation of OR gate is called as the NOR gate in Pre A* – algebra.

There are two additional gates which are equivalent to combinations of the above basic gates.

a. NAND gate, pictured in Fig (XI) is equivalent to an AND gate followed by a NOT gate.

b. NOR gate, pictured in Fig (XII) is equivalent to an OR gate followed by a NOT gate.

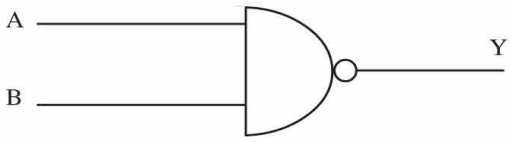

Fig (XI)

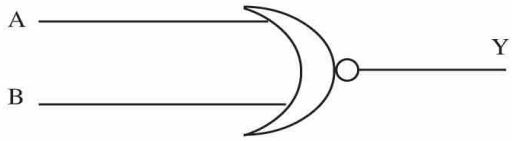

Fig (XII)

The truth tables for these gates (using two inputs A ad B) appear in Fig (c). The NAND and NOR gates can actually have two or more inputs just like the corresponding AND ad OR gates. Furthermore, the output of a NAND gate is 0 if and only if all the inputs are1, and the output of a NOR gate is 1 if ad only if all inputs are 0. also the output of a NAND and NOR gate is 2 if one of the input is 2.

A	B	NAND	NOR
1	1	0	0
1	0	1	0
1	2	2	2
0	1	1	0
0	0	1	1
0	2	2	2
2	1	2	2
2	0	2	2
2	2	2	2

Fig (XIII)

Observe that the only difference between the AND and NAND gates and between the OR and NOR gates is that the NAND ad NOR gates are each followed by a circle. We can also use such a small circle to indicate a complement before a gate. For example, the Boolean expressions corresponding to the two logic circuits in following Fig (XIV) are as follows.

a). $Y = (p^{\sim} \wedge q)^{\sim}$

b). $Y = (p^{\sim} \vee q^{\sim} \vee r)^{\sim}$

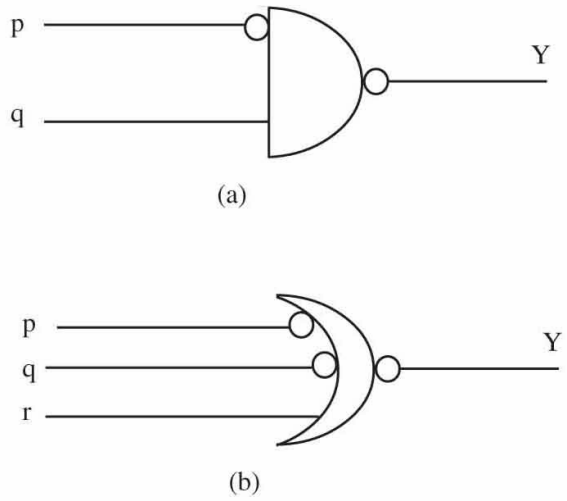

Fig (XIV)

6.3 Logic Circuits:-

The study of Logic circuits is motivated mostly by their use in digital computers.But such circuits also form the foundation of many other digital systems where performing arithmetic operations on numbers . Logic circuits perform operations on digital signals and are usually implemented as electronic circuits where the signal values are restricted to a few discrete values.In binary logic circuits there are only two levels 0 and 1.In decimal logic circuits there are 10 values,from 0 to 9.Since each signal is represented by a digit,such a logic circuit is referred to as digital circuits.In contrast,there exist analog circuits where the signals may take on a continuous range of values between some minimum and some maximum levels.

The dominance of binary circuits in digital systems is a consequence of their simplicity,which results frim consraning the signals to assume only two possible values. The simplest binary element in a switch that has two states.If a given switch is controlled by an input variable x,then we can say that switch is open if x=0,and closed if x=1.

6.3.1 Definition of Logic circuits:

A computer switching circuit that consists of a number of logic gates and performs logical operations on data. Electronic circuits which process information encoded as one of a limited set of voltage or current levels. Logical circuits are the basic building blocks used to realize consumer and industrial products that incorporate digital electronics. Such products include digital computes, video games, voice synthesizers, pocket calculators and robot controls.

6.3.2 Logic circuits on Pre A*-algebra:

For Pre A*-algebra,we can assume three values 0,1,2 each is restricted to a particular state we can say that switch is open if x=0,and closed if x=1 and we choos some specific state if x=2.

Logic circuits (are also called logic networks) are structures which are built up from certain elementary circuits called logic gates. Each logic circuit may be viewed as a machine L which contains one or more input devices and exactly one output device. Each input device in L sends a signal specifically, 0,1, or 2 to the circuit L, and L proceses the set of {0,1,2} to yield an output. Accordingly, a sequence may be assigned to each input device, and L processes the input sequences one digit at a time to produce an output sequence.

A logic circuit L is a well formed structure whose elementary components are the above OR, AND ad NOT gates. Fig (ix) is an example of a logic circuit with inputs p, q, r ad output A. A dot (.) indicates a place where the input line splits so that its digit signal is sent in more than one direction.

Working from left to right, we express Y interms of the inputs p, q, r as follows. The output of the AND gate is pΛq which is then negated to yield (p q)~

The output of the lower OR gate is p~∨ r, which is then negated to yield (p~∨ r)~. The output of the OR gate on the right, with inputs (p ∧ q)~ and (p~ ∨ r)~ gives us our desired representation, that is

$A = (p \wedge q)^{\sim} \vee (p^{\sim} \vee r)^{\sim}$

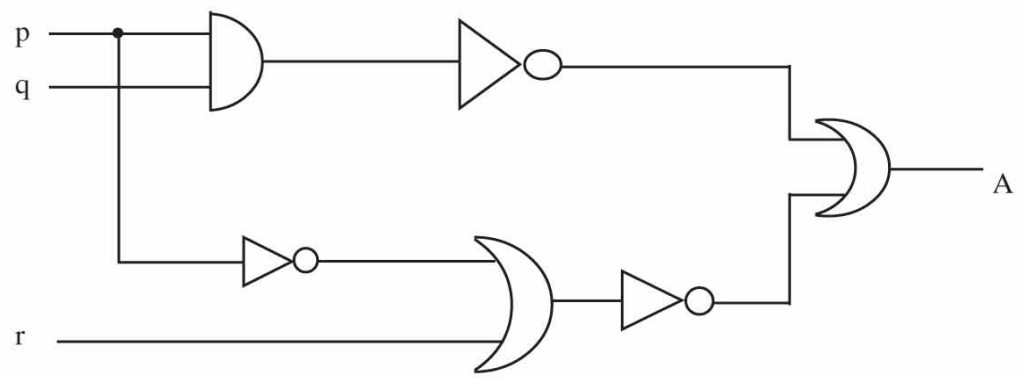

Fig (ix)

6.3.3 AND – OR Circuits On Pre A* – algebra:

The logic circuit L which corresponds to a join of meet expression is called an AND – OR circuit in

Pre A* – algebra:

Such a circuit L has several inputs where:

1.Some of the inputs or their complements are fed into each AND gate.

2.The outputs of all the AND gates are fed into a single OR gate.

3.The output of the OR gate is the output for the circuit L

The following illustrates this type of a logic circuit in Pre A* – algebra:

Ex : Fig (X) is a typical AND – OR circuit with three inputs p, q, r ad output A. We can easily express A as a Boolean expression in the inputs p, q, r as follows. First we find the output of each AND gate.

a.The inputs of the first AND gate are p, q, r hence $p \wedge q \wedge r$ is the output.

b.The inputs of the second AND gate are p, $q^{\sim}$, r hence $p \wedge q^{\sim} \wedge r$ is the output

c.The inputs of the third AND gate are $p^{\sim}$ ad q hence $p^{\sim} \wedge q$ is the output.

Then the sum of the outputs of the AND gates is the output of the OR gate, which is the output A of the circuit. Thus $A = (p \wedge q \wedge r) \vee (p \wedge q^{\sim} \wedge r) \vee (p^{\sim} \wedge q)$

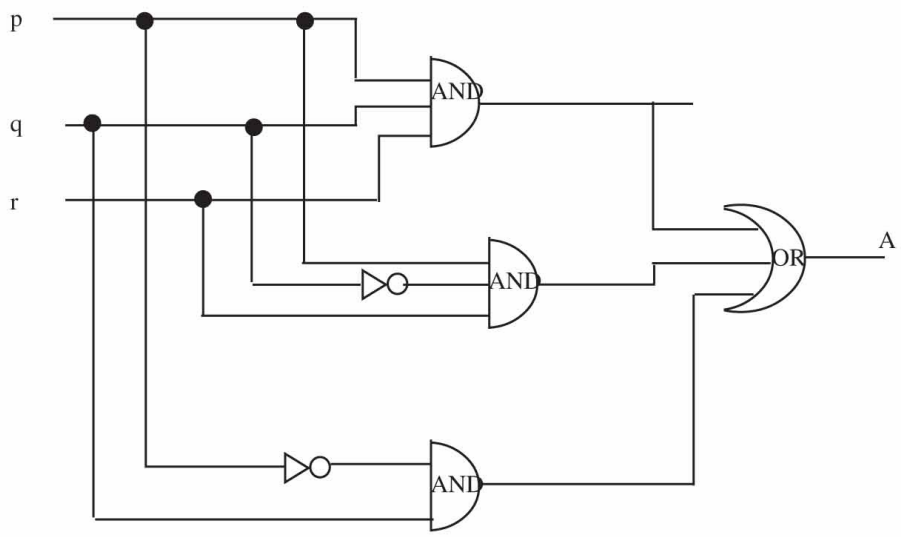

Fig (X)

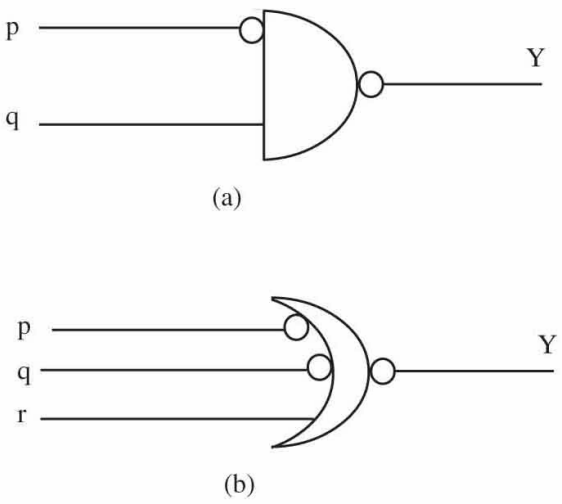

Fig (XIV)

6.3.4 Propositions:

Let P (p, q,…..)denote an expression constructed from logic variable p,q,… which take on the value TRUE (T), FALSE (F) NEITHER TRUE NOR FALSE (N), and the logical connectives ∧, V and ~, such an expression P (p, q,….) will be called a proposition.

Observe that the truth tables for the OR, AND and NOT gates are respectively identical to the truth tables for the propositions p v q (disjunction “p or q”) p∧q (conjuction “p ad q”) and ~ p (negation, not p)

Note that in Pre A* – algebra, " p and q", "p or q" are same when 2 appears as one of the truth value.

Also ~p remains same when 2 is the truth value.

6.3.5 Theorem : Logic gates forms a Pre A* – algebra.

Proof: Let A be a a Pre A* – algebra.

(i) Then by using NOT gate we can prove that $x \%\% x, \quad \forall x \in A$

If x=0 then $x \%\% x, \quad \forall x \in A$

if x=1 then $x \wedge x = x, \quad \forall x \in A$

if x=2 then $x \wedge x = x, \quad \forall x \in A$

(ii)By using AND gate,

we can prove that $x \wedge x = x, \quad \forall x \in A$

If x=0 then $x \wedge x = x, \quad \forall x \in A$

if x=1 then $x \wedge x = x, \quad \forall x \in A$

if x=2 then $x \wedge x = x, \quad \forall x \in A$

(iii) By using AND gate,we can prove that

$x \wedge y = y \wedge x, \quad \forall x, y \in A$

The truth table is the following:

X	y	x ∧ y	y∧ x
0	0	0	0
0	1	0	0
0	2	2	2
1	0	0	0
1	1	1	1
1	2	2	2
2	0	2	2
2	1	2	2
2	2	2	2

iv)) By using AND gate and NOT gate we can prove that

$(x \wedge y)' = x' \vee y', \quad \forall x, y \in A$

The truth table is the following:

x	y	x ∧y	(x ∧y)~	x ~	y~	x ~ v y~
0	0	0	1	1	1	1
0	1	0	1	1	0	1
0	2	2	2	1	2	2
1	0	0	1	0	1	1
1	1	1	0	0	0	0
1	2	2	2	0	2	2
2	0	2	2	2	1	2
2	1	2	2	2	0	2
2	2	2	2	2	2	2

(v)) By using AND gate we can prove that

$$x \wedge (y \wedge z) = (x \wedge y) \wedge z, \quad \forall x, y, z \in A$$

(vi) By using AND gate and OR gate we can prove that

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z), \quad \forall x, y, z \in A$$

(vii)) By using AND gate and NOT gate we can prove that

$$x \wedge y = x \wedge (x \% y), \quad \forall x, y \in A$$

Hence logic gates forms the Pre A* – algebra

6.3.6 Theorem : Logic circuits forms a Pre A* – algebra.

Proof: Let A be a a Pre A* – algebra.

By theorem 6.3.5,we have logic gates forms the Pre A* – algebra.

Since logic circuits consists of a number of logic gates and performs operations on Pre A* – algebra

Hence Logic circuits forms a Pre A* – algebra.

Conclusion: We investigated the concept of logic on Pre A*-algebra. Pre A*-algebra is regular extension of Boolean logic to 3 truth-values, where 0 stands for false,1 stands for true but the third truth-value stands for an undefined truth-value. We give truth tables for 2 inputs, 3 inputs on Pre A*-algebra. By well known definitions of logic gates, AND gate, OR gate, NOT gate, NAND gate, NOR gates in the Boolean algebra, we defined logic gates, AND gate, OR gate, NOT gate, NAND gate, NOR gates in Pre A*-algebra. We use the operations Ù,Ú for AND gate, OR gate respectively where the complementation is used by NOT gate. We introduced the concept of logic circuits on Pre A*-algebra and we establish the logic circuits on Pre A*-algebra .We proved logic gates forms a Pre A*-algebra .In a similar way, we proved that logic circuits forms a Pre A*-algebra.