Module 6: Design and Analysis of Combinational Circuit

1. Problem

ABCD-to-seven-segment converter is a combinational circuit that accepts a decimal digit in BCD and generates the appropriate outputs for selection of segments in a display indicator used for displaying the decimal digit. The outputs of the converter select the corresponding segments in the display as shown in Figure 1. No segments should be selected for invalid inputs. Design the BCD-to-seven-segment circuit.

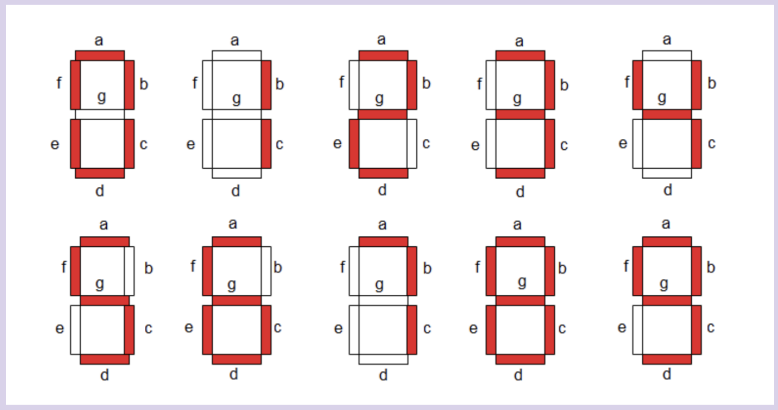


Figure 1. BCD-to-Seven-Segment Converter Output

2. Solution

2.1 Truth Table

To design the BCD-to-seven-segment converter combinational circuit, a truth table must first be constructed (refer to Table 1). The table columns are partitioned accordingly. The first column group, 4-Bit Binary Inputs, lists all possible 4-bit binary inputs, ranging from 0000 to 1111. The second column group, In BCD, identifies inputs with valid BCD representations and provides their corresponding BCD values. On the other hand, inputs that do not have a valid BCD representation are marked as "invalid." The third column group, Decimal Values, represents the equivalent decimal digits of all 4-bit binary inputs, ranging from 0 to 15. Finally, the last column group, 7-Segment Output Display, displays the corresponding activation pattern for the seven-segment display. For invalid inputs, all segment activation patterns are denoted as 'x,' representing a Don't Care condition.

2.2 Karnaugh Map to SOP function

Having constructed the corresponding truth table, writing the SOP functions follows. To do so, Karnaugh Maps will be constructed to derive the SOP functions for every segment there is (refer to Figures 2 to 8).

2.3 SOP Function to Circuit Diagram

Having derived the SOP functions for each segment (a to f) (refer to Figure 1), the corresponding circuit diagrams will be constructed using Logisim Evolution version 3.90.

Each segment's SOP function will be implemented separately using the software's "Add Circuit" feature. While the BCD-to-seven-segment converter combinational circuit can be implemented without this separation, doing so helps reduce workspace clutter and improve organization.

The main circuit will be placed in the "main" tab, with each segment's SOP function assigned to its respective subcircuit:

- segment-a → "circuit_a" tab
- segment-b → "circuit_b" tab
- up to segment-f → "circuit_f" tab

(refer to Figures 9 to 15)

Angela Denise Z. Almazan
CMSC 130: Logic Design & Digital Computer Circuits

2.4 BCD-to-seven-segment Converter Combinational Circuit

Having converted the SOP functions into their corresponding circuit diagrams and assembled the main circuit for the BCD-to-seven-segment converter, the design is now complete. The main circuit is now ready for input-output testing, where it accepts valid BCD inputs and generates the corresponding decimal digit on the display.

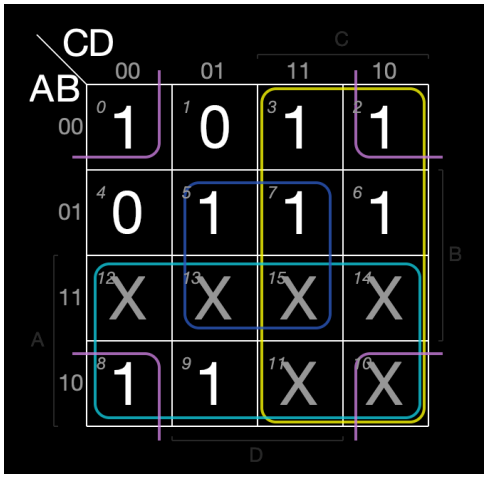
As stated in the problem, for invalid inputs, no segments should be selected. However, due to software constraints, Logisim Evolution v3.90 merely uses a basic decoder for BCD-to-seven-segment conversion, which may still generate unintended segment activation patterns for invalid inputs.

3. Final Output Documentation

Documented below are some snapshots of the output display from the designed BCD-to-seven-segment circuit (refer to Figures 16 to 18).

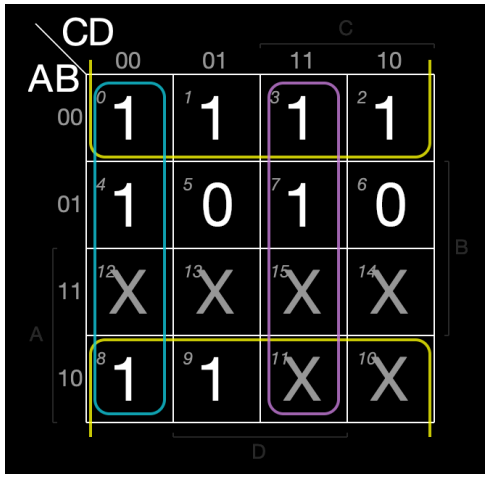
4-Bit Binary Inputs				In BCD	Decimal Values	7-Segment Output Display						
A	B	C	D			a	b	c	d	e	f	g
0	0	0	0	0000	0	1	1	1	1	1	1	0
0	0	0	1	0001	1	0	1	1	0	0	0	0
0	0	1	0	0010	2	1	1	0	1	1	0	1
0	0	1	1	0011	3	1	1	1	1	0	0	1
0	1	0	0	0100	4	0	1	1	0	0	1	1
0	1	0	1	0101	5	1	0	1	1	0	1	1
0	1	1	0	0110	6	1	0	1	1	1	1	1
0	1	1	1	0111	7	1	1	1	0	0	0	0
1	0	0	0	1000	8	1	1	1	1	1	1	1
1	0	0	1	1001	9	1	1	1	1	0	1	1
1	0	1	0	invalid	10	x	x	x	x	x	x	x
1	0	1	1	invalid	11	x	x	x	x	x	x	x
1	1	0	0	invalid	12	x	x	x	x	x	x	x
1	1	0	1	invalid	13	x	x	x	x	x	x	x
1	1	1	0	invalid	14	x	x	x	x	x	x	x
1	1	1	1	invalid	15	x	x	x	x	x	x	x

Table 1. Truth Table



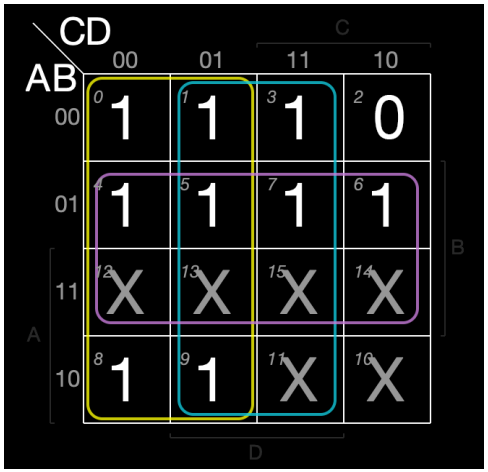
a = B'D' + BD + C + A

Figure 2. a-segment Karnaugh Map



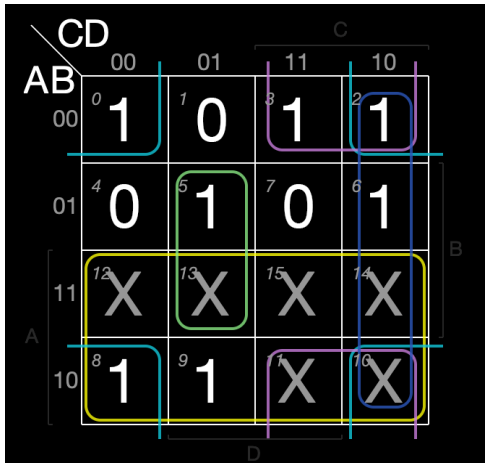
b = B' + C'D' + CD

Figure 3. b-segment Karnaugh Map



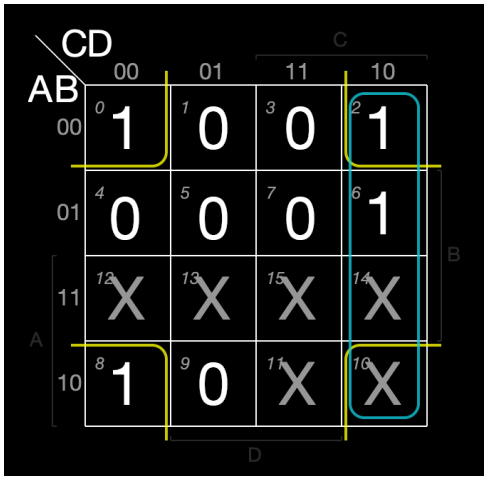
c = C' + D + B

Figure 4. c-segment Karnaugh Map



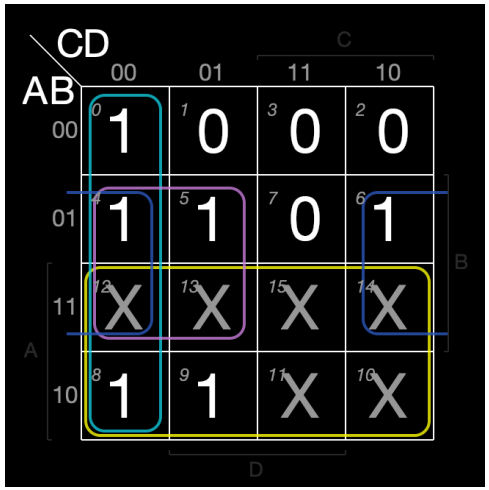
d = B'D' + B'C + CD' + BC'D + A

Figure 5. d-segment Karnaugh Map



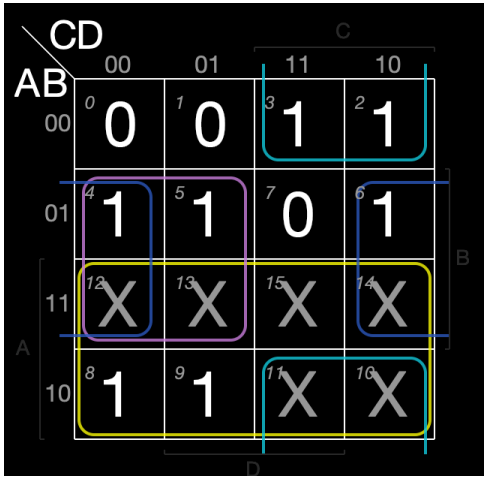
e = B'D' + CD'

Figure 6. e-segment Karnaugh Map



f = A + BD' + BC' + C'D'

Figure 7. f-segment Karnaugh Map



g = A + BC' + B'C + BD'

Figure 8. g-segment Karnaugh Map

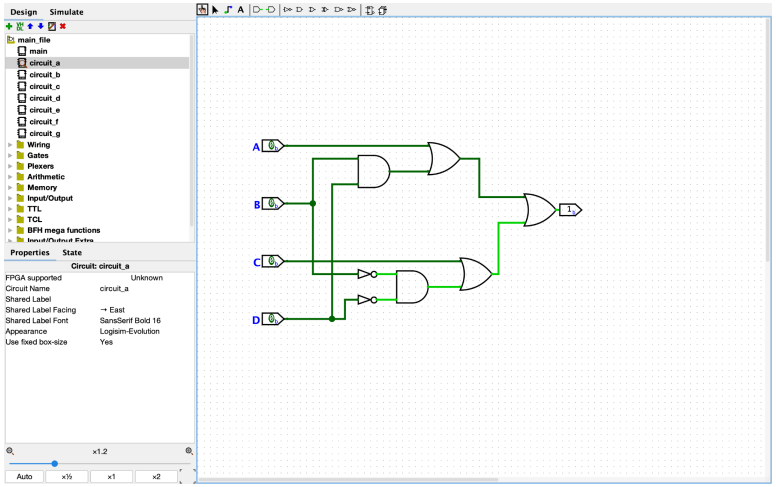


Figure 9. circuit_a

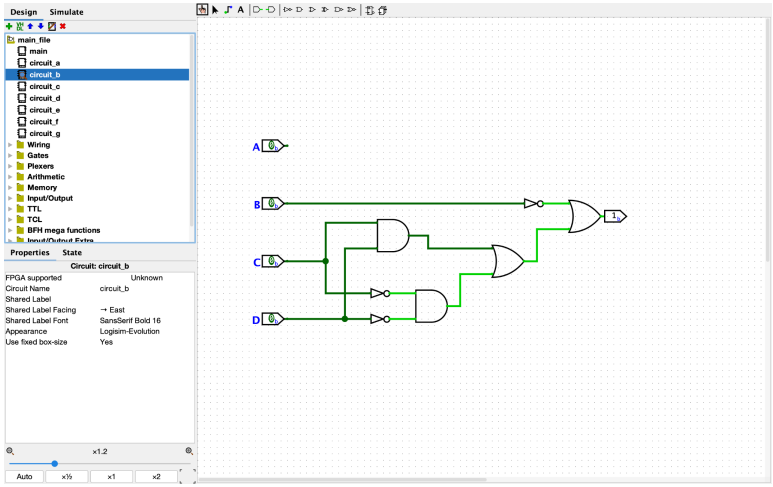


Figure 10. circuit_b

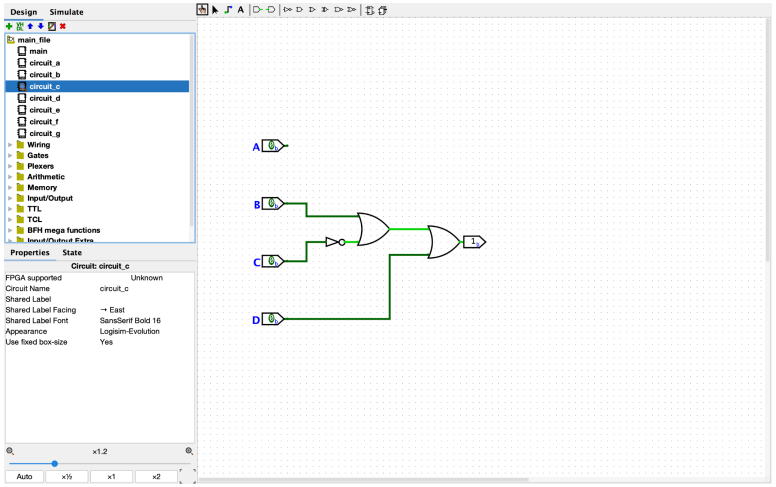


Figure 11. circuit_c

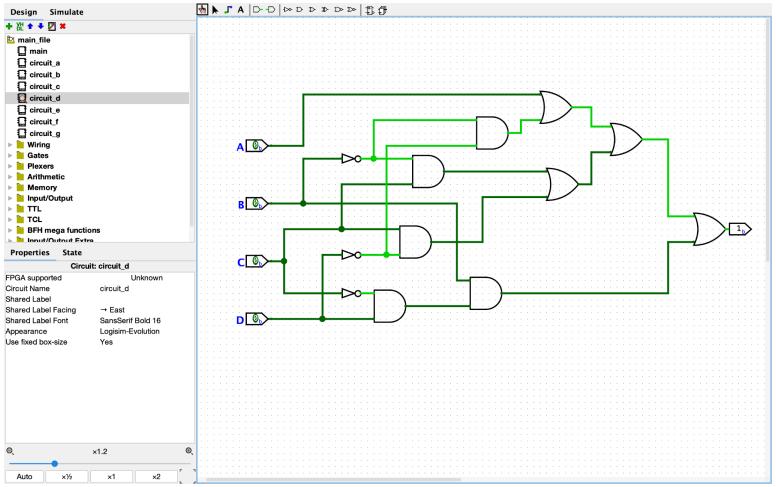


Figure 12. circuit_d

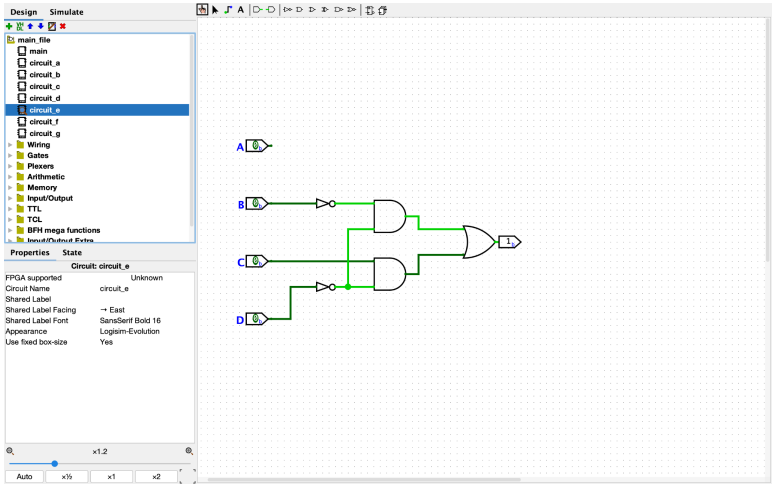


Figure 13. circuit_e

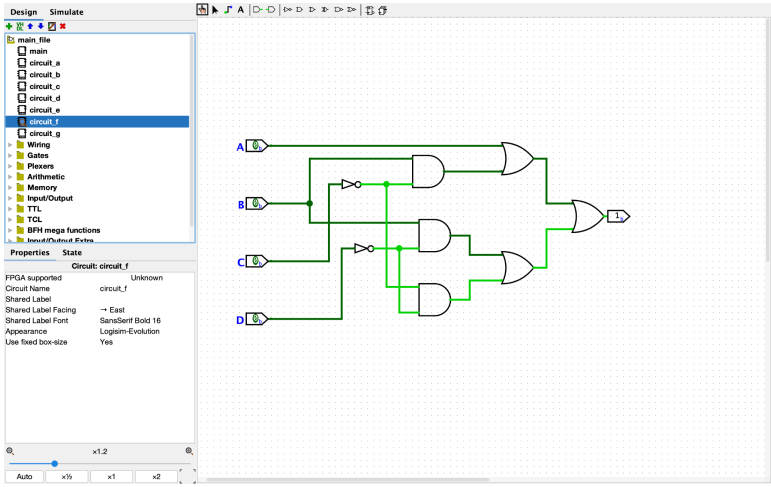


Figure 14. circuit_f

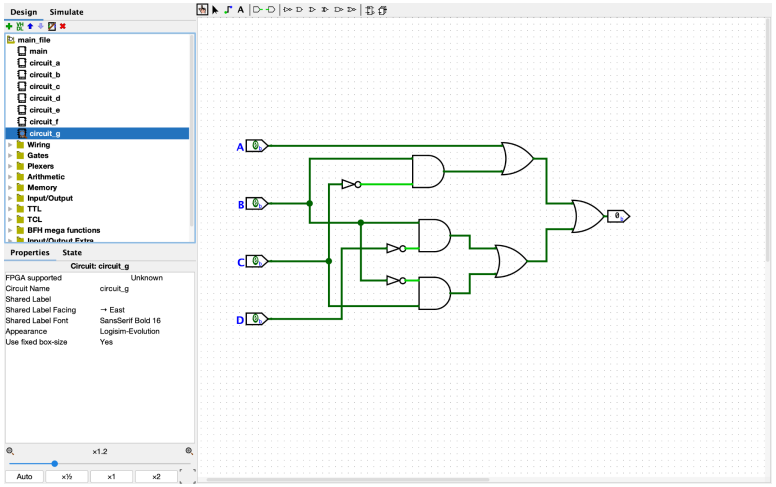


Figure 15. circuit_g

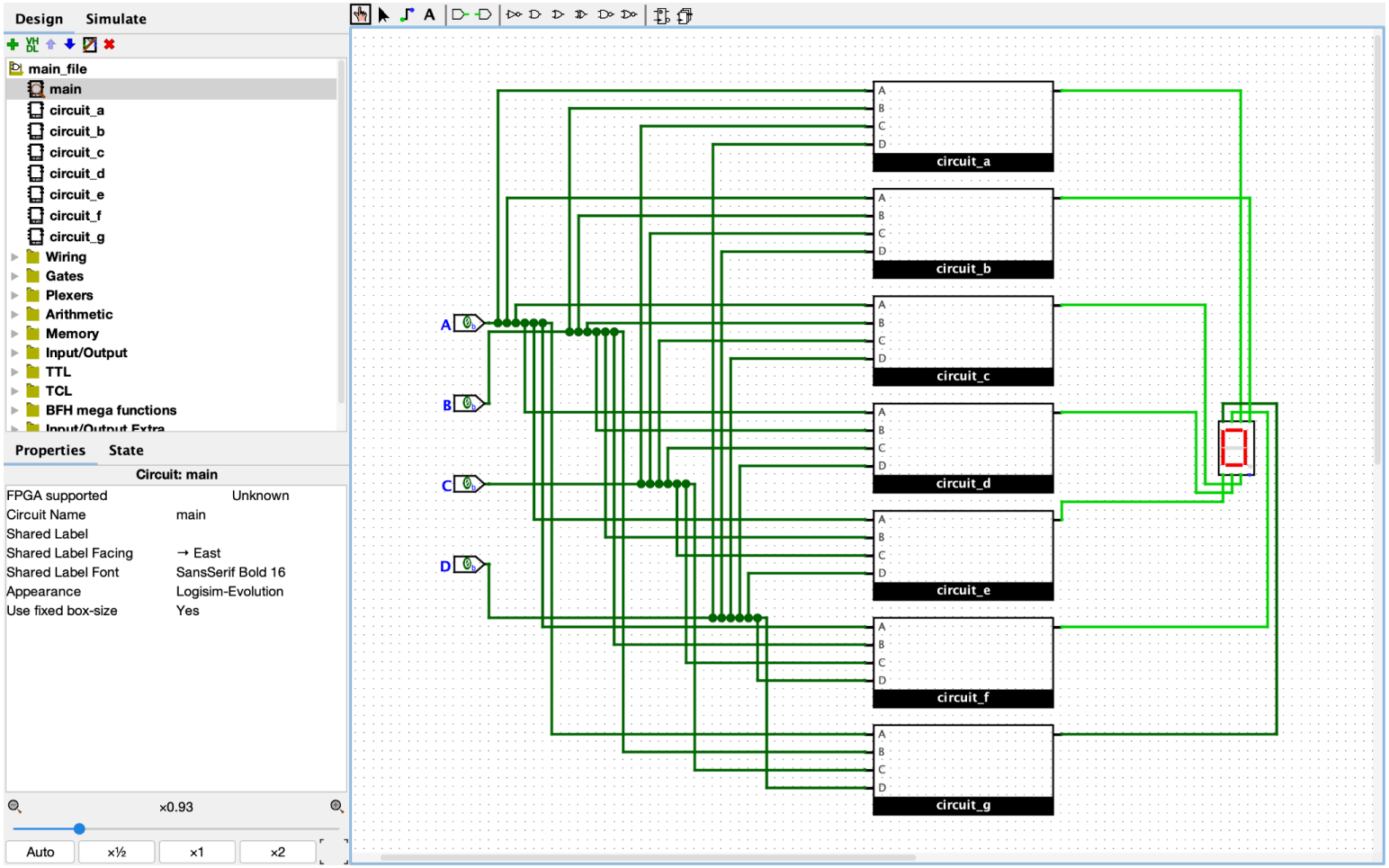


Figure 16. 0000

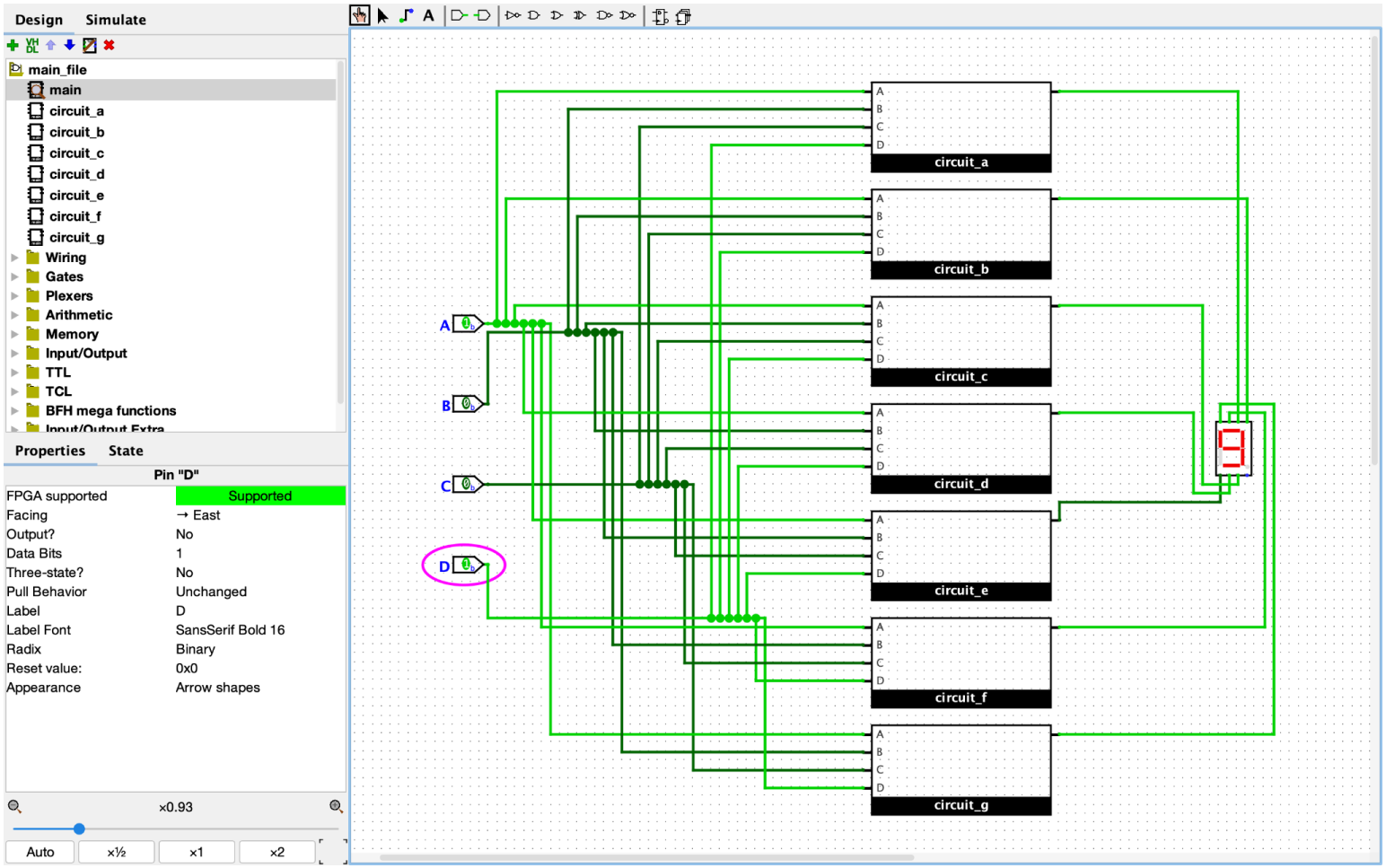


Figure 17. 1001

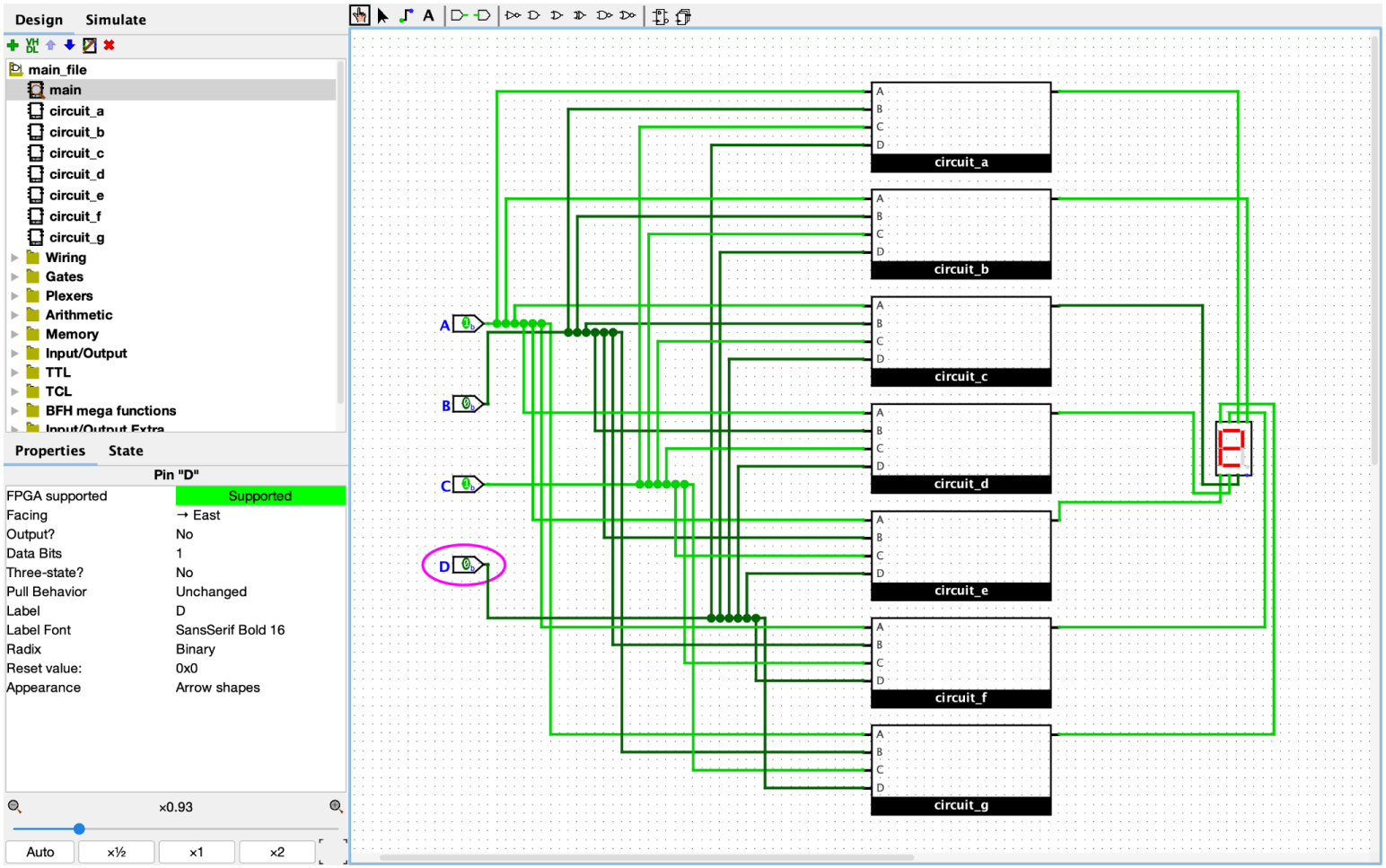


Figure 18. Invalid BCD Sample