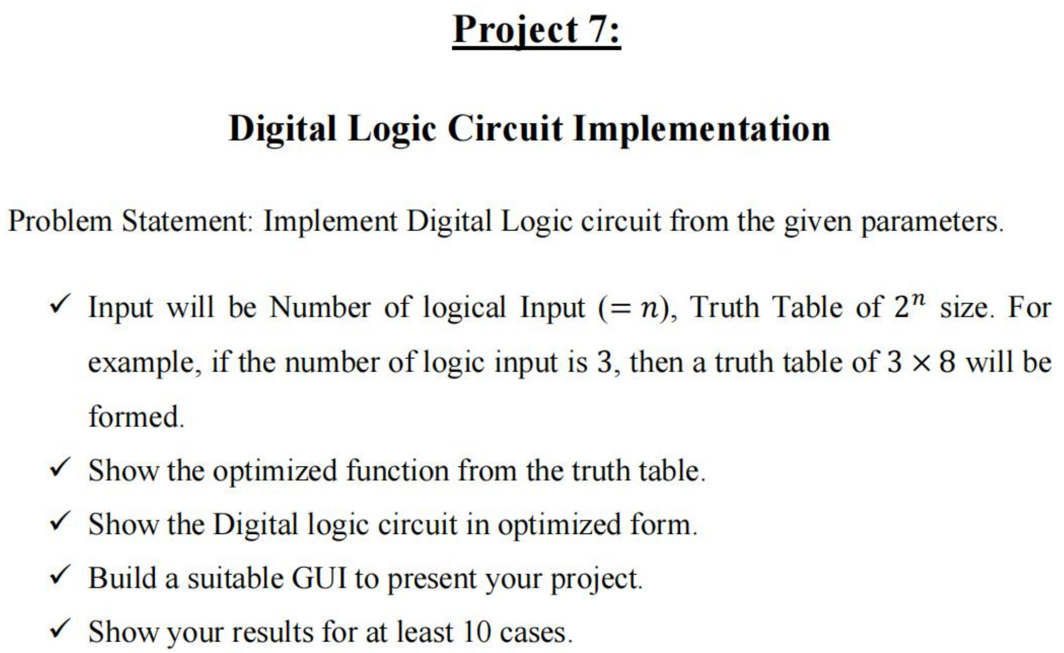
Name: Manzoor Elahi Tamjeed

Roll: 1906046

Dept: EEE

Final Project (Project 7)

EEE 212



Project 7:

Digital Logic Circuit Implementation

Problem Statement: Implement Digital Logic circuit from the given parameters.

- ✓ Input will be Number of logical Input ($= n$), Truth Table of 2^n size. For example, if the number of logic input is 3, then a truth table of 3×8 will be formed.
- ✓ Show the optimized function from the truth table.
- ✓ Show the Digital logic circuit in optimized form.
- ✓ Build a suitable GUI to present your project.
- ✓ Show your results for at least 10 cases.

PROBLEM:

Here, the user will input a number (integer). Then, the integer will be the number of Boolean variables.

So, for that, a Boolean truth table will be formed with the size of n*(2^n). So, the user have to input 2^n of 1 or zeros as the output of the truth table.

We have to identify the function from that table and optimize that. Then we have to draw the digital logic circuit from that function.

Then, we have to build a GUI(Graphical User Interface) to present the project.

The code for function:

```
clc
clear all;
n=input('Enter the number of variables:');
N=2^n;
Tab = zeros(N,n);
for i=1:n
   y = [zeros(N/2^i,1); ones(N/2^i,1)];
   Tab(:,i) = repmat(y,2^(i-1),1);
end
Tab(:,n+1)=input('Enter the output of truth
table:');
table=Tab;
disp('Truth Table');
disp(table);
i=0;
j=1;
m=1;
u=0;
v=0;
l=0;
temp=1;
s=char(zeros(1,N));
for j=1:N
    if(table(j,n+1)==1)
        u=u+1;
        l=j;
    else
        v=v+1;
    end
end
if(u<=v)
  for j=1:N
    if(table(j,n+1)==1)
        for k=1:n
            if(table(j,m)==1)
                s(temp)=char('A'+i);
                temp=temp+1;
            else
                s(temp)=char('A'+i);
                temp=temp+1;
                s(temp)=char('?');
                temp=temp+1;
            end
              i=i+1;
              m=m+1;
        end
        if(j==l)
            s(temp)=char('.');
            temp=temp+1;
        else
            s(temp)=char('+');
            temp=temp+1;
        end
    end
    i=0;
    m=1;
  end
end
if(v<u)
   for j=1:N
    if(table(j,n+1)==0)
        s(temp)=char('(');
        temp=temp+1;
        for k=1:n
            if(table(j,m)==1)
                s(temp)=char('A'+i);
                temp=temp+1;
                s(temp)=char('?');
                temp=temp+1;
            else
                s(temp)=char('A'+i);
                temp=temp+1;
            end
            if(k==n)
        s(temp)=char(')');
        temp=temp+1;
      else
        s(temp)=char('+');
        temp=temp+1;
       i=i+1;
       m=m+1;
      end
    end
  end
  i=0;
  m=1;
 end
end
if sum(Tab(:,n+1))== (2^n)
  fprintf("The Boolean Function is: 1");
elseif sum(Tab(:,n+1))==0
  fprintf("The Boolean Function is: 1");
else
fprintf('The Boolean Function is: %s \n',s);
end
```

Then, we have to build the gui for the project. I did it in matlab app designer. The code for GUI:

```
 classdef finalapp < matlab.apps.AppBase


% Properties that correspond to app components
properties (Access = public)
UIFigure matlab.ui.Figure
thenumberofvariablesEditFieldLabel matlab.ui.control.Label
thenumberofvariablesEditField matlab.ui.control.NumericEditField
UITable matlab.ui.control.Table
runButton matlab.ui.control.Button
functionEditFieldLabel matlab.ui.control.Label
functionEditField matlab.ui.control.EditField
entertheoutputEditFieldLabel matlab.ui.control.Label
entertheoutputEditField matlab.ui.control.EditField
UIAxes matlab.ui.control.UIAxes
end
```

```
% Callbacks that handle component events
methods (Access = private)


% Button pushed function: runButton
function runButtonPushed(app, event)
n=app.thenumberofvariablesEditField.Value;
N=2^n;
Tab = zeros(N,n);
for i=1:n
y = [zeros(N/2^i,1); ones(N/2^i,1)];
Tab(:,i) = repmat(y,2^(i-1),1);
end
a=app.entertheoutputEditField.Value;
Tab(:,n+1)=str2double(strsplit(a,','));
table=Tab;
app.UITable.Data=table;
app.UITable.ColumnName{n+1}='X';
i=0;
j=1;
m=1;
u=0;
v=0;
l=0;
temp=1;
s=char(zeros(1,N));
for j=1:N
if(table(j,n+1)==1)
u=u+1;
l=j;
else
v=v+1;
end
end
if(u<=v)
for j=1:N
if(table(j,n+1)==1)
for k=1:n
if(table(j,m)==1)
s(temp)=char('A'+i);
temp=temp+1;
else
s(temp)=char('A'+i);
temp=temp+1;
s(temp)=char('′');
temp=temp+1;
end
i=i+1;
```

```
m=m+1;
end
if(j==l)
s(temp)=char('.');
temp=temp+1;
else
s(temp)=char('+');
temp=temp+1;
end
end
i=0;
m=1;
end
end
if(v<u)
for j=1:N
if(table(j,n+1)==0)
s(temp)=char('(');
temp=temp+1;
for k=1:n
if(table(j,m)==1)
s(temp)=char('A'+i);
temp=temp+1;
s(temp)=char('′');
temp=temp+1;
else
s(temp)=char('A'+i);
temp=temp+1;
end
if(k==n)
s(temp)=char(')');
temp=temp+1;
else
s(temp)=char('+');
temp=temp+1;
i=i+1;
m=m+1;
end
end
end
i=0;
m=1;
end
end
if sum(Tab(:,n+1))== (2^n)
app.functionEditField.Value=char(49);
elseif sum(Tab(:,n+1))==0
app.functionEditField.Value=”zero”;
else
app.functionEditField.Value=s;
```

```
end
xp=0:36;
yp=-2:0.1:2;
t=u;
out=Tab(:,n+1);
rows=2^n;
cols=n;
m=ones(rows,cols);
for i = 1:cols
j=1;
while j<=rows
m(j,i)=0;
if mod(j,2^(cols-i))==0
j=j+2^(cols-i);
end
j=j+1;
end
end
k=1;
a=[];
for i=1:2^n
if out(i)==1
for j = 1:n
if m(i,j)==0
start=-17.5+(j-1)*5;
elseif m(i,j)==1
start=-20+(j-1)*5;
end
a(k,:)= linspace(start,2*n+2,3);
k=k+1;
end
end
end
for i=1:n
ya=15:-1:-t*32;
xa=i*ones(1,length(ya));
plot(app.UIAxes, 5*xa-25,ya,'k');
axis([-30 80 -80 30])
hold (app.UIAxes,'on');
yb=6.6:-1:-t*32 ;
xb=i*ones(1,length(yb))+0.5;
plot(app.UIAxes,5*xb-25,yb,'k');
yc=[10:-1:8] ;
xc=i*ones(1,length(yc))+0.5;
plot(app.UIAxes,5*xc-25,yc,'c');
xd=i+0.25:0.05:i+0.25+0.5;
yd=4*abs(xd-i-0.5)+7;
plot(app.UIAxes,5*xd-25,yd,'g');
f=0.2;
trig = 0:0.1:2*pi;
```

```
xtrig = f * cos(trig);
ytrig = f * sin(trig);
plot(app.UIAxes,xtrig+((5*xd(1)-25)+(5*xd(end)-25))/2, ytrig+yd(1)-1.2,'b');
yf=yc(end)*ones(1,length(xd));
plot(app.UIAxes,5*xd-25,yf,'k');
xe=i:0.1:i+0.5;
y5=yc(1)*ones(1,length(xe));
plot(app.UIAxes,5*xe-25,y5,'g');
end
app.UIAxes.Visible=true;
k=1;
for i=0:t-1
yg=(0:-1:-8);
xg=(2*n)*ones(1,length(yg))+2;
plot(app.UIAxes,xg,yg-i*32,'k');
plot(app.UIAxes,a(k,:),zeros(length(a(k,:)))+yg(1)-i*32,'c')
k=k+1;
plot(app.UIAxes,a(k,:),zeros(length(a(k,:)))+yg(end)-i*32,'g')
k=k+1;
yc=-4:0.1:4;
xc=sqrt(16-(yc).^2)+xg(1);
plot(app.UIAxes,xc,yc+(yg(1)+yg(end))/2 - i*32,'k')
plot(app.UIAxes,xc-4,zeros(1,length(xc))- i*32,'m')
plot(app.UIAxes,xc-4,zeros(1,length(xc))- i*32-8,'c')
for j=1:n-2
plot(app.UIAxes,xg+j*8,yg-i*32-j*4,'k');
plot(app.UIAxes,a(k,:),zeros(length(a(k,:)))+yg(end)-i*32-j*4,'k')
k=k+1;
plot(app.UIAxes,xc+j*8,yc+(yg(1)+yg(end))/2 - i*32-j*4,'c')
plot(app.UIAxes,xc+j*8,yc+(yg(1)+yg(end))/2 - i*32-j*4,'g')
plot(app.UIAxes,xc-4+j*8,zeros(1,length(xc))- i*32-j*4,'g')
plot(app.UIAxes,linspace(0,xg(1)+j*8,5),zeros(1,5)- i*32-8-j*4,'g')
end
end
for i=0:t-2
for k=1:t-1
xpa=-yp.^2;
xpb=-0.1*yp.^2-3.6;
plot(app.UIAxes,xpa+(n-2)*10+10+(i+1)*40,2*yp-4*i-8-(n-2)*4,'b')
plot(app.UIAxes,xpb+(n-2)*10+10+(i+1)*40,2*yp-4*i-8-(n-2)*4,'b')
plot(app.UIAxes,yp+(n-2)*10+44+40*(k-1),zeros(1,length(yp))- k*4-8 -(n-2)*4,'m')
plot(app.UIAxes,xp+(n-2)*10+10+40*(k-1),zeros(1,length(xp))- k*4-0-(n-2)*4,'m')
s=xp(1)+(n-2)*10+10;
f=yp(1)+(n-2)*10+44+40*(k-1);
xpp=s:f;
plot(app.UIAxes,xpp,zeros(1,length(xpp))- (k-1)*(32)-36-(n-2)*4,'m')
yf = (-(k-1)*(32)-36-(n-2)*4):(- k*4-8-(n-2)*4);
plot(app.UIAxes,zeros(1,length(yf))+xpp(end),yf,'m')
end
end
```

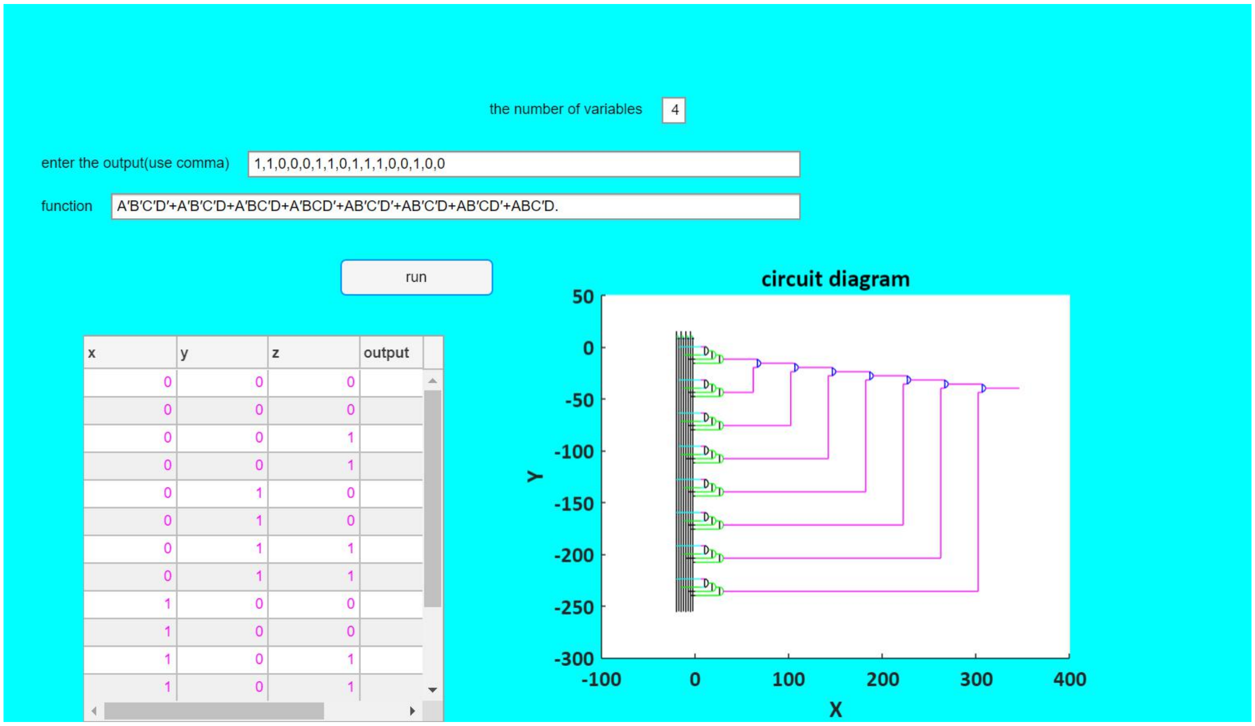
```
plot(app.UIAxes,xp+(n-2)*10+10+40*(t-1),zeros(1,length(xp))- t*4-0-(n-2)*4,'m')
end
end


% Component initialization
methods (Access = private)


% Create UIFigure and components
function createComponents(app)


% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Color = [0 1 1];
app.UIFigure.Position = [100 100 640 480];
app.UIFigure.Name = 'MATLAB App';
app.UIFigure.WindowStyle = 'modal';
app.UIFigure.WindowState = 'maximized';


% Create thenumberofvariablesEditFieldLabel
app.thenumberofvariablesEditFieldLabel = uilabel(app.UIFigure);
app.thenumberofvariablesEditFieldLabel.HorizontalAlignment = 'right';
app.thenumberofvariablesEditFieldLabel.Position = [430 491 131 22];
app.thenumberofvariablesEditFieldLabel.Text = 'the number of variables';


% Create thenumberofvariablesEditField
app.thenumberofvariablesEditField = uieditfield(app.UIFigure, 'numeric');
app.thenumberofvariablesEditField.Position = [576 491 20 22];


% Create UITable
app.UITable = uitable(app.UIFigure);
app.UITable.ColumnName = {'x'; 'y'; 'z'; 'output'};
app.UITable.RowName = {};
app.UITable.ForegroundColor = [1 0 1];
app.UITable.Position = [102 1 296 317];


% Create runButton
app.runButton = uibutton(app.UIFigure, 'push');
app.runButton.ButtonPushedFcn = createCallbackFcn(app, @runButtonPushed, true);
app.runButton.Position = [438 391 125 30];
app.runButton.Text = 'run';


% Create functionEditFieldLabel
```

```matlab
app.functionEditFieldLabel = uilabel(app.UIFigure);
app.functionEditFieldLabel.HorizontalAlignment = 'right';
app.functionEditFieldLabel.Position = [51 391 48 22];
app.functionEditFieldLabel.Text = 'function';

% Create functionEditField
app.functionEditField = uieditfield(app.UIFigure, 'text');
app.functionEditField.Position = [114 391 224 22];

            % Create entertheoutputEditFieldLabel
            app.entertheoutputEditFieldLabel = uilabel(app.UIFigure);
            app.entertheoutputEditFieldLabel.HorizontalAlignment = 'right';
            app.entertheoutputEditFieldLabel.Position = [132 447 90 22];
            app.entertheoutputEditFieldLabel.Text = 'enter the output';

            % Create entertheoutputEditField
            app.entertheoutputEditField = uieditfield(app.UIFigure, 'text');
            app.entertheoutputEditField.Position = [237 447 453 22];

            % Create UIAxes
            app.UIAxes = uiaxes(app.UIFigure);
            title(app.UIAxes, 'circuit diagram')
            xlabel(app.UIAxes, 'X')
            ylabel(app.UIAxes, 'Y')
            zlabel(app.UIAxes, 'Z')
            app.UIAxes.FontName = 'Calibri';
            app.UIAxes.FontWeight = 'bold';
            app.UIAxes.FontSize = 18;
            app.UIAxes.Position = [460 1 465 379];

            % Show the figure after all components are created
            app.UIFigure.Visible = 'on';
        end
    end

    % App creation and deletion
    methods (Access = public)

        % Construct app
        function app = finalapp

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.UIFigure)

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
        end
    end
```

The 10 cases for the test:

1. No of variables: 4

   Input numbers: 1,1,0,0,0,1,1,0,1,1,1,0,0,1,0,0



the number of variables: 4

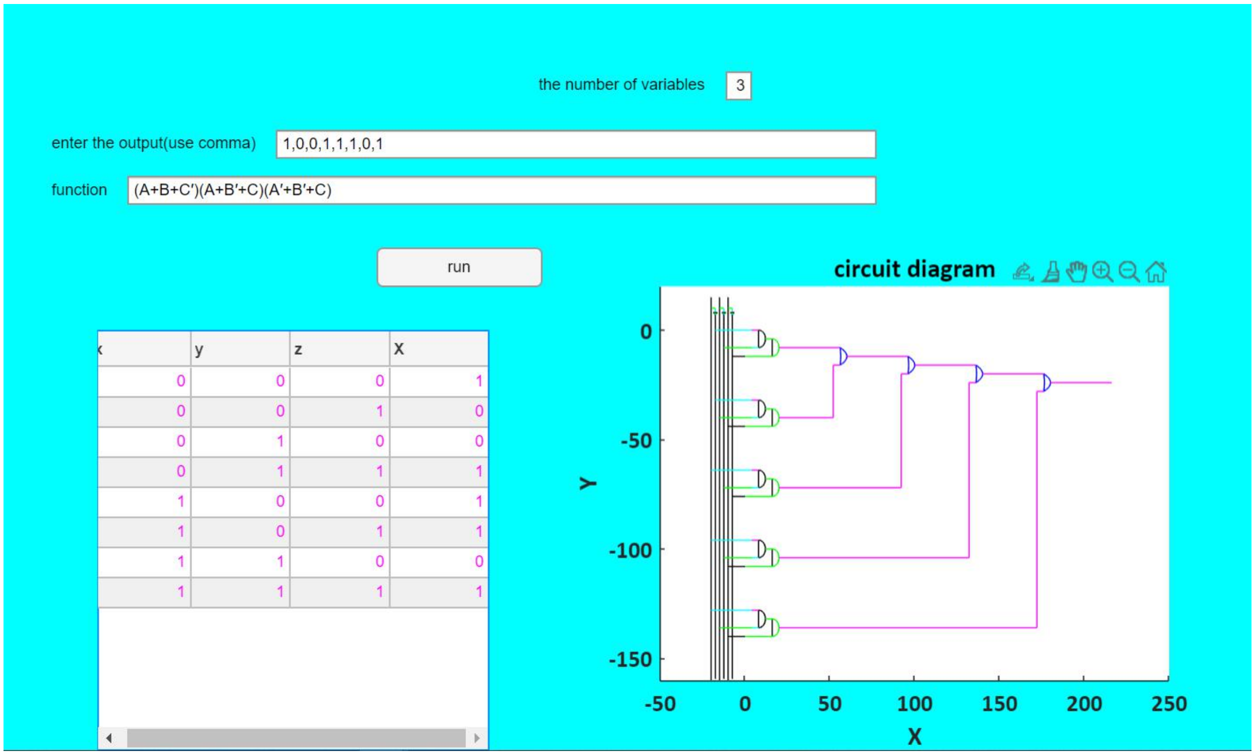
enter the output(use comma): 1,1,0,0,0,1,1,0,1,1,1,0,0,1,0,0

function: A'B'C'D'+A'B'C'D+A'BC'D+A'BCD'+AB'C'D'+AB'C'D+AB'CD'+ABC'D.

x	y	z	output
0	0	0	
0	0	0	
0	0	1	
0	0	1	
0	1	0	
0	1	0	
0	1	1	
0	1	1	
1	0	0	
1	0	0	
1	0	1	
1	0	1	

2. No of variables: 3

Output=1,0,0,1,1,1,0,1



the number of variables 3

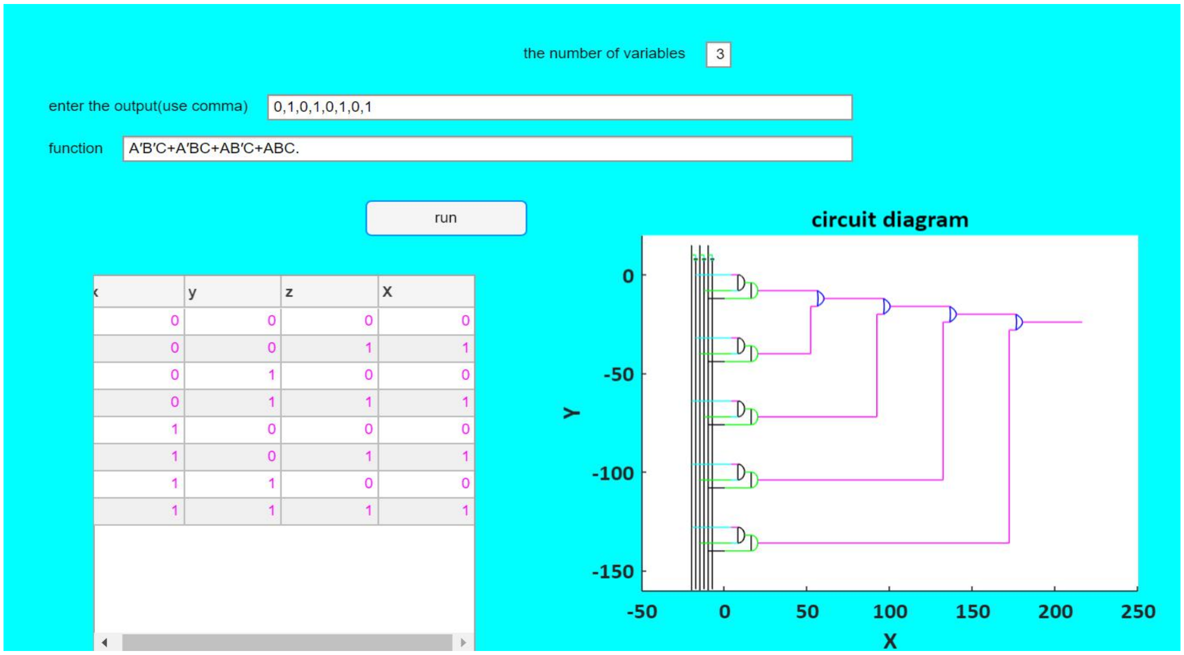
enter the output(use comma) 1,0,0,1,1,1,0,1

function (A+B+C')(A+B'+C)(A'+B'+C)

x	y	z	X
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

3. No of variables: 3

Output=0,1,0,1,0,1,0,1



the number of variables 3

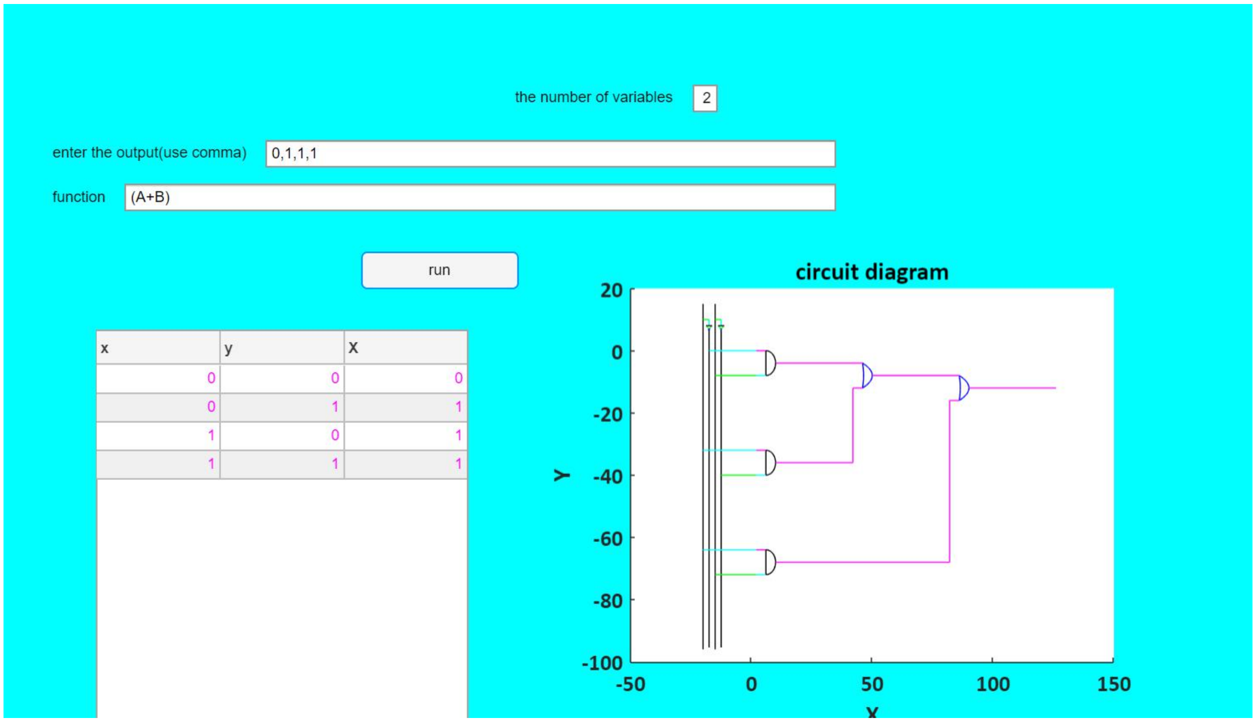
enter the output(use comma) 0,1,0,1,0,1,0,1

function A'B'C+A'BC+AB'C+ABC.

x	y	z	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

4. No of variables: 2

   Output=0,1,1,1



the number of variables 2

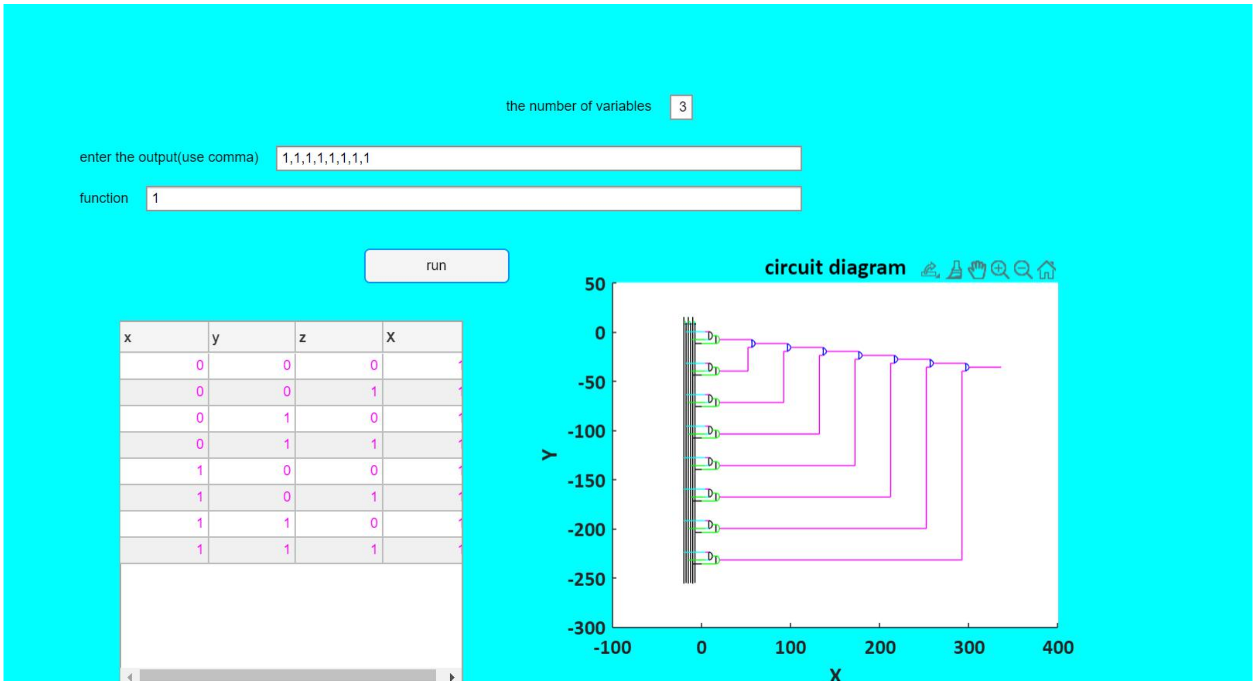
enter the output(use comma) 0,1,1,1

function (A+B)

x	y	X
0	0	0
0	1	1
1	0	1
1	1	1

5. No of variables: 3

   Output=1,1,1,1,1,1,1,1



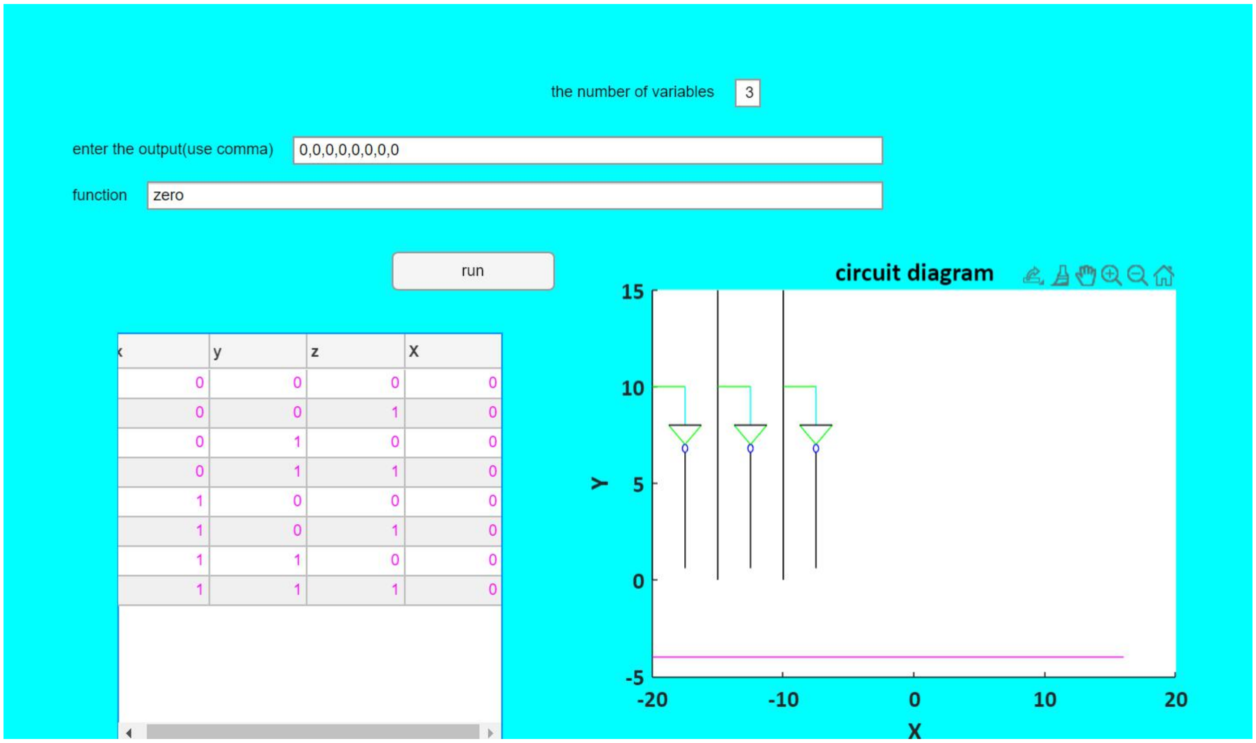
the number of variables 3

enter the output(use comma) 1,1,1,1,1,1,1,1

function 1

x	y	z	X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

6. No of variables: 3
   Output=0,0,0,0,0,0,0,0



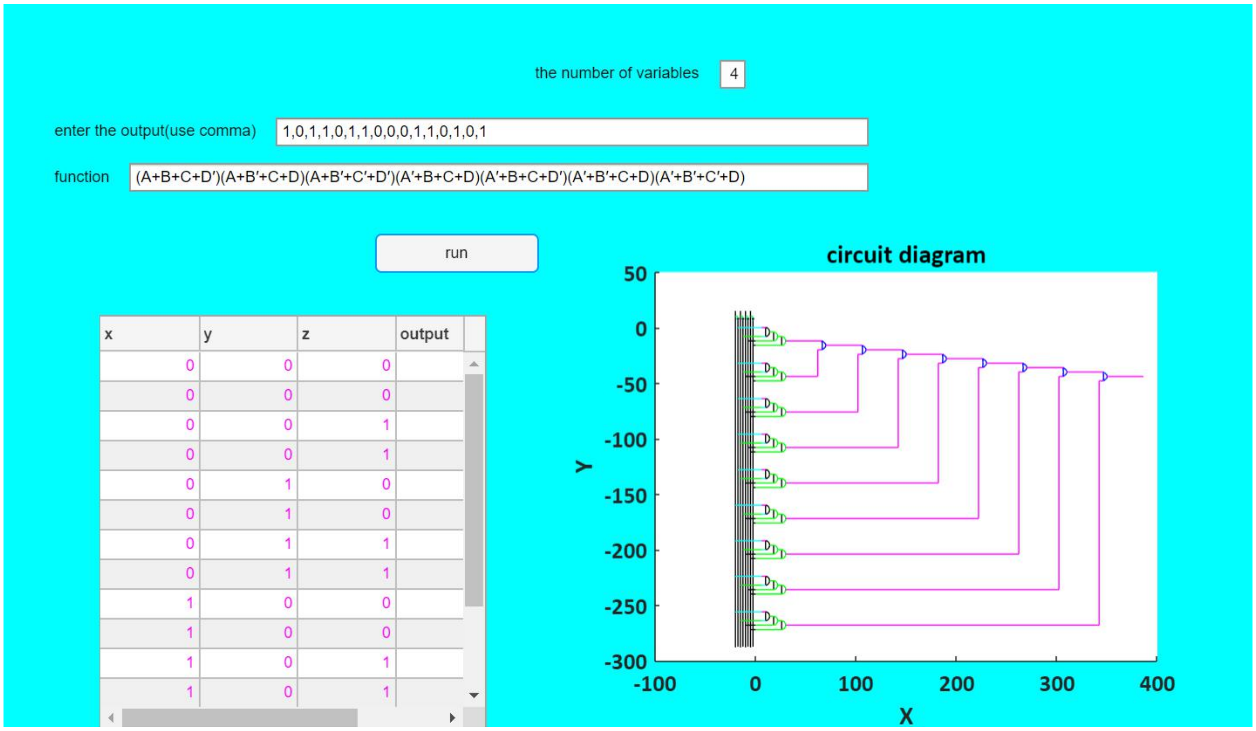
the number of variables 3

enter the output(use comma) 0,0,0,0,0,0,0,0

function zero

x	y	z	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

7. No of variables: 4
   Output=1,0,1,1,0,1,1,0,0,0,1,1,0,1,0,1



the number of variables 4

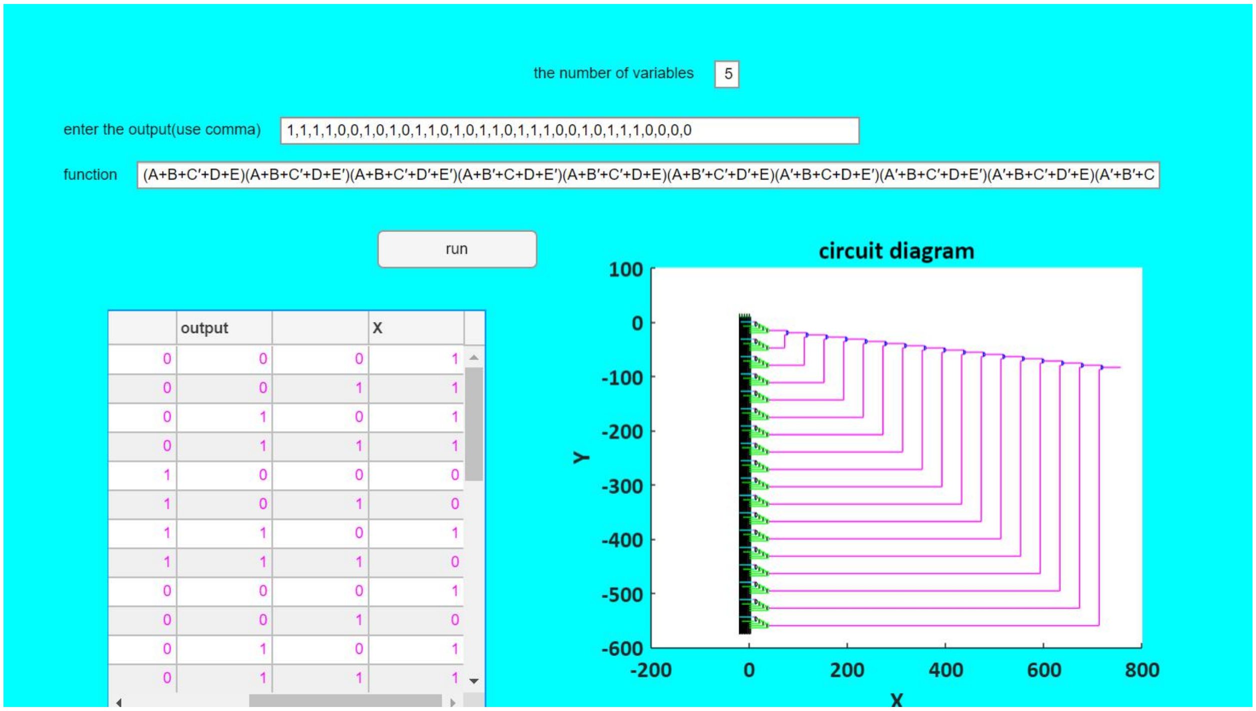
enter the output(use comma) 1,0,1,1,0,1,1,0,0,0,1,1,0,1,0,1

function (A+B+C+D′)(A+B′+C+D)(A+B′+C′+D′)(A′+B+C+D)(A′+B+C+D′)(A′+B′+C+D)(A′+B′+C′+D)

x	y	z	output
0	0	0	
0	0	0	
0	0	1	
0	0	1	
0	1	0	
0	1	0	
0	1	1	
0	1	1	
1	0	0	
1	0	0	
1	0	1	
1	0	1	

8. No of variables: 5

   Output=1,1,1,1,0,0,1,0,1,0,1,1,0,1,0,1,1,0,1,1,1,0,0,1,0,1,1,1,0,0,0,0



the number of variables 5

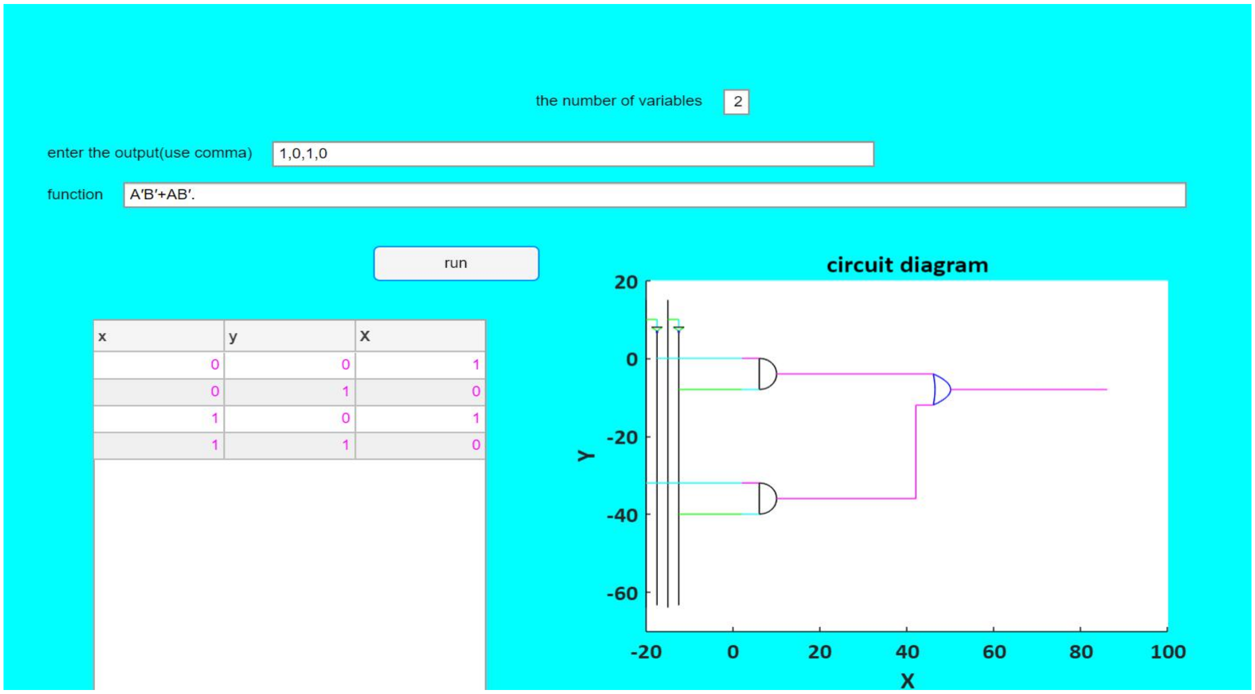
enter the output(use comma) 1,1,1,1,0,0,1,0,1,0,1,1,0,1,0,1,1,0,1,1,1,0,0,1,0,1,1,1,0,0,0,0

function (A+B+C′+D+E)(A+B+C′+D+E′)(A+B+C′+D′+E′)(A+B′+C+D+E′)(A+B′+C′+D+E)(A+B′+C′+D′+E)(A′+B+C+D+E′)(A′+B+C′+D+E′)(A′+B+C′+D′+E)(A′+B′+C

	output		X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1

9. No of variables: 2

   Output=1,0,1,0



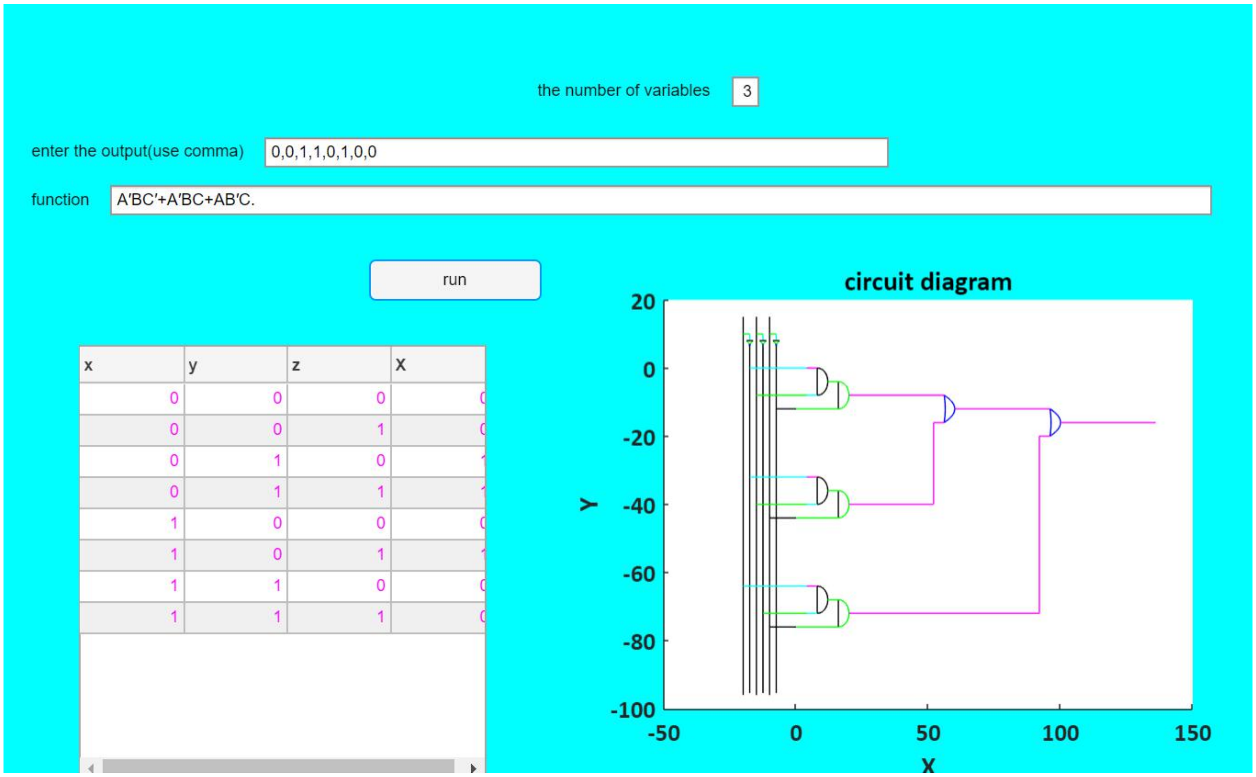
the number of variables 2

enter the output(use comma) 1,0,1,0

function A′B′+AB′.

x	y	X
0	0	1
0	1	0
1	0	1
1	1	0

10. No of variables: 3
Output=0,0,1,1,0,1,0,0



the number of variables 3

enter the output(use comma) 0,0,1,1,0,1,0,0

function A′BC′+A′BC+AB′C.

x	y	z	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

DISCUSSION:

Here, we tried to be as neat and clean as possible. The boundary cases are tested. The logic equation was not simplified that much. But I tried as much as I could.

For the circuit draw, for long values, it may be very hard to visible.

But I tried to minimize the error as much as I could.