1. Take input Synonymous Codons
2. Assign/ Encode the codon in binary
3. Enter no of input line, no of gate and input sequence for the circuit
4. Initialize Codon Matrix, gate array and take input t

Codonarr[i][j] = codon

Codonbinary[i][j] = 0/1

1. For k <= no of gate

begin

For j=1 to no of input line

begin

if(j == t[j] && t3[j] == t[-1])

gatearr[k][j]=-1

else if (j==t[j] && t[j]!= last )

gatearr[k][j]=0

else

gatearr[k][j]=1

end

end

1. Take position of faulty gate

n=faulty gate

1. For k=0 to no. of gate

begin

if(k==n)

codonarr[k+1][] = codonarr[k][]

codonbinary[k+1][] = codonbinary[k][]

else

for i=1 to no of input

begin

if (gatearr[k][i]==0 or 1)

codonarr[k+1][i]= codon[k][i]

codonbinary[k+1][i]= codonbinary[k][i]

else if (gatearr[k][i] == -1)

x=AND(k)

codonbinary[k+1][i] XOR (codonbinary[k][],x)

if (codonbinary[k+1][i] == 1)

y=replace\_last(codonarr[k][i])

codonarr[k+1][i] = concatenate(k,y)

else

codonarr[k+1][i] = concatenate(k)

end

end

1. For i=0 to inputline

Begin

Print(codonarr[gateno+1][i], ”(“ ,codonbinary[gateno+1][i], ”)” )

Acid+= findamino(codonarr[gateno+1][i])

Findsimilar\_string(Acid)

1. Findamino(x)

begin

if( x == dict[])

return dict[value]

end