EIE2810 Digital Systems Design Laboratory

Laboratory Report #5

Name: Student ID:
Date: 2024.4.19

The Chinese University of Hong Kong, Shenzhen

Student ID: Name:

- Experiment A: use D flip-flop to upgrade a function generator to a word generator
- Experiment B: design a looped counter by D flip-flop
- Experiment C: design a responder module based on D flip-flop, 8-3 line encoder and 7-segment module

1. Experiment A

1.1 Design

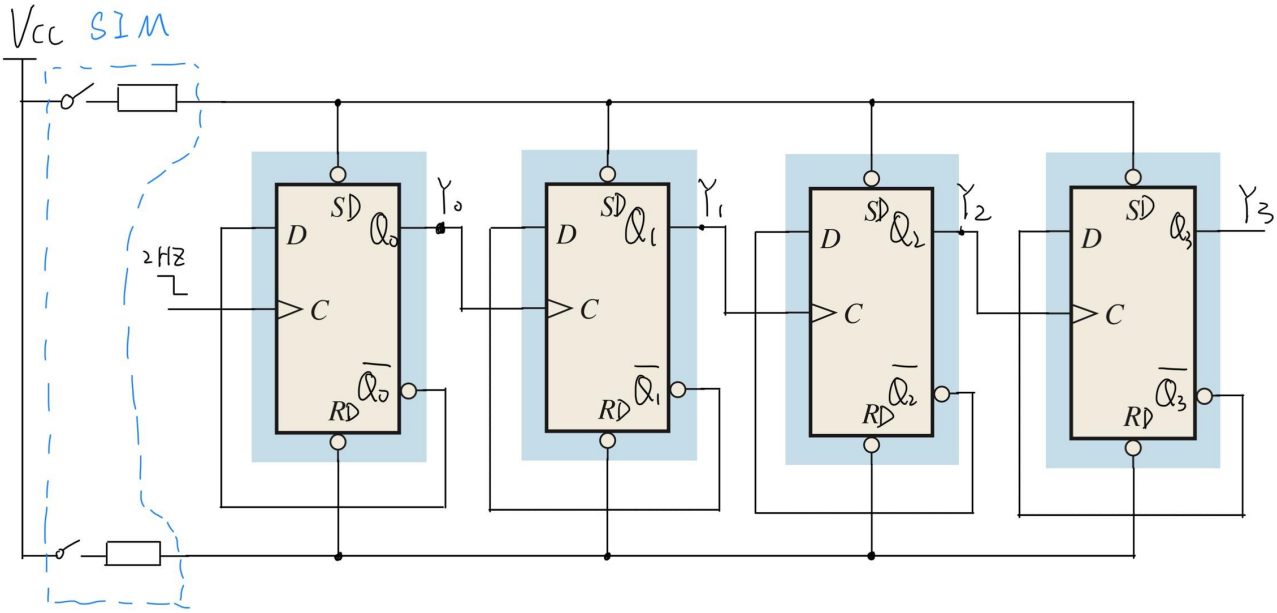


Figure 1. Design of 4 channel word generator (gate level)

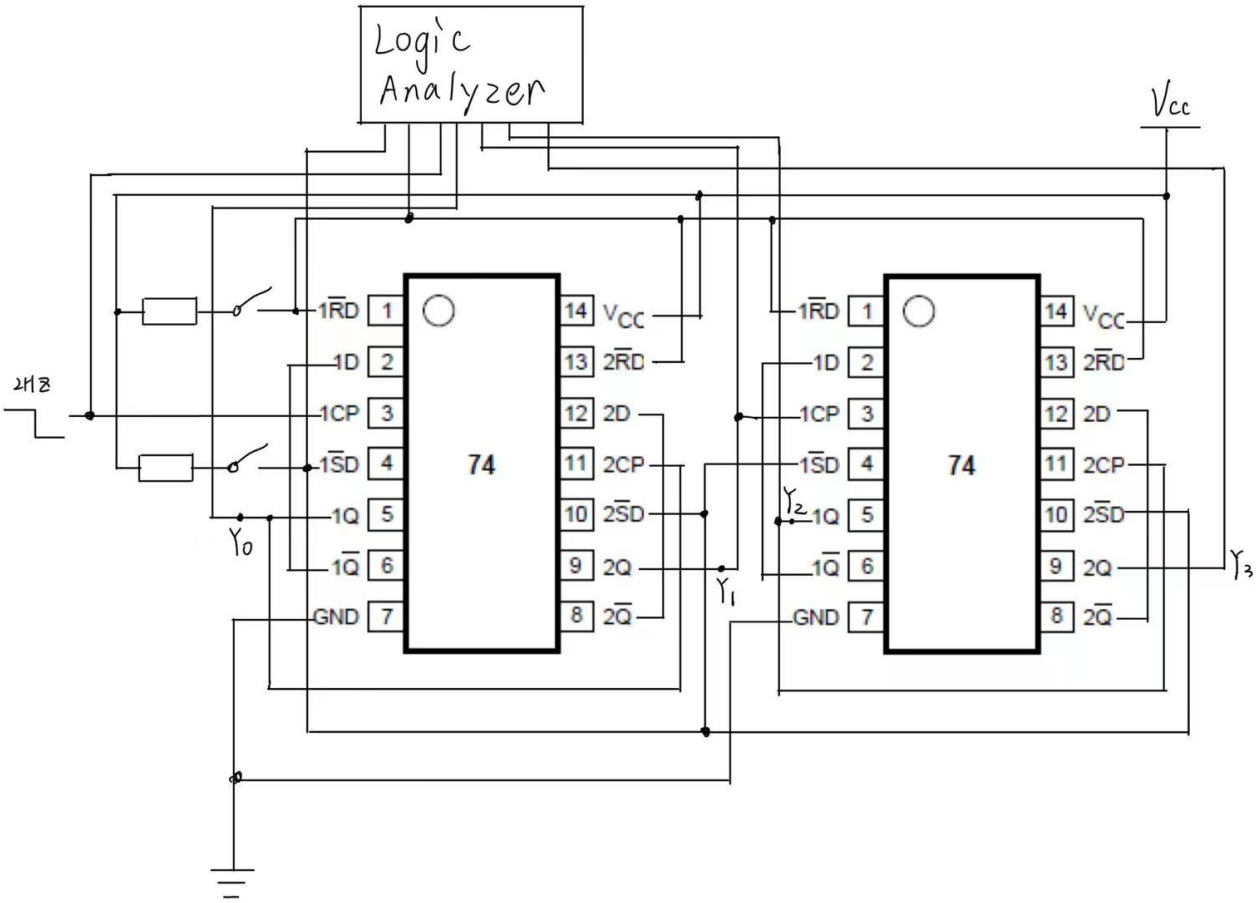


Figure 2. Design of 4 channel word generator (chip level)

1.2 Result

In the four below figures, the waveforms show $\overline{SD}$, $\overline{RD}$, 2HZ square wave, Y0, Y1,

Y2, Y3 respectively, from D0 to D6.

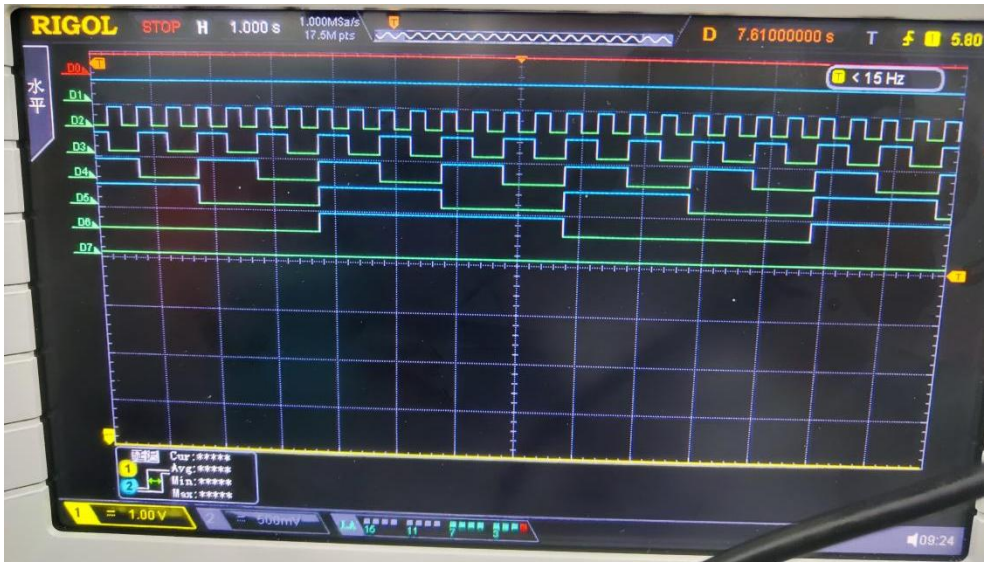


Figure 3. Waveform when $\overline{SD}$ and $\overline{RD}$ are both HIGH

Figure 4. Waveform when $\overline{SD}$ is HIGH and $\overline{RD}$ is LOW

Figure 5. Waveform when $\overline{SD}$ is LOW and $\overline{RD}$ is HIGH

Figure 6. Waveform when $\overline{SD}$ and $\overline{RD}$ are both LOW

This result is consistent with the function table of D flip-flop.

INPUT				OUTPUT	
$\overline{SD}$	$\overline{RD}$	CP	D	Q	$\overline{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H	H

INPUT				OUTPUT	
$\overline{SD}$	$\overline{RD}$	CP	D	Qn+1	$\overline{Q}$n+1
H	H	↑	L	L	H
H	H	↑	H	H	L

Table 1. Function tables of D flip-flop

When $\overline{SD}$ and $\overline{RD}$ are both HIGH, the function of word generator is successfully implemented.

2. Experiment B

2.1 Design .

- Count down

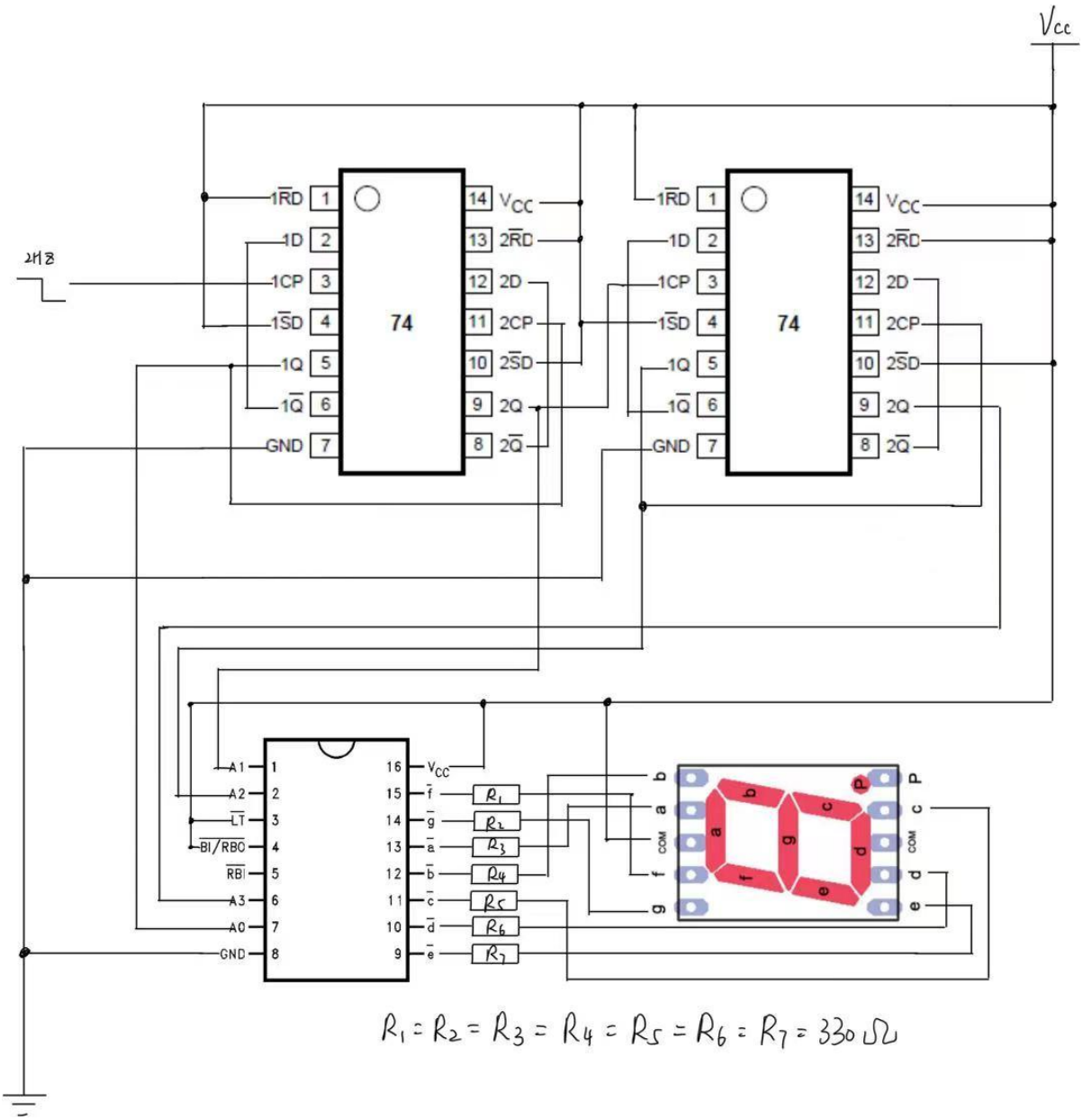


Figure 7. Design of count-down counter (chip level)

- Count up

Compared to the count-down counter where Q outputs are connected to the 7-segment display (in Figure 7), here $\bar{Q}$ outputs are connected to the display.

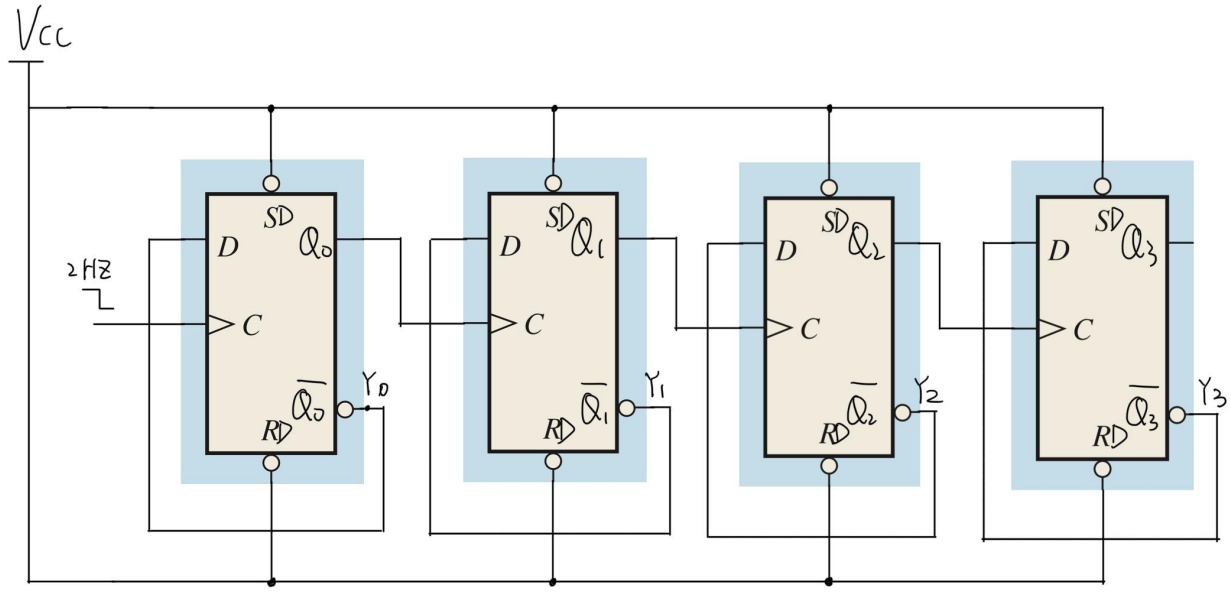


Figure 8. Design of count-up counter (gate level) (flip-flop part)

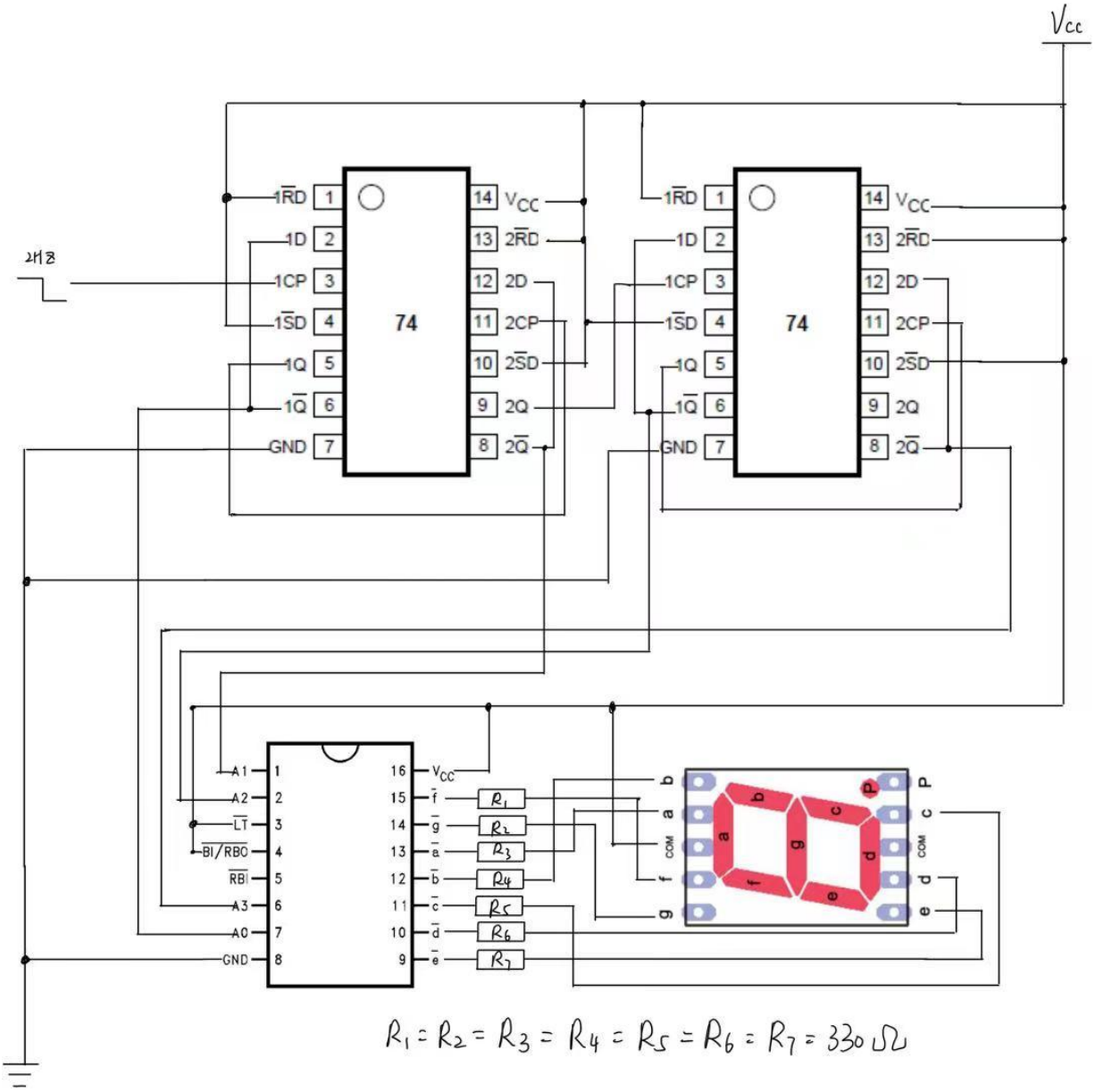


Figure 9. Design of count-up counter (chip level)

- Count up in loop

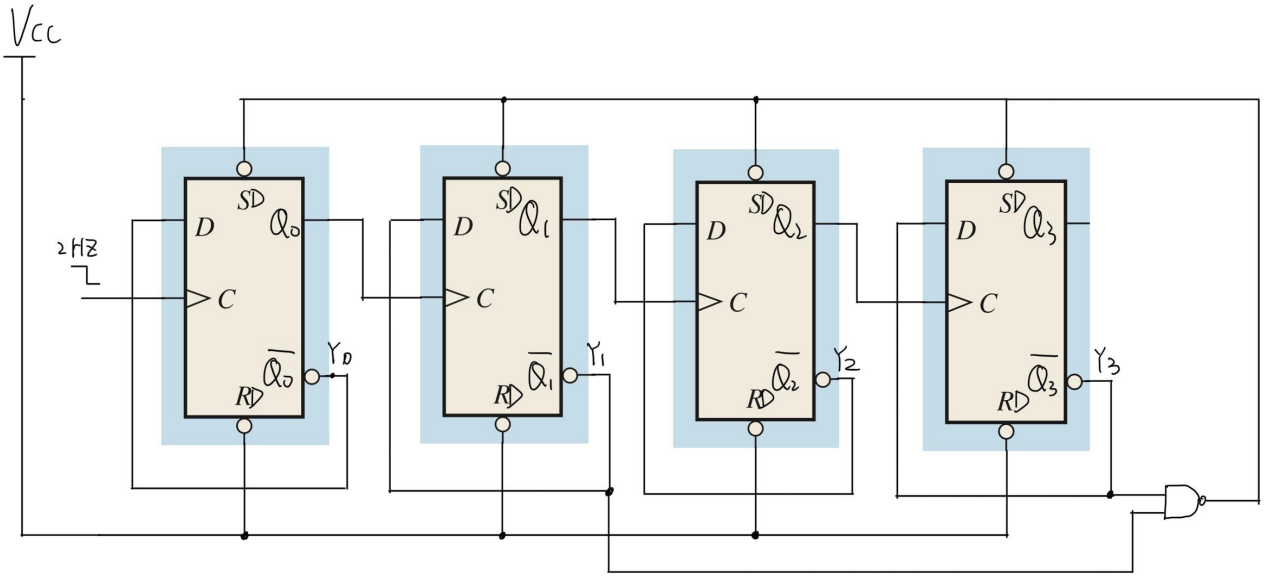


Figure 10. Design of count-up-in-loop counter (gate level) (flip-flop part)

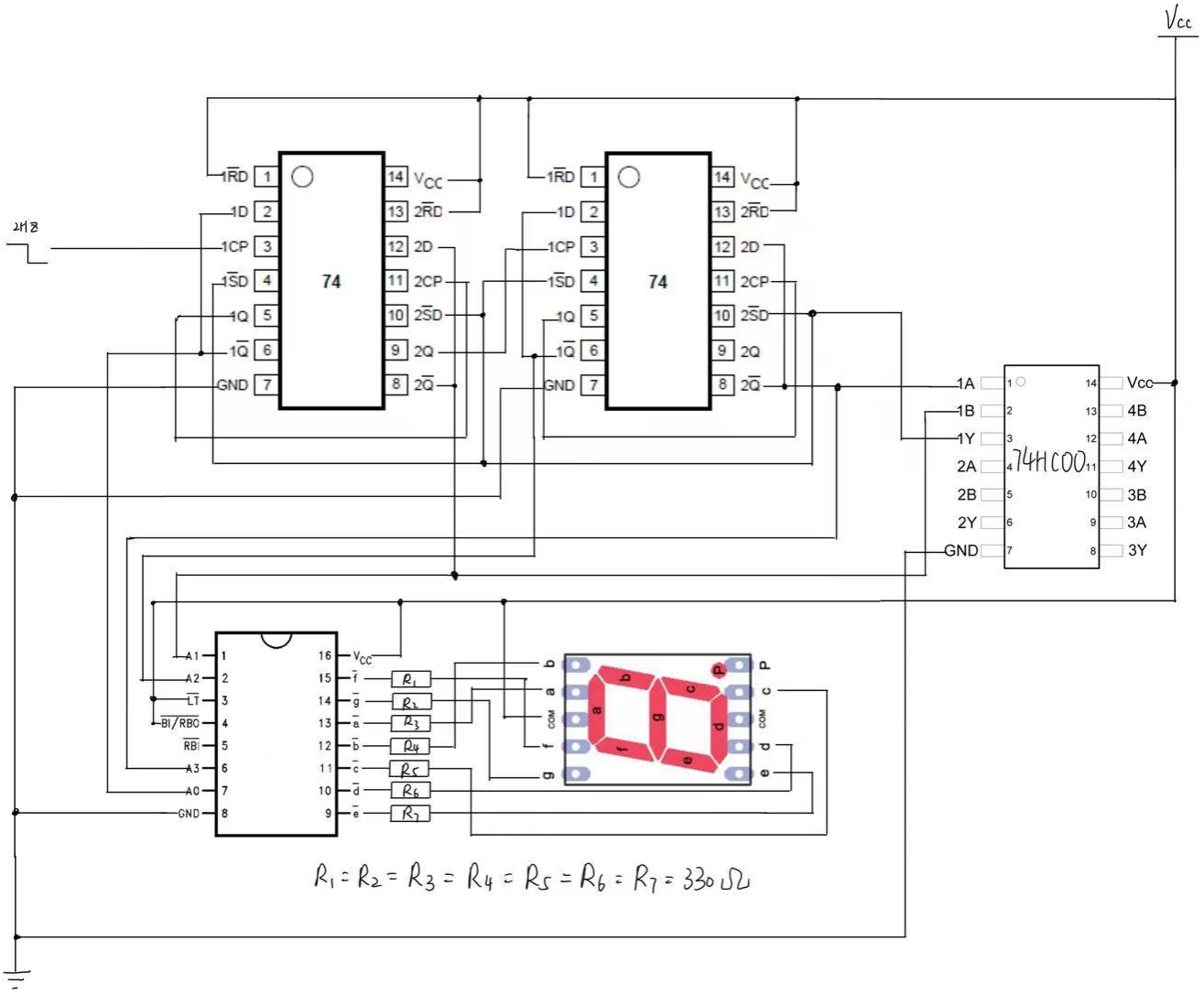


Figure 11. Design of count-up-in-loop counter (chip level)

2.2 Result

- Count-down from 9 to 0

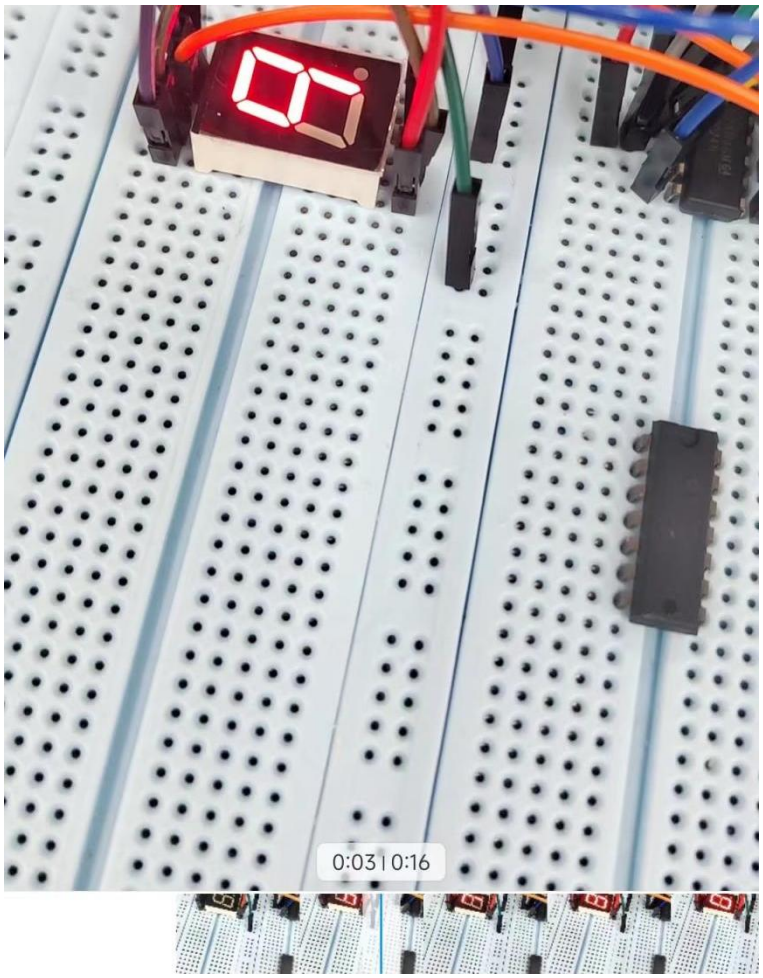


Figure 12. Displays “9”

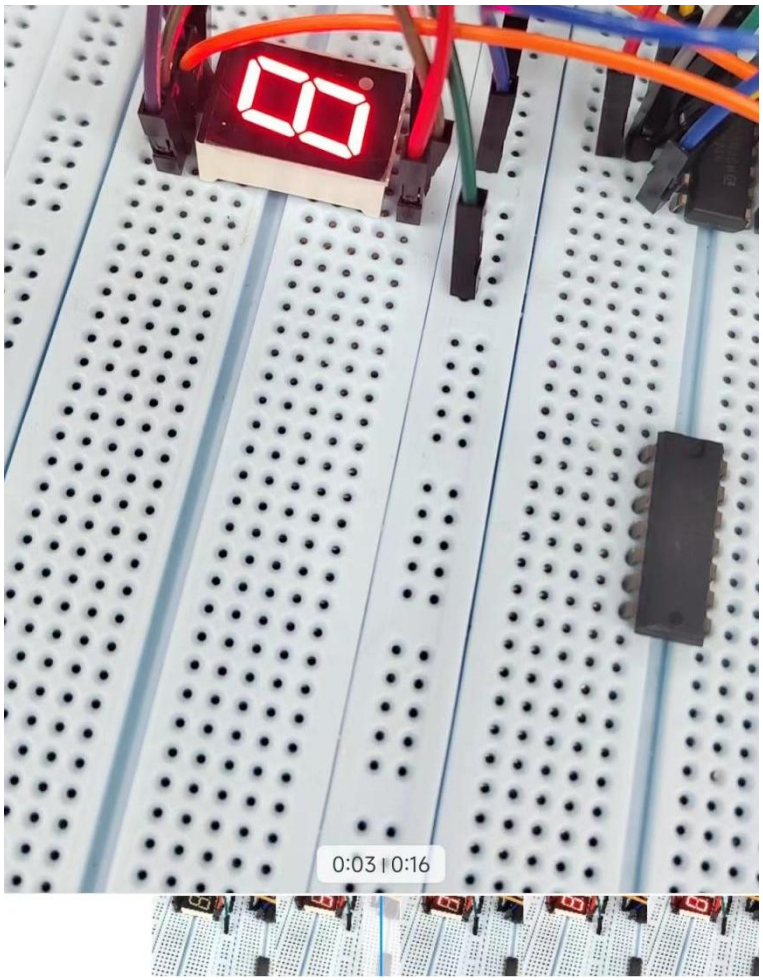


Figure 13. Displays “8”

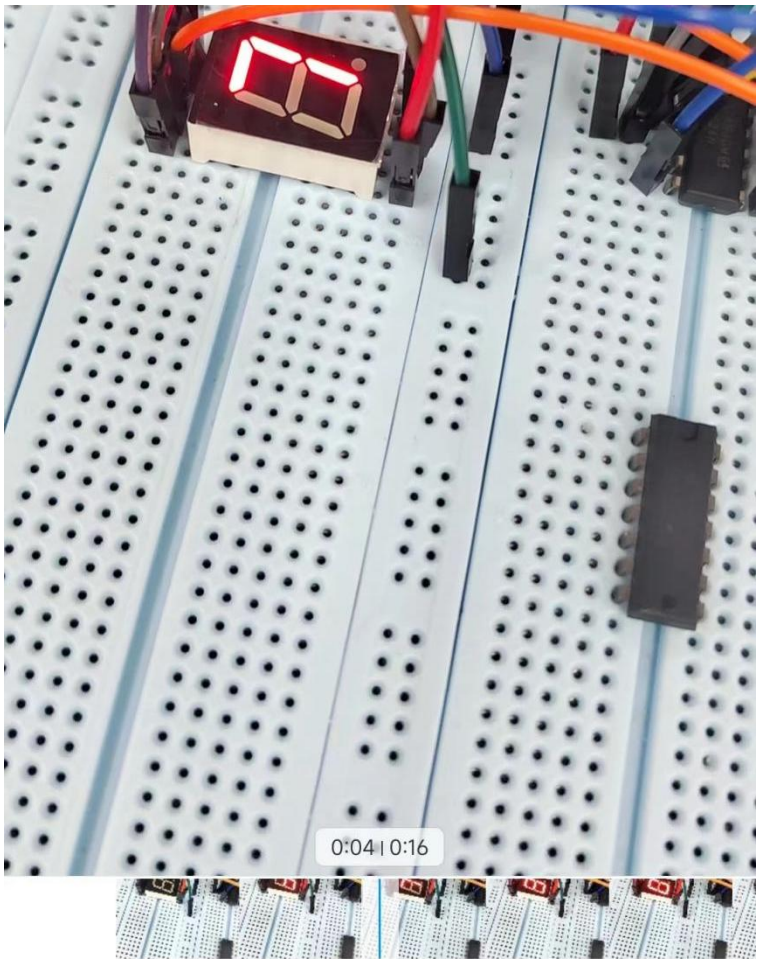


Figure 14. Displays “7”

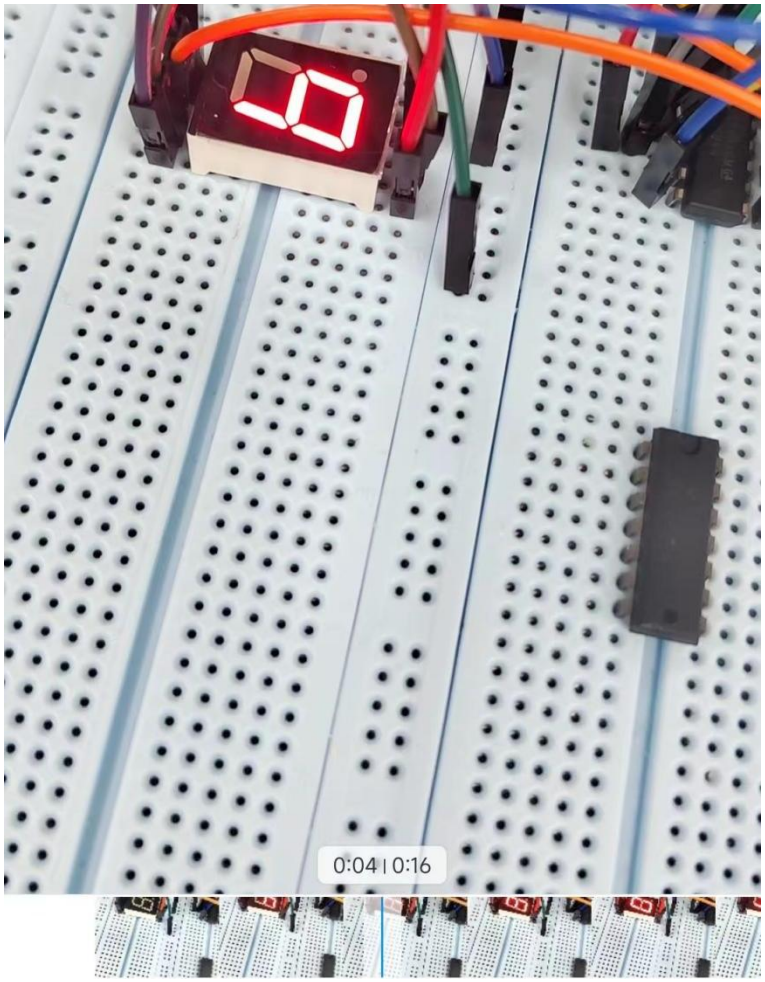


Figure 15. Displays “6”

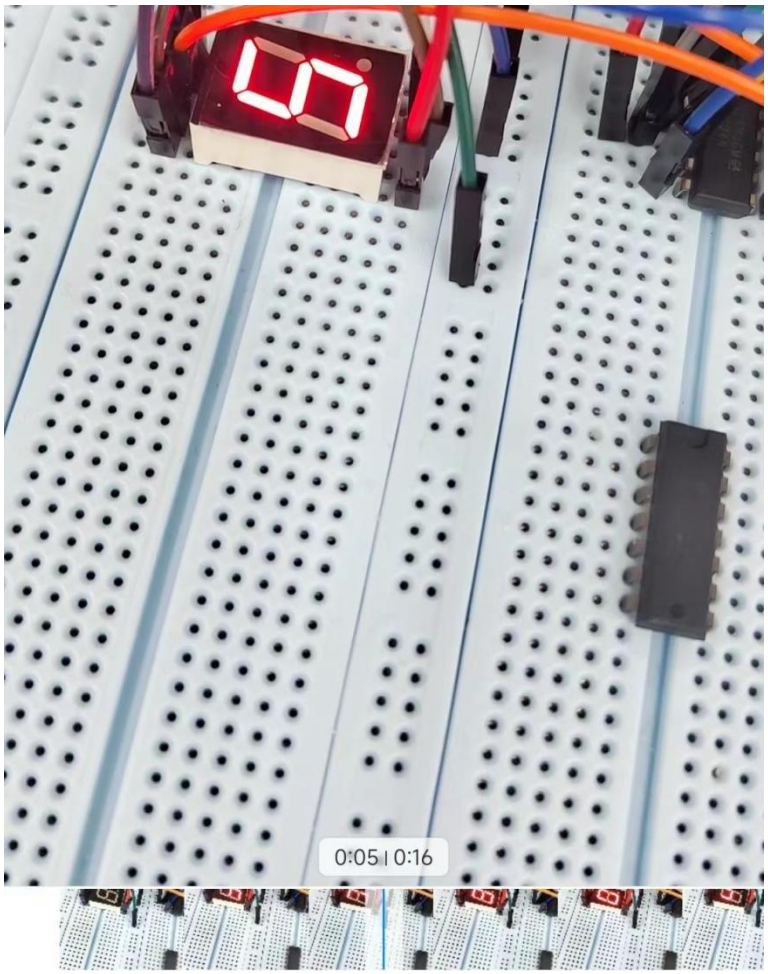


Figure 16. Displays “5”

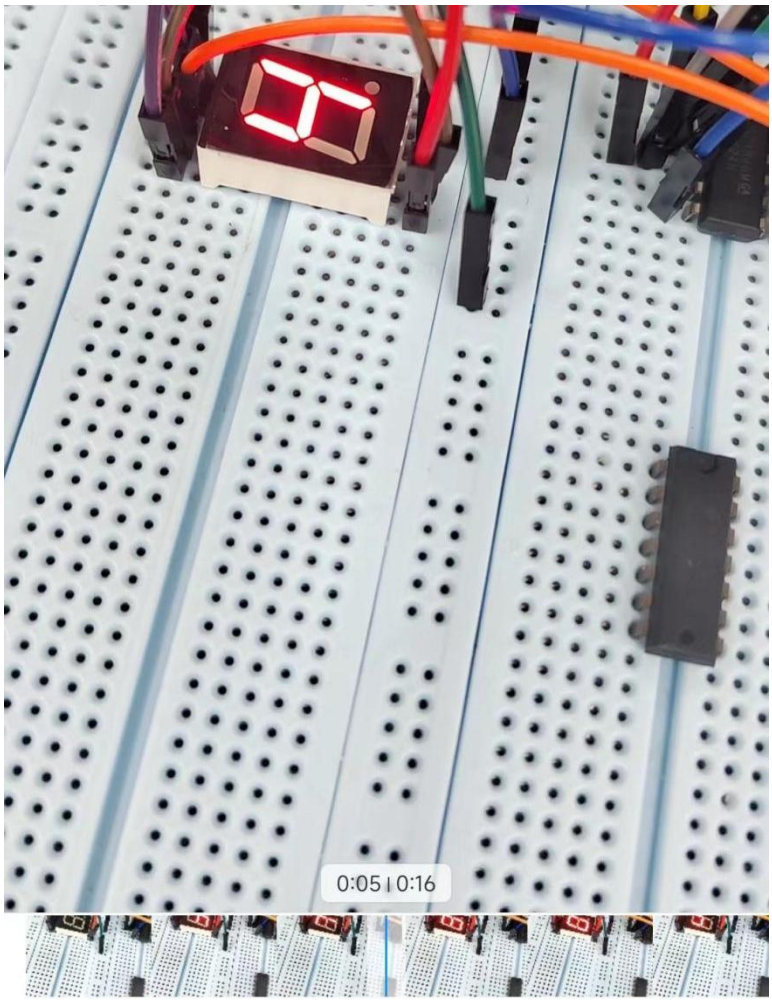


Figure 17. Displays “4”

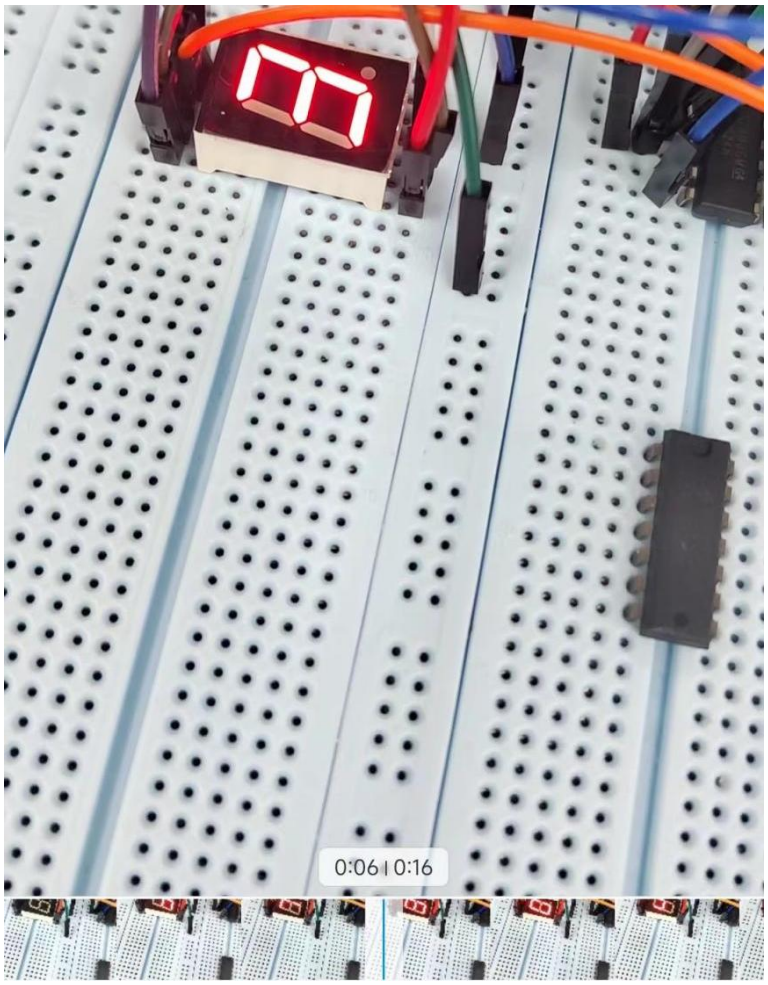


Figure 18. Displays “3”

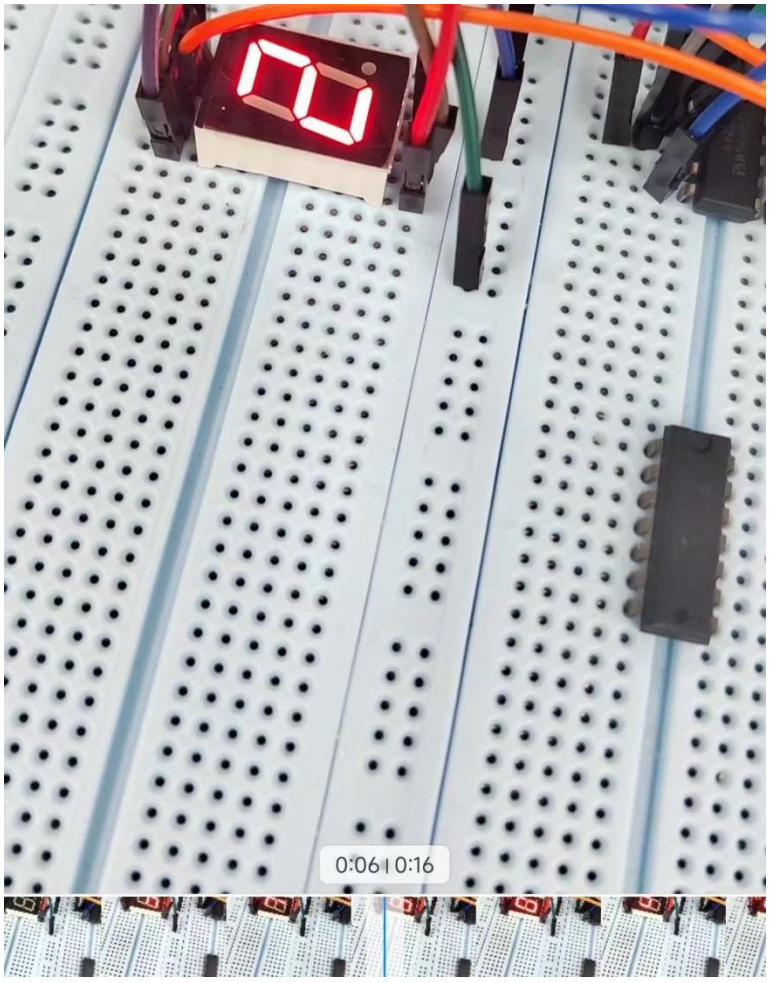


Figure 19. Displays “2”

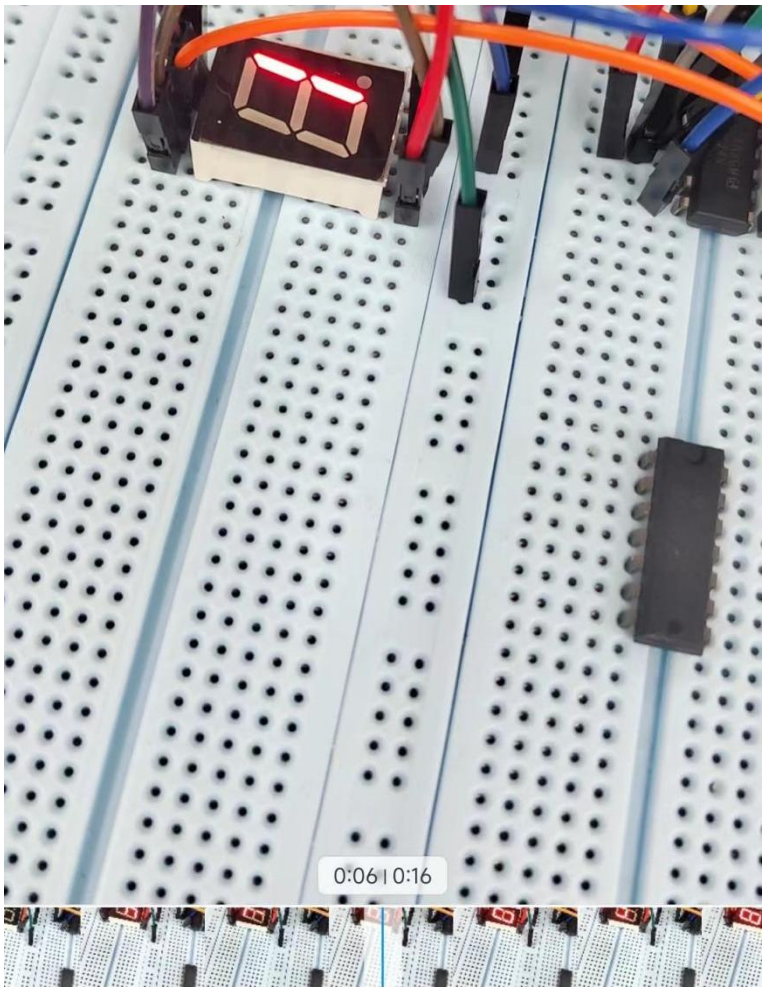


Figure 20. Displays “1”

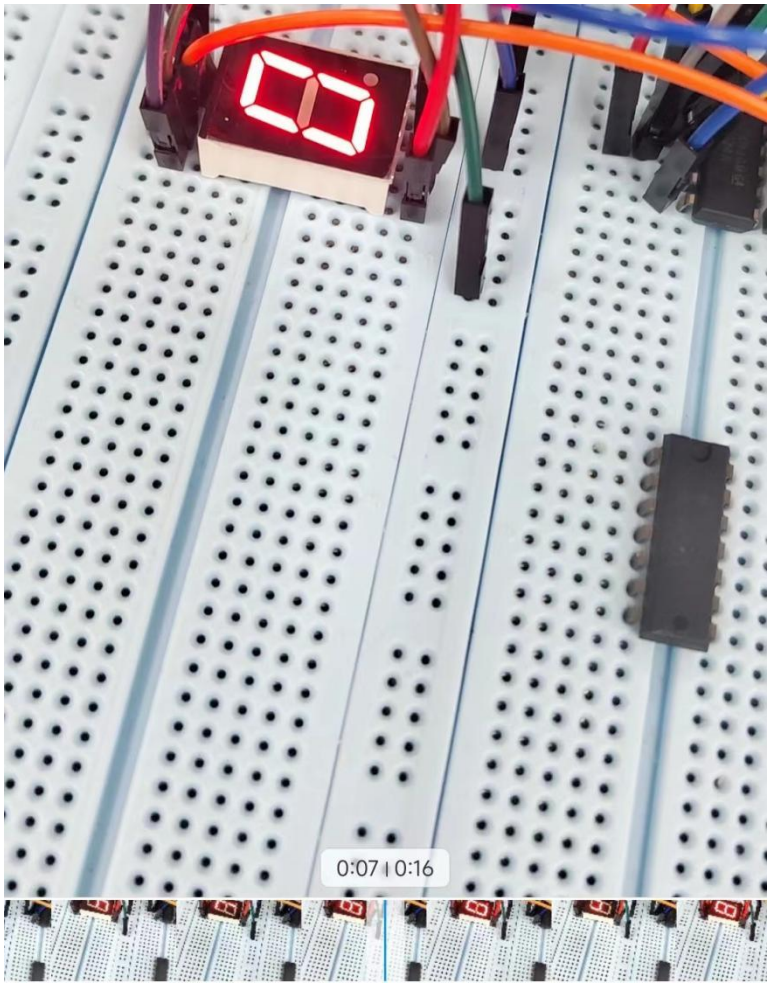


Figure 21. Displays “0”

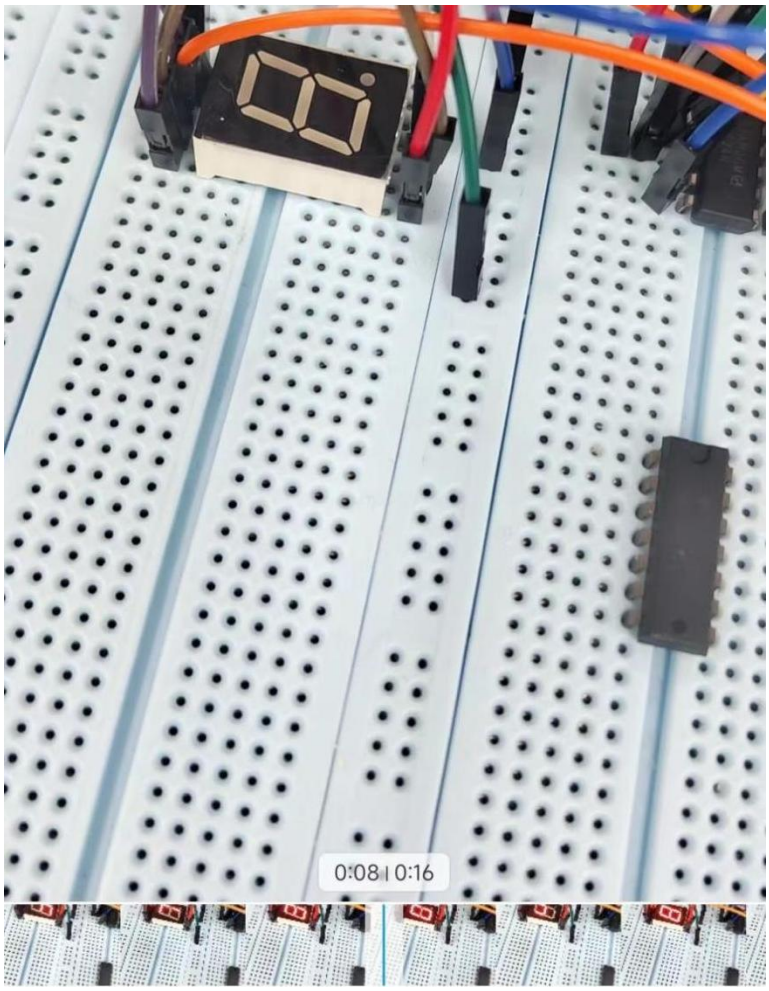


Figure 22. Displays oddly (BCD code is 1111)

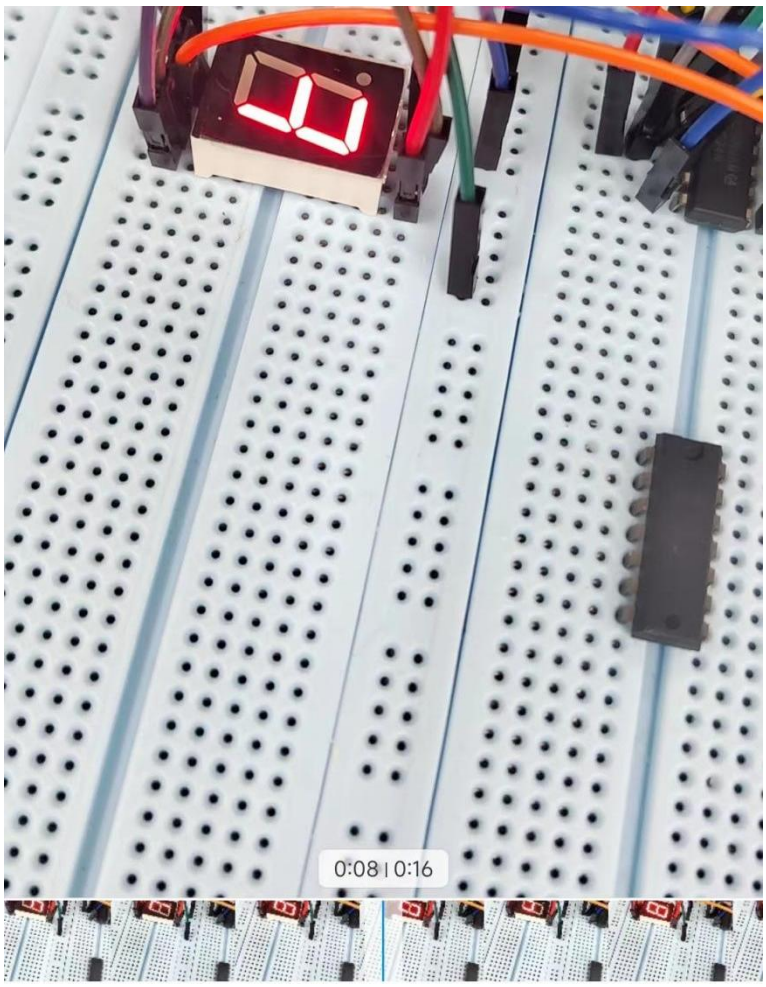


Figure 23. Displays oddly (BCD code is 1110)

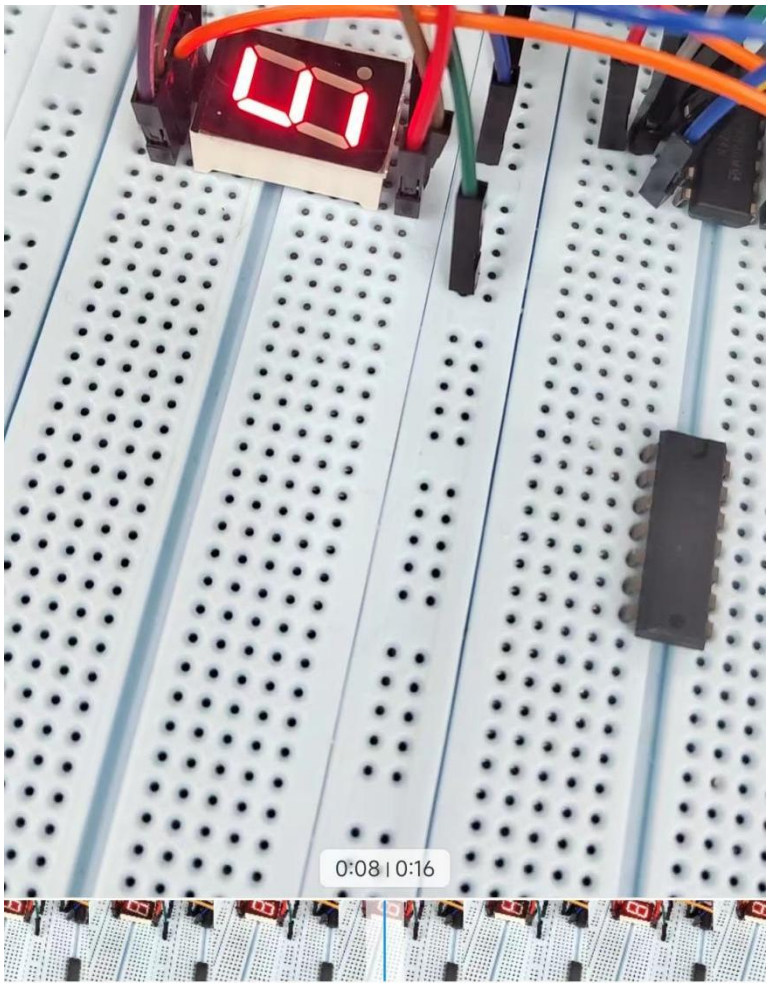


Figure 24. Displays oddly (BCD code is 1101)

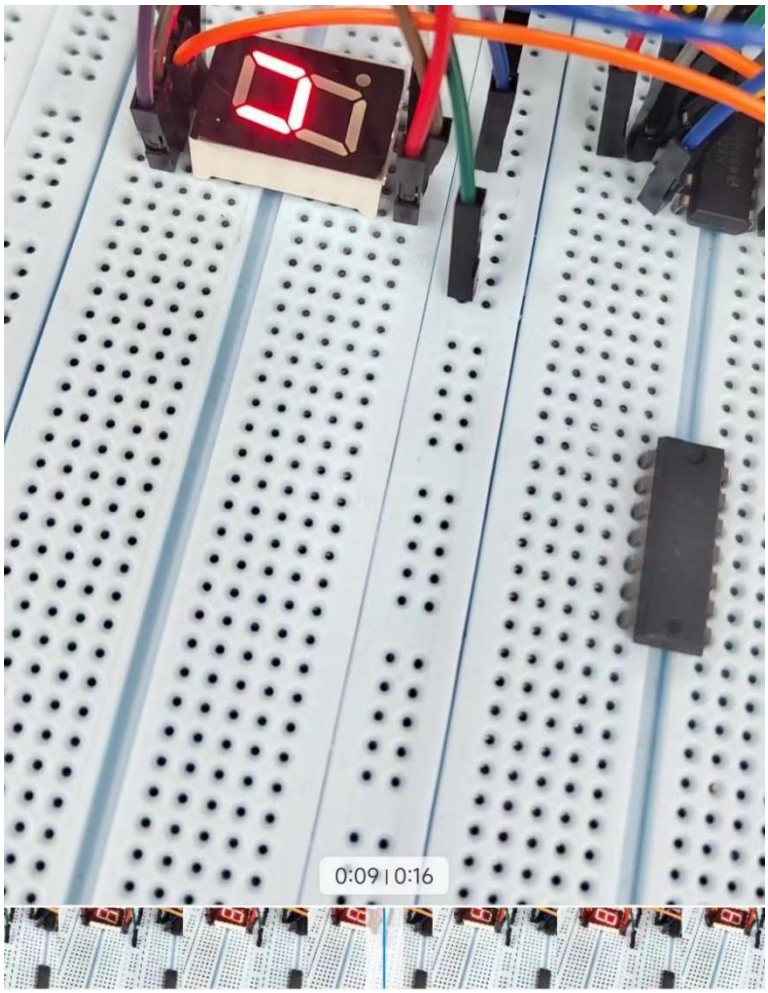


Figure 25. Displays oddly (BCD code is 1100)

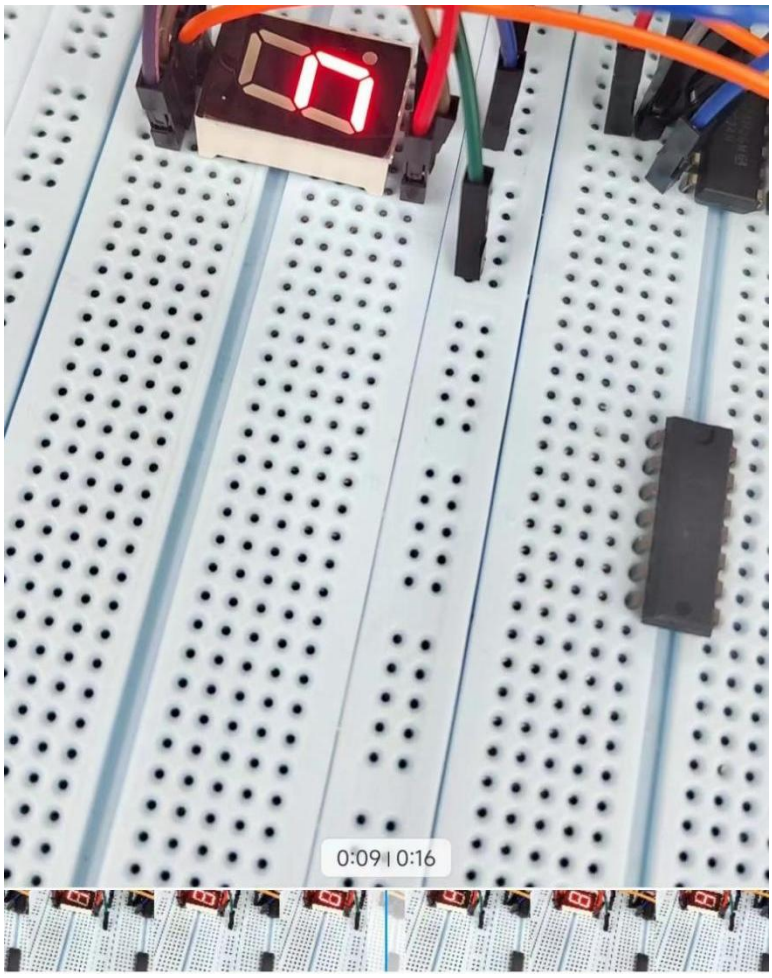


Figure 26. Displays oddly (BCD code is 1011)

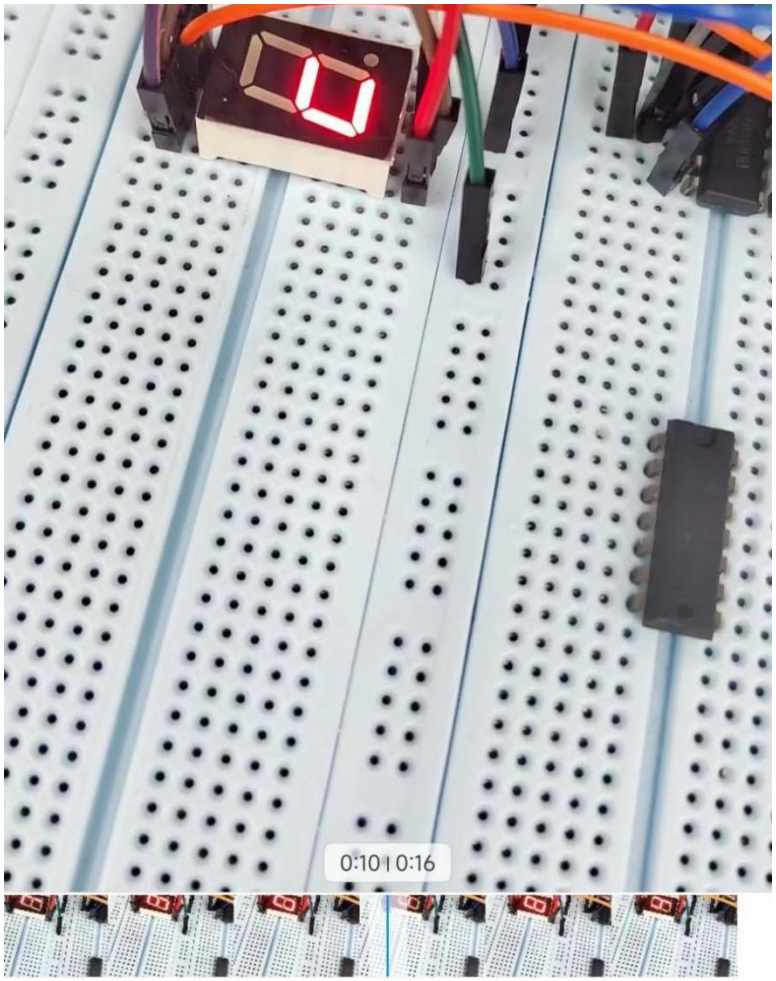


Figure 27. Displays oddly (BCD code is 1010)

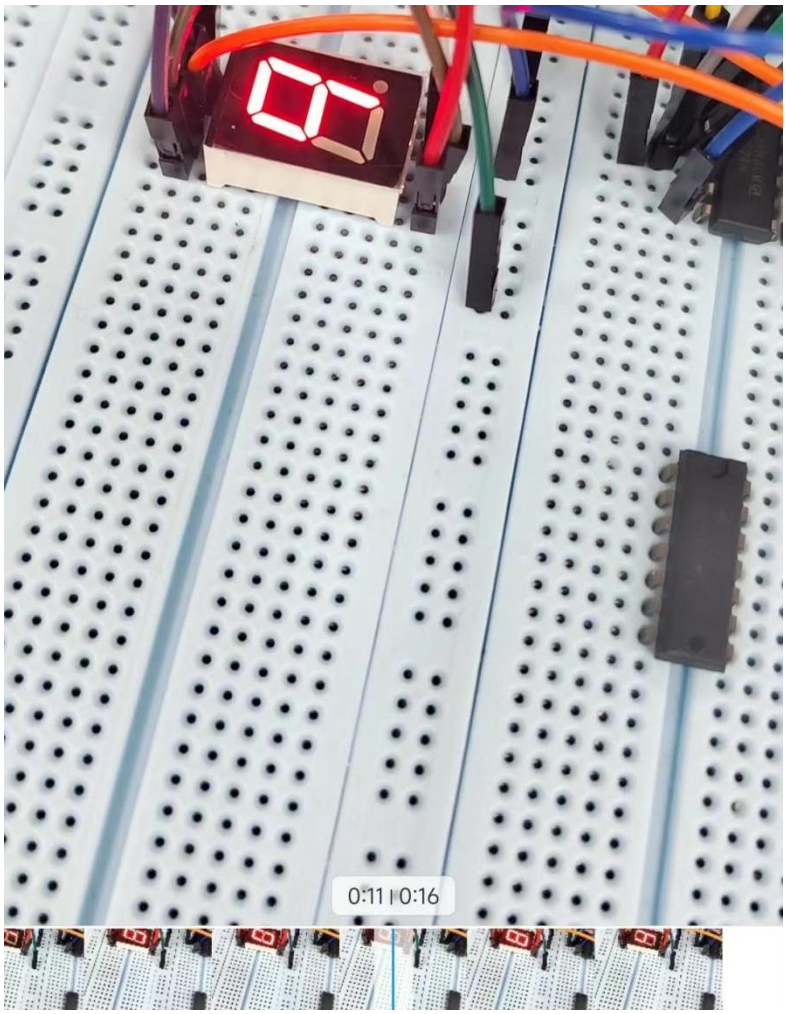


Figure 28. Displays “9” again

Figure 12~ Figure 28 shows the display of the 7-segement display, it flashes from 9 to 0, and some weird symbols are displayed when the BCD code is from 1111 to 1010.

- Count-up from 0 to 9

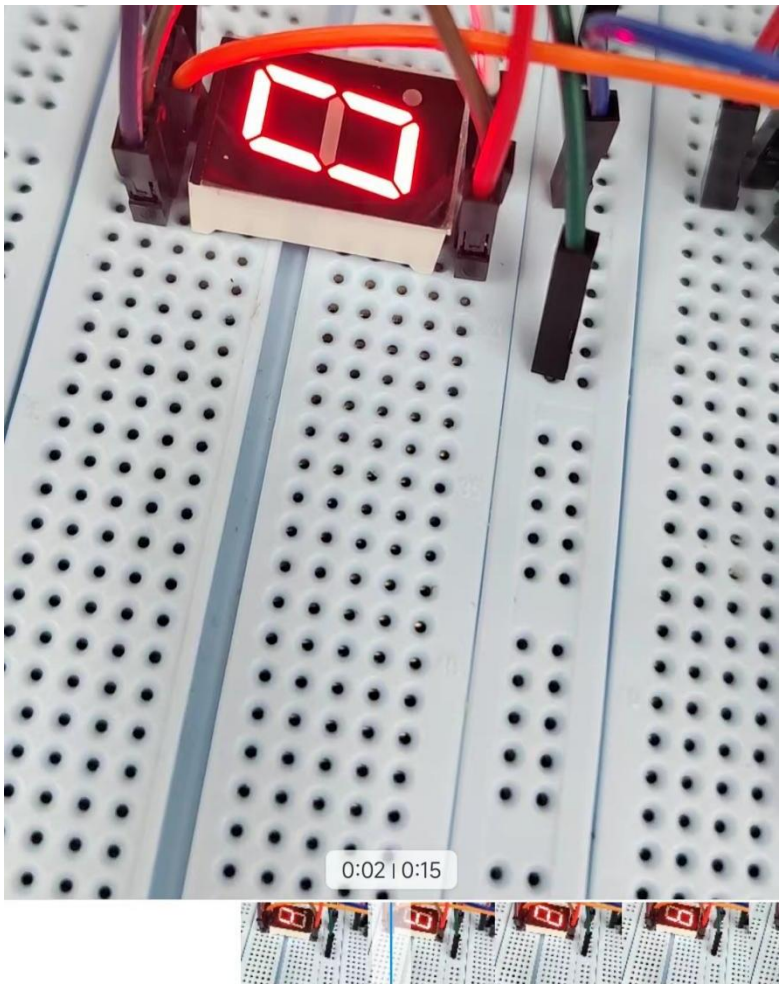


Figure 29. Displays “0”

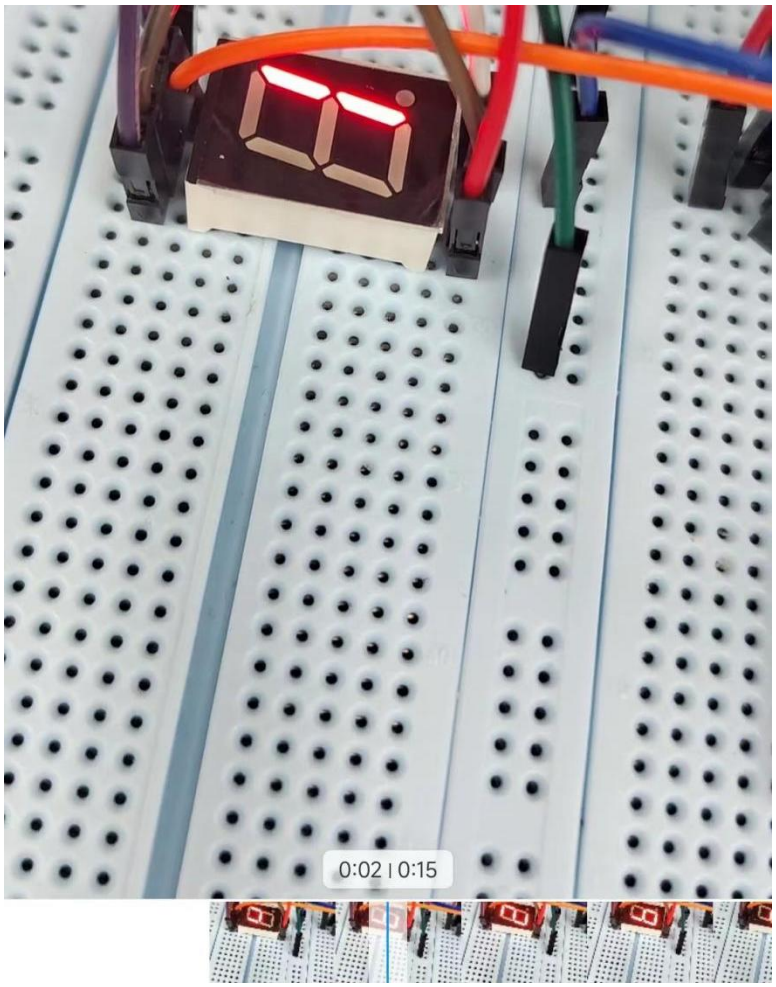


Figure 30. Displays “1”

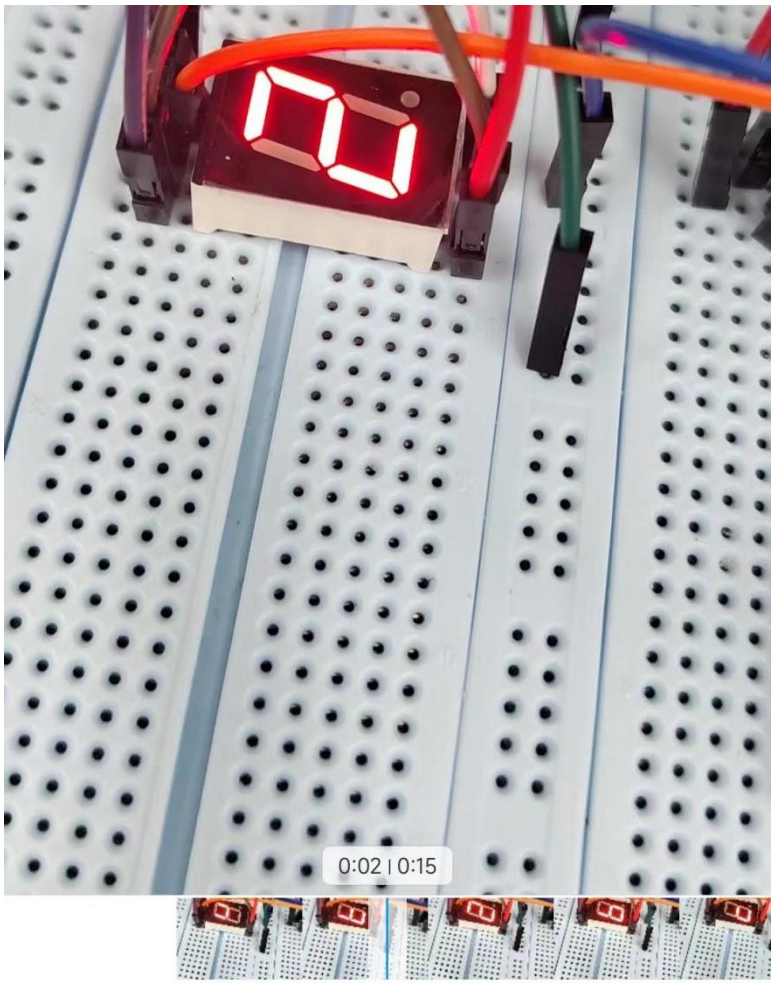


Figure 31. Displays “2”

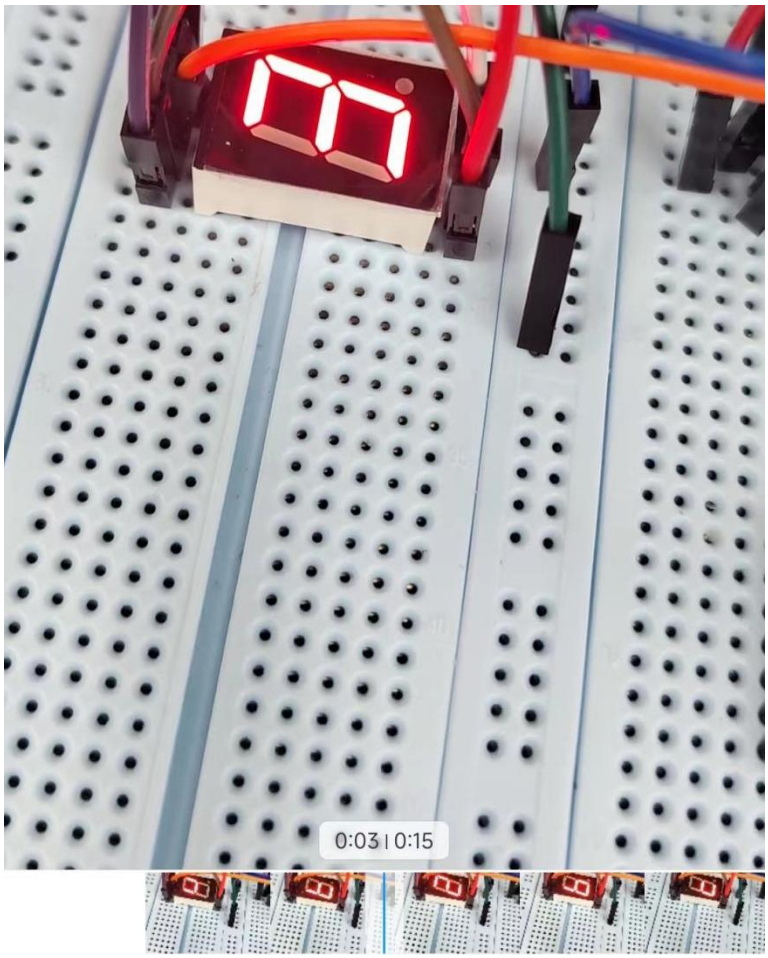


Figure 32. Displays “3”

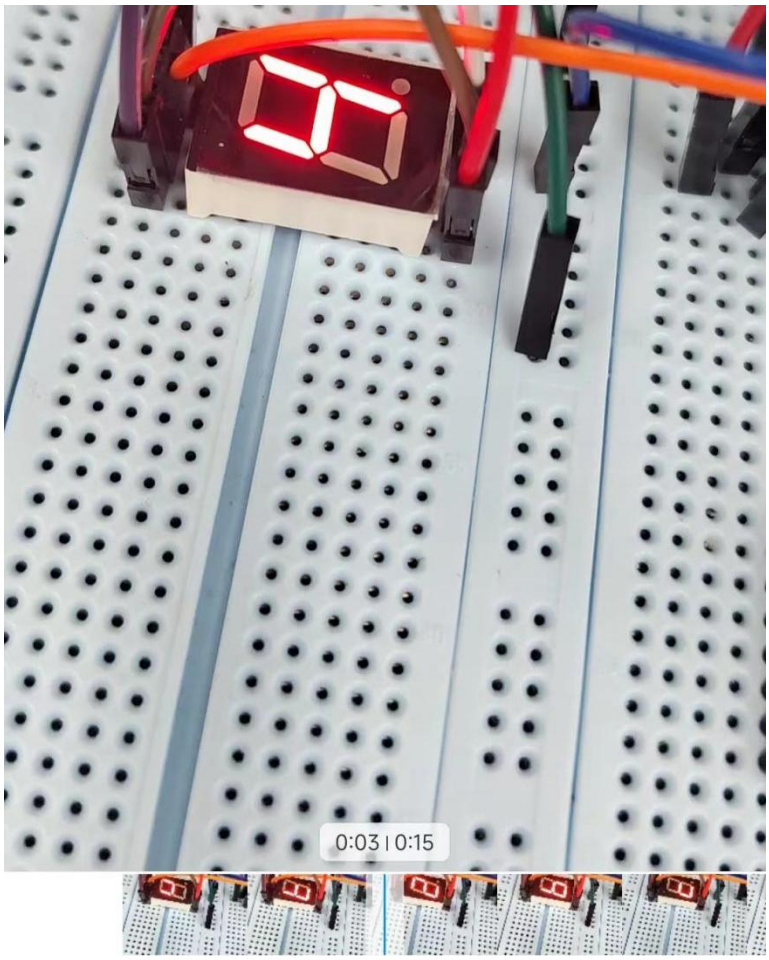


Figure 33. Displays “4”

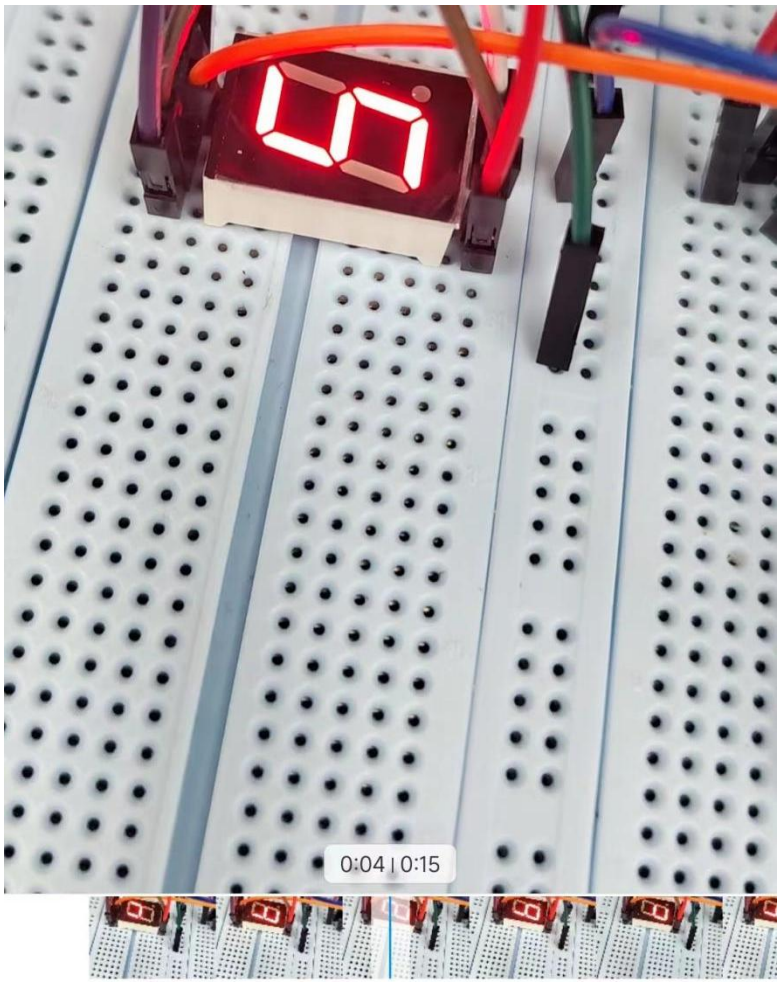


Figure 34. Displays “5”

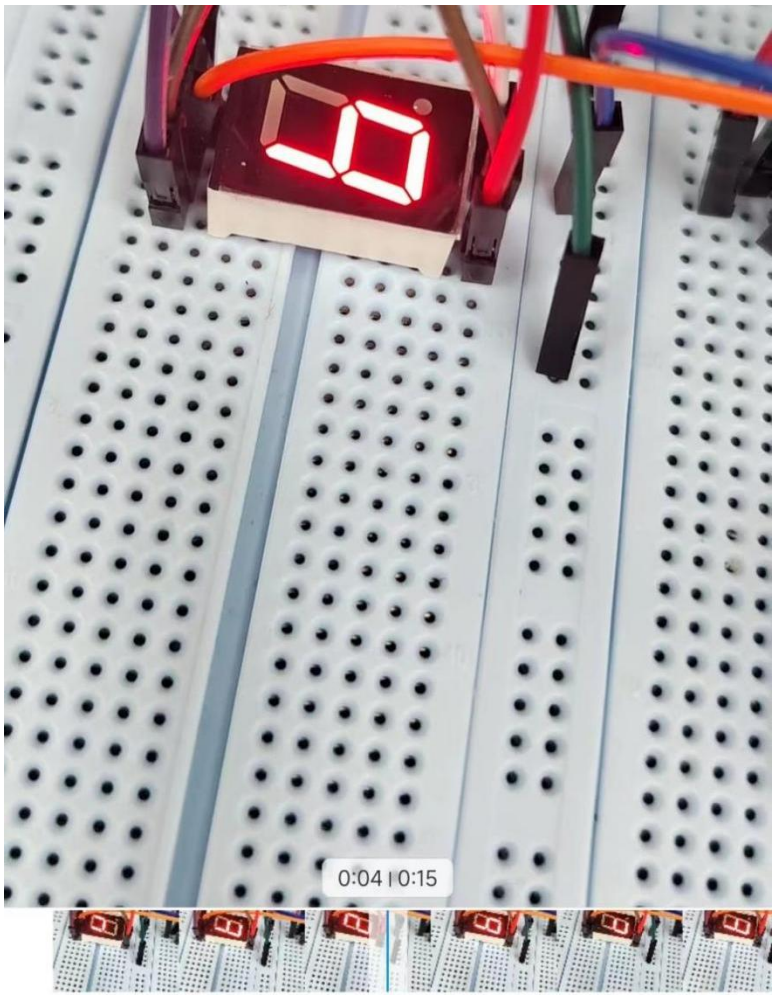


Figure 35. Displays “6”

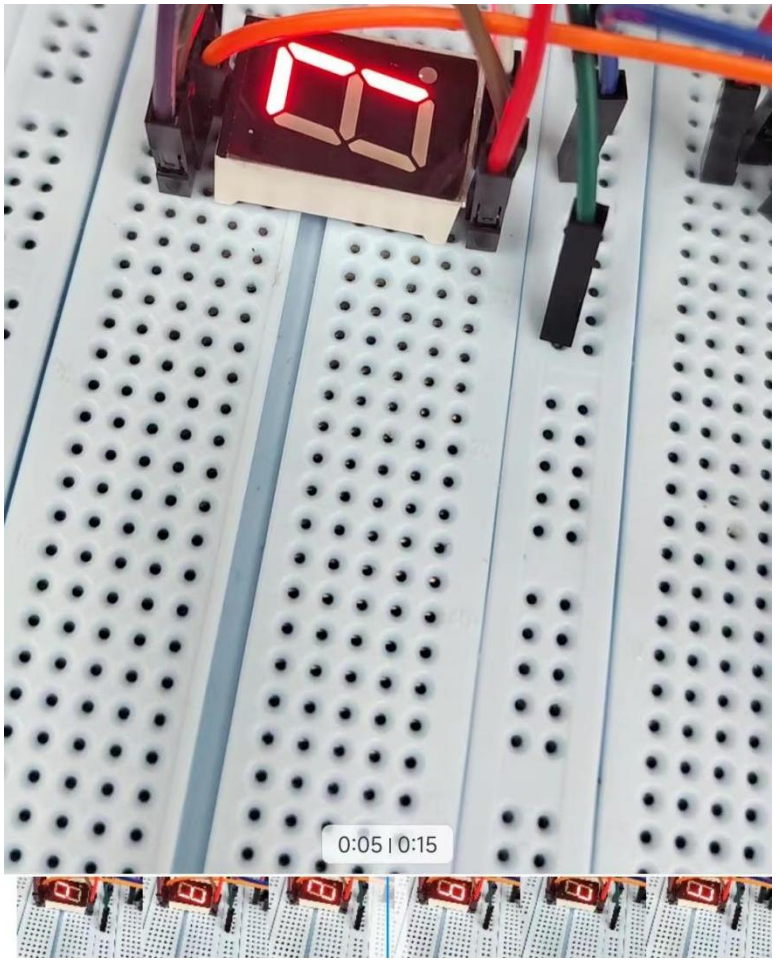


Figure 36. Displays “7”

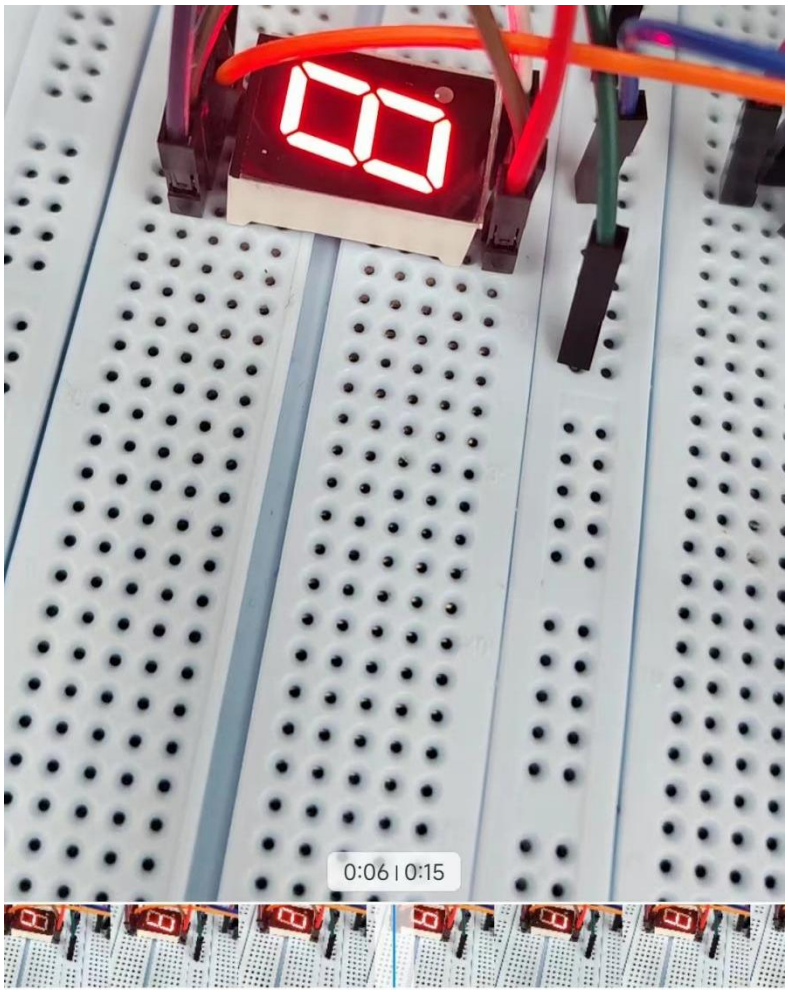


Figure 37. Displays “8”

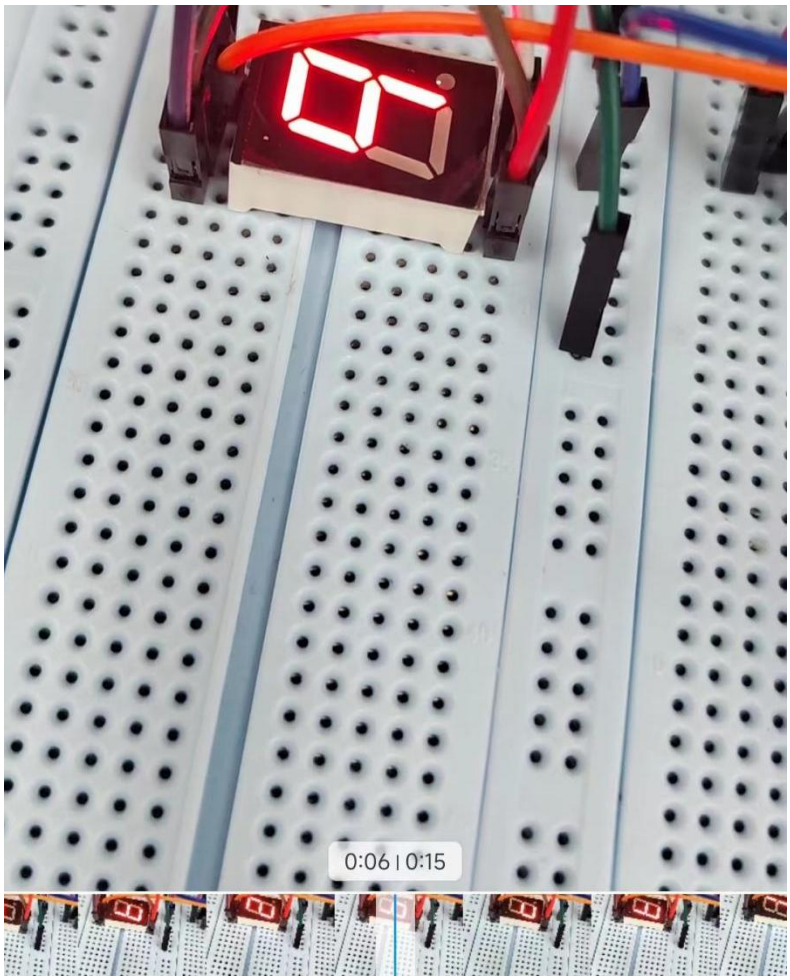


Figure 38. Displays “9”

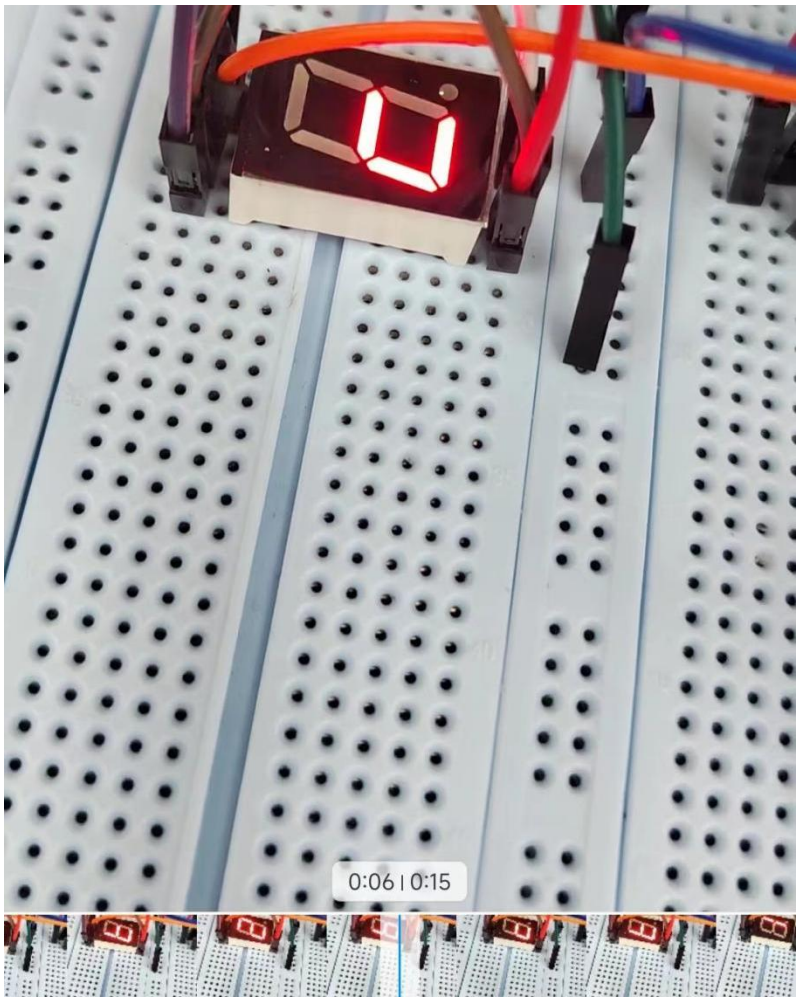


Figure 39. Displays oddly (BCD code is 1010)

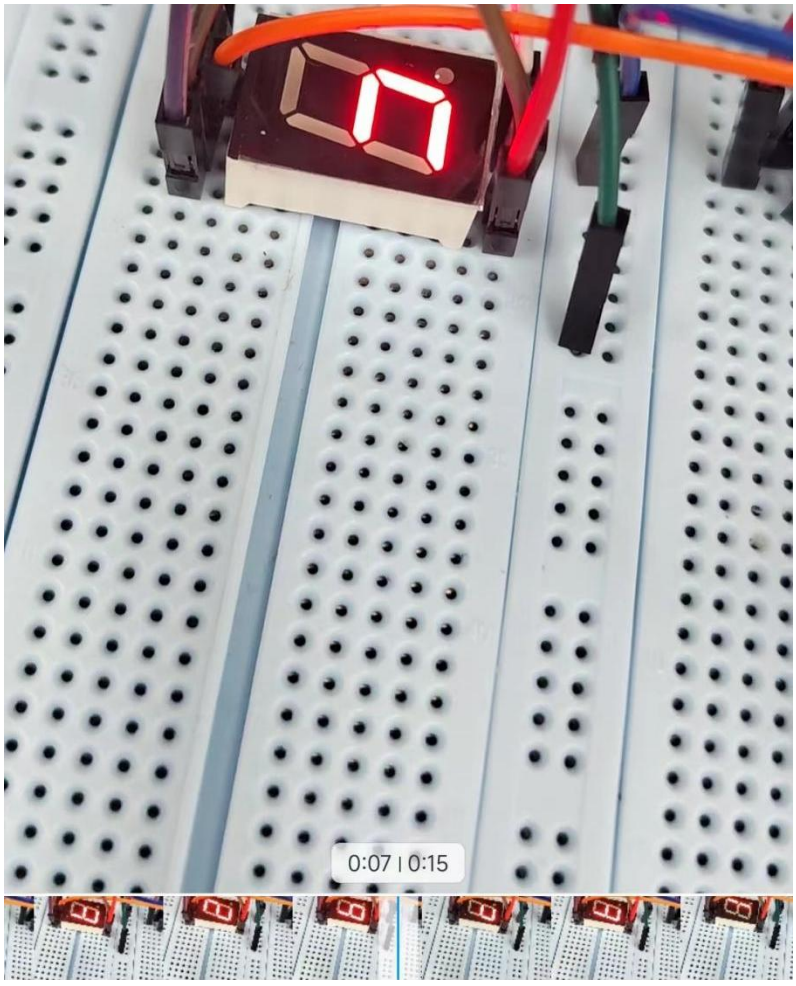


Figure 40. Displays oddly (BCD code is 1011)

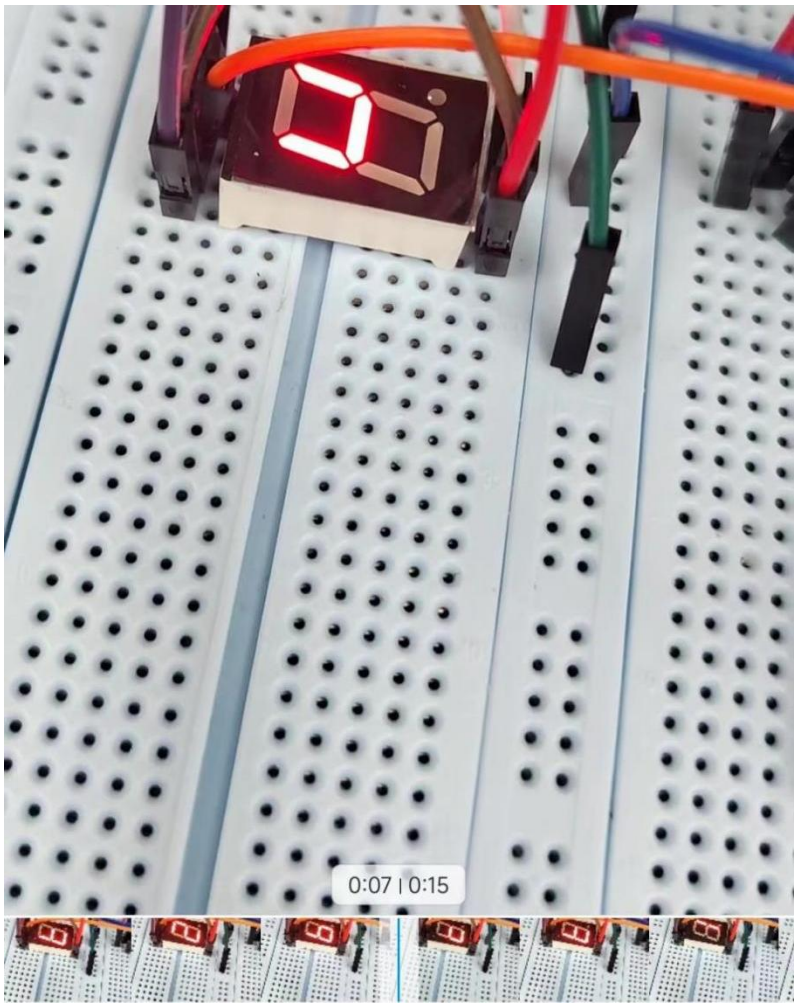


Figure 41. Displays oddly (BCD code is 1100)

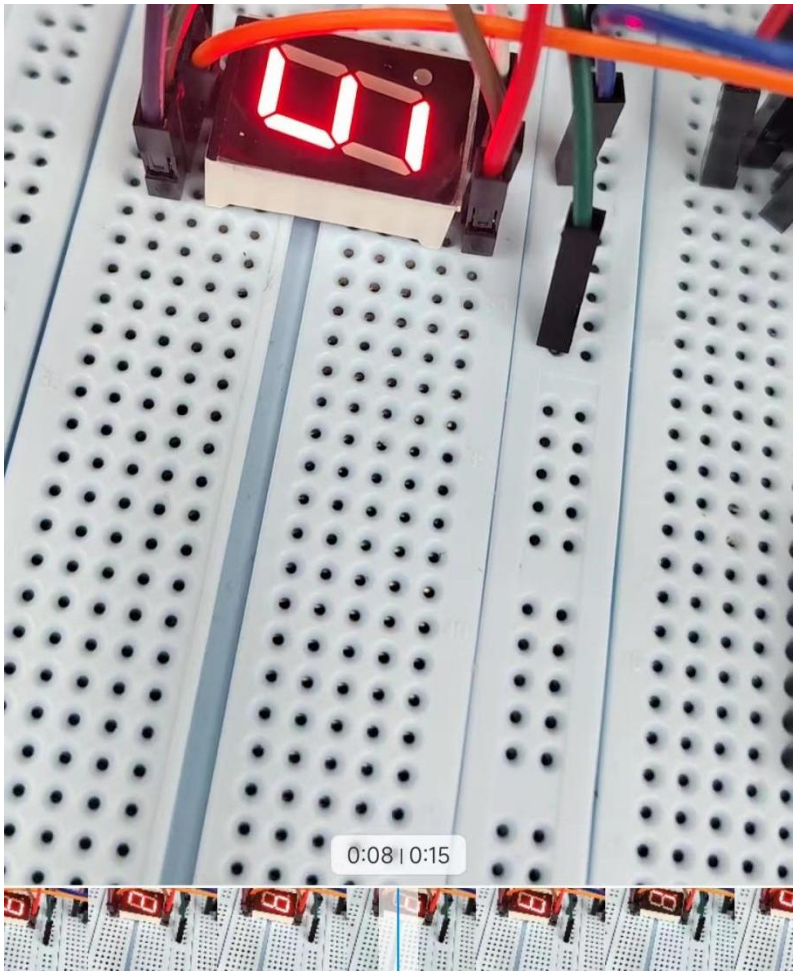


Figure 42. Displays oddly (BCD code is 1101)

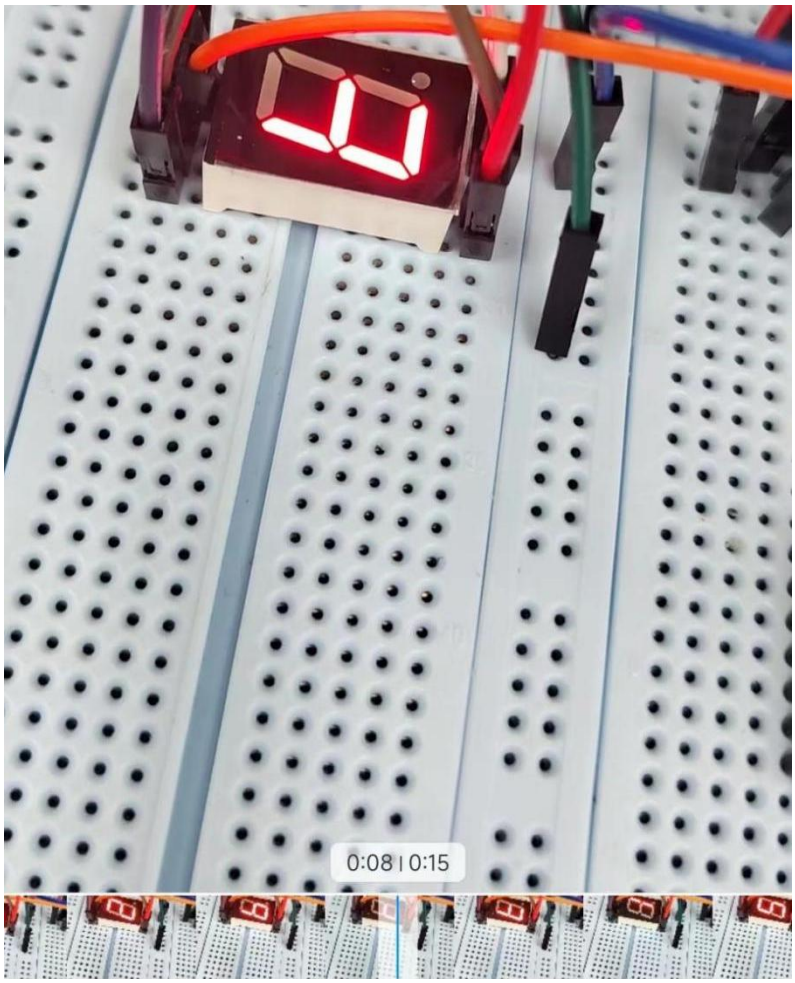


Figure 43. Displays oddly (BCD code is 1110)

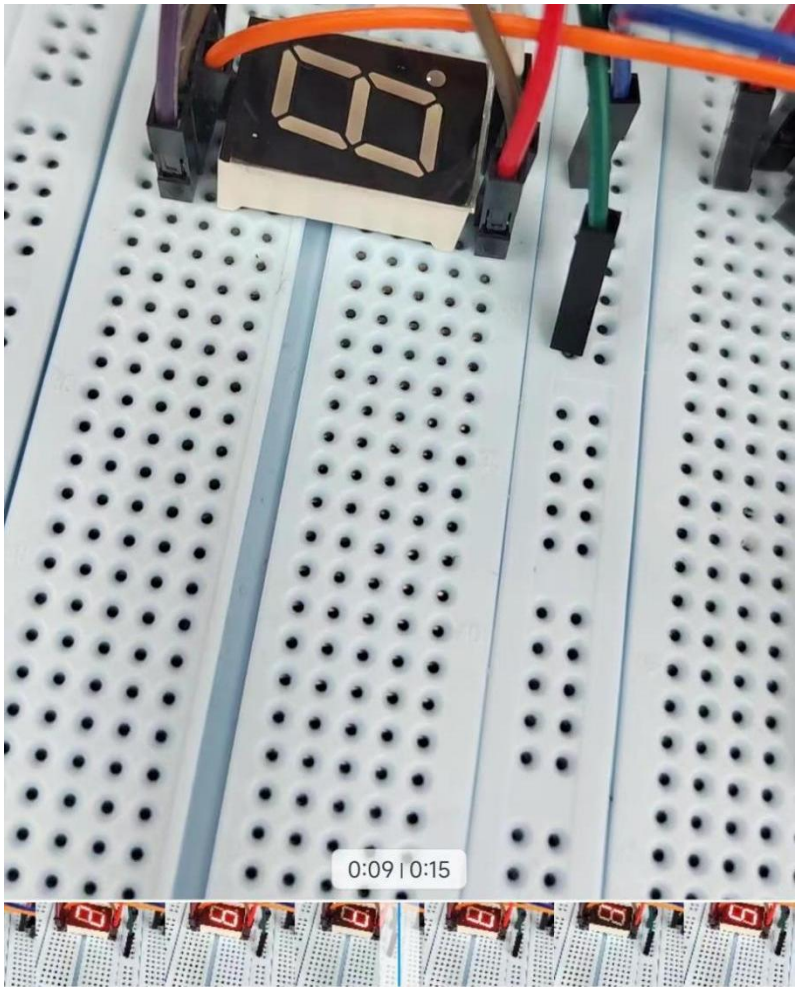


Figure 44. Displays oddly (BCD code is 1111)

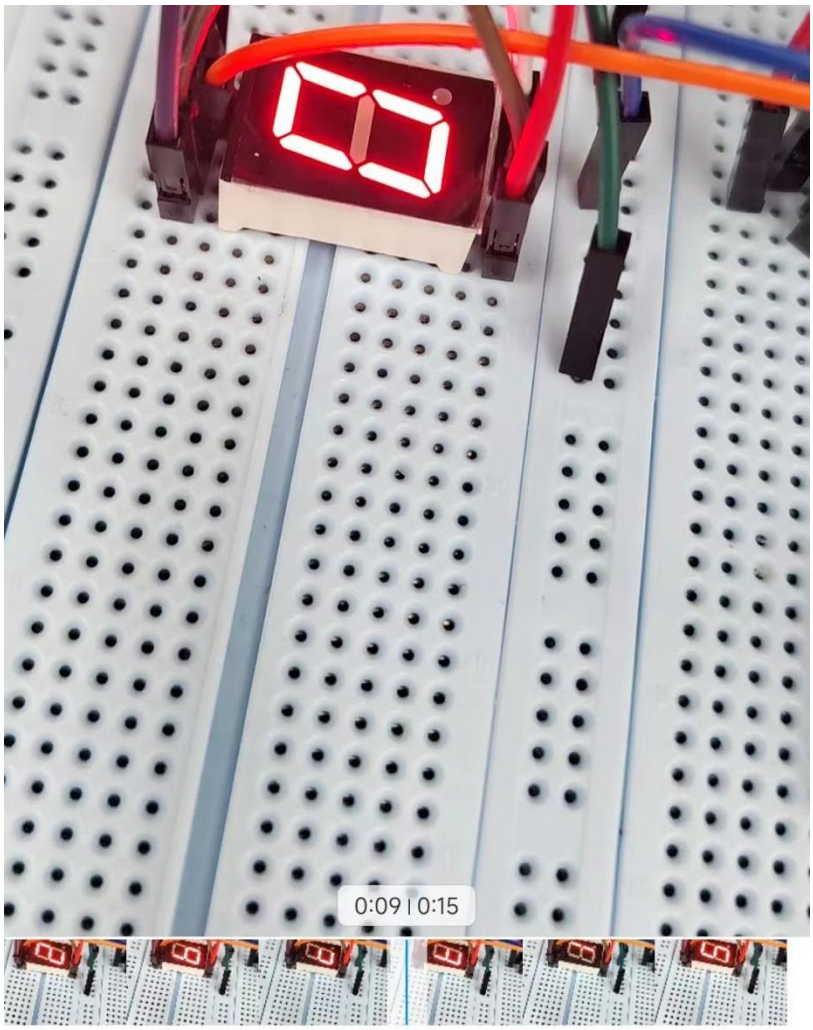


Figure 45. Displays “0” again

Figure 29 ~ Figure 45 shows the display of the 7-segement display, it flashes from 0 to 9, and some weird symbols are displayed when the BCD code is from 1010 to 1111.

- Count-up from 0 to 9 in a loop

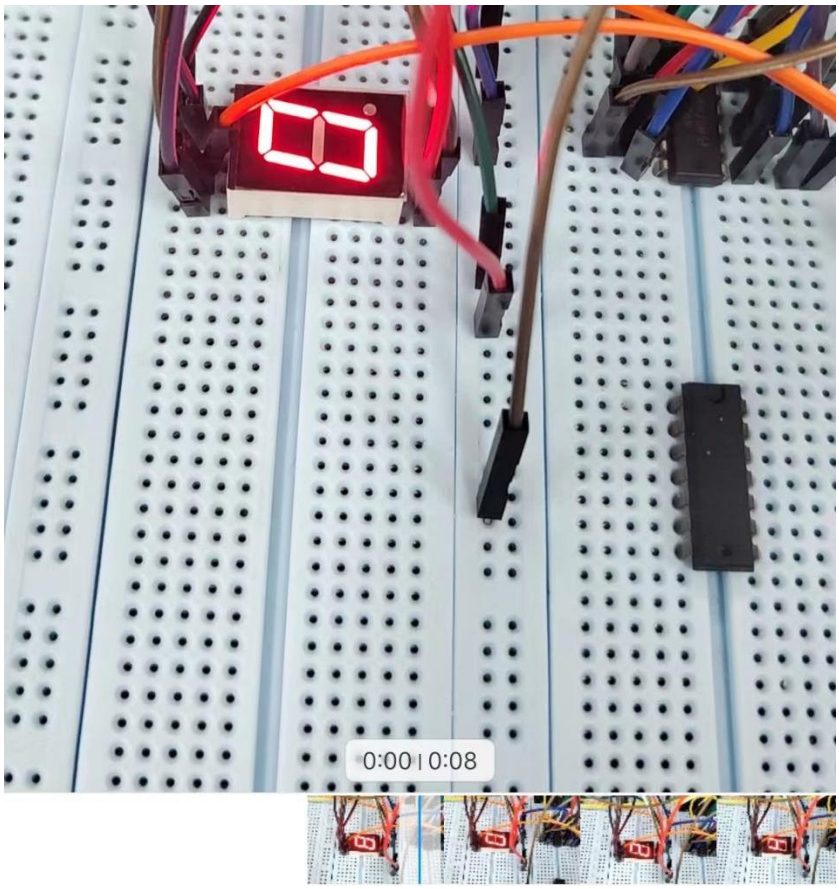


Figure 46. Displays “0”

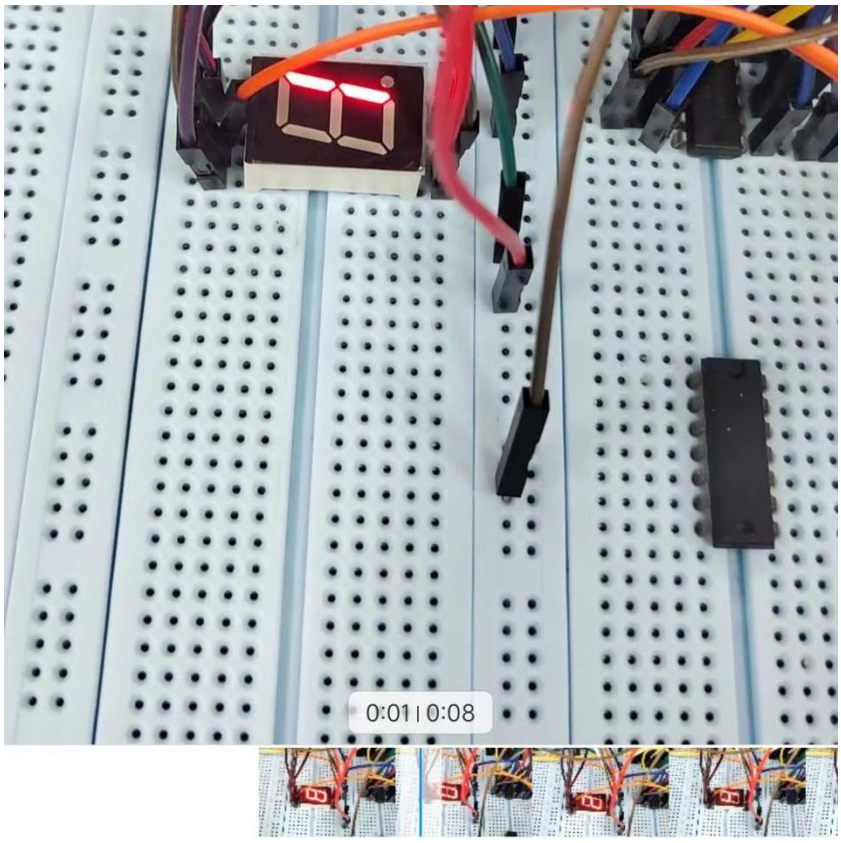


Figure 47. Displays “1”

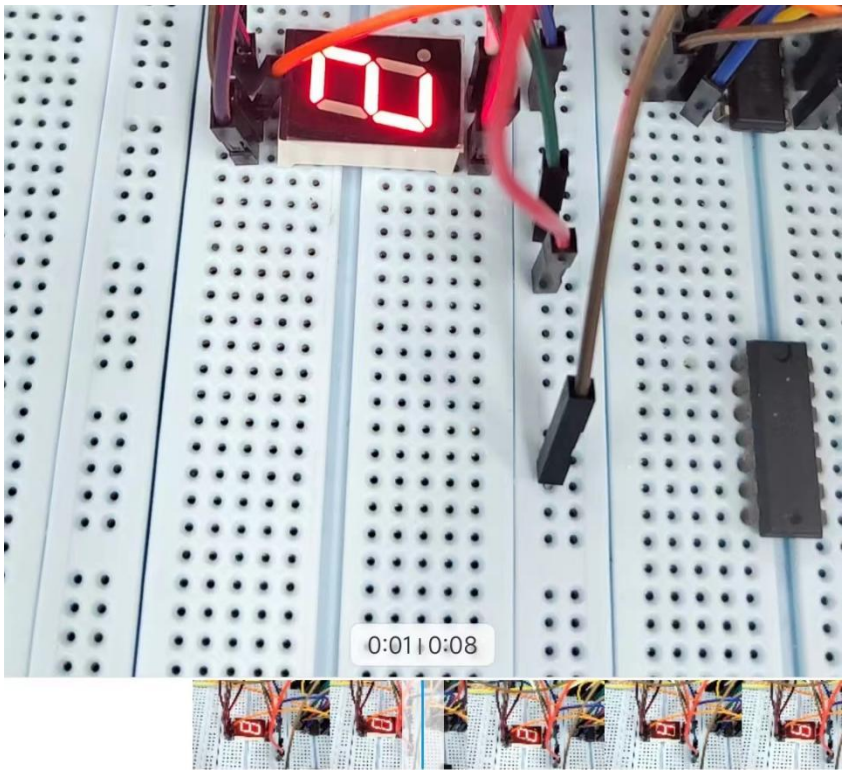


Figure 48. Displays “2”

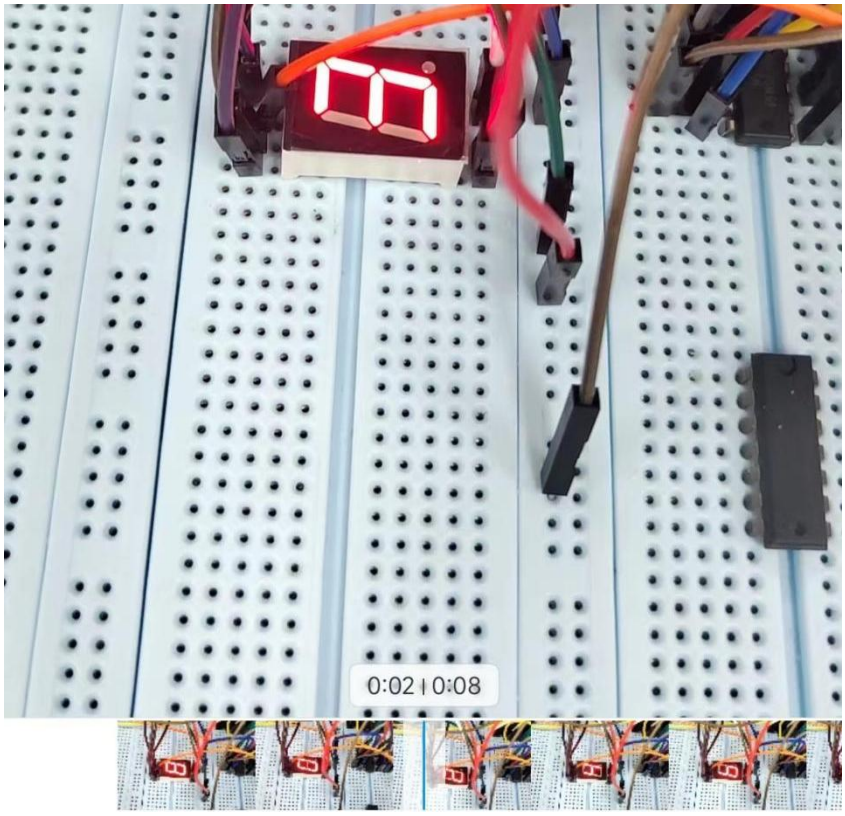


Figure 49. Displays “3”

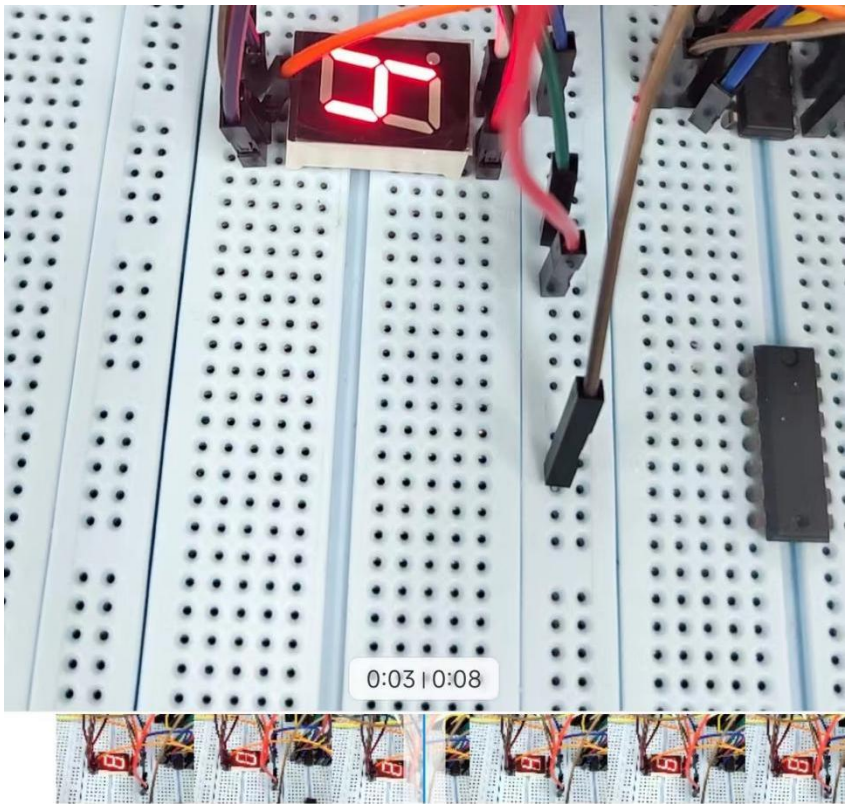


Figure 50. Displays “4”

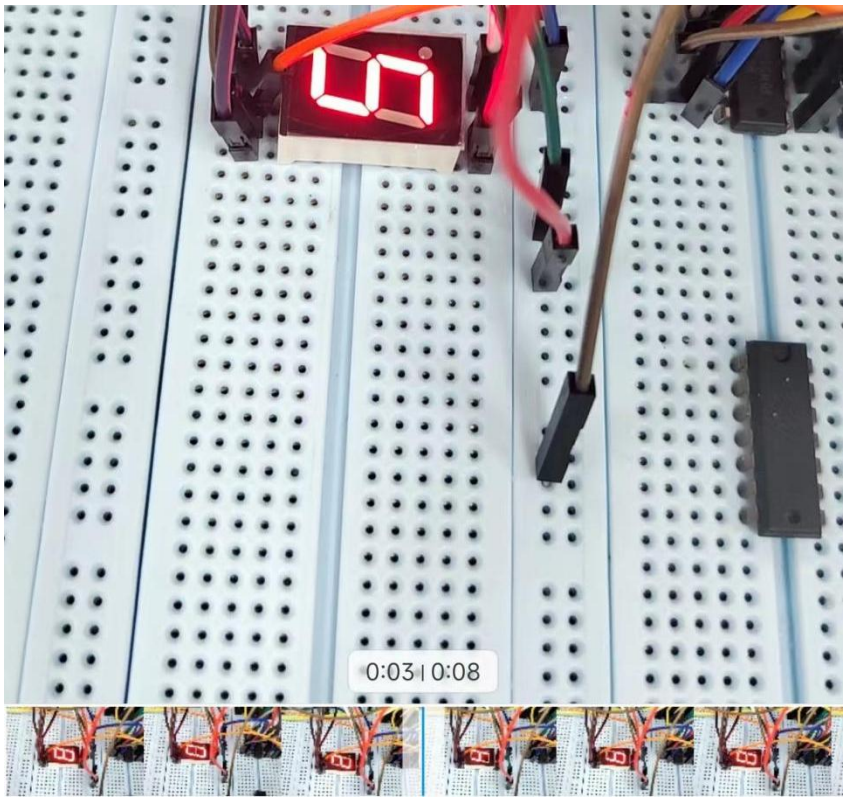


Figure 51. Displays “5”

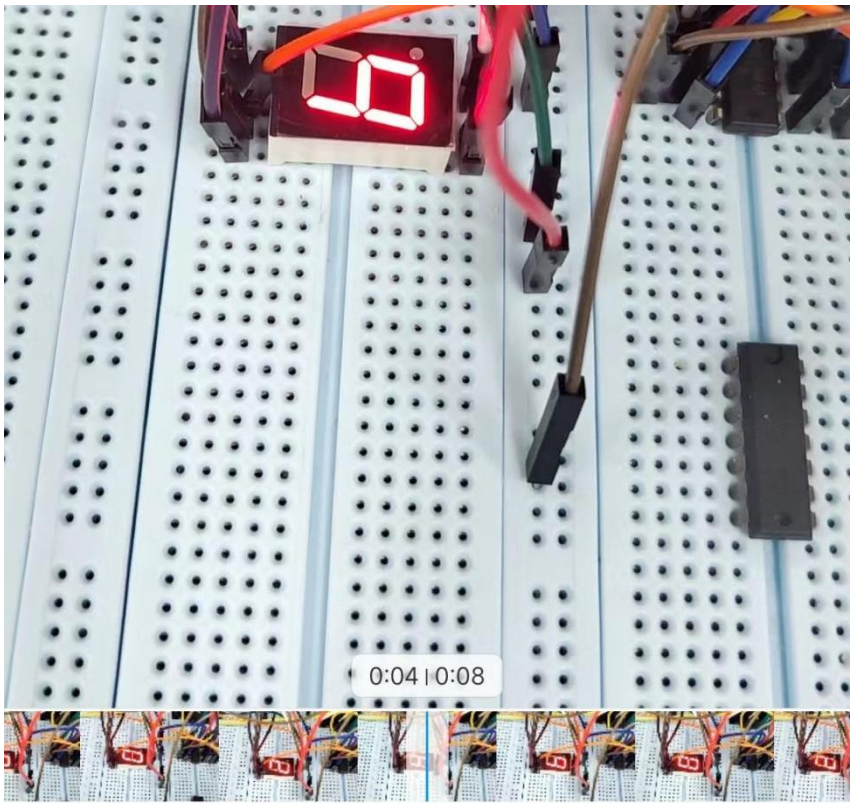


Figure 52. Displays “6”

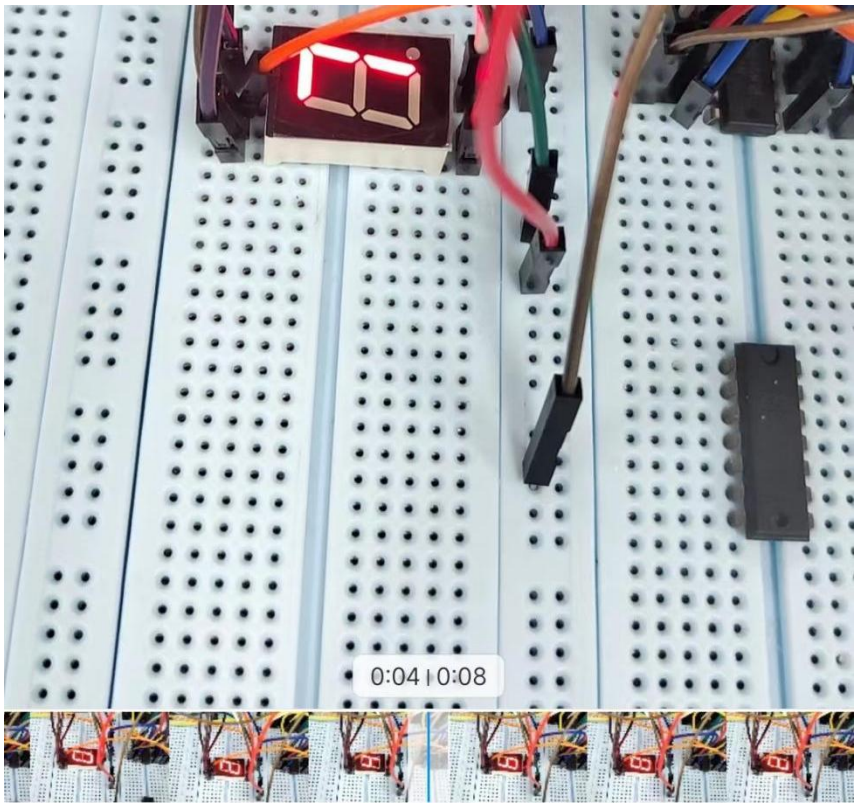


Figure 53. Displays “7”

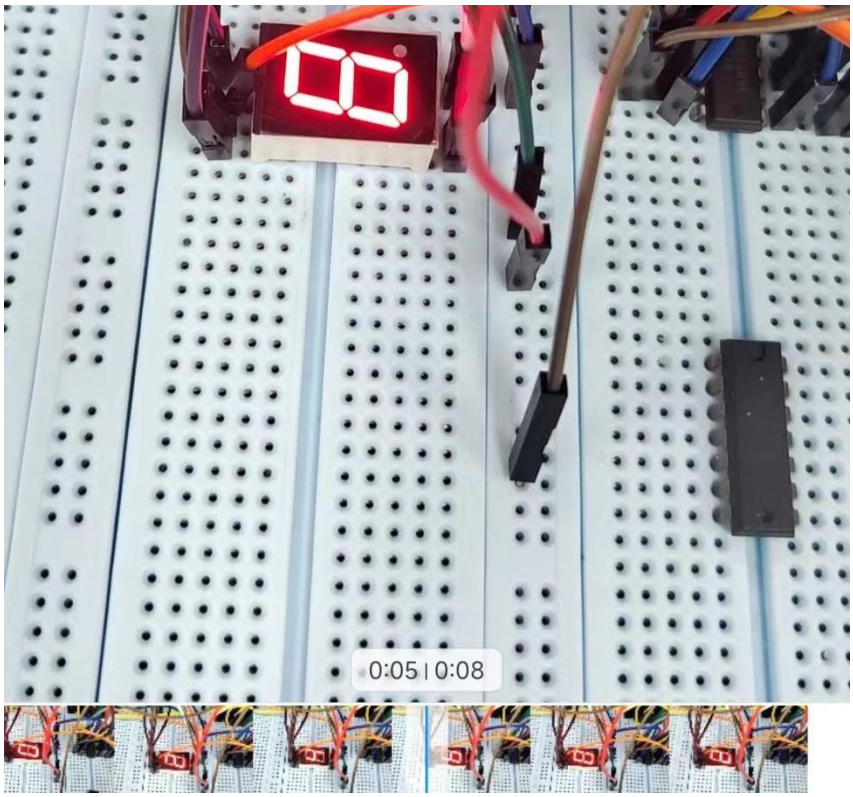


Figure 54. Displays “8”

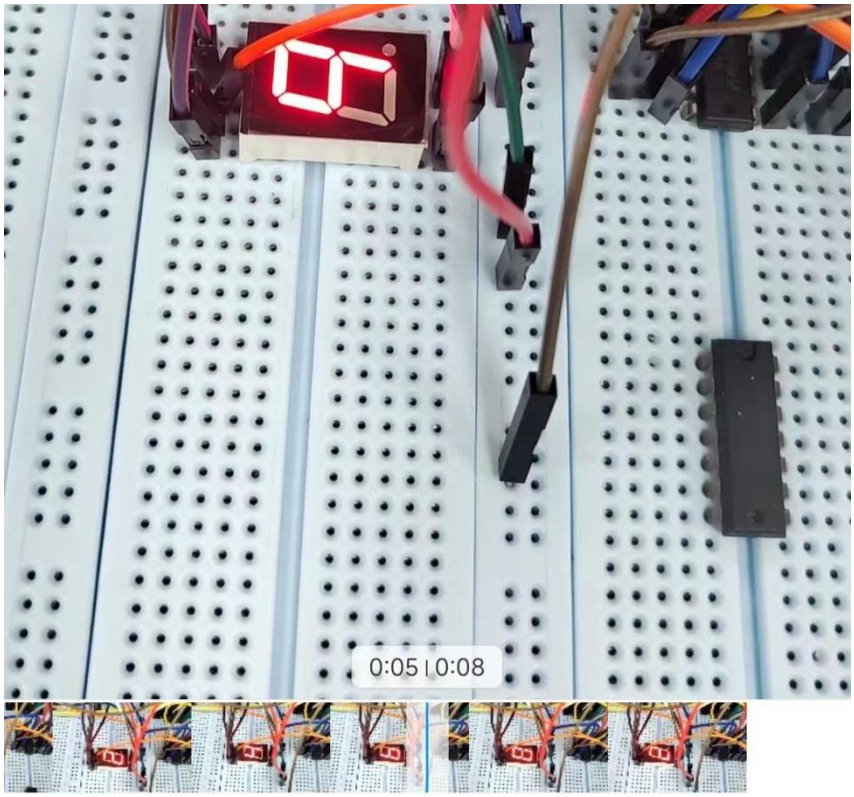


Figure 55. Displays “9”

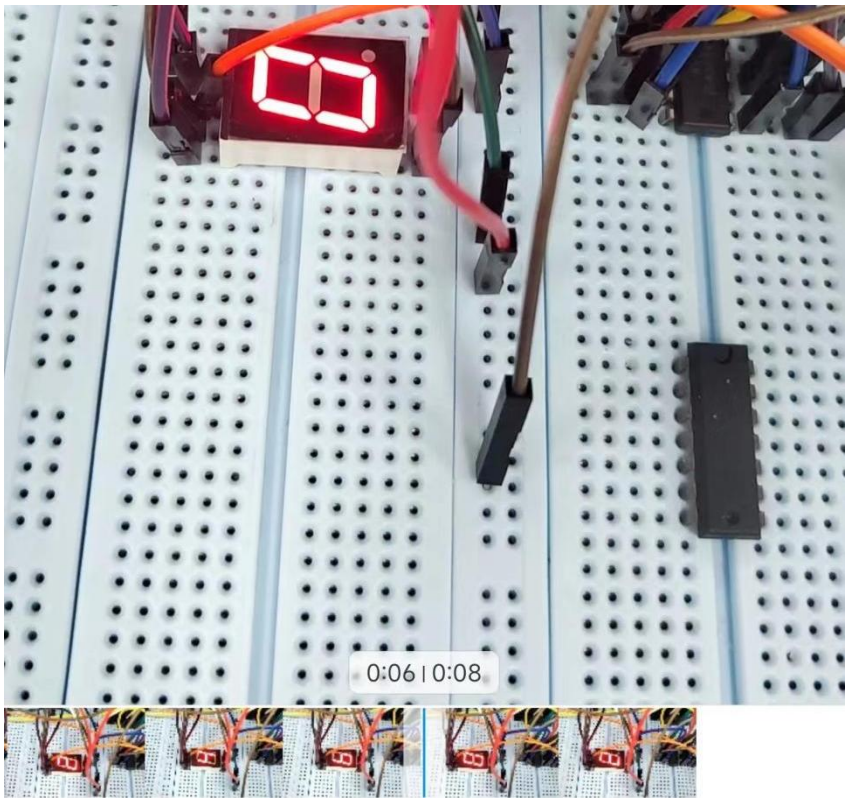


Figure 56. Displays “0” again

Figure 46 ~ Figure 56 shows the display of the 7-segement display, it flashes from 0 to 9, and back to 0 again. The weird symbols have been eliminated.

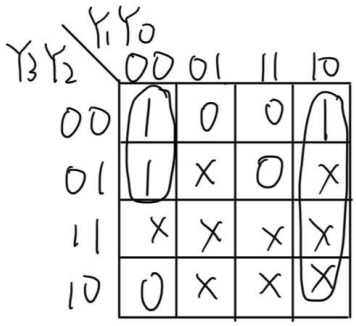
2.3 Questions

Q1.How to design a circuit generating a circuit of BCD codes from 0000=>0001=>0010=>0011=>0100=>0111=>1000=>0000=>0001=>…?

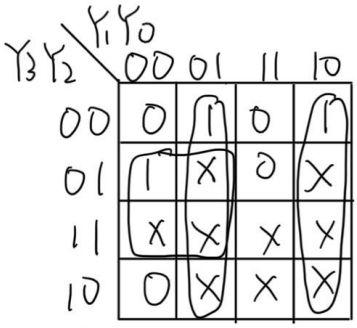
A1. The previous designs all use asynchronous counter. In this design, synchronous counter is used.

current state $Y_3\ Y_2\ Y_1\ Y_0$	next state $Y_3\ Y_2\ Y_1\ Y_0$	$D_0\ D_1\ D_2\ D_3$
0 0 0 0	0 0 0 1	1 0 0 0
0 0 0 1	0 0 1 0	0 1 0 0
0 0 1 0	0 0 1 1	1 1 0 0
0 0 1 1	0 1 0 0	0 0 1 0
0 1 0 0	0 1 1 1	1 1 1 0
0 1 1 1	1 0 0 0	0 0 0 1
1 0 0 0	0 0 0 0	0 0 0 0

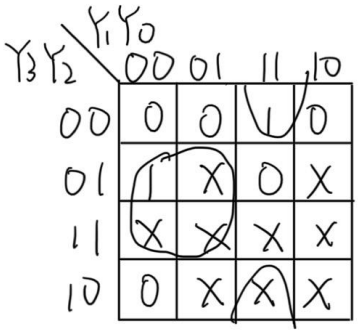
Figure 57. Next-state table



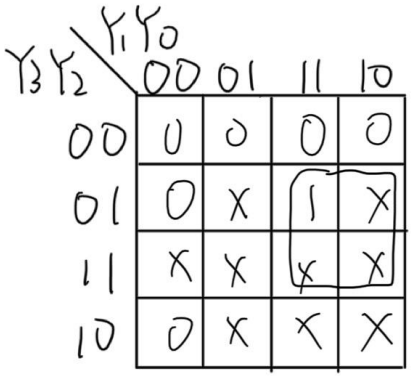
$$D_0 = \overline{Y_3}\,Y_1\,\overline{Y_0} + Y_1\,\overline{Y_0}$$
$$= \overline{Y_0}\,(\overline{Y_3} + Y_1)$$



$$D_1 = Y_2\,\overline{Y_1} + \overline{Y_1}\,Y_0 + Y_1\,\overline{Y_0}$$
$$= Y_2\,\overline{Y_1} + Y_1 \oplus Y_0$$



$$D_2 = Y_2\,\overline{Y_1} + \overline{Y_2}\,Y_1\,Y_0$$



$$D_3 = Y_2\,Y_1$$

Figure 58. Use K-map to specify Boolean expressions for each D input

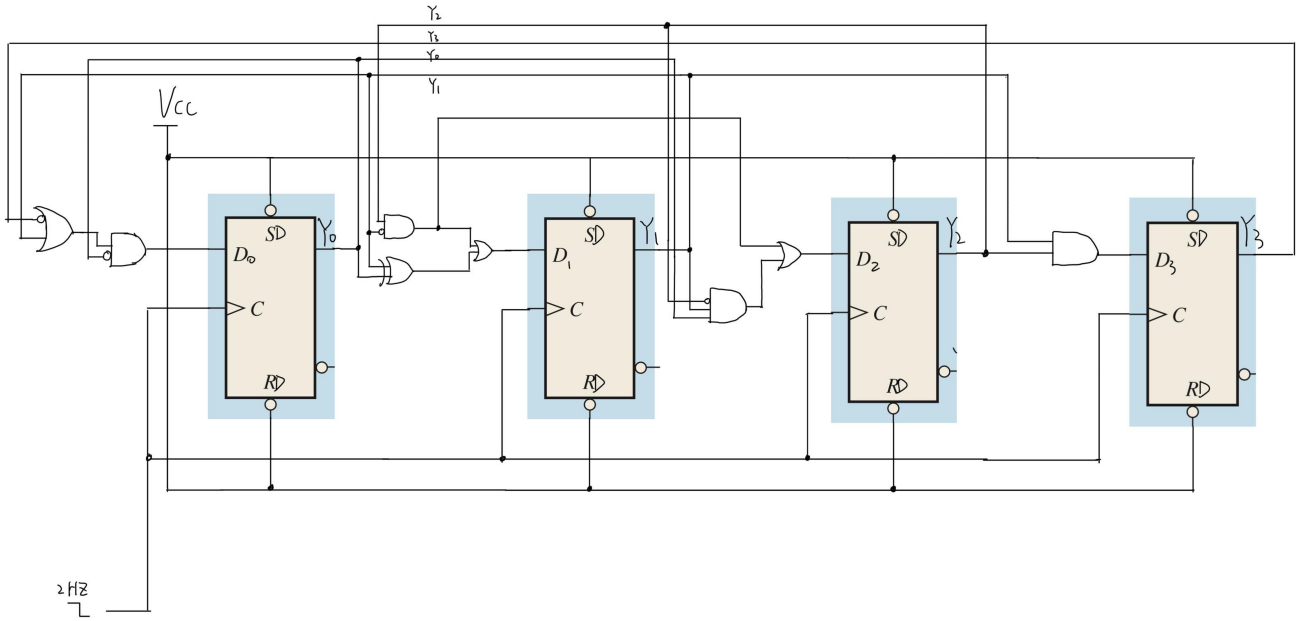


Figure 59. Designed circuit (only flip-flop part)

In Figure 59, Y0, Y1, Y2,Y3 outputs should be connected to 74HC47 and then connected to the 7-segment display to show the BCD code.

Q2. In step 3) we have provided chips which are not necessary. Can you find the least number of basic gates?
A2. Only need one NAND gate.

3. Experiment C

3.1 Design

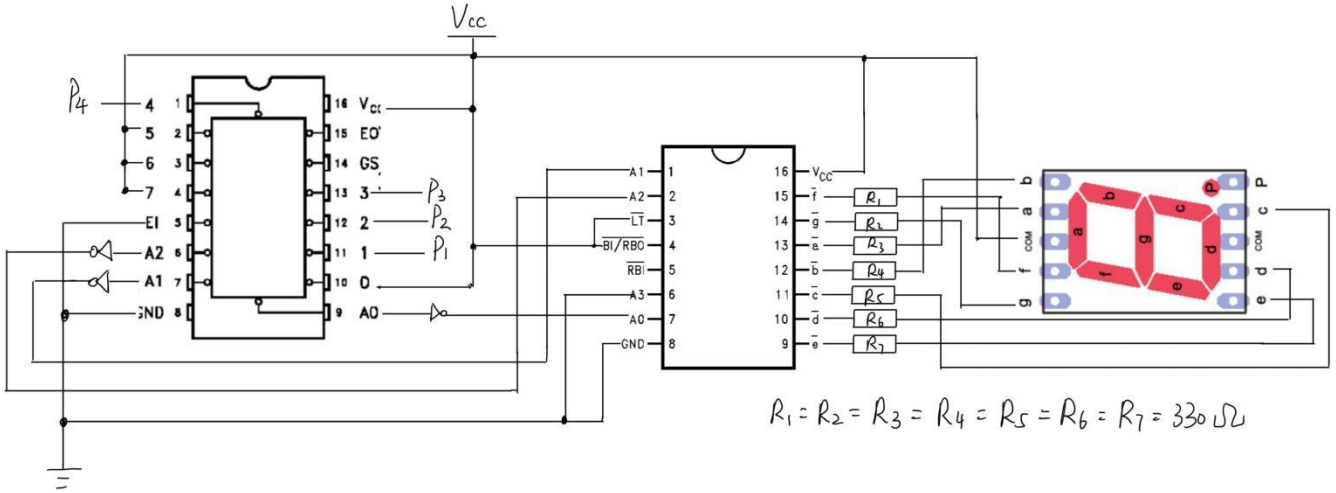


Figure 60. Designed circuit

In Figure 60, P1, P2, P3, P4 represent the $\bar{Q}$ output of each flip-flop respectively, as Figure 61 shows.

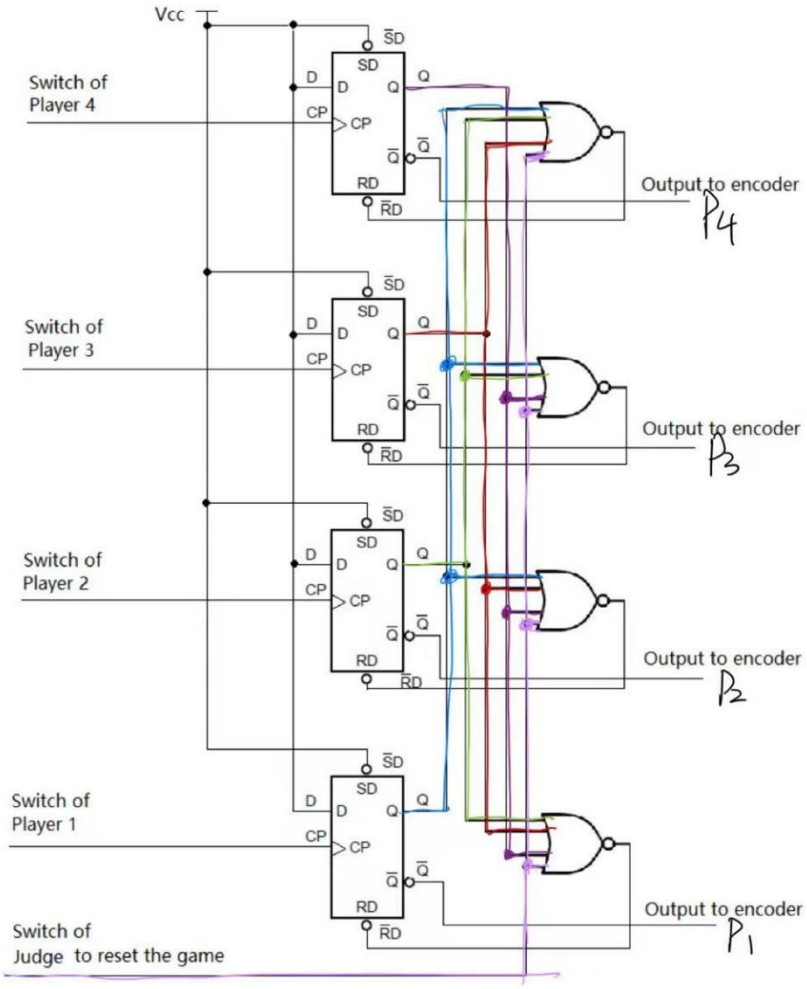


Figure 61. P1, P2, P3, P4 are $\bar{Q}$ output of each flip-flop

3.2 Result

- Build the circuit of responder module and verify

In the below figures, the SIM are connected to judge, Player 4, Player 3, Player 2, Payer 1 (from left to right), the LOM are connected to the lights corresponding to Payer 4, Payer 3, Payer 2, Payer 1 respectively (from left to right), represented by L4, L3, L2, L1. Initially, the four lights are all on. If one player presses the responder (turn on the corresponding switch of SIM), his /her light will go out.

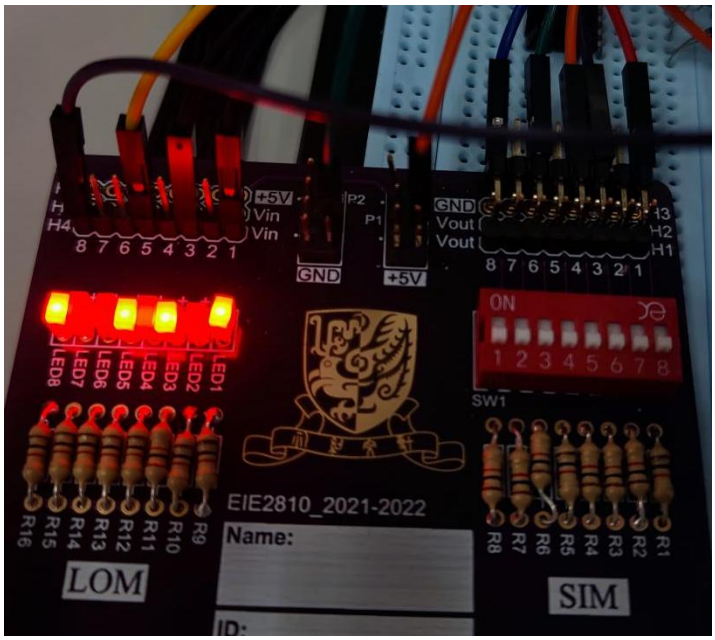


Figure 62. Initial state

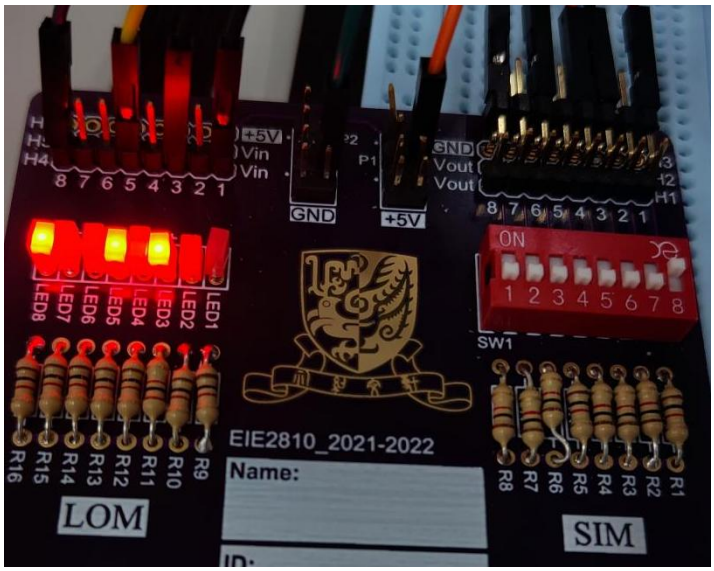


Figure 63. When player 1 presses the responder, L1 goes out

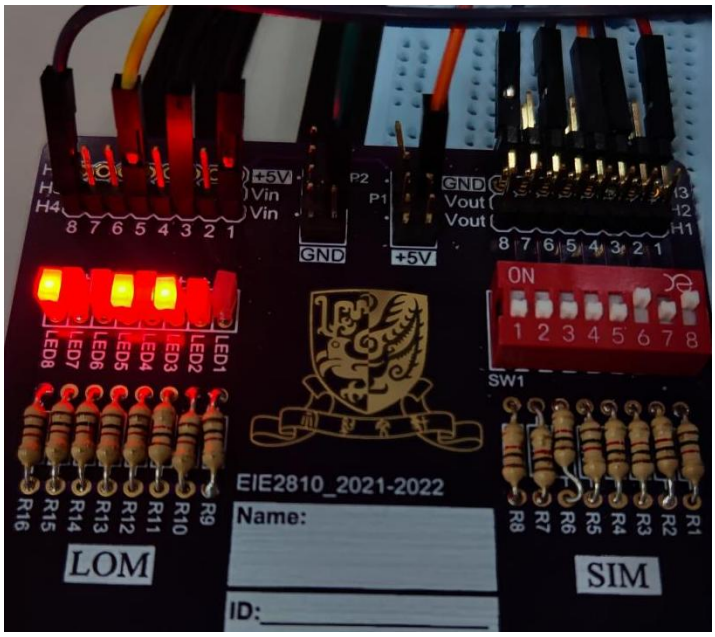


Figure 64. Player 2 presses the responder after player 1, no change

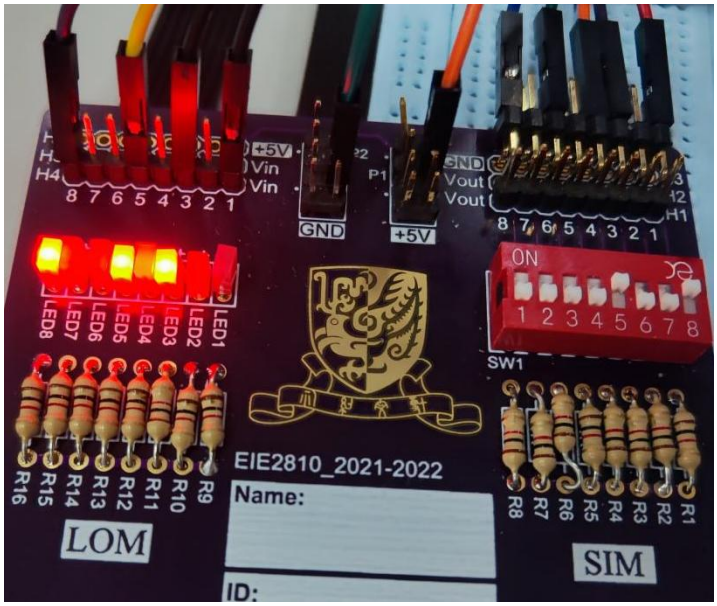


Figure 65. Player 3 presses the responder after player 1, no change

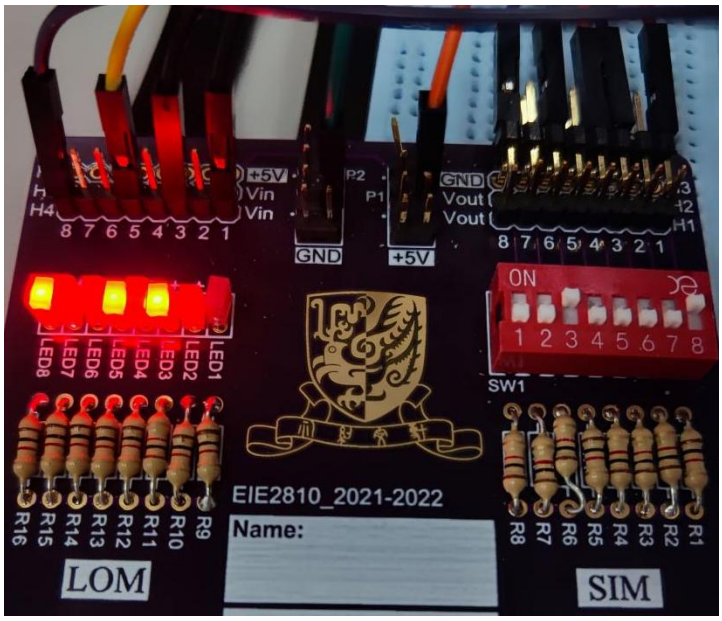


Figure 66. Player 4 presses the responder after player 1, no change

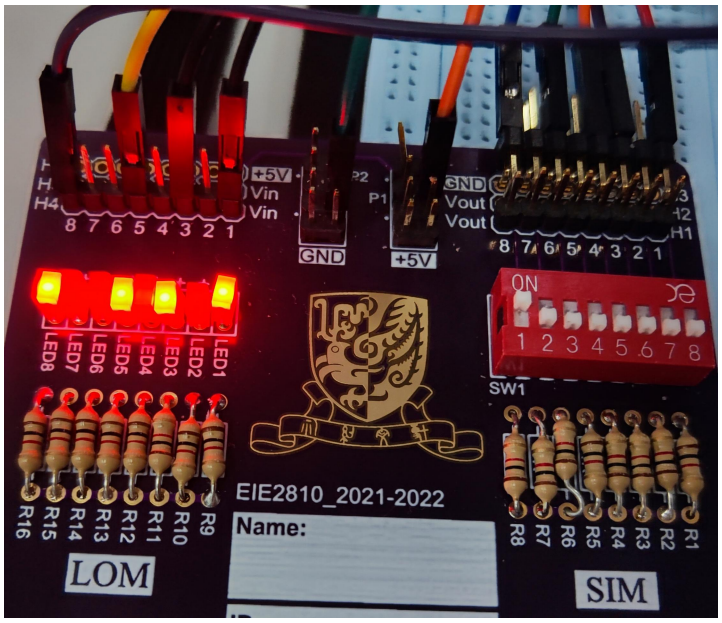


Figure 67. Judge clears the signal

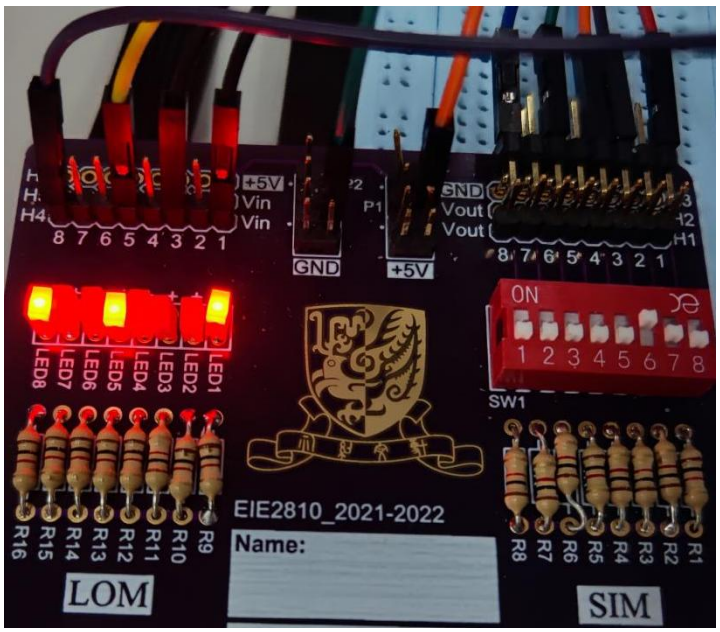


Figure 68. When player 2 presses the responder, L2 goes out

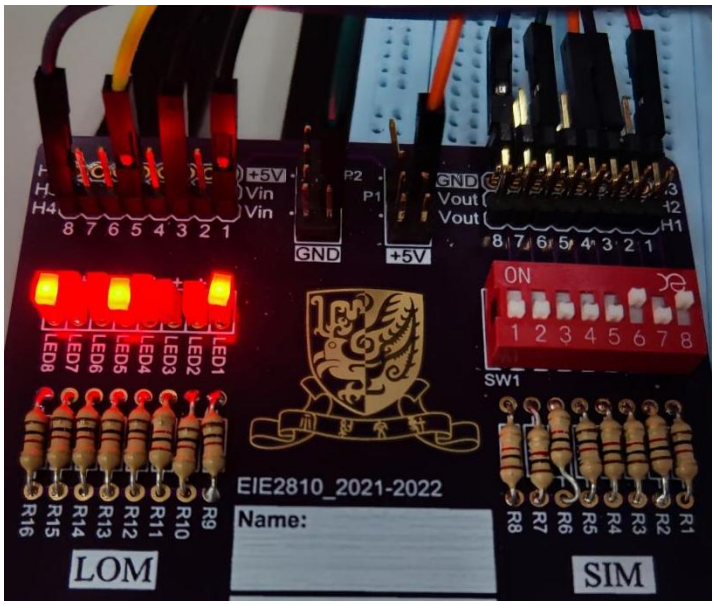


Figure 69. Player 1 presses the responder after player 2, no change

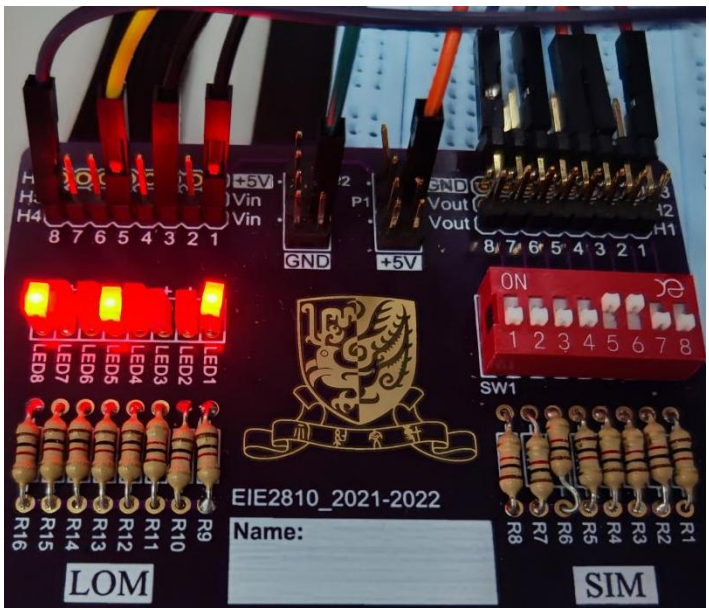


Figure 70. Player 3 presses the responder after player 2, no change

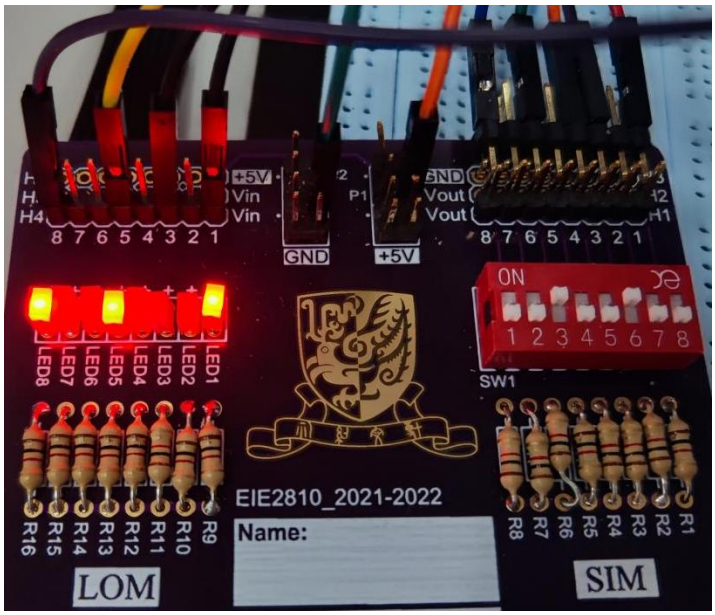


Figure 71. Player 4 presses the responder after player 2, no change

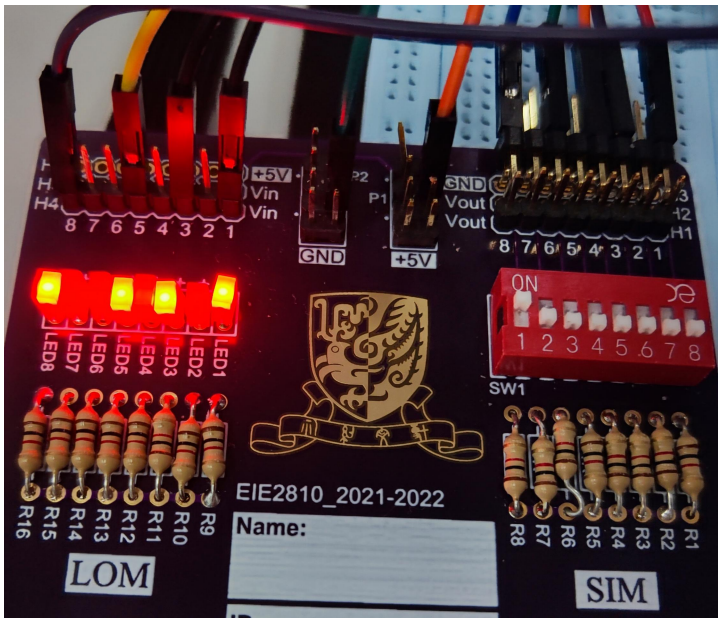


Figure 72. Judge clears the signal

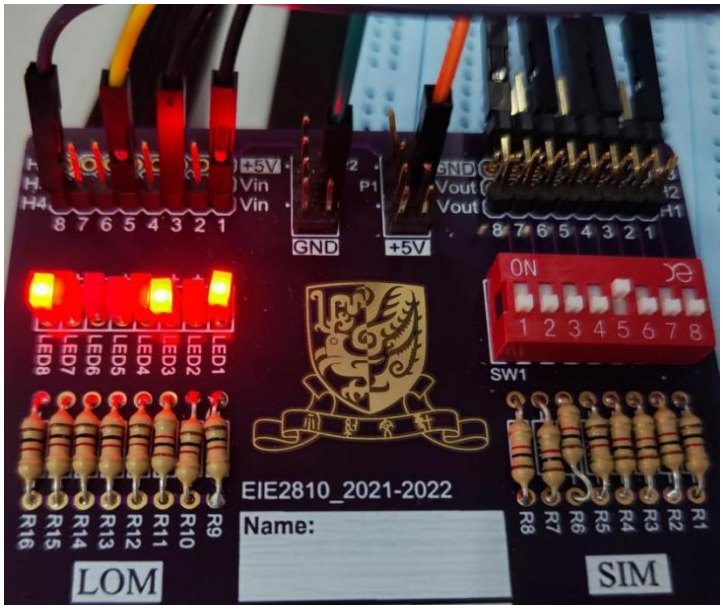


Figure 73. When player 3 presses the responder, L3 goes out

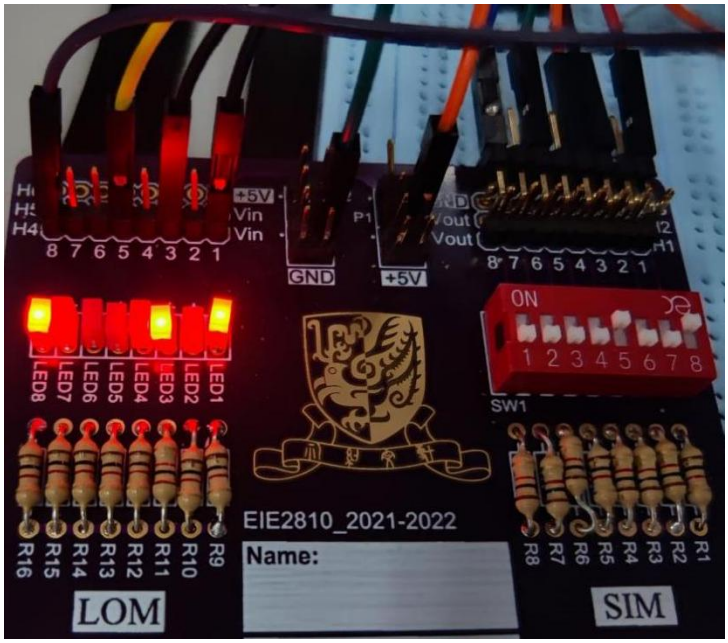


Figure 74. Player 1 presses the responder after player 3, no change

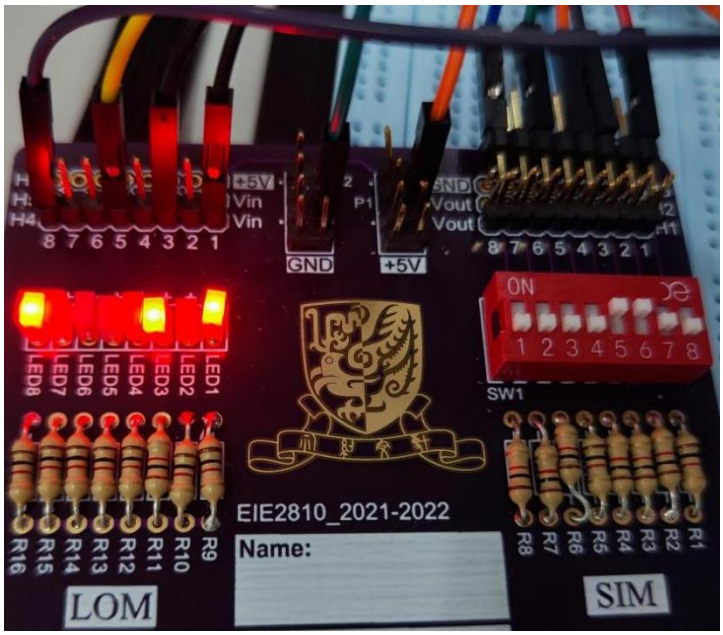


Figure 75. Player 2 presses the responder after player 3, no change

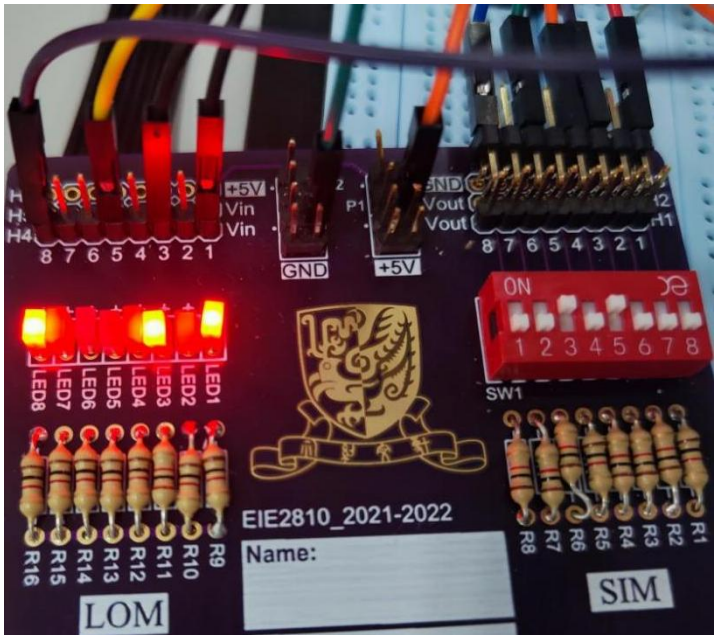


Figure 76. Player 4 presses the responder after player 3, no change

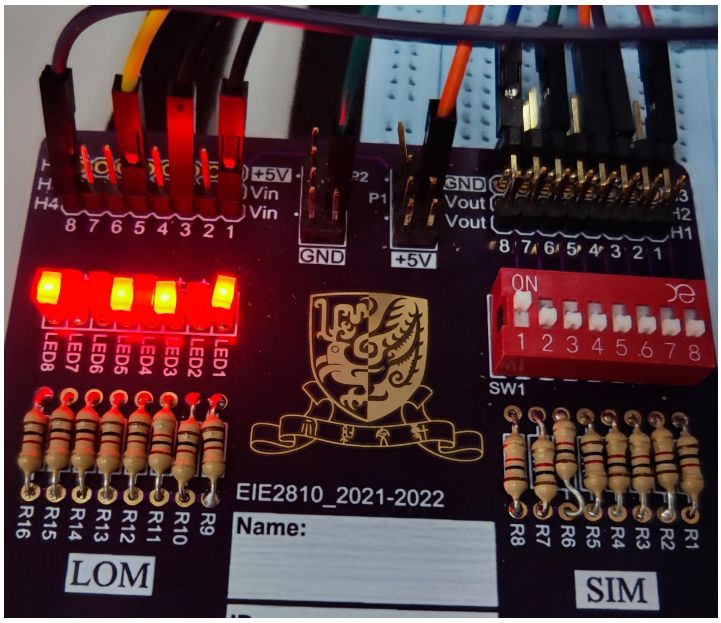


Figure 77. Judge clears the signal

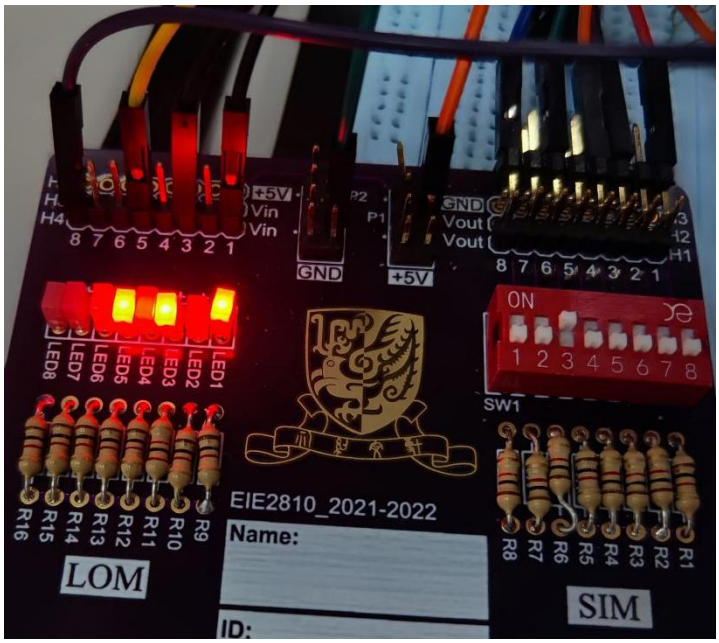


Figure 78. When player 4 presses the responder, L4 goes out

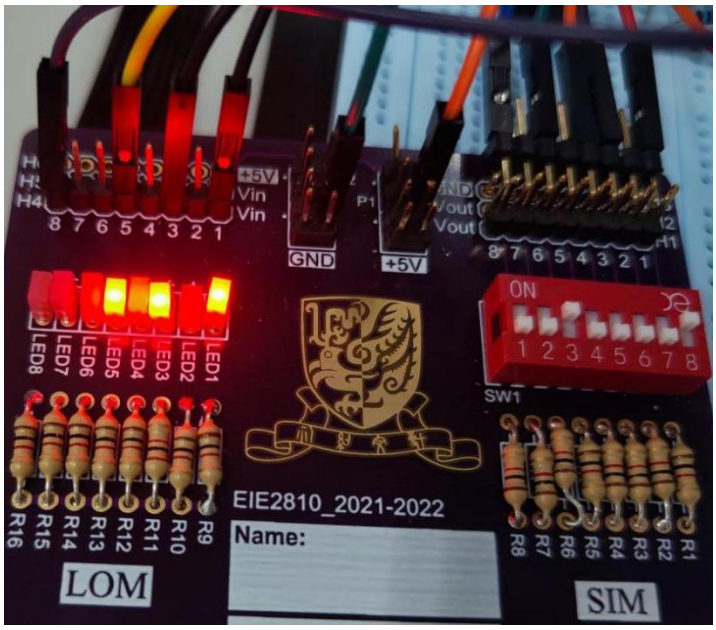


Figure 79. Player 1 presses the responder after player 4, no change

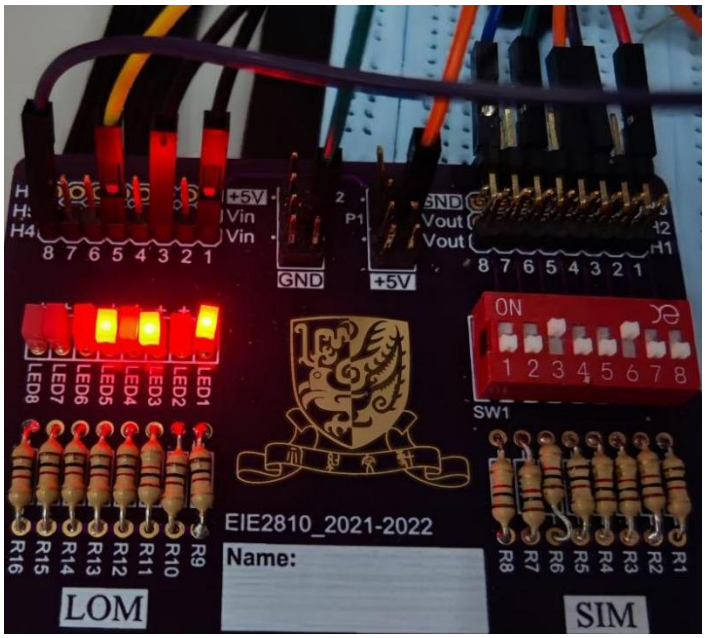


Figure 80. Player 2 presses the responder after player 4, no change

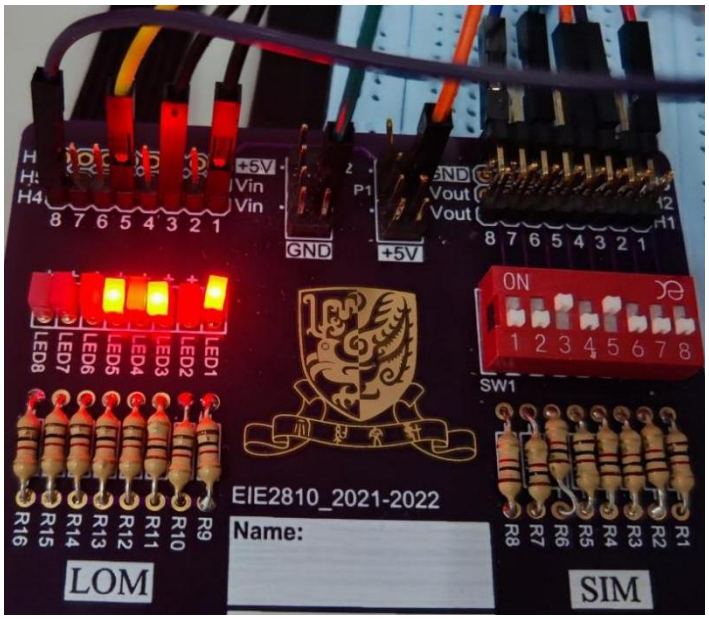


Figure 81. Player 3 presses the responder after player 4, no change

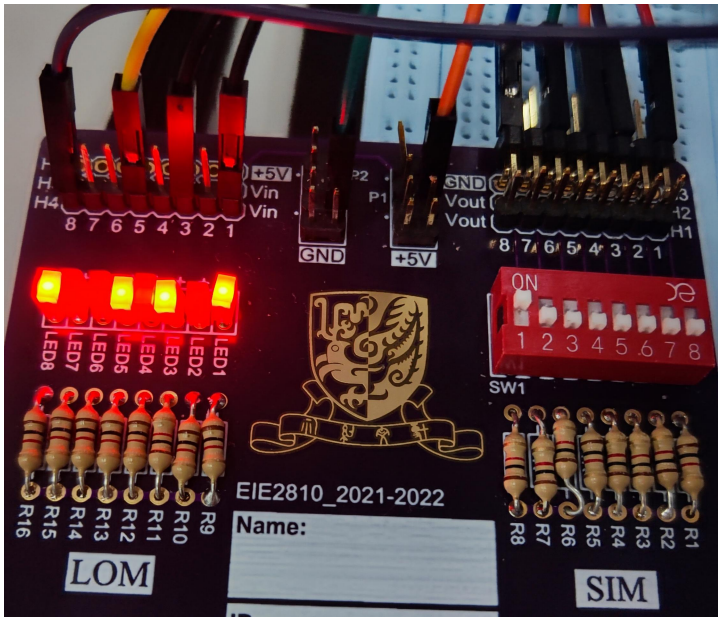


Figure 82. Judge clears the signal

- Connect the 4 outputs to 8-3 line priority encoder. The fastest player's ID will be shown as BCD code in LOM

In the below figures, the SIM are connected to judge, Player 4, Player 3, Player 2, Payer 1 (from left to right), the LOM shows the MSB to LSB of the BCD code (from left to right).

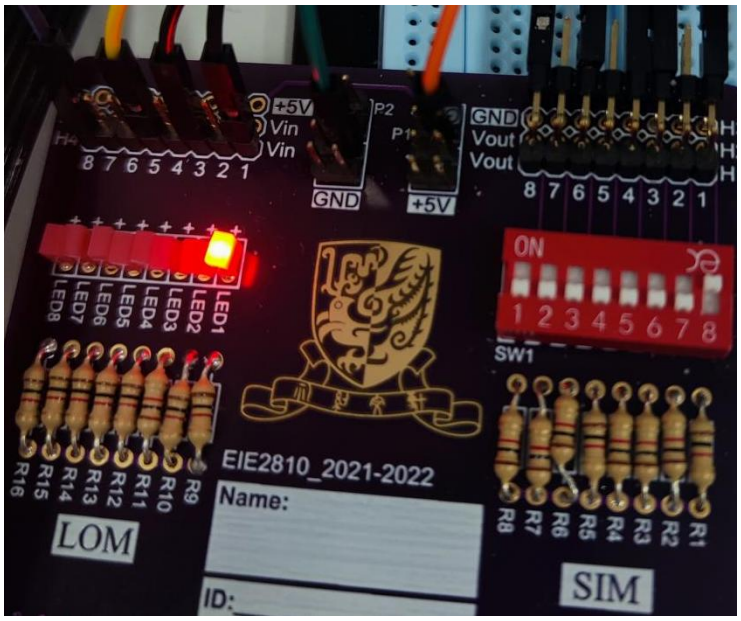


Figure 83. When player 1 presses the responder, BCD 0001 is shown

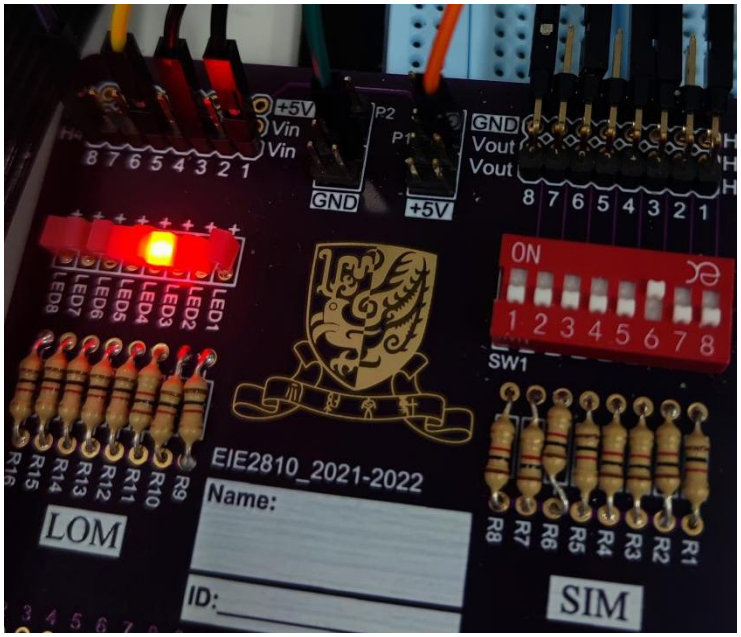


Figure 84. When player 2 presses the responder, BCD 0010 is shown

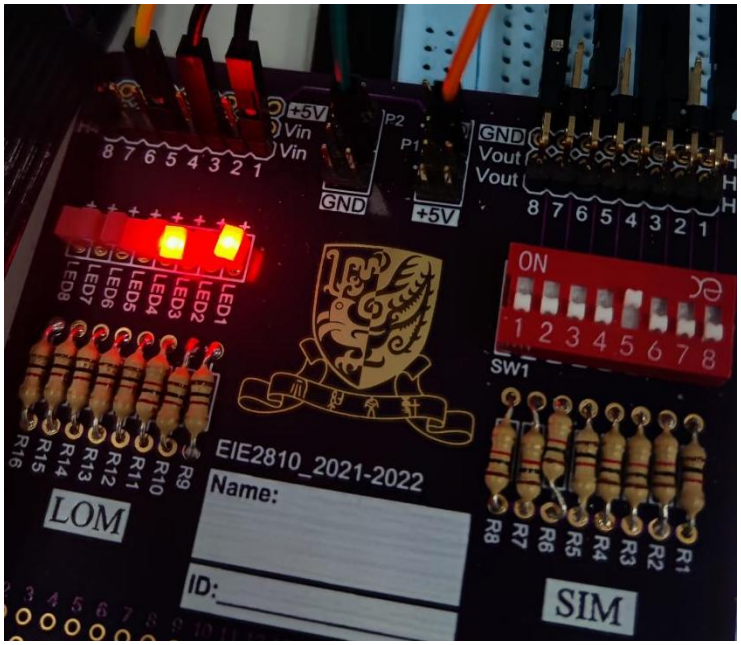


Figure 85. When player 3 presses the responder, BCD 0011 is shown

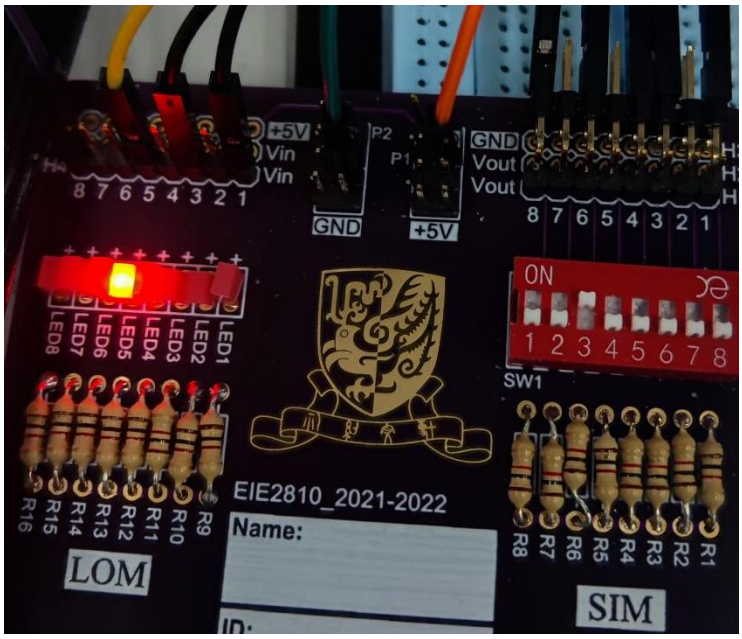


Figure 86. When player 4 presses the responder, BCD 0100 is shown

- Show the BCD code into the 7-segment display

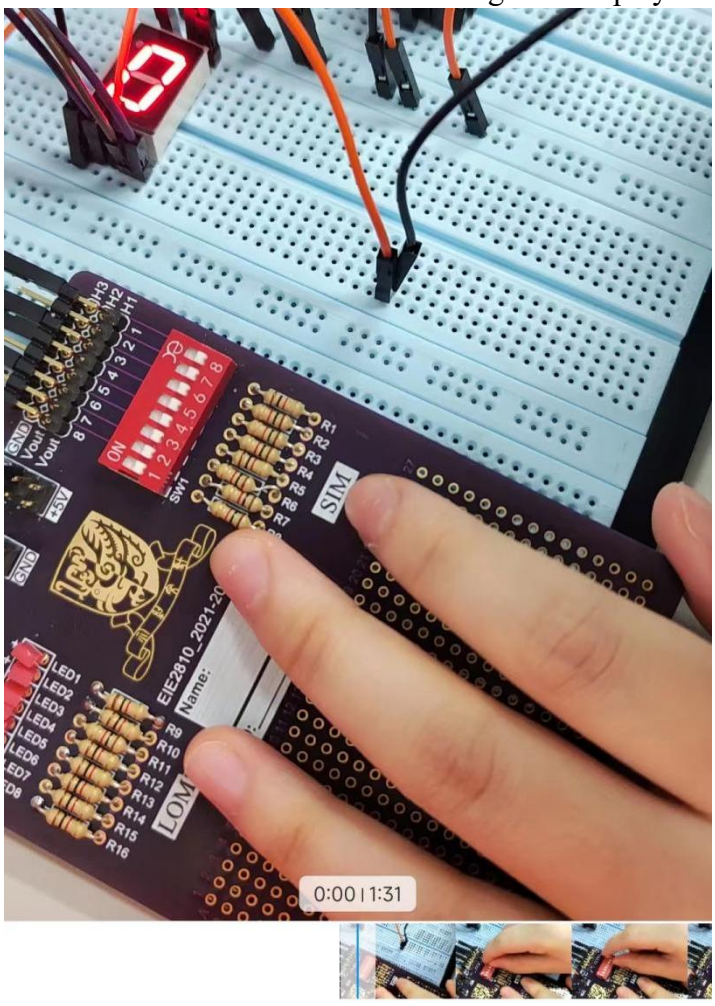


Figure 87. Initial state, “0” is displayed

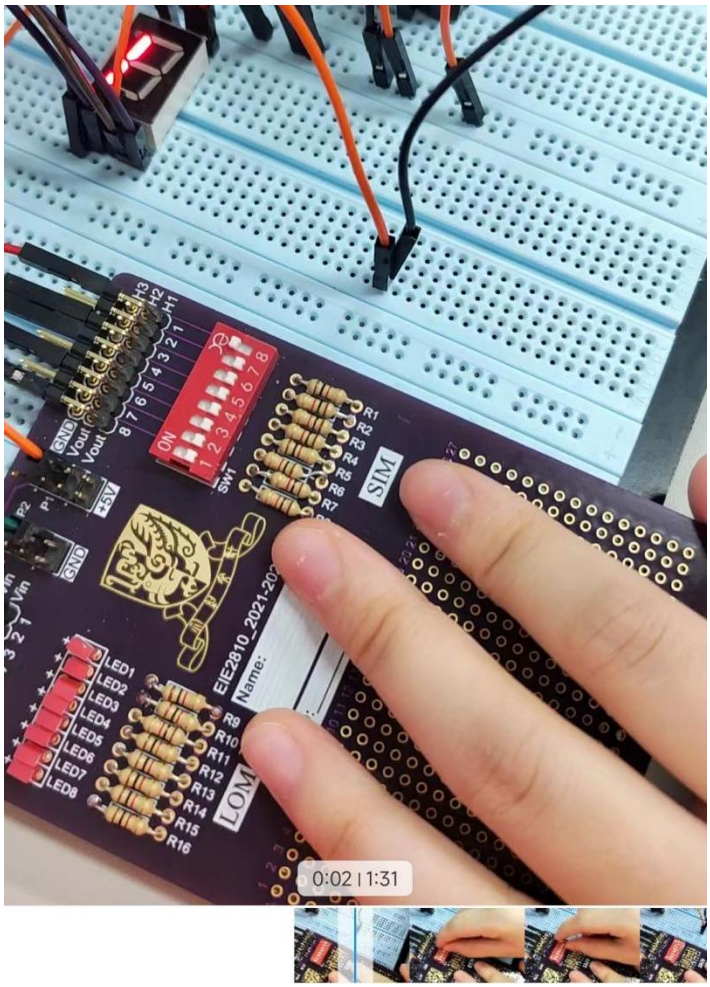


Figure 88. When player 1 presses the responder, “1” is displayed

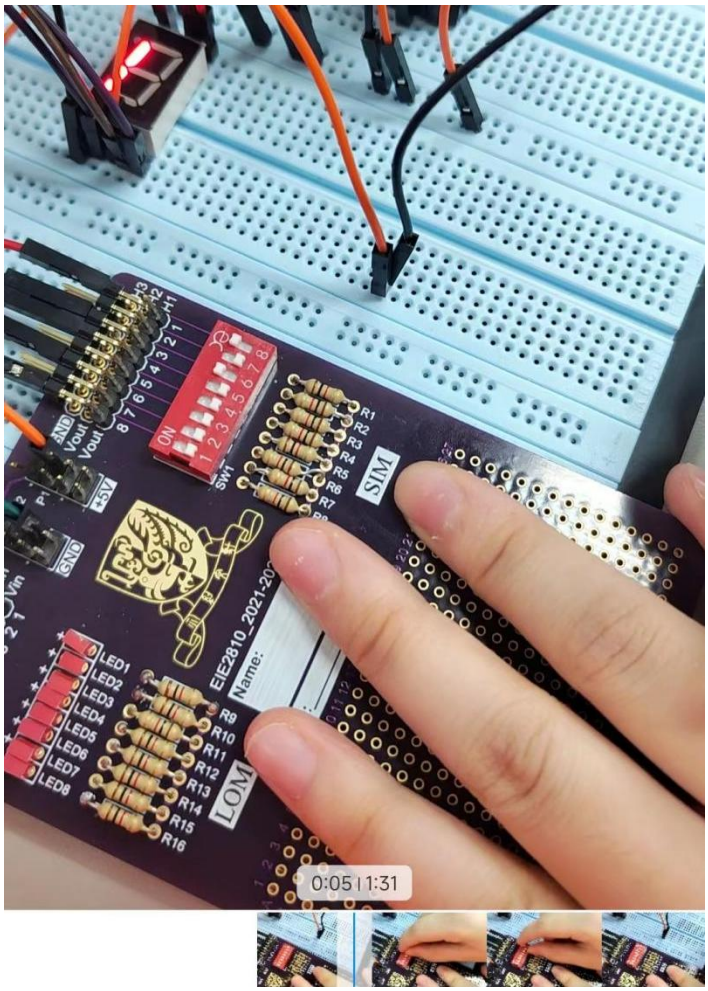


Figure 89. Player 2 presses the responder after player 1, no change

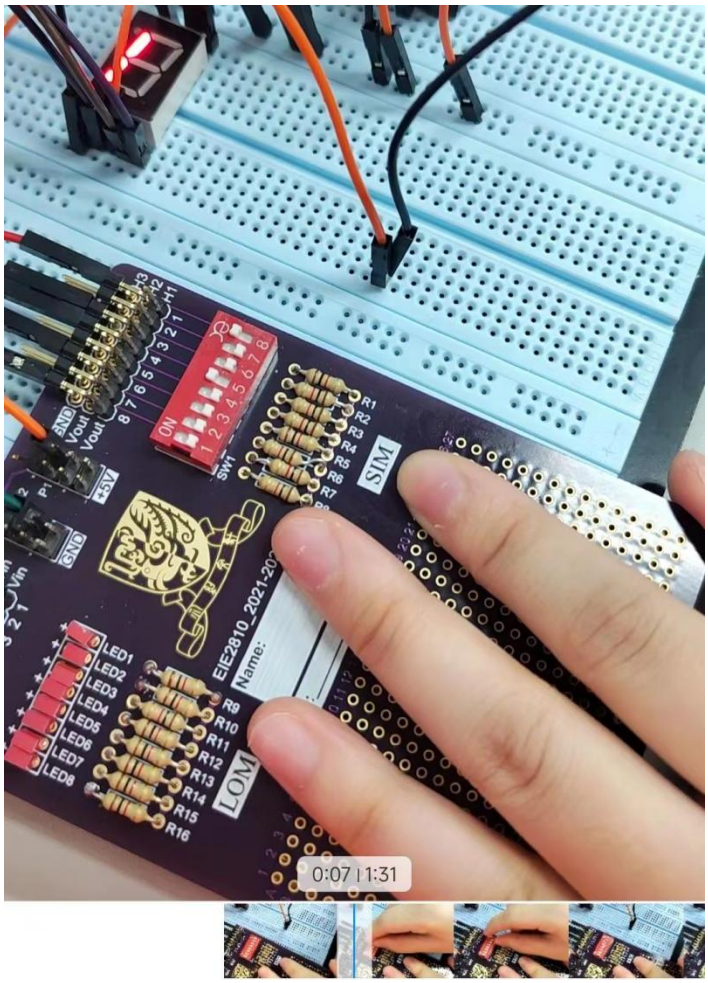


Figure 90. Player 3 presses the responder after player 1, no change

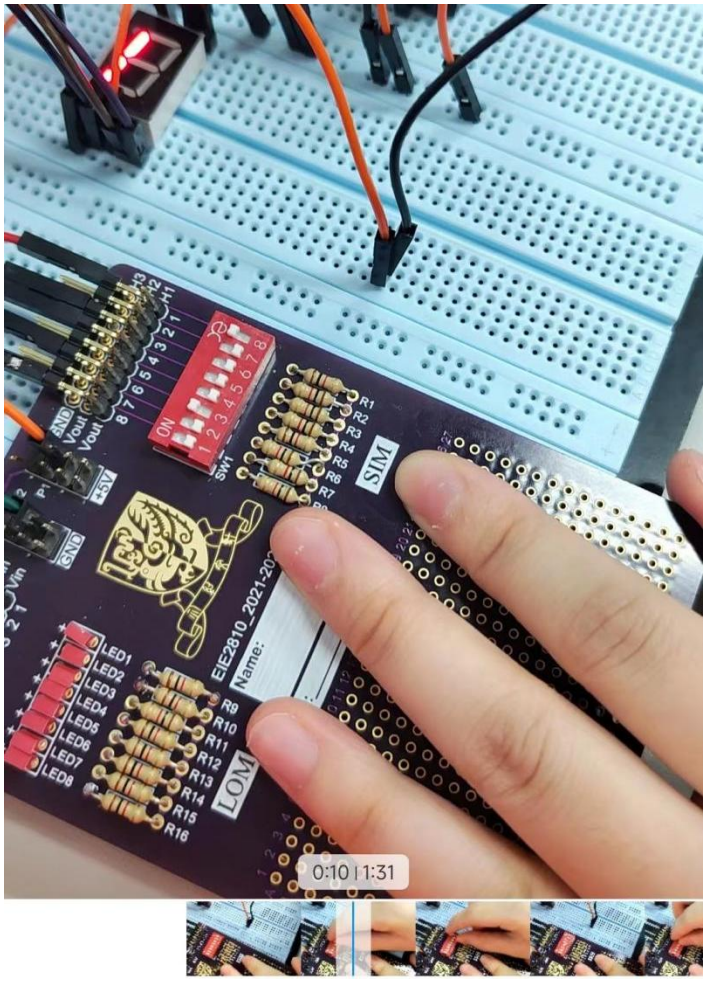


Figure 91. Player 4 presses the responder after player 1, no change

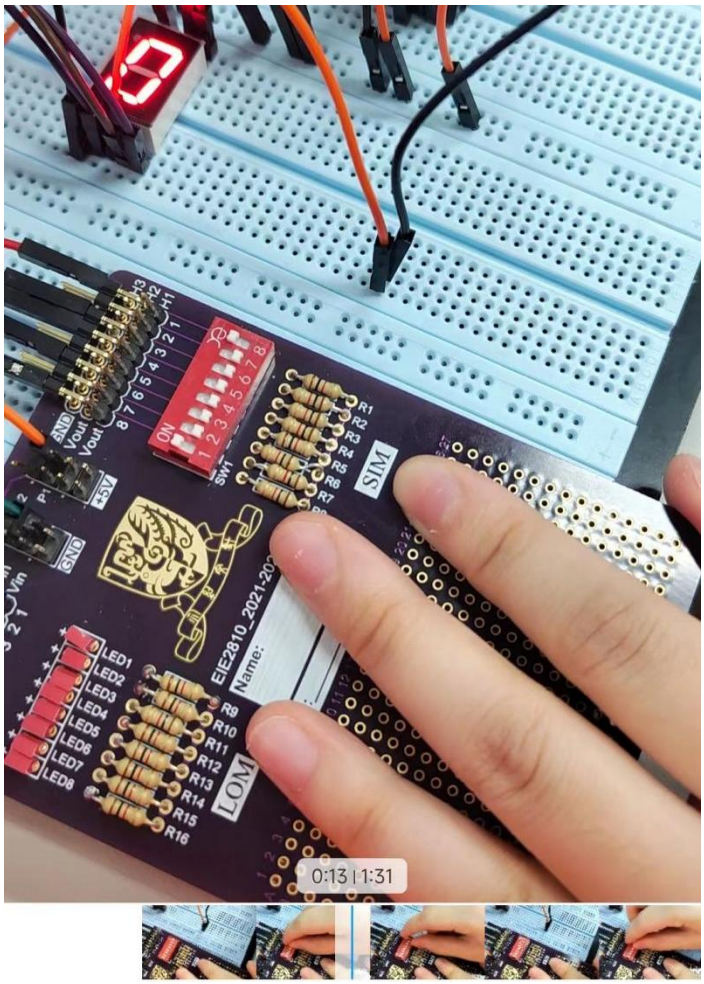


Figure 92. Judge clears the signal

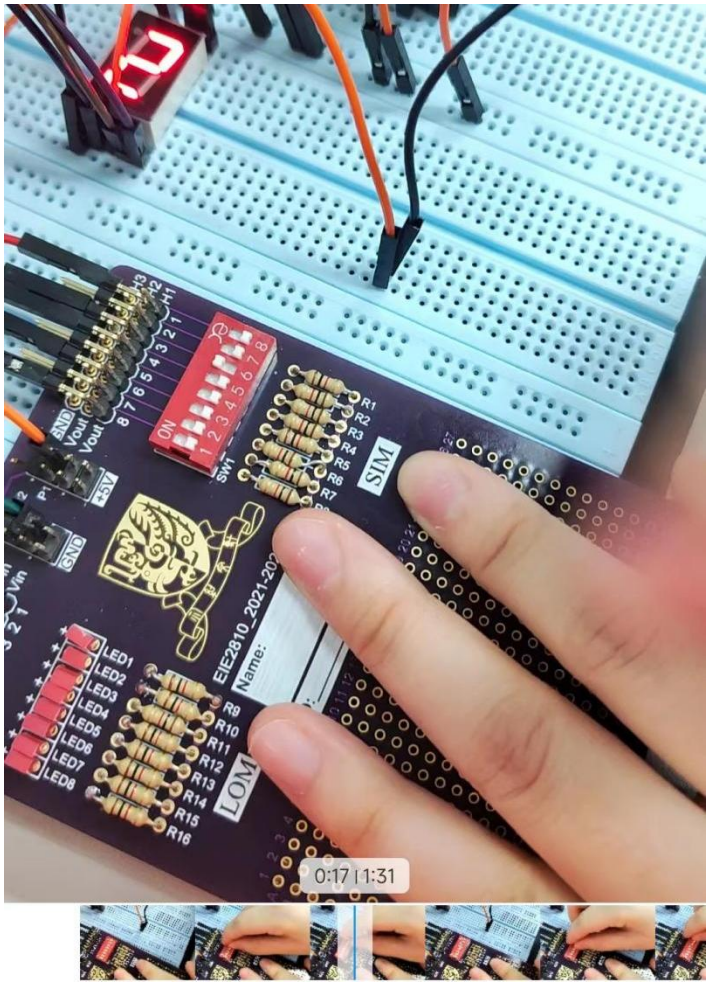


Figure 93. When player 2 presses the responder, “2” is displayed

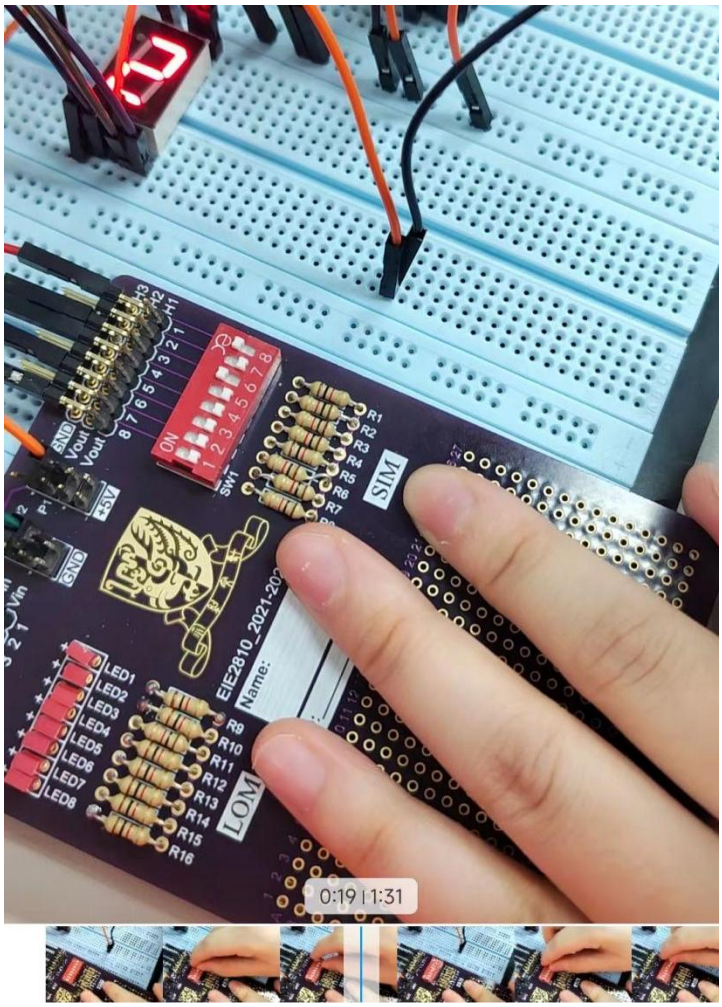


Figure 94. Player 1 presses the responder after player 2, no change

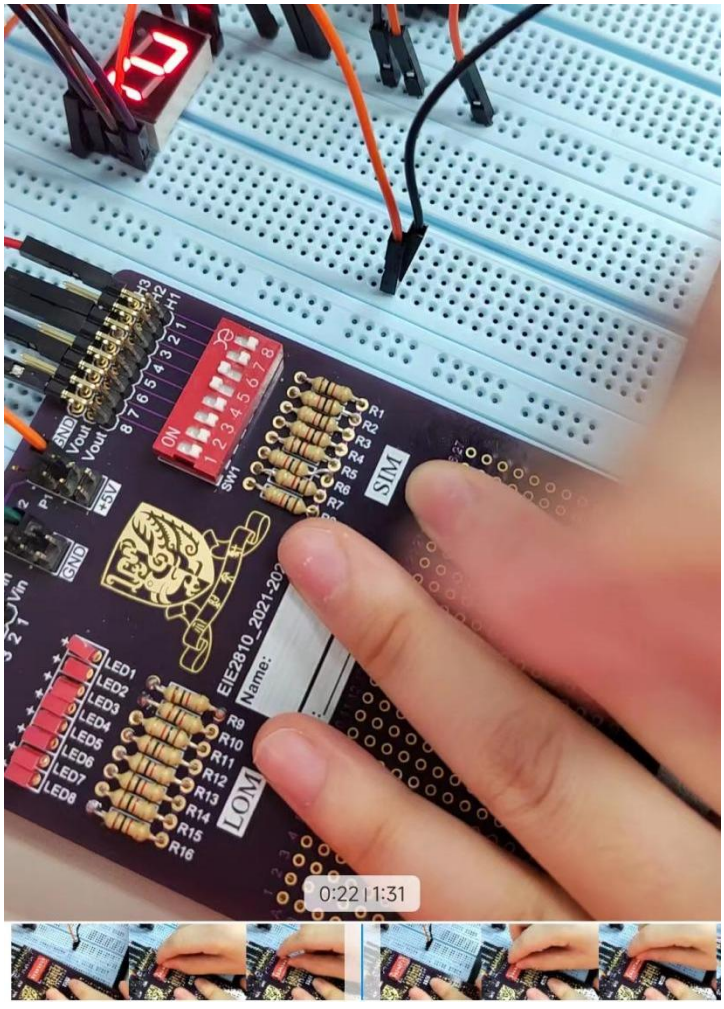


Figure 95. Player 3 presses the responder after player 2, no change

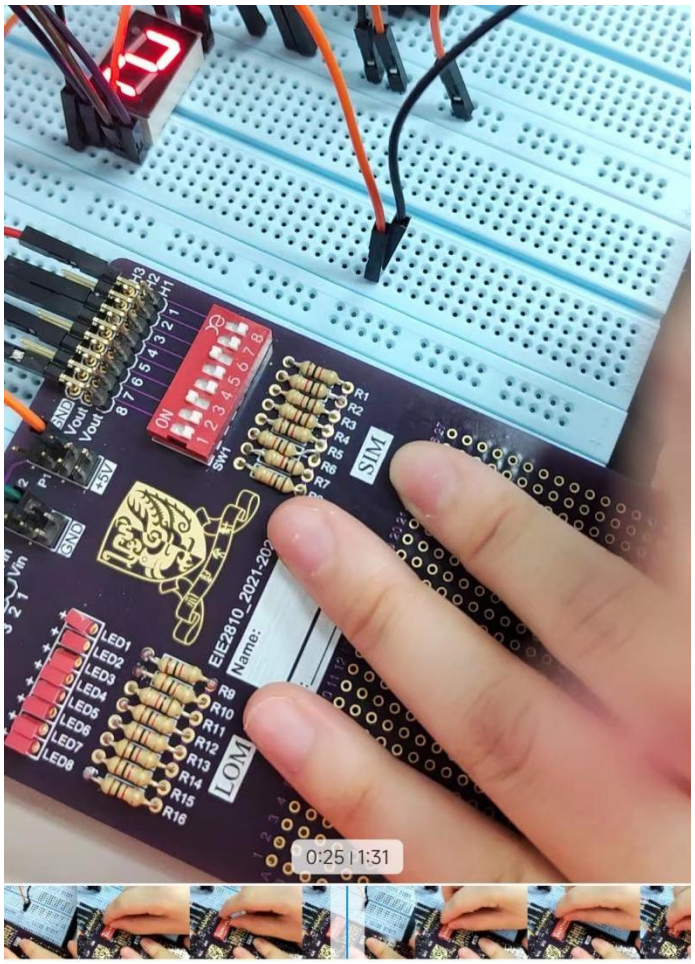


Figure 96. Player 4 presses the responder after player 2, no change

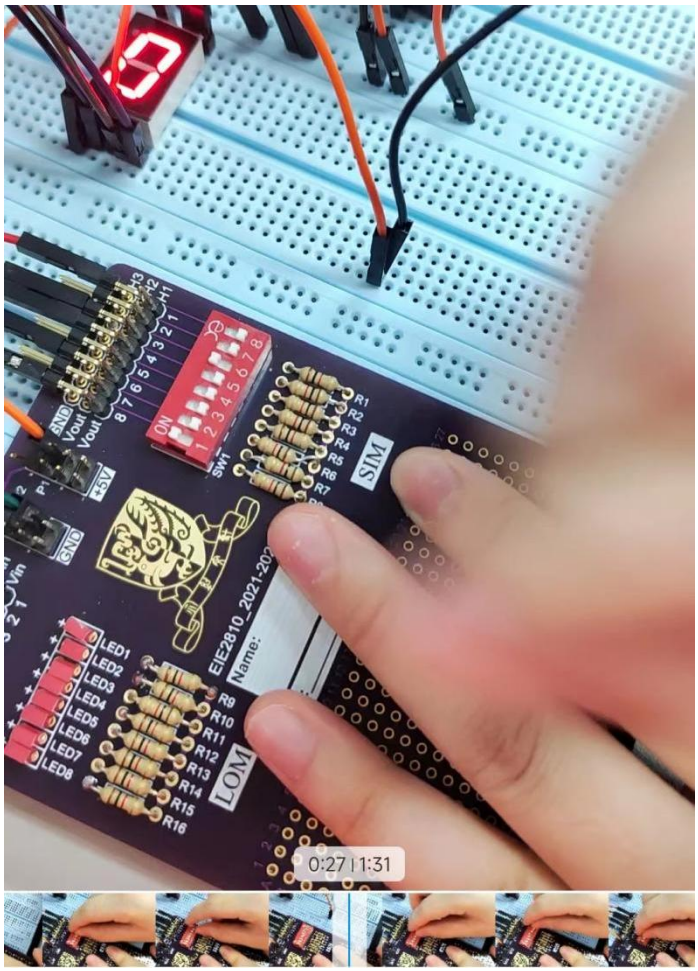


Figure 97. Judge clears the signal

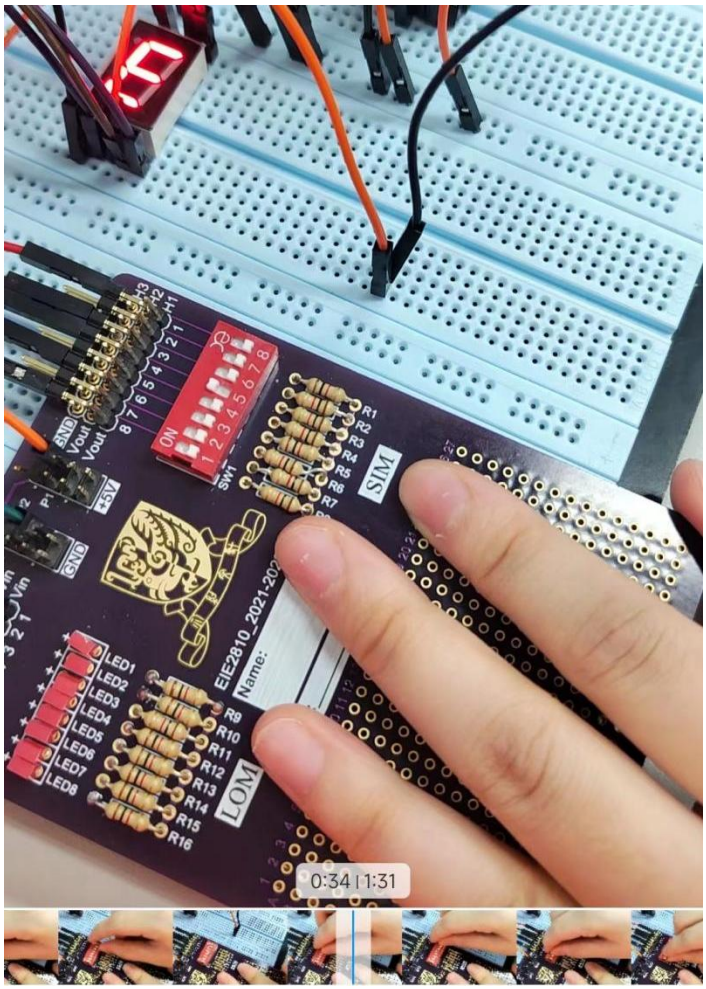


Figure 98. When player 3 presses the responder, “3” is displayed

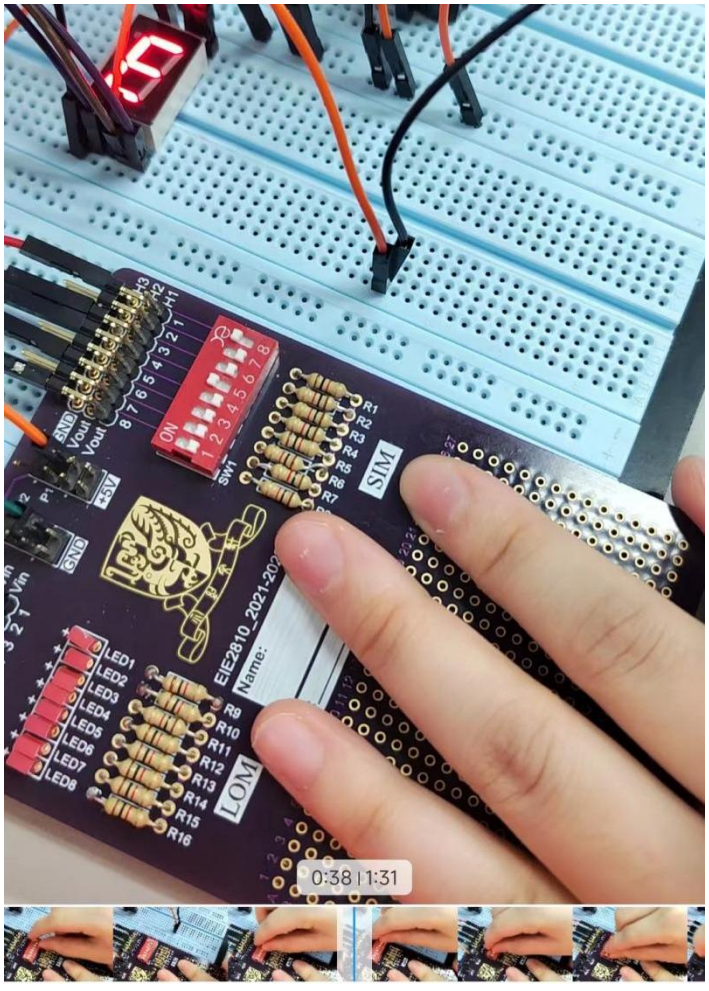


Figure 99. Player 1 presses the responder after player 3, no change

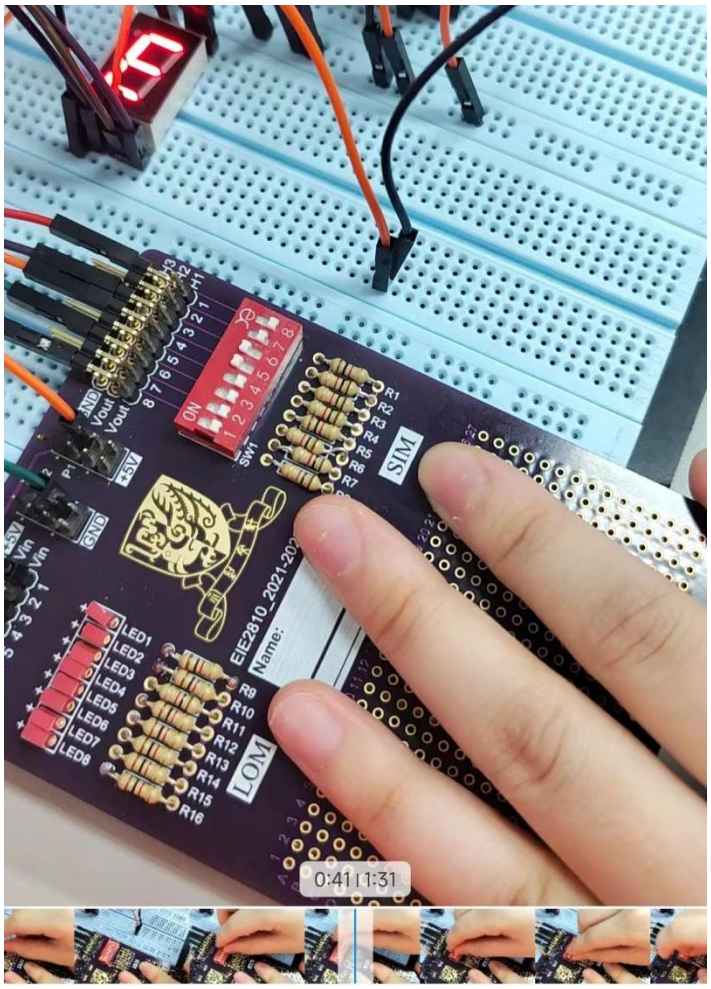


Figure 100. Player 2 presses the responder after player 3, no change

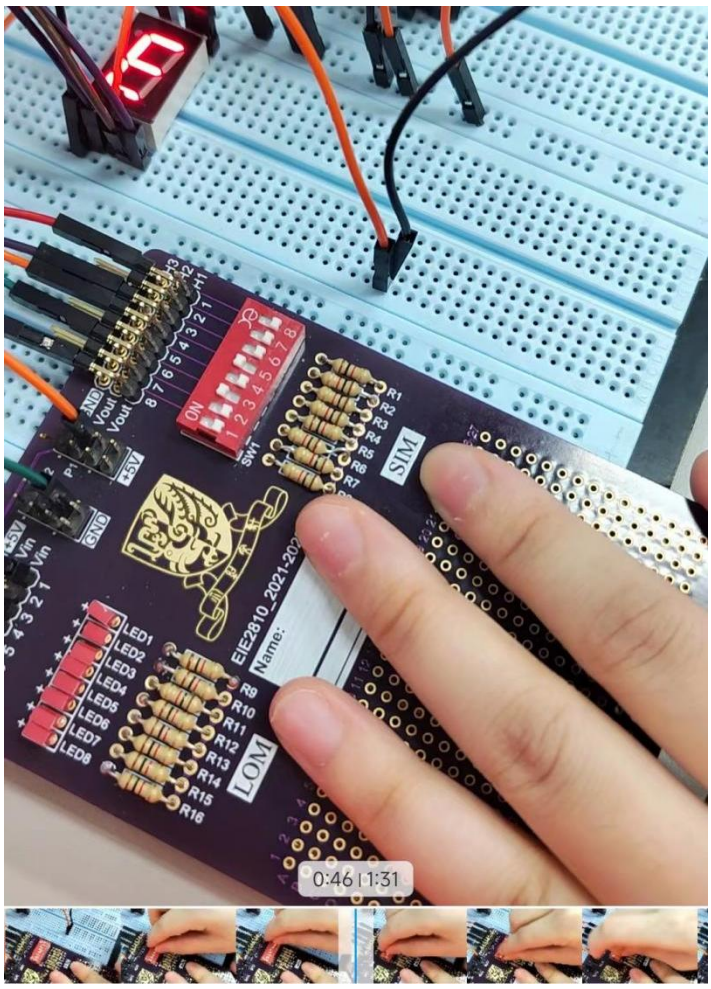


Figure 101. Player 4 presses the responder after player 3, no change

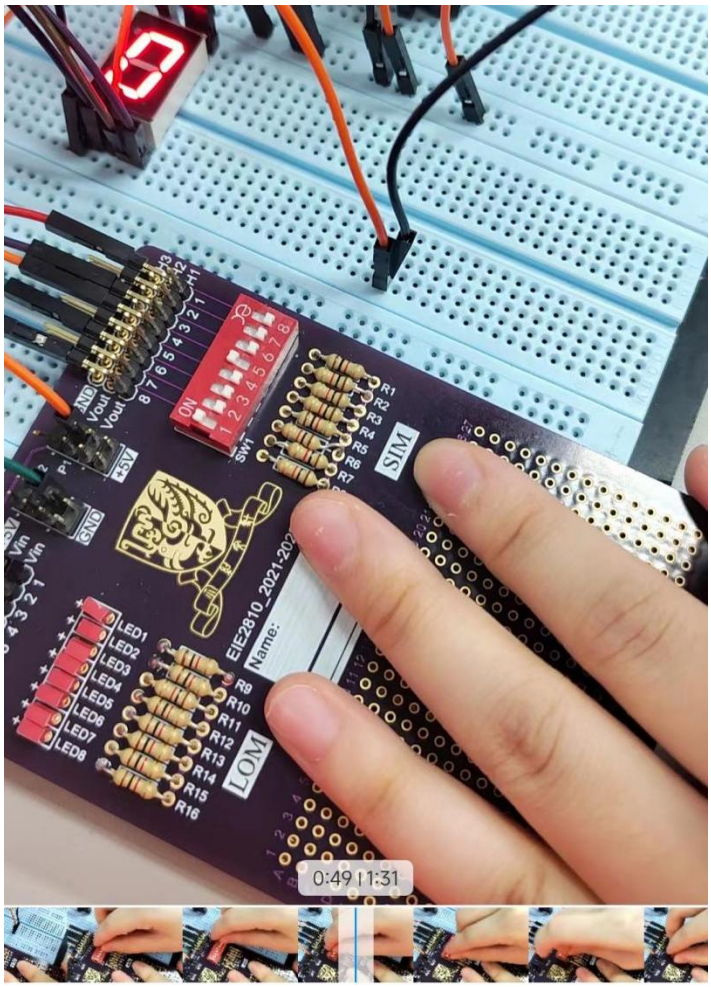


Figure 102. Judge clears the signal

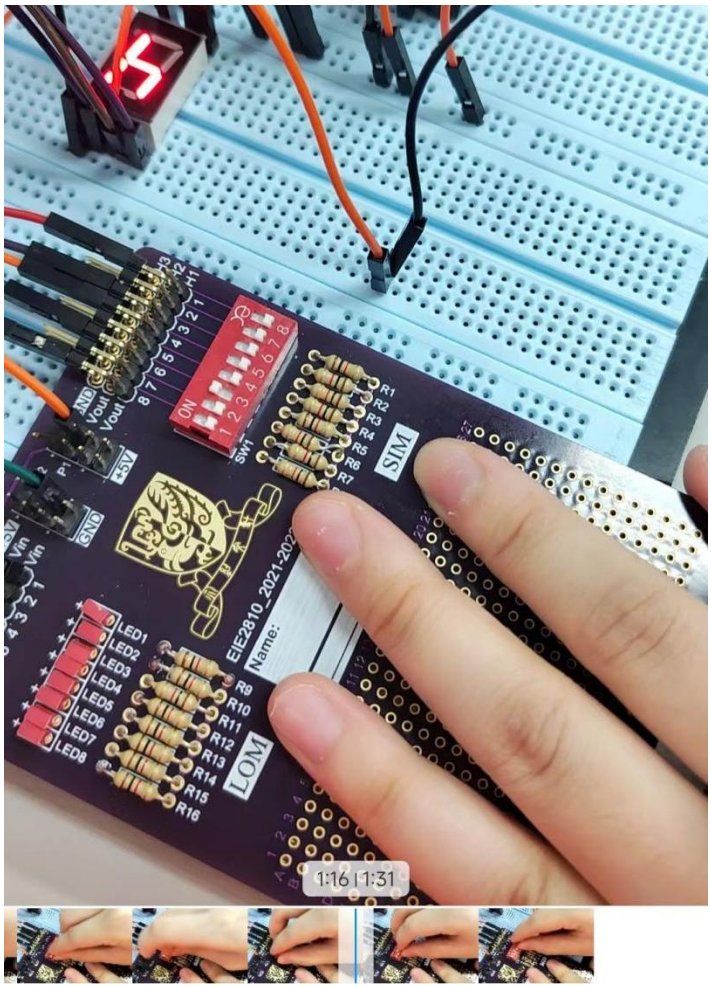


Figure 103. When player 3 presses the responder, “3” is displayed

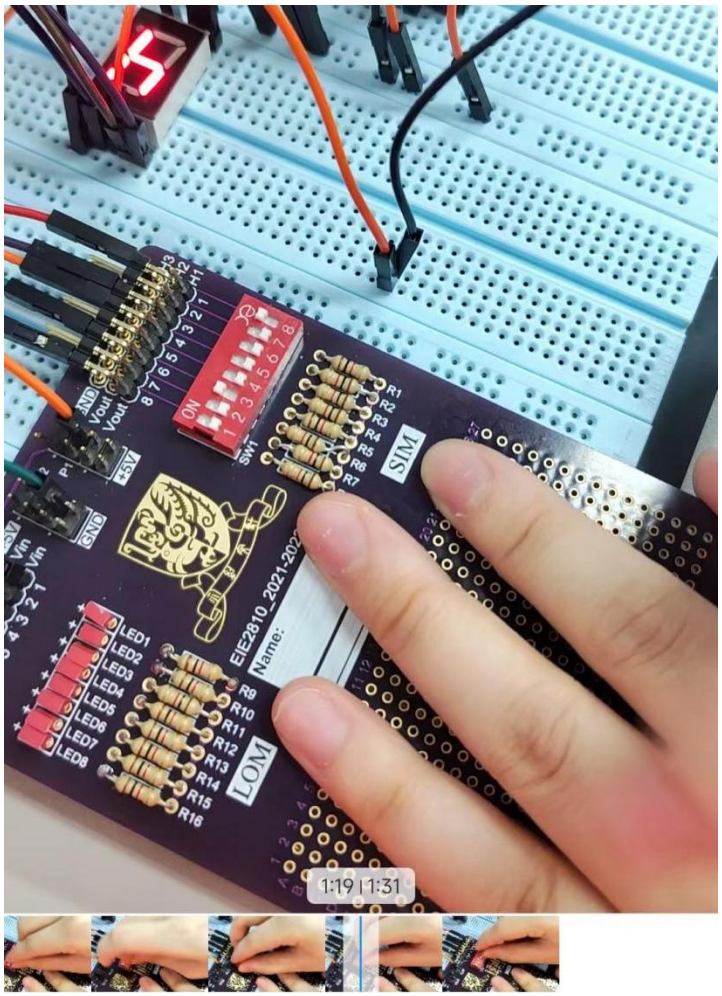


Figure 104. Player 1 presses the responder after player 4, no change

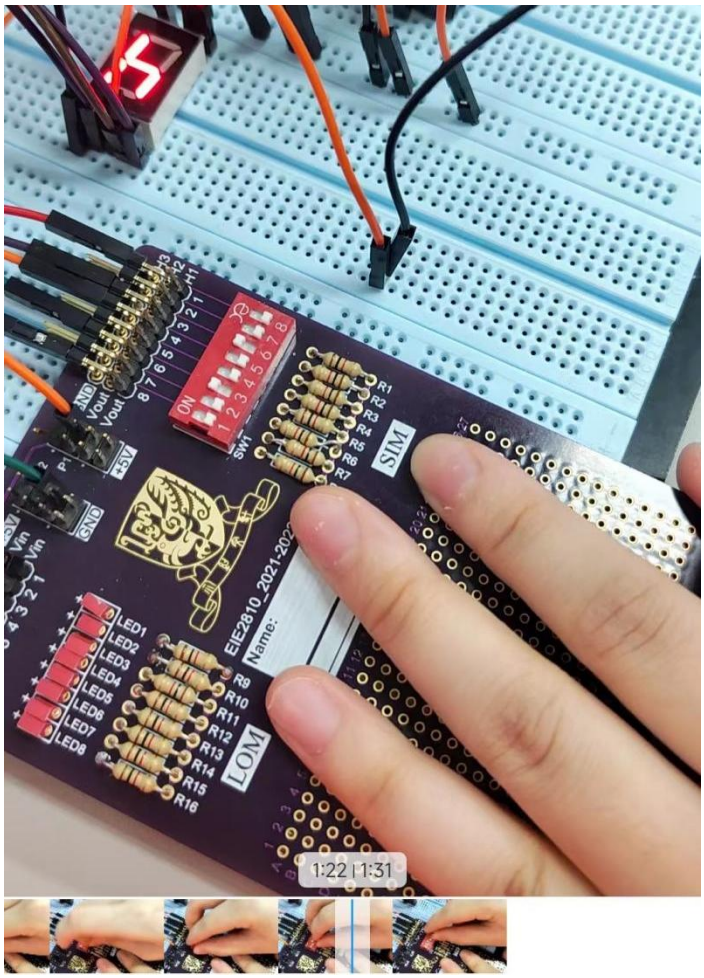


Figure 105. Player 2 presses the responder after player 4, no change

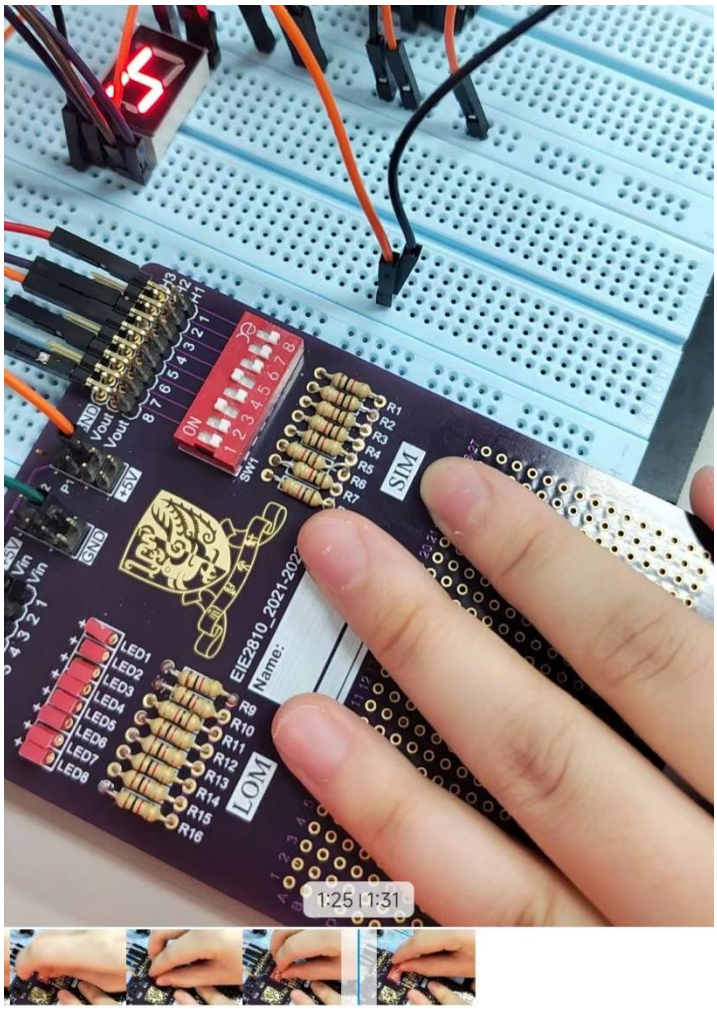


Figure 106. Player 3 presses the responder after player 4, no change

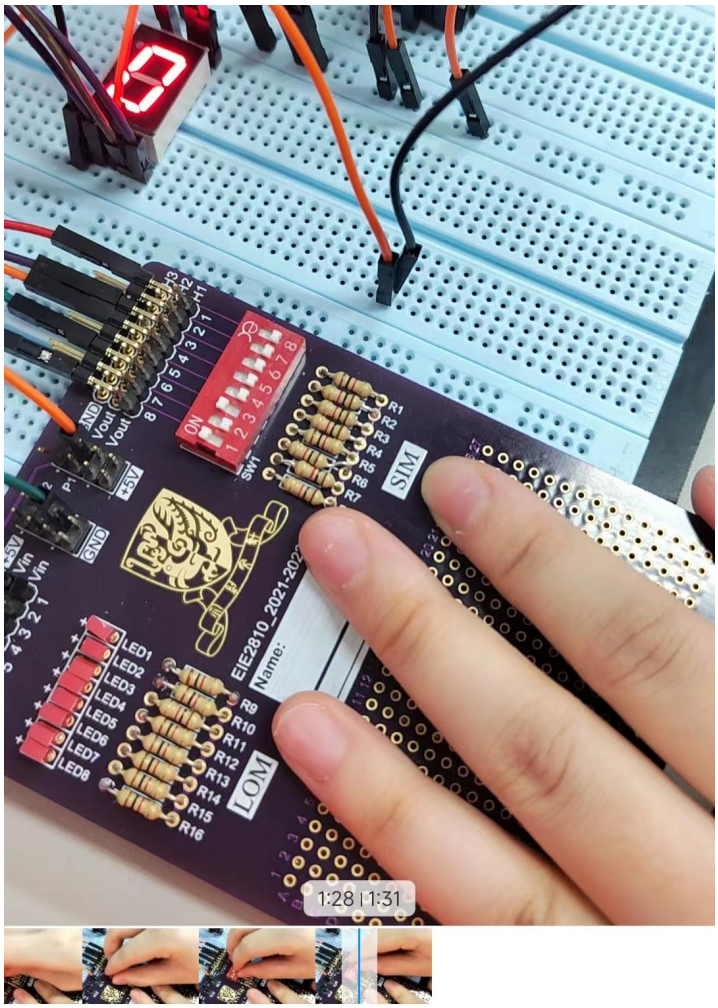


Figure 107. Judge clears the signal

3.3 Questions

Q.Currently, when the judge resets the circuit, the display shows “0”. But there is no player with ID “0”. It is better that no segment in the display is lightened. There is a way that you change your circuit by changing only 1 wire. Can you find out how? (Hint: observe the truth tables of 74HC148 and 74HC47.)

A.

E1	0	1	2	3	4	5	6	7	A2	A1	A0	GS	E0
L	H	H	H	H	H	H	H	H	H	H	H	H	L

Table 2. 74HC148

$\overline{LT}$	$\overline{RBI}$	A3	A2	A1	A0	$\overline{BI/RBO}$	$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	$\bar{e}$	$\bar{f}$	$\bar{g}$
X	X	X	X	X	X	L	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H

Table 3. 74HC74

Notice that for 74HC148, when the 8 inputs are all HIGH, the EO output is LOW; For 74HC47, when $\overline{BI/RBO}$ is LOW, the 7 outputs will all be HIGH. So connect $\overline{BI/RBO}$ pin of 74HC47 to the EO pin of 74HC148.

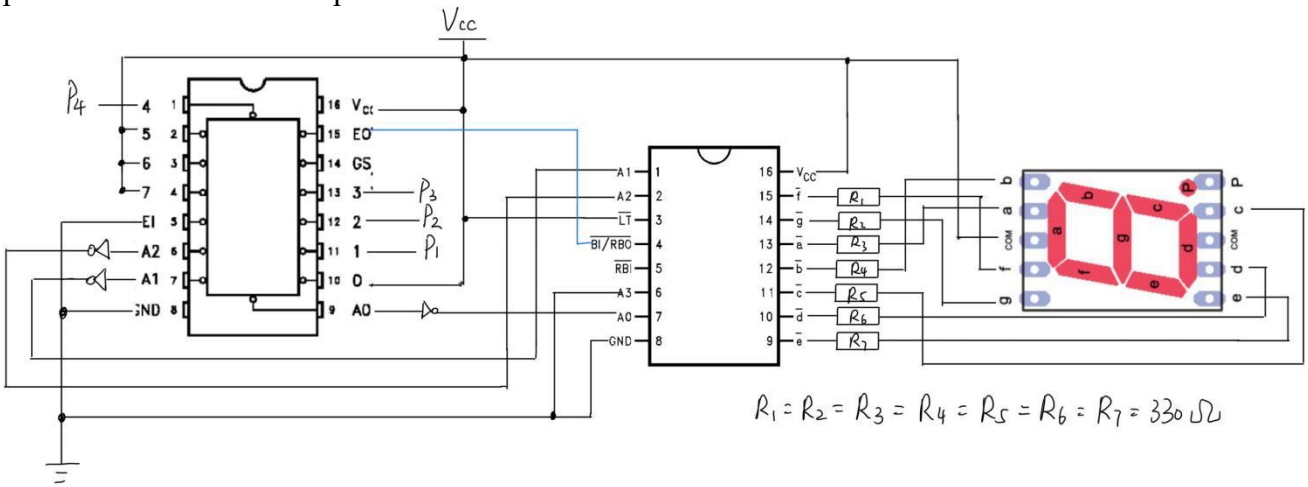


Figure 108. Designed circuit

4. Conclusion

In this lab, I

- Used D flip-flop to upgrade a function generator to a word generator.
- Built a looped counter by D flip-flop.
- Built a responder module based on D flip-flop, 8-3 line encoder and 7-segment module.

I learned:

- When $\overline{BI/RBO}$ is LOW, all segments of 74HC47 will be turned off, which can be used to display “blank”.
- How to design a counter with truncated sequence:

  1. The SET and RESET of D flip-flop can be used to change the looping state when designing a counter.
  2. If the counter need to be reset or truncated the counting sequence when it reaches a particular binary value (in this case, 1010), use the outputs corresponding to this value. For example, If the sequence required to change from 1010 to 0000, and no code greater than 1010 appears in the sequence,

instead of using all the four outputs, only the two “1” outputs are needed. Connect them to a NAND gate to form the logic to detect this condition (since NAND of two high (1) outputs yields a low (0)). And then connect to $\overline{SET}$.

3. Synchronous counter are generally more flexible and easier to design for specific sequences like truncated sequences than asynchronous counter.