Islamic University of Technology

Name : Shazzad Ahmed

Student ID : 220021108

Section : A-(02)

Department :EEE

Course No : EEE-4416

Course Title : Matlab Simulation Lab

Experiment : MATLAB GUI Project

Date of Submission : 30/09/2025

Digital Logic & Digital Modulation Simulator (MATLAB App)

Introduction

This project is a MATLAB App Designer-based application that integrates Digital Logic Design and Digital Modulation concepts into a single interactive platform. The app provides a user-friendly graphical interface for simulating logic gates, constructing truth tables, and visualizing digital modulation schemes.

Features

- **Digital Logic Design Tab**: Includes buttons for logic gates (AND, OR, NOT, NAND, NOR, XOR, XNOR).
- Generates truth tables based on selected logic gates.
- Displays corresponding waveforms for input and output signals.

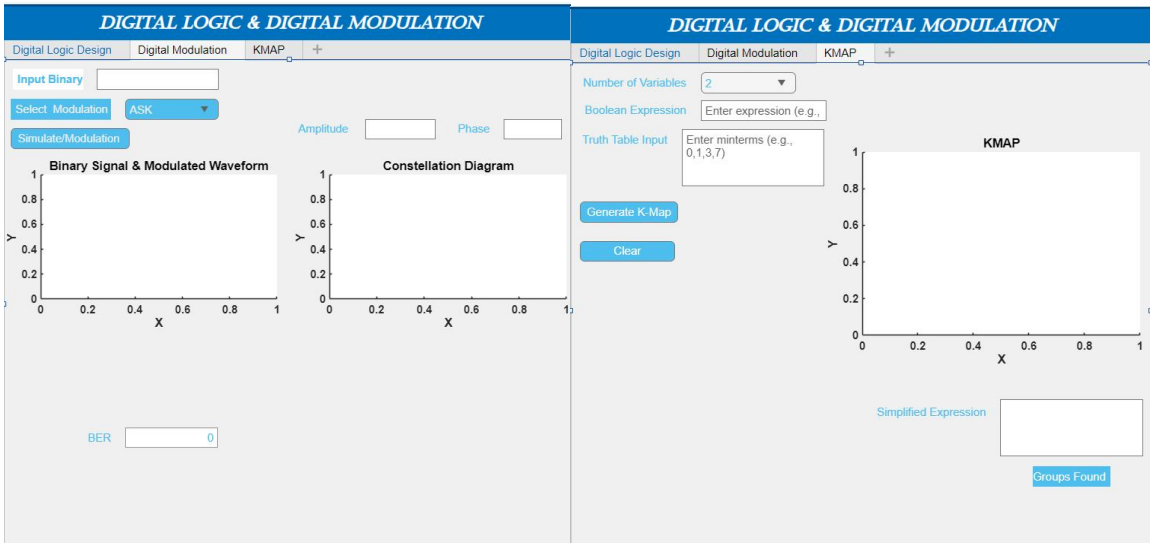
- **Digital Modulation Tab**: Allows selection of modulation schemes (ASK, PSK, FSK, etc.).(Not completed yet. It's a prototype)
- User inputs binary streams for modulation.
- Displays output waveform, constellation diagram, and eye diagram.
- Includes fields for amplitude, phase, BER, and modulation parameters.

Objective

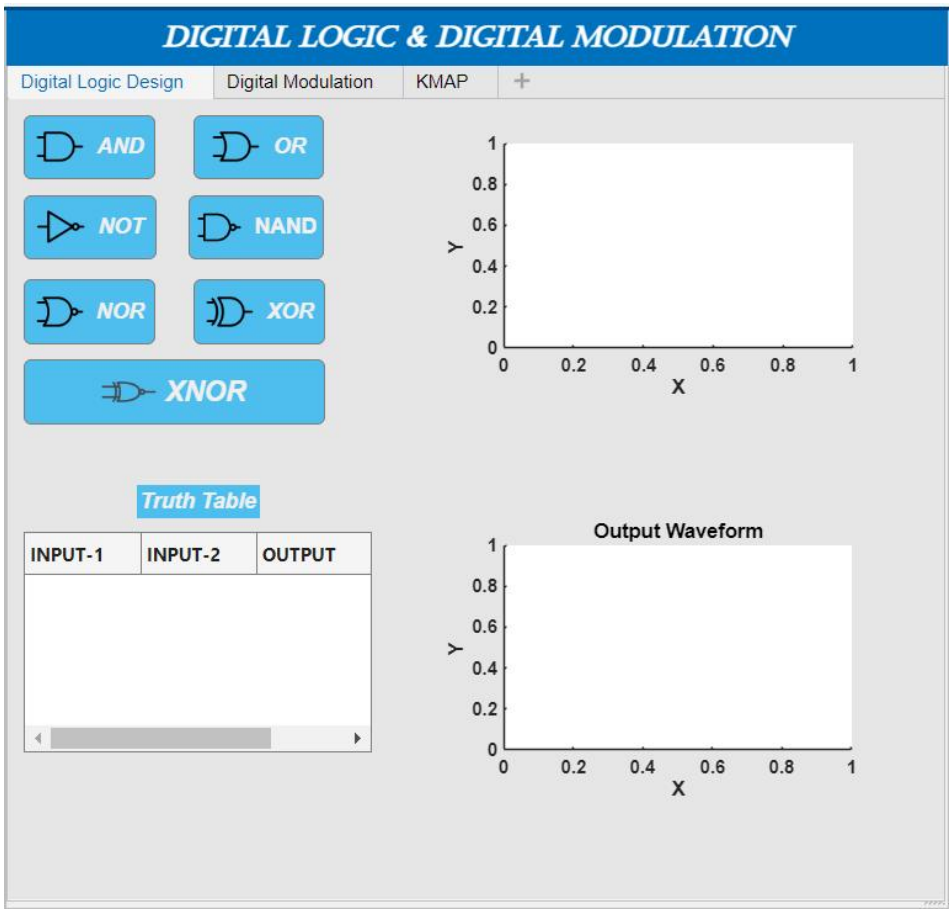
The main objective of this project is to provide an educational and simulation tool that helps students and researchers understand the fundamentals of digital logic , digital modulation and Kmap of logic design. By combining truth table generation, logic gate simulation, Kmap, and modulation visualization, the app serves as an interactive laboratory environment.

APP's Design View

Prototype



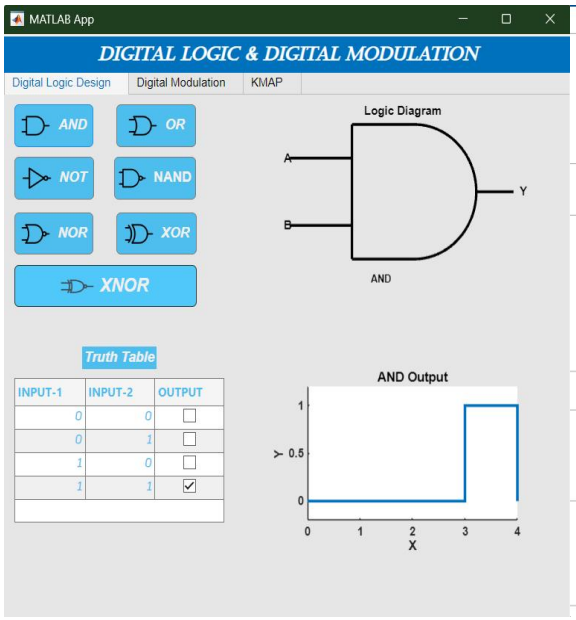
Working



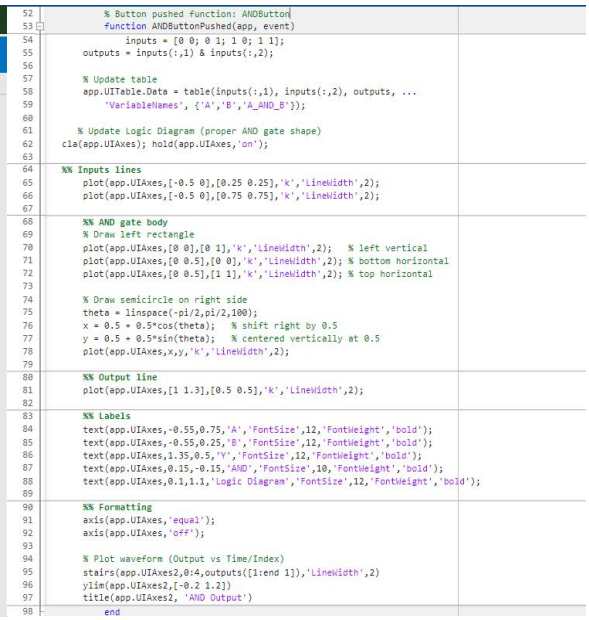
Different Segments and Outputs

And Gate:

Design view Output



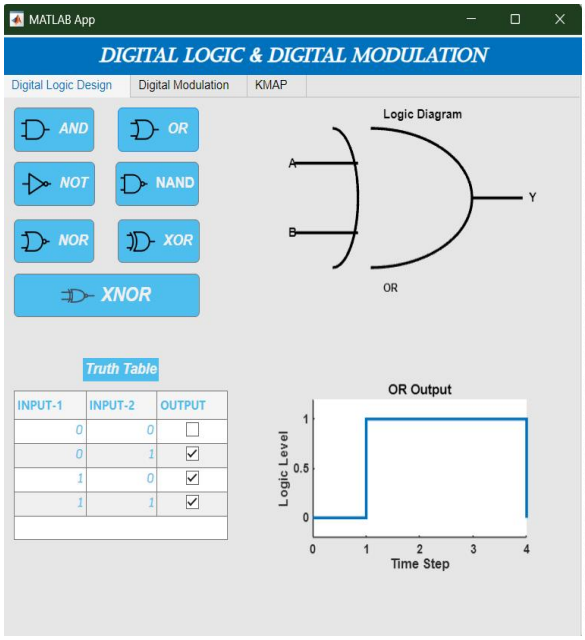
Code view



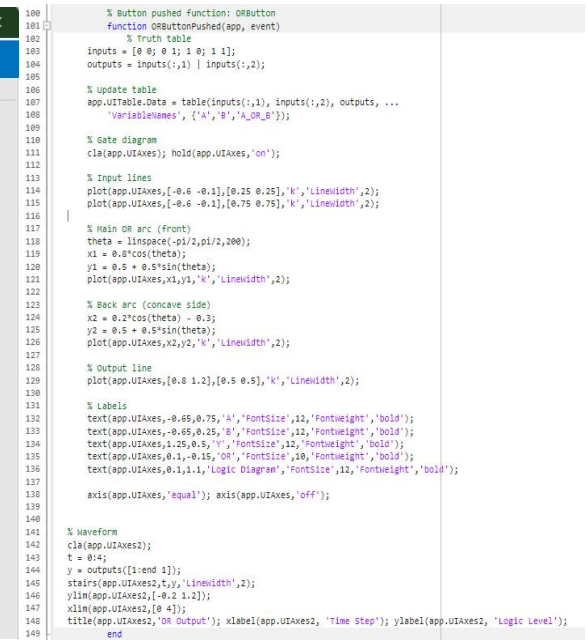
```
        % Button pushed function: ANDButton
        function ANDButtonPushed(app, event)
            inputs = [0 0; 0 1; 1 0; 1 1];
    outputs = inputs(:,1) & inputs(:,2);

    % Update table
    app.UITable.Data = table(inputs(:,1), inputs(:,2), outputs, ...
        'VariableNames', {'A','B','A_AND_B'});

  % Update Logic Diagram (proper AND gate shape)
cla(app.UIAxes); hold(app.UIAxes,'on');

%% Inputs lines
    plot(app.UIAxes,[-0.5 0],[0.25 0.25],'k','LineWidth',2);
    plot(app.UIAxes,[-0.5 0],[0.75 0.75],'k','LineWidth',2);

    %% AND gate body
    % Draw left rectangle
    plot(app.UIAxes,[0 0],[0 1],'k','LineWidth',2);   % left vertical
    plot(app.UIAxes,[0 0.5],[0 0],'k','LineWidth',2); % bottom horizontal
    plot(app.UIAxes,[0 0.5],[1 1],'k','LineWidth',2); % top horizontal

    % Draw semicircle on right side
    theta = linspace(-pi/2,pi/2,100);
    x = 0.5 + 0.5*cos(theta);   % shift right by 0.5
    y = 0.5 + 0.5*sin(theta);   % centered vertically at 0.5
    plot(app.UIAxes,x,y,'k','LineWidth',2);

    %% Output line
    plot(app.UIAxes,[1 1.3],[0.5 0.5],'k','LineWidth',2);

    %% Labels
    text(app.UIAxes,-0.55,0.75,'A','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,-0.55,0.25,'B','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,1.35,0.5,'Y','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,0.15,-0.15,'AND','FontSize',10,'FontWeight','bold');
    text(app.UIAxes,0.1,1.1,'Logic Diagram','FontSize',12,'FontWeight','bold');

    %% Formatting
    axis(app.UIAxes,'equal');
    axis(app.UIAxes,'off');

    % Plot waveform (Output vs Time/Index)
    stairs(app.UIAxes2,0:4,outputs([1:end 1]),'LineWidth',2)
    ylim(app.UIAxes2,[-0.2 1.2])
    title(app.UIAxes2, 'AND Output')
        end
```

OR Gate:

Design view Output



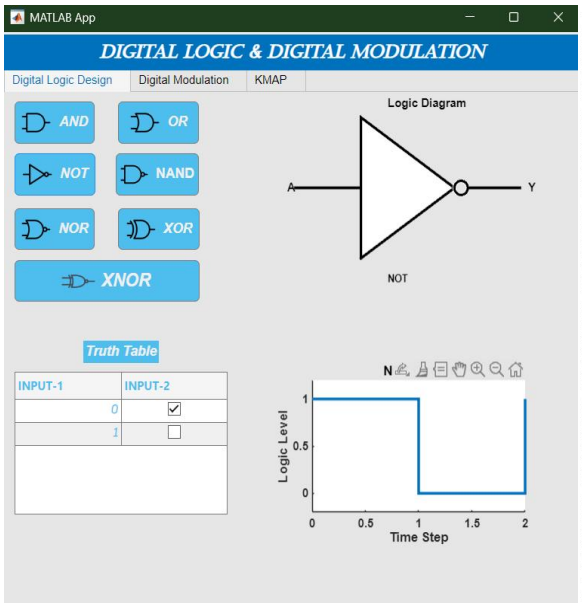
Code view



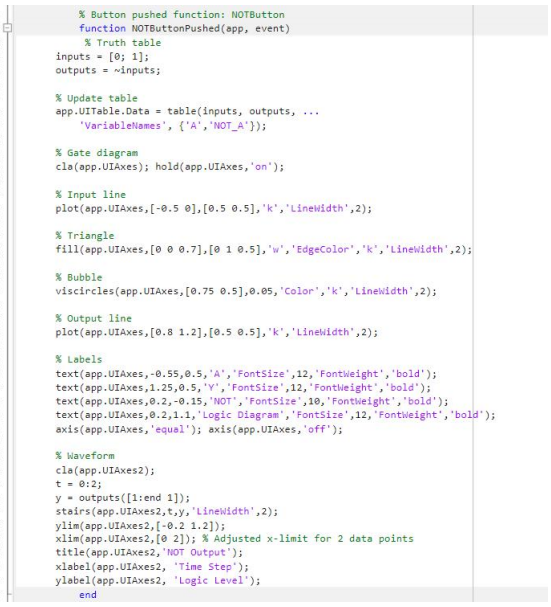
```
        % Button pushed function: ORButton
        function ORButtonPushed(app, event)
            % Truth table
    inputs = [0 0; 0 1; 1 0; 1 1];
    outputs = inputs(:,1) | inputs(:,2);

    % Update table
    app.UITable.Data = table(inputs(:,1), inputs(:,2), outputs, ...
        'VariableNames', {'A','B','A_OR_B'});

    % Gate diagram
    cla(app.UIAxes); hold(app.UIAxes,'on');

    % Input lines
    plot(app.UIAxes,[-0.6 -0.1],[0.25 0.25],'k','LineWidth',2);
    plot(app.UIAxes,[-0.6 -0.1],[0.75 0.75],'k','LineWidth',2);

    % Main OR arc (front)
    theta = linspace(-pi/2,pi/2,200);
    x1 = 0.8*cos(theta);
    y1 = 0.5 + 0.5*sin(theta);
    plot(app.UIAxes,x1,y1,'k','LineWidth',2);

    % Back arc (concave side)
    x2 = 0.2*cos(theta) - 0.3;
    y2 = 0.5 + 0.5*sin(theta);
    plot(app.UIAxes,x2,y2,'k','LineWidth',2);

    % Output line
    plot(app.UIAxes,[0.8 1.2],[0.5 0.5],'k','LineWidth',2);

    % Labels
    text(app.UIAxes,-0.65,0.75,'A','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,-0.65,0.25,'B','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,1.25,0.5,'Y','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,0.1,-0.15,'OR','FontSize',10,'FontWeight','bold');
    text(app.UIAxes,0.1,1.1,'Logic Diagram','FontSize',12,'FontWeight','bold');

    axis(app.UIAxes,'equal'); axis(app.UIAxes,'off');


% Waveform
cla(app.UIAxes2);
t = 0:4;
y = outputs([1:end 1]);
stairs(app.UIAxes2,t,y,'LineWidth',2);
ylim(app.UIAxes2,[-0.2 1.2]);
xlim(app.UIAxes2,[0 4]);
title(app.UIAxes2,'OR Output'); xlabel(app.UIAxes2, 'Time Step'); ylabel(app.UIAxes2, 'Logic Level');
        end
```

NOT Gate:

Design view Output



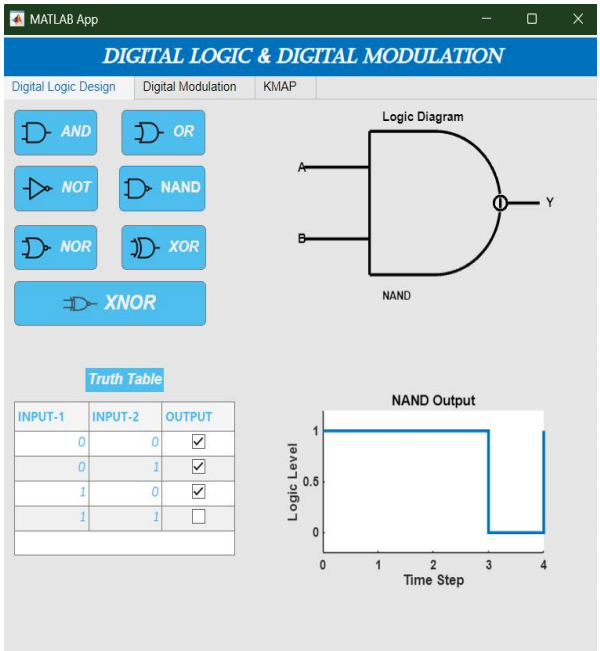
Code view



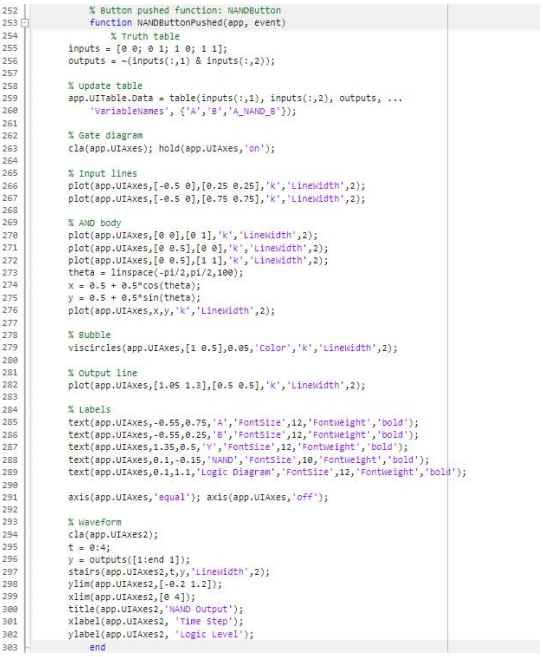
```
        % Button pushed function: NOTButton
        function NOTButtonPushed(app, event)
         % Truth table
    inputs = [0; 1];
    outputs = ~inputs;

    % Update table
    app.UITable.Data = table(inputs, outputs, ...
        'VariableNames', {'A','NOT_A'});

    % Gate diagram
    cla(app.UIAxes); hold(app.UIAxes,'on');

    % Input line
    plot(app.UIAxes,[-0.5 0],[0.5 0.5],'k','LineWidth',2);

    % Triangle
    fill(app.UIAxes,[0 0 0.7],[0 1 0.5],'w','EdgeColor','k','LineWidth',2);

    % Bubble
    viscircles(app.UIAxes,[0.75 0.5],0.05,'Color','k','LineWidth',2);

    % Output line
    plot(app.UIAxes,[0.8 1.2],[0.5 0.5],'k','LineWidth',2);

    % Labels
    text(app.UIAxes,-0.55,0.5,'A','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,1.25,0.5,'Y','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,0.2,-0.15,'NOT','FontSize',10,'FontWeight','bold');
    text(app.UIAxes,0.2,1.1,'Logic Diagram','FontSize',12,'FontWeight','bold');
    axis(app.UIAxes,'equal'); axis(app.UIAxes,'off');

    % Waveform
    cla(app.UIAxes2);
    t = 0:2;
    y = outputs([1:end 1]);
    stairs(app.UIAxes2,t,y,'LineWidth',2);
    ylim(app.UIAxes2,[-0.2 1.2]);
    xlim(app.UIAxes2,[0 2]); % Adjusted x-limit for 2 data points
    title(app.UIAxes2,'NOT Output');
    xlabel(app.UIAxes2, 'Time Step');
    ylabel(app.UIAxes2, 'Logic Level');
        end
```

NAND Gate:

Design view Output



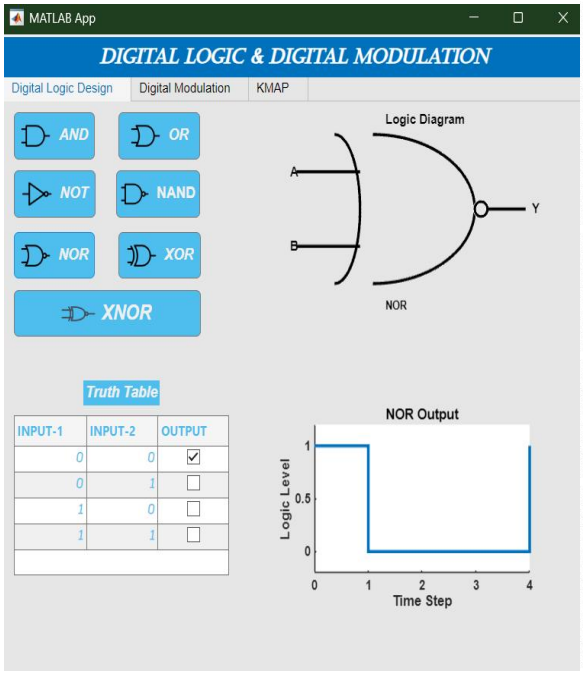
Code view



```
        % Button pushed function: NANDButton
        function NANDButtonPushed(app, event)
             % Truth table
    inputs = [0 0; 0 1; 1 0; 1 1];
    outputs = ~(inputs(:,1) & inputs(:,2));

    % Update table
    app.UITable.Data = table(inputs(:,1), inputs(:,2), outputs, ...
        'VariableNames', {'A','B','A_NAND_B'});

    % Gate diagram
    cla(app.UIAxes); hold(app.UIAxes,'on');

    % Input lines
    plot(app.UIAxes,[-0.5 0],[0.25 0.25],'k','LineWidth',2);
    plot(app.UIAxes,[-0.5 0],[0.75 0.75],'k','LineWidth',2);

    % AND body
    plot(app.UIAxes,[0 0],[0 1],'k','LineWidth',2);
    plot(app.UIAxes,[0 0.5],[0 0],'k','LineWidth',2);
    plot(app.UIAxes,[0 0.5],[1 1],'k','LineWidth',2);
    theta = linspace(-pi/2,pi/2,100);
    x = 0.5 + 0.5*cos(theta);
    y = 0.5 + 0.5*sin(theta);
    plot(app.UIAxes,x,y,'k','LineWidth',2);

    % Bubble
    viscircles(app.UIAxes,[1 0.5],0.05,'Color','k','LineWidth',2);

    % Output line
    plot(app.UIAxes,[1.05 1.3],[0.5 0.5],'k','LineWidth',2);

    % Labels
    text(app.UIAxes,-0.55,0.75,'A','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,-0.55,0.25,'B','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,1.35,0.5,'Y','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,0.1,-0.15,'NAND','FontSize',10,'FontWeight','bold');
    text(app.UIAxes,0.1,1.1,'Logic Diagram','FontSize',12,'FontWeight','bold');

    axis(app.UIAxes,'equal'); axis(app.UIAxes,'off');

    % Waveform
    cla(app.UIAxes2);
    t = 0:4;
    y = outputs([1:end 1]);
    stairs(app.UIAxes2,t,y,'LineWidth',2);
    ylim(app.UIAxes2,[-0.2 1.2]);
    xlim(app.UIAxes2,[0 4]);
    title(app.UIAxes2,'NAND Output');
    xlabel(app.UIAxes2, 'Time Step');
    ylabel(app.UIAxes2, 'Logic Level');
        end
```

NOR Gate:

Design view Output

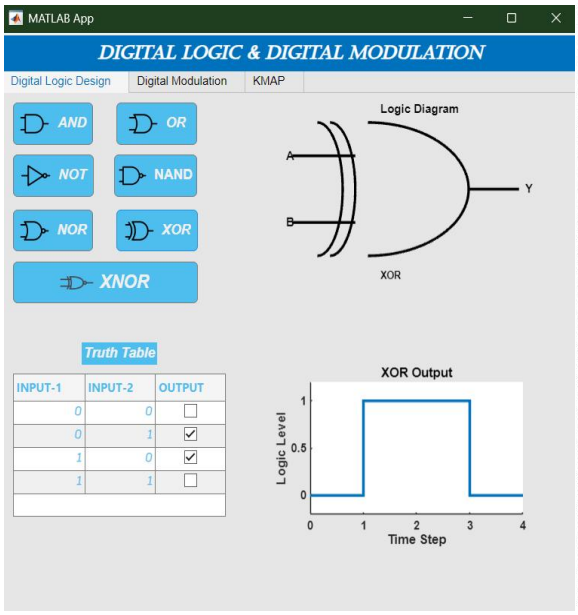


Code view

```
        % Button pushed function: NORButton
        function NORButtonPushed(app, event)
            % Truth table
    inputs = [0 0; 0 1; 1 0; 1 1];
    outputs = ~(inputs(:,1) | inputs(:,2));

    % Update table
    app.UITable.Data = table(inputs(:,1), inputs(:,2), outputs, ...
        'VariableNames', {'A','B','A_NOR_B'});

    % Gate diagram
    cla(app.UIAxes); hold(app.UIAxes,'on');

    % Input lines
    plot(app.UIAxes,[-0.6 -0.1],[0.25 0.25],'k','LineWidth',2);
    plot(app.UIAxes,[-0.6 -0.1],[0.75 0.75],'k','LineWidth',2);

    % OR body
    theta = linspace(-pi/2,pi/2,200);
    x1 = 0.8*cos(theta);
    y1 = 0.5 + 0.5*sin(theta);
    plot(app.UIAxes,x1,y1,'k','LineWidth',2);

    x2 = 0.2*cos(theta) - 0.3;
    y2 = 0.5 + 0.5*sin(theta);
    plot(app.UIAxes,x2,y2,'k','LineWidth',2);

    % Bubble
    viscircles(app.UIAxes,[0.85 0.5],0.05,'Color','k','LineWidth',2);

    % Output line
    plot(app.UIAxes,[0.9 1.2],[0.5 0.5],'k','LineWidth',2);

    % Labels
    text(app.UIAxes,-0.65,0.75,'A','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,-0.65,0.25,'B','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,1.25,0.5,'Y','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,0.1,-0.15,'NOR','FontSize',10,'FontWeight','bold');
    text(app.UIAxes,0.1,1.1,'Logic Diagram','FontSize',12,'FontWeight','bold');

    axis(app.UIAxes,'equal'); axis(app.UIAxes,'off');

     % Waveform
    cla(app.UIAxes2);
    t = 0:4;
    y = outputs([1:end 1]);
    stairs(app.UIAxes2,t,y,'LineWidth',2);
    ylim(app.UIAxes2,[-0.2 1.2]);
    xlim(app.UIAxes2,[0 4]);
    title(app.UIAxes2,'NOR Output');
    xlabel(app.UIAxes2, 'Time Step');
    ylabel(app.UIAxes2, 'Logic Level');
        end
```

XOR Gate:

Design view Output

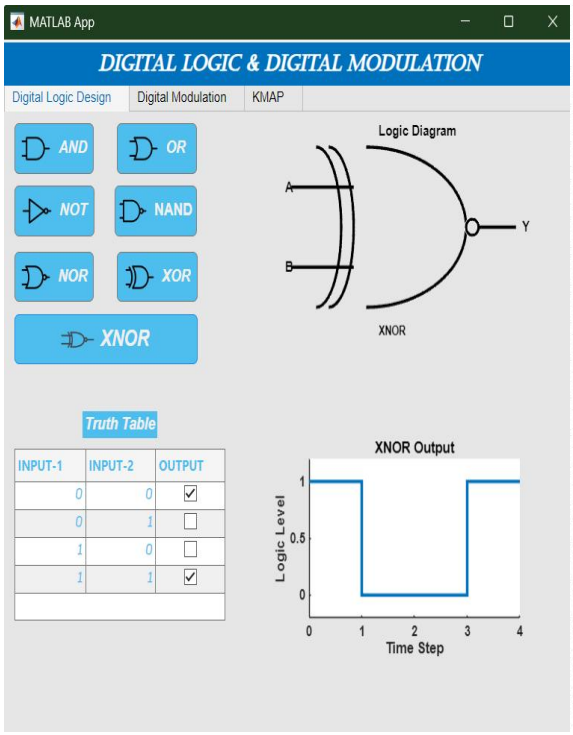


Code view

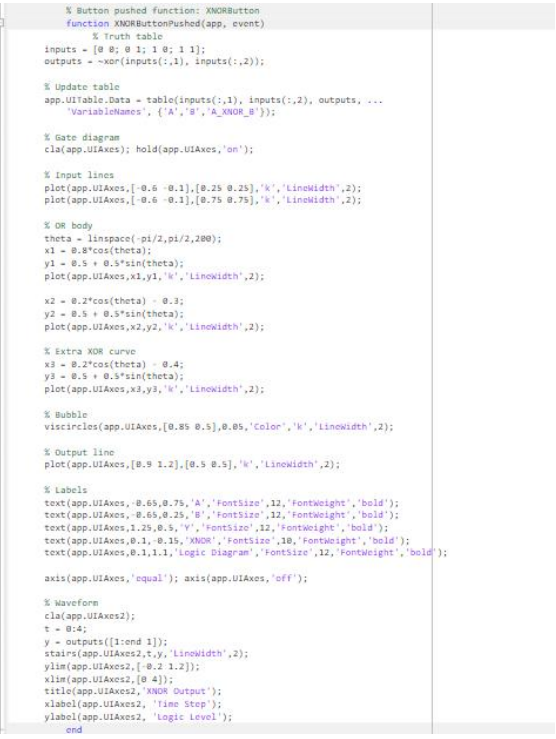
```
        % Button pushed function: XORButton
        function XORButtonPushed(app, event)
            % Truth table
    inputs = [0 0; 0 1; 1 0; 1 1];
    outputs = xor(inputs(:,1), inputs(:,2));

    % Update table
    app.UITable.Data = table(inputs(:,1), inputs(:,2), outputs, ...
        'VariableNames', {'A','B','A_XOR_B'});

    % Gate diagram
    cla(app.UIAxes); hold(app.UIAxes,'on');

    % Input lines
    plot(app.UIAxes,[-0.6 -0.1],[0.25 0.25],'k','LineWidth',2);
    plot(app.UIAxes,[-0.6 -0.1],[0.75 0.75],'k','LineWidth',2);

    % Main OR arc (front)
    theta = linspace(-pi/2,pi/2,200);
    x1 = 0.8*cos(theta);
    y1 = 0.5 + 0.5*sin(theta);
    plot(app.UIAxes,x1,y1,'k','LineWidth',2);

    % Back arc (concave side)
    x2 = 0.2*cos(theta) - 0.3;
    y2 = 0.5 + 0.5*sin(theta);
    plot(app.UIAxes,x2,y2,'k','LineWidth',2);

    % Extra XOR arc (parallel to back curve)
    x3 = 0.2*cos(theta) - 0.4;
    y3 = 0.5 + 0.5*sin(theta);
    plot(app.UIAxes,x3,y3,'k','LineWidth',2);

    % Output line
    plot(app.UIAxes,[0.8 1.2],[0.5 0.5],'k','LineWidth',2);

    % Labels
    text(app.UIAxes,-0.65,0.75,'A','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,-0.65,0.25,'B','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,1.25,0.5,'Y','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,0.1,-0.15,'XOR','FontSize',10,'FontWeight','bold');
    text(app.UIAxes,0.1,1.1,'Logic Diagram','FontSize',12,'FontWeight','bold');

    axis(app.UIAxes,'equal'); axis(app.UIAxes,'off');

    % Waveform
    cla(app.UIAxes2);
    t = 0:4;
    y = outputs([1:end 1]);
    stairs(app.UIAxes2,t,y,'LineWidth',2);
    ylim(app.UIAxes2,[-0.2 1.2]);
    xlim(app.UIAxes2,[0 4]);
    title(app.UIAxes2,'XOR Output');
    xlabel(app.UIAxes2, 'Time Step');
    ylabel(app.UIAxes2, 'Logic Level');
        end
```

XNOR Gate:

Design view Output



Code view



```
        % Button pushed function: XNORButton
        function XNORButtonPushed(app, event)
            % Truth table
    inputs = [0 0; 0 1; 1 0; 1 1];
    outputs = ~xor(inputs(:,1), inputs(:,2));

    % Update table
    app.UITable.Data = table(inputs(:,1), inputs(:,2), outputs, ...
        'VariableNames', {'A','B','A_XNOR_B'});

    % Gate diagram
    cla(app.UIAxes); hold(app.UIAxes,'on');

    % Input lines
    plot(app.UIAxes,[-0.6 -0.1],[0.25 0.25],'k','LineWidth',2);
    plot(app.UIAxes,[-0.6 -0.1],[0.75 0.75],'k','LineWidth',2);

    % OR body
    theta = linspace(-pi/2,pi/2,200);
    x1 = 0.8*cos(theta);
    y1 = 0.5 + 0.5*sin(theta);
    plot(app.UIAxes,x1,y1,'k','LineWidth',2);

    x2 = 0.2*cos(theta) - 0.3;
    y2 = 0.5 + 0.5*sin(theta);
    plot(app.UIAxes,x2,y2,'k','LineWidth',2);

    % Extra XOR curve
    x3 = 0.2*cos(theta) - 0.4;
    y3 = 0.5 + 0.5*sin(theta);
    plot(app.UIAxes,x3,y3,'k','LineWidth',2);

    % Bubble
    viscircles(app.UIAxes,[0.85 0.5],0.05,'Color','k','LineWidth',2);

    % Output line
    plot(app.UIAxes,[0.9 1.2],[0.5 0.5],'k','LineWidth',2);

    % Labels
    text(app.UIAxes,-0.65,0.75,'A','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,-0.65,0.25,'B','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,1.25,0.5,'Y','FontSize',12,'FontWeight','bold');
    text(app.UIAxes,0.1,-0.15,'XNOR','FontSize',10,'FontWeight','bold');
    text(app.UIAxes,0.1,1.1,'Logic Diagram','FontSize',12,'FontWeight','bold');

    axis(app.UIAxes,'equal'); axis(app.UIAxes,'off');

    % Waveform
    cla(app.UIAxes2);
    t = 0:4;
    y = outputs([1:end 1]);
    stairs(app.UIAxes2,t,y,'LineWidth',2);
    ylim(app.UIAxes2,[-0.2 1.2]);
    xlim(app.UIAxes2,[0 4]);
    title(app.UIAxes2,'XNOR Output');
    xlabel(app.UIAxes2, 'Time Step');
    ylabel(app.UIAxes2, 'Logic Level');
        end
```

Conclusion

The Digital Logic & Digital Modulation Simulator is a versatile learning and teaching tool. It enhances practical understanding of core concepts in Digital Electronics and Communication Systems through interactive simulations.