**Experiment No:4**

Aim: To implement 2-input AND gate & 2-input OR gate logic using MP neuron

Theory:

**McCullogh-Pitts Model**

Every neuron model consists of a processing element with synaptic input connection and a single

input. The "neurons" operated under the following assumptions:

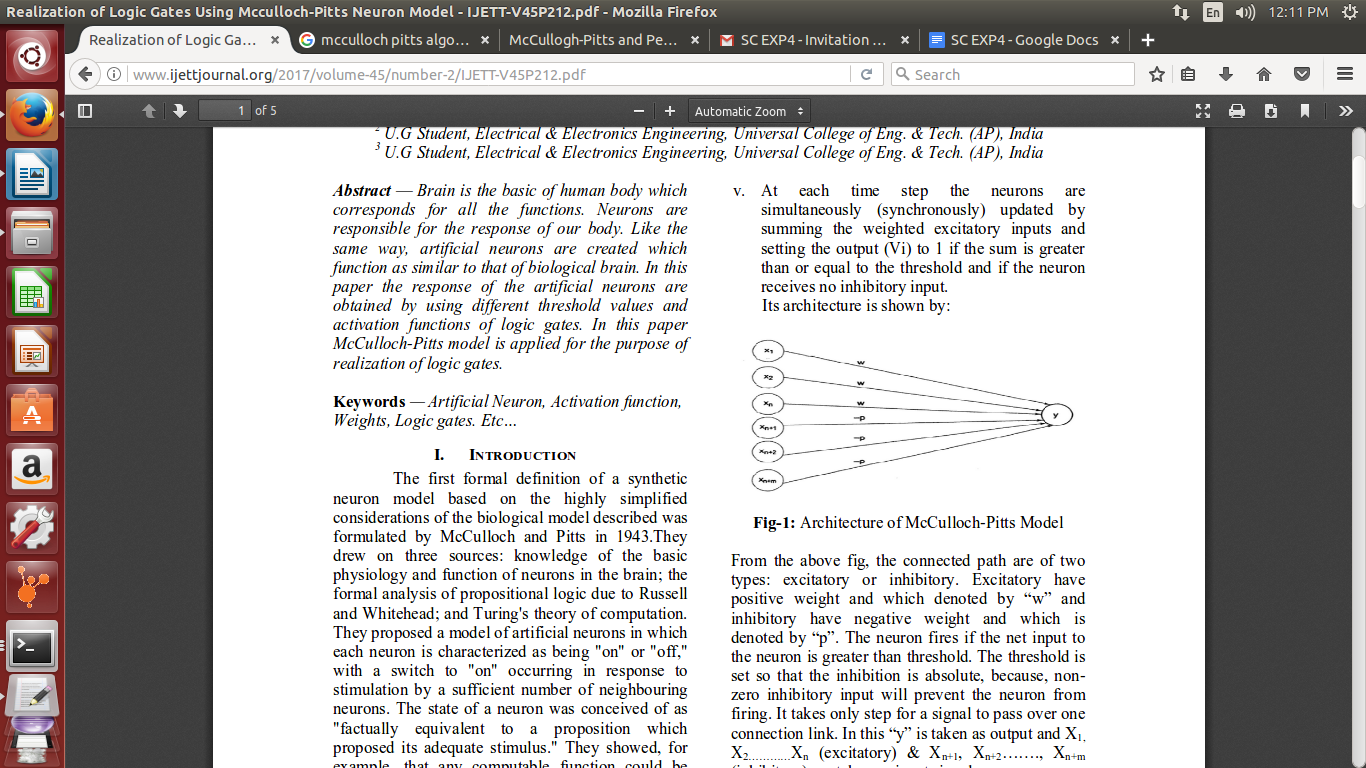
i.They are binary devices (Vi = [0,1])

ii.Each neuron has a fixed threshold, theta values.

iii.The neuron receives inputs from excitatory synapses, all having identical weights.

iv. Inhibitory inputs have an absolute veto power over any excitatory inputs.

v.At each time step the neurons are simultaneously (synchronously) updated by summing the weighted excitatory inputs and setting the output (Vi) to 1 if the sum is greater than or equal to the threshold and if the neuron receives no inhibitory input.



Program:

import java.util.\*;

public class qwerty{

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int w1, w2,b,th;

int[] yin = new int[10];

int[] yout = new int[10];

int[] x1w1 = new int[10];

int[] x2w2 = new int[10];

int i;

int[] x1 = {0,0,1,1};

int[] x2 = {0,1,0,1};

System.out.println("Enter Weights");

w1 =sc.nextInt();

w2 =sc.nextInt();

for (i=0;i<4 ;i++ ) {

yin[i]=x1[i]\*w1+x2[i]\*w2;

}

System.out.println("the net input yin= ");

for (i=0;i<4 ;i++ ) {

System.out.println(""+yin[i]);

}

System.out.println("Enter the Threshold Value");

th=sc.nextInt();

System.out.println("Enter the Operation to be performed\n1)AND \n2)OR");

b=sc.nextInt();

if (b==1) {

System.out.println("And using MP");

System.out.println();

for (i=0;i<4 ;i++ ) {

x1w1[i]=x1[i]&w1;

x2w2[i]=x2[i]&w2;

}

for (i=0;i<4 ;i++ ) {

yin[i]=x1w1[i]+x2w2[i];

if(yin[i]>=th)

yout[i]=1;

else yout[i]=0;

}

System.out.println("x1\tx2\tx1w1\tx2w2\tyin\tyout");

for (i=0;i<4 ;i++ ) {

System.out.println(""+x1[i]+"\t"+x2[i]+"\t"+x1w1[i]+"\t"+x2w2[i]+"\t"+yin[i]+"\t"+yout[i]);

}

}

else

{

System.out.println("OR function ");

System.out.println();

for(i=0;i<4;i++)

{

x1w1[i]=x1[i]\*w1;

x2w2[i]=x2[i]\*w2;

}

for(i=0;i<4;i++)

{

yin[i]=x1w1[i]+x2w2[i];

if(yin[i]>=th)

yout[i]=1;

else

yout[i]=0;

}

System.out.println("x1\tx2\tx1w1\tx2w2\tyin\tyout");

for(i=0;i<4;i++)

{

System.out.println(""+x1[i]+"\t"+x2[i]+"\t" +x1w1[i]+ "\t" +x2w2[i]+ "\t" +yin[i]+ "\t" +yout[i]);

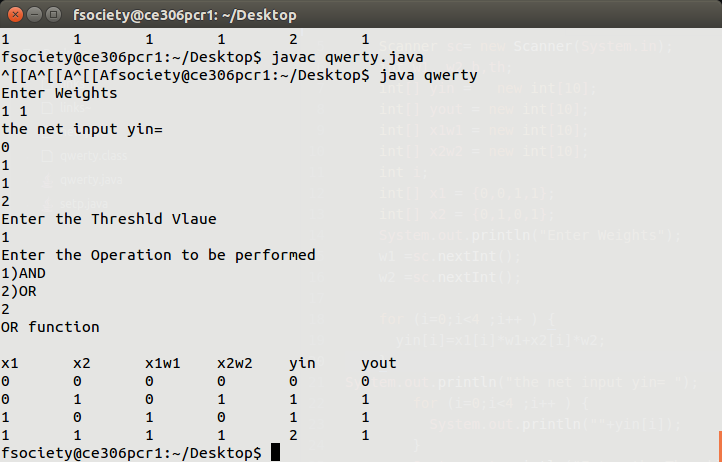
}

}

}

}

Output:





Conclusion: Hence the Implementation of AND & or operator using MP-neuron is implemented