



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

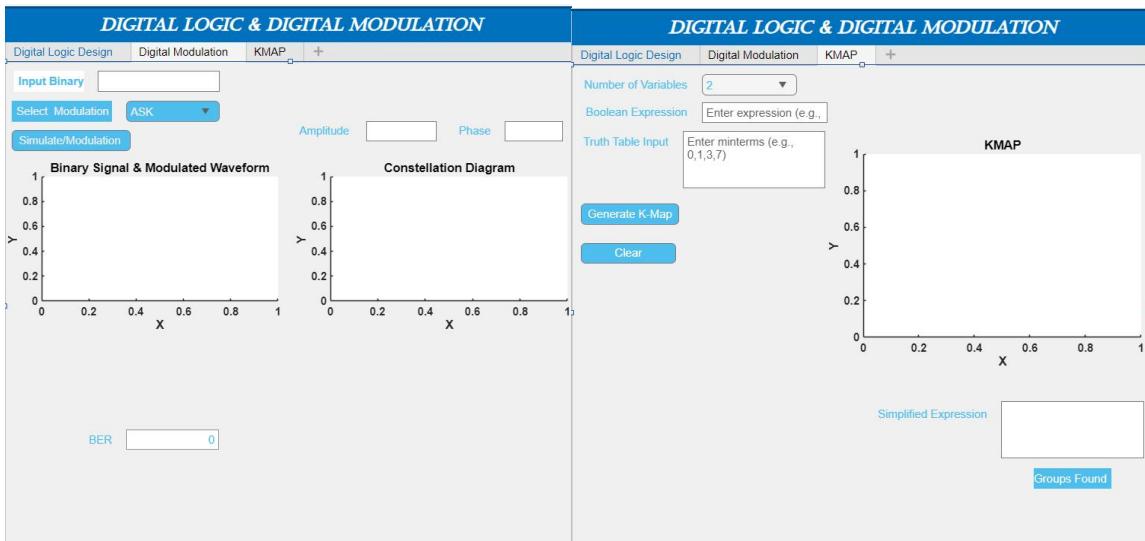
- **Digital Logic Design Tab**: Includes buttons for logic gates (AND, OR, NOT, NAND, NOR, XOR, XNOR).
- Generates truth tables based on selected logic gates.
- Displays corresponding waveforms for input and output signals.
- **Digital Modulation Tab**: Allows selection of modulation schemes (ASK, PSK,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

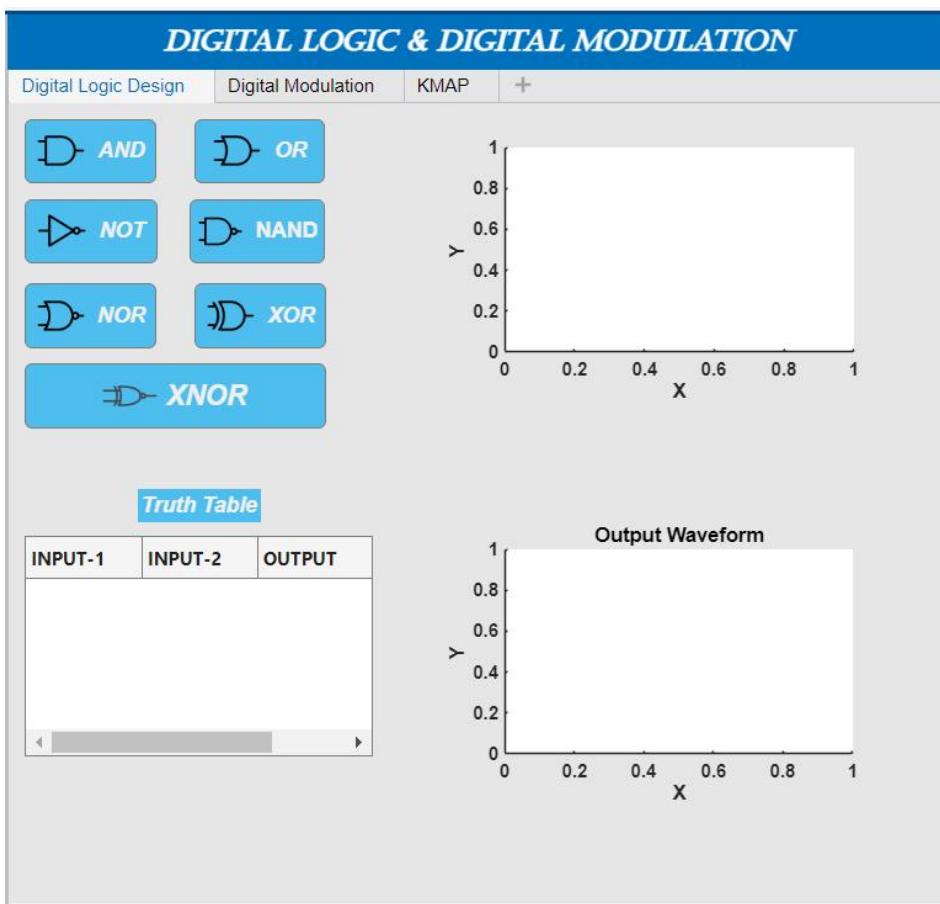
The main objective of this project is to provide an educational and simulation tool that helps students and researchers understand the fundamentals of digital logic , digital modulation and Kmap of logic design. By combining truth table generation, logic gate simulation, Kmap, and modulation visualization,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

APP's Design View

Prototype



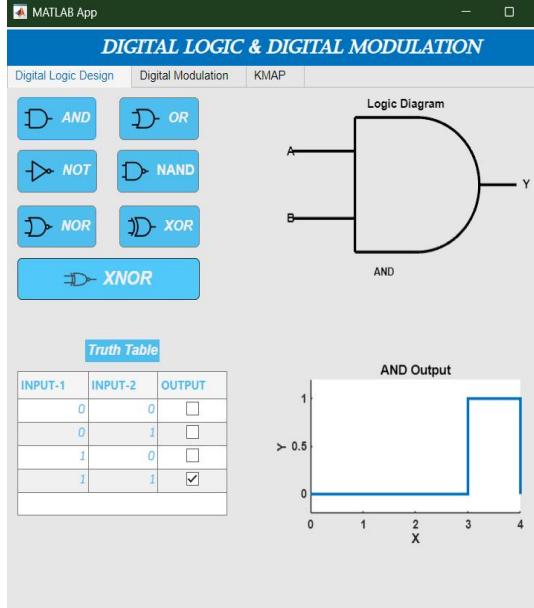
Working



Different Segments and Outputs

And Gate:

Design view Output



Code view

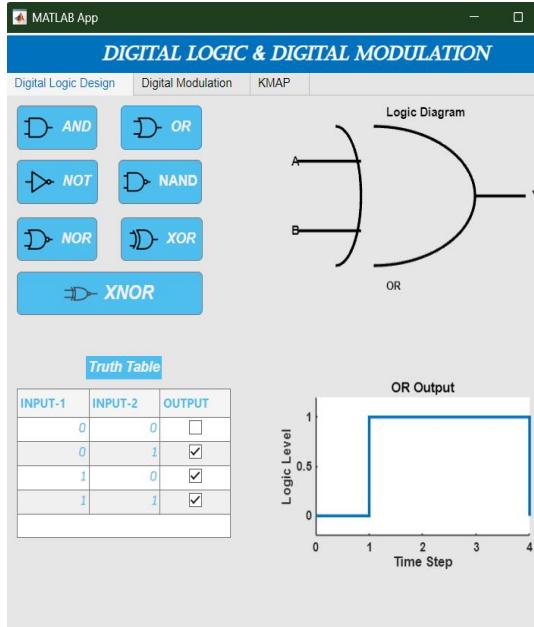
```

52 % Button pushed function: ANDButton
53 function ANDButtonPushed(app, event)
54     inputs = [0 0; 0 1; 1 0; 1 1];
55     outputs = inputs(:,1) & inputs(:,2);
56
57 % Update table
58 app.UITable.Data = table(inputs(:,1), inputs(:,2), outputs, ...
59    'VariableNames', {'A','B','A_AND_B'});
60
61 % Update Logic Diagram (proper AND gate shape)
62 cla(app.UIAxes); hold(app.UIAxes,'on');
63
64 %% Inputs lines
65 plot(app.UIAxes,[0 0],[0.25 0.25],'k','LineWidth',2);
66 plot(app.UIAxes,[0 0],[0.75 0.75],'k','LineWidth',2);
67
68 %% AND gate body
69 % Draw left rectangle
70 plot(app.UIAxes,[0 0],[0 1],'k','LineWidth',2); % left vertical
71 plot(app.UIAxes,[0 0.5],[0 0],'k','LineWidth',2); % bottom horizontal
72 plot(app.UIAxes,[0 0.5],[1 1],'k','LineWidth',2); % top horizontal
73
74 % Draw semicircle on right side
75 theta = linspace(-pi/2,pi/2,100);
76 x = 0.5 + 0.5*cos(theta); % shift right by 0.5
77 y = 0.5 + 0.5*sin(theta); % centered vertically at 0.5
78 plot(app.UIAxes,x,y,'k','LineWidth',2);
79
80 %% Output line
81 plot(app.UIAxes,[1 1.3],[0.5 0.5],'k','LineWidth',2);
82
83 %% Labels
84 text(app.UIAxes,-0.55,0.75,'A','FontSize',12,'FontWeight','bold');
85 text(app.UIAxes,-0.55,0.25,'B','FontSize',12,'FontWeight','bold');
86 text(app.UIAxes,0.35,0.5,'Y','FontSize',12,'FontWeight','bold');
87 text(app.UIAxes,0.15,-0.15,'AND','FontSize',10,'FontWeight','bold');
88 text(app.UIAxes,0.1,1.1,'Logic Diagram','FontSize',12,'FontWeight','bold');
89
90 %% Formatting
91 axis(app.UIAxes,'equal');
92 axis(app.UIAxes,'off');
93
94 % Plot waveform (Output vs Time/Index)
95 stairs(app.UIAxes2,1:4,outputs([1:end 1]),'LineWidth',2)
96 ylim(app.UIAxes2,[-0.2 1.2]);
97 title(app.UIAxes2, 'AND Output')
98 end

```

OR Gate:

Design view Output



Code view

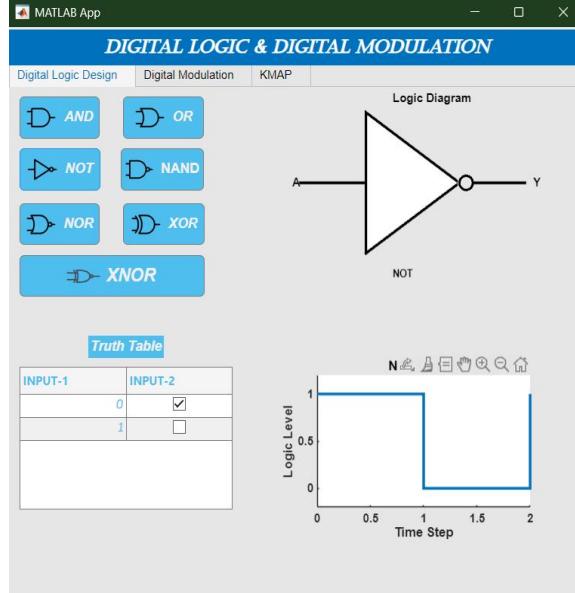
```

100 % Button pushed function: ORButton
101 function ORButtonPushed(app, event)
102     % Truth table
103     inputs = [0 0; 0 1; 1 0; 1 1];
104     outputs = inputs(:,1) | inputs(:,2);
105
106 % update table
107 app.UITable.Data = table(inputs(:,1), inputs(:,2), outputs, ...
108    'VariableNames', {'A','B','A_OR_B'});
109
110 %% diagram
111 cla(app.UIAxes); hold(app.UIAxes,'on');
112
113 %% Input lines
114 plot(app.UIAxes,[-0.6 -0.1],[0.25 0.25],'k','LineWidth',2);
115 plot(app.UIAxes,[-0.6 -0.1],[0.75 0.75],'k','LineWidth',2);
116
117 %% main OR arc (front)
118 theta = linspace(-pi/2,pi/2,200);
119 x1 = 0.8*cos(theta);
120 y1 = 0.5 + 0.5*sin(theta);
121 plot(app.UIAxes,x1,y1,'k','LineWidth',2);
122
123 %% back arc (concave side)
124 x2 = 0.2*cos(theta) - 0.3;
125 y2 = 0.5 + 0.5*sin(theta);
126 plot(app.UIAxes,x2,y2,'k','LineWidth',2);
127
128 %% Output line
129 plot(app.UIAxes,[0.8 1.2],[0.5 0.5],'k','LineWidth',2);
130
131 %% Labels
132 text(app.UIAxes,-0.65,0.75,'A','FontSize',12,'FontWeight','bold');
133 text(app.UIAxes,-0.65,0.25,'B','FontSize',12,'FontWeight','bold');
134 text(app.UIAxes,0.25,0.5,'Y','FontSize',12,'FontWeight','bold');
135 text(app.UIAxes,0.1,-0.15,'OR','FontSize',10,'FontWeight','bold');
136 text(app.UIAxes,0.1,1.1,'Logic Diagram','FontSize',12,'FontWeight','bold');
137
138 axis(app.UIAxes,'equal'); axis(app.UIAxes,'off');
139
140 %% Waveform
141 cla(app.UIAxes2);
142 t = 1:4;
143 y = outputs([1:end 1]);
144 stairs(app.UIAxes2,t,y,'LineWidth',2);
145 ylim(app.UIAxes2,[-0.2 1.2]);
146 xlim(app.UIAxes2,[0 4]);
147 title(app.UIAxes2, 'OR Output'); xlabel(app.UIAxes2, 'Time Step'); ylabel(app.UIAxes2, 'Logic Level');
148 end

```

NOT Gate:

Design view Output



Code view

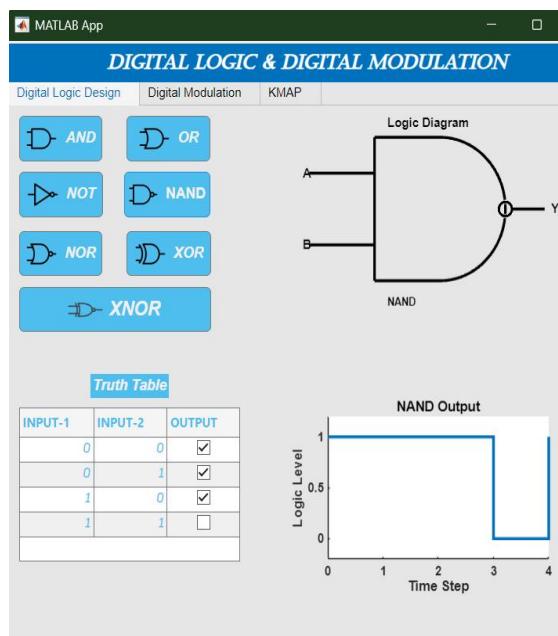
```

151 % Button pushed function: NOTButtonPushed
152 function NOTButtonPushed(app, event)
153 % Truth table
154 inputs = [0; 1];
155 outputs = ~inputs;
156
157 % Update table
158 app.UITable.Data = table(inputs, outputs, ...
159 'VariableNames', {'A','NOT_A'});
160
161 % Gate diagram
162 cla(app.UIAxes); hold(app.UIAxes,'on');
163
164 % Input line
165 plot(app.UIAxes,[0.5 0],[0.5 0.5],'k','LineWidth',2);
166
167 % Triangle
168 fill(app.UIAxes,[0 0 0.7],[0 1 0.5],'w','EdgeColor','k','LineWidth',2);
169
170 % Bubble
171 viscircles(app.UIAxes,[0.75 0.5],0.05,'Color','k','LineWidth',2);
172
173 % Output line
174 plot(app.UIAxes,[0.8 1.2],[0.5 0.5],'k','LineWidth',2);
175
176 % Labels
177 text(app.UIAxes,-0.55,0.5,'A','FontSize',12,'FontWeight','bold');
178 text(app.UIAxes,1.25,0.5,'Y','FontSize',12,'FontWeight','bold');
179 text(app.UIAxes,0.2,-0.15,'NOT','FontSize',10,'FontWeight','bold');
180 text(app.UIAxes,0.2,1.1,'Logic Diagram','FontSize',12,'FontWeight','bold');
181 axis(app.UIAxes,'equal'); axis(app.UIAxes,'off');
182
183 % Waveform
184 cla(app.UIAxes2);
185 t = 0:2;
186 y = outputs([1:end 1]);
187 stairs(app.UIAxes2,t,y,'LineWidth',2);
188 ylim(app.UIAxes2,[0 2.1]);
189 xlim(app.UIAxes2,[0 2]); % adjusted x-limit for 2 data points
190 title(app.UIAxes2,'NOT Output');
191 xlabel(app.UIAxes2,'Time Step');
192 ylabel(app.UIAxes2,'Logic Level');
193 end

```

NAND Gate:

Design view Output



Code view

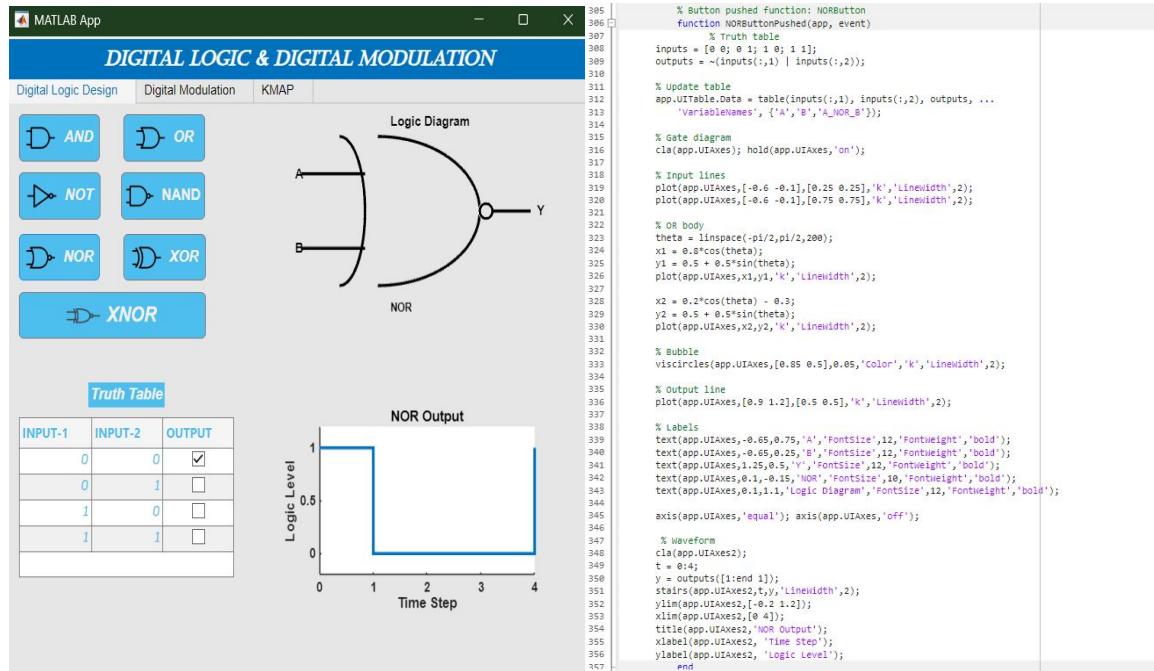
```

252 % Button pushed function: NANDButtonPushed
253 function NANDButtonPushed(app, event)
254 % Truth table
255 inputs = [0 0; 0 1; 1 0; 1 1];
256 outputs = ~inputs(:,1) & inputs(:,2);
257
258 % Update table
259 app.UITable.data = table(inputs(:,1), inputs(:,2), outputs, ...
260 'VariableNames', {'A','B','A_NAND_B'});
261
262 % Gate diagram
263 cla(app.UIAxes); hold(app.UIAxes,'on');
264
265 % Input lines
266 plot(app.UIAxes,[0.5 0],[0.25 0.25],'k','LineWidth',2);
267 plot(app.UIAxes,[0.5 0],[0.75 0.75],'k','LineWidth',2);
268
269 % AND body
270 plot(app.UIAxes,[0 0],[0 1],'k','LineWidth',2);
271 plot(app.UIAxes,[0 0.5],[0 0.5],'k','LineWidth',2);
272 plot(app.UIAxes,[0 0.5],[1 1],'k','LineWidth',2);
273 theta = linspace(-pi/2,pi/2,100);
274 x = 0.5 + 0.5*cos(theta);
275 y = 0.5 + 0.5*sin(theta);
276 plot(app.UIAxes,x,y,'k','LineWidth',2);
277
278 % Bubble
279 viscircles(app.UIAxes,[1 0.5],0.05,'Color','k','LineWidth',2);
280
281 % Output line
282 plot(app.UIAxes,[1.05 1.3],[0.5 0.5],'k','LineWidth',2);
283
284 % Labels
285 text(app.UIAxes,-0.55,0.75,'A','FontSize',12,'FontWeight','bold');
286 text(app.UIAxes,-0.55,0.25,'B','FontSize',12,'FontWeight','bold');
287 text(app.UIAxes,1.35,0.5,'Y','FontSize',12,'FontWeight','bold');
288 text(app.UIAxes,0.1,0.15,'NAND','FontSize',10,'FontWeight','bold');
289 text(app.UIAxes,0.1,1.1,'Logic Diagram','FontSize',12,'FontWeight','bold');
290 axis(app.UIAxes,'equal'); axis(app.UIAxes,'off');
291
292 % Waveform
293 cla(app.UIAxes2);
294 t = 0:4;
295 y = outputs([1:end 1]);
296 stairs(app.UIAxes2,t,y,'LineWidth',2);
297 ylim(app.UIAxes2,[0 2.1]);
298 xlim(app.UIAxes2,[0 4]);
299 title(app.UIAxes2,'NAND Output');
300 xlabel(app.UIAxes2,'Time Step');
301 ylabel(app.UIAxes2,'Logic Level');
302 end

```

NOR Gate:

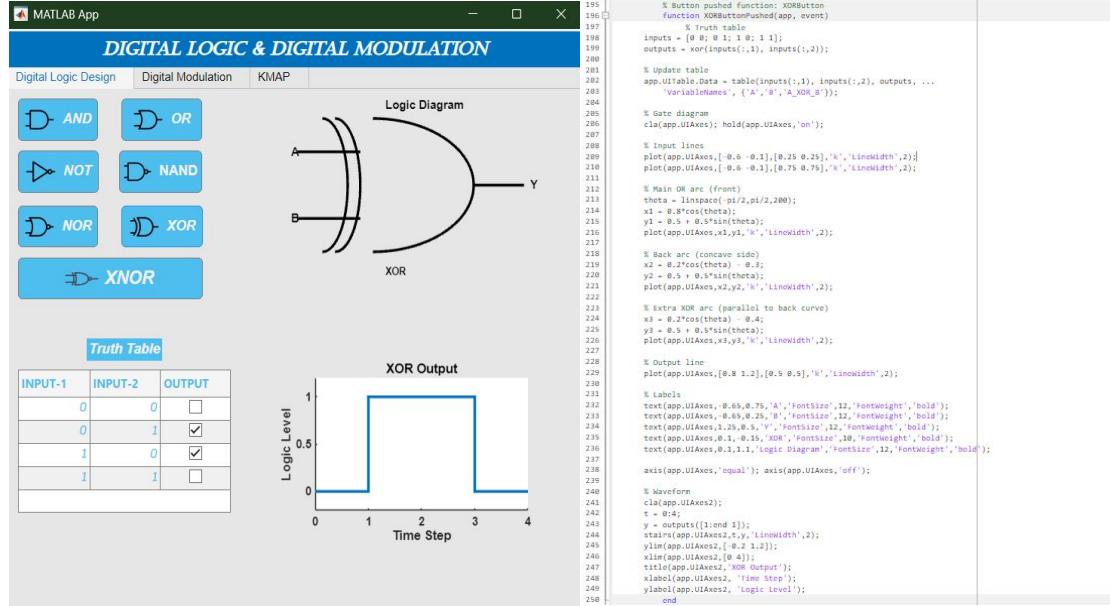
Design view Output



Code view

XOR Gate:

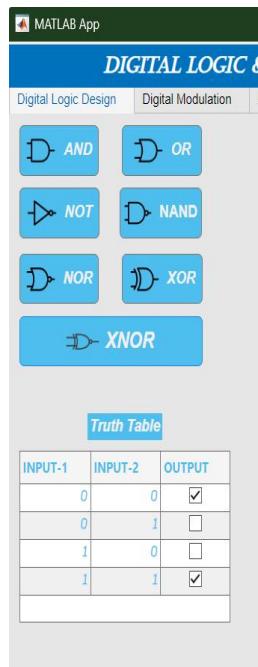
Design view Output



Code view

XNOR Gate:

Design view Output



Code view

```

359 % Button pushed function: XNORButton
360 function XNORButtonPushed(app, event)
361 % Truth table
362 inputs = [0 0; 0 1; 1 0; 1 1];
363 outputs = ~xor(inputs(:,1), inputs(:,2));
364
365 % Update table
366 app.UITable.Data = table(inputs(:,1), inputs(:,2), outputs, ...
367 'VariableNames', {'A','B','A_XNOR_B'});
368
369 % Gate diagram
370 cla(app.UIAxes); hold(app.UIAxes,'on');
371
372 % Input lines
373 plot(app.UIAxes,[-0.6 -0.1],[0.25 0.25],'k','LineWidth',2);
374 plot(app.UIAxes,[0.6 -0.1],[0.75 0.75],'k','LineWidth',2);
375
376 % OR body
377 theta = linspace(-pi/2,pi/2,200);
378 x1 = 0.8*cos(theta);
379 y1 = 0.5 + 0.5*sin(theta);
380 plot(app.UIAxes,x1,y1,'k','LineWidth',2);
381
382 x2 = 0.2*cos(theta) - 0.3;
383 y2 = 0.5 + 0.5*sin(theta);
384 plot(app.UIAxes,x2,y2,'k','LineWidth',2);
385
386 % Extra XNOR curve
387 x3 = 0.2*cos(theta) - 0.4;
388 y3 = 0.5 + 0.5*sin(theta);
389 plot(app.UIAxes,x3,y3,'k','LineWidth',2);
390
391 % Bubble
392 viscircles(app.UIAxes,[0.85 0.5],0.05,'Color','k','LineWidth',2);
393
394 % Output line
395 plot(app.UIAxes,[0.9 1.2],[0.5 0.5],'k','LineWidth',2);
396
397 % Labels
398 text(app.UIAxes, 0.65,0.75,'A','FontSize',12,'FontWeight','bold');
399 text(app.UIAxes, 0.65,0.25,'B','FontSize',12,'FontWeight','bold');
400 text(app.UIAxes,1.25,0.5,'Y','FontSize',12,'FontWeight','bold');
401 text(app.UIAxes,0.1, 0.15,'XNOR','FontSize',16,'FontWeight','bold');
402 text(app.UIAxes,0.1,1.1,'Logic Diagram','FontSize',12,'FontWeight','bold');
403
404 axis(app.UIAxes, 'equal'); axis(app.UIAxes, 'off');
405
406 % Waveform
407 cla(app.UIAxes2);
408 t = 0:4;
409 y = outputs([1:end 1]);
410 stem(app.UIAxes2,t,y,'LineWidth',2);
411 ylim(app.UIAxes2,[0 1.2]);
412 xlim(app.UIAxes2,[0 4]);
413 title(app.UIAxes2,'XNOR Output');
414 xlabel(app.UIAxes2,'Time Step');
415 ylabel(app.UIAxes2,'Logic level');
416 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.