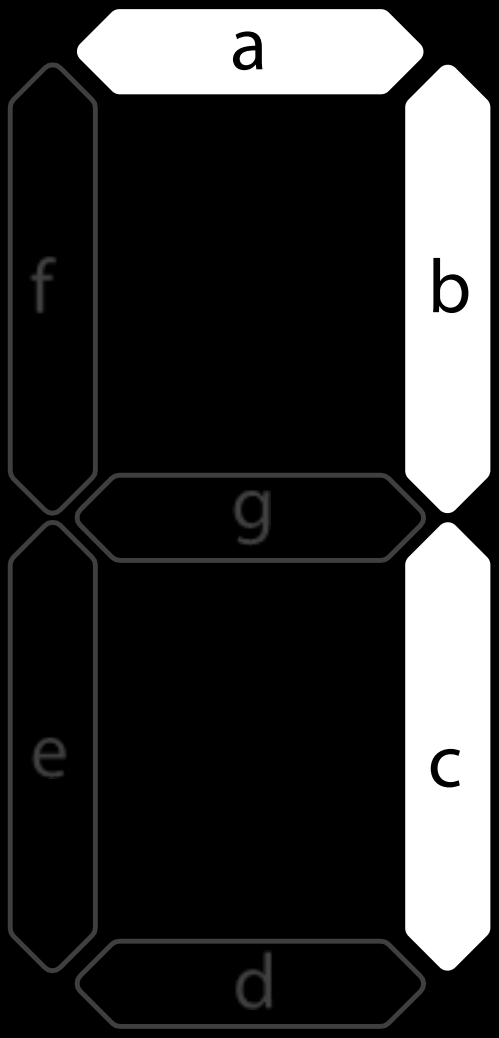


7 Segments DISPLAY

INVINCIBLE SQUAD(2A)

WALEED ASHRAF
bcsm-s21-011

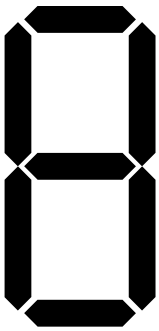
MUNEEB AHMAD
bcsm-s21-047

AHMED SHAHBAZ
bcsm-s21-022

Table Of Content

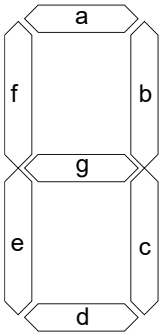
1. Introduction:

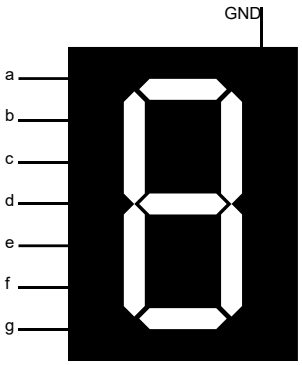
The 7-segment display, also written as “seven segment display”, consists of seven LEDs (hence its name) arranged in a rectangular fashion as shown below. Each of the seven LEDs is called a **segment** because when illuminated the segment forms part of a numerical digit to be displayed.



An additional 8th LED is sometimes used within the same package thus allowing the indication of a decimal point, (DP) when two or more 7-segment displays are connected together to display numbers greater than ten.

Each one of the seven LEDs in the display is given a positional segment with one of its connection pins being brought straight out of the rectangular plastic package. These individually LED pins are labelled from **a** through to **g** representing each individual LED. The other LED pins are connected together and wired to form a common pin.

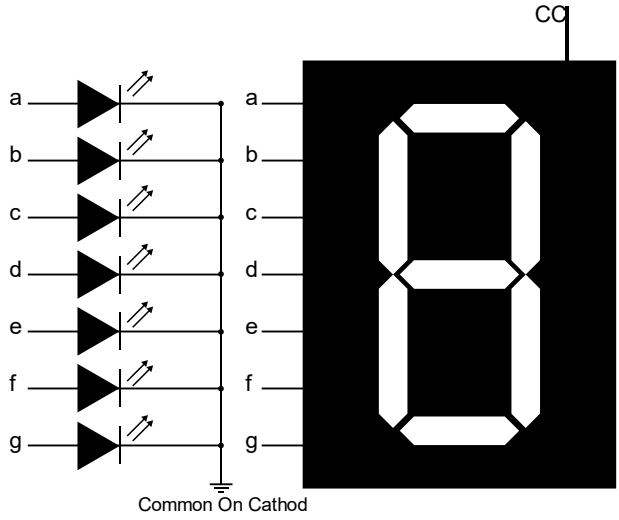




The displays common pin is generally used to identify which type of 7-segment display it is. As each LED has two connecting pins, one called the “Anode” and the other called the “Cathode”, there are therefore two types of LED 7-segment display called: **Common Cathode (CC)** and **Common Anode (CA).**

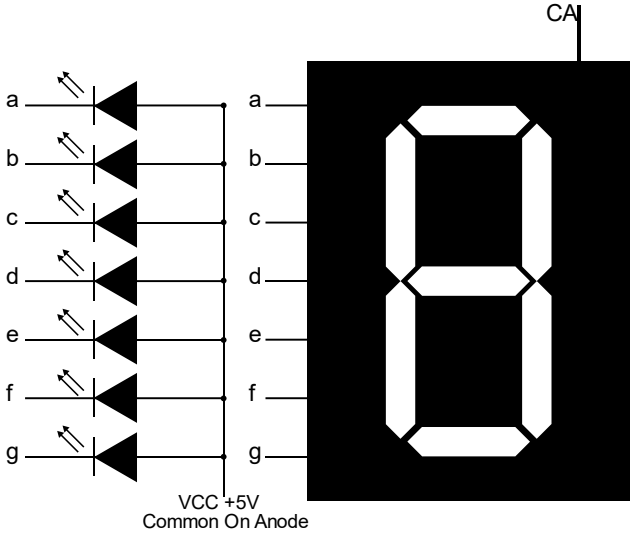
- ## Common Cathode Display (CC):

In the **common cathode display**, all the cathode connections of the LED segments are joined together to logic “0” or ground. The individual segments are illuminated by application of a “HIGH”, or logic “1” signal via a current limiting resistor to forward bias the individual Anode terminals (a-g).



- **Common Anode Display (CA):**

In the **common anode display**, all the anode connections of the LED segments are joined together to logic "1". The individual segments are illuminated by applying a ground, logic "0" or "LOW" signal via a suitable current limiting resistor to the Cathode of the particular segment (a-g).

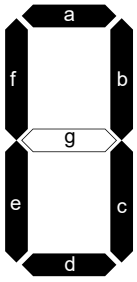


2. 7-Segment Display for all Numbers.

In order to display a particular number on seven segments display we need specific combination of segments that will be High (1) and other should be Low (0).

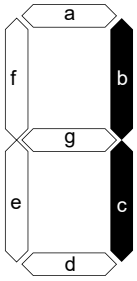
- **To Display 0:**

In order to display decimal value "0" on the display, all the segment will be High (1) except g.



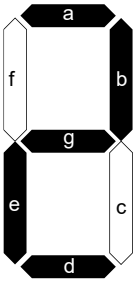
- **To Display 1:**

  In order to display decimal value “1” on the display we need to make segment **b** and **c** High (1) and all the other should be low.



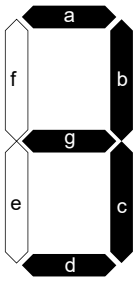
- **To Display 2:**

  In order to display a numeric value “2” on the display we need to make segment **a, b, g, e** and **d** High (1) and all the other should be low.



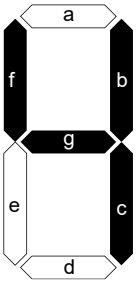
- **To Display 3:**

  In order to display decimal value “3” on the display we need to make segment **a, b, c, d** and **g** High (1) and all the other should be low.



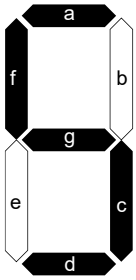
- **To Display 4:**

  In order to display decimal value “4” on the display we need to make segment **b, c, f** and **g** High (1) and all the other should be low.



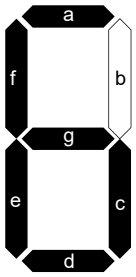
- **To Display 5:**

  In order to display decimal value “5” on the display we need to make segment **a, c, d, f** and **g** High (1) and all the other should be low.



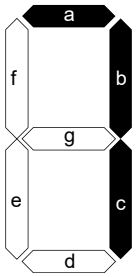
- **To Display 6:**

  In order to display decimal value “6” on the display we need to make segment **a, c, d, e, f** and **g** High (1) and all the other should be low.



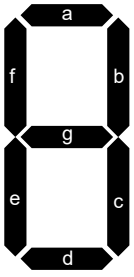
- **To Display 7:**

  In order to display decimal value “7” on the display we need to make segment **a, b,** and **c** High (1) and all the other should be low.



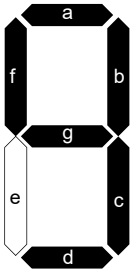
- **To Display 8:**

  In order to display decimal value “8” on the display we need to make all segment High (1).



- **To Display 9:**

  In order to display decimal value “9” on the display we need to make all segment High (1) except segment **e**.



3. Truth Table.

No	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
A	1	0	1	0	X	X	X	X	X	X	X
B	1	0	1	1	X	X	X	X	X	X	X
C	1	1	0	0	X	X	X	X	X	X	X
D	1	1	0	1	X	X	X	X	X	X	X
E	1	1	1	0	X	X	X	X	X	X	X
F	1	1	1	1	X	X	X	X	X	X	X

In this we only need to display decimal values from 0 to 9 using there BCD codes. As in hexadecimal number system after decimal value 9 there is no decimal value 10. After that alphabetic values starts A to F.

In this project we are only concerned with the decimal values from 0-9, so from A-F are the **don’t care conditions.**

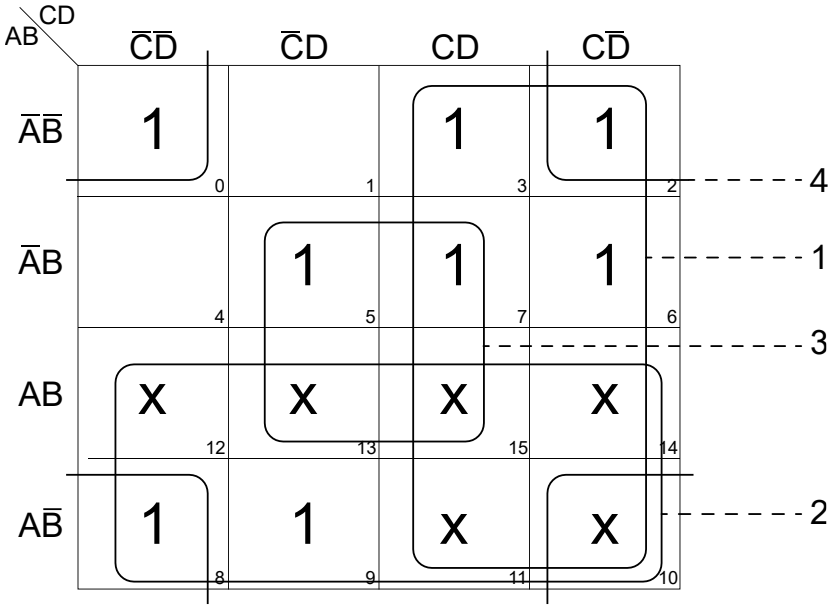
4. Logic Expressions For Each Segment.

- **For Segment a:**

To derive logic expression for segment **"a"** we need construct K-Map using truth table we constructed above and then we can easily extract logic expression.

Using the truth table, we will map our High (1) values for segment **"a"** and after that we will make groups on K-Map and then extract a SOP expression to construct a logic circuit for the specific segment.

- **K-Map**:

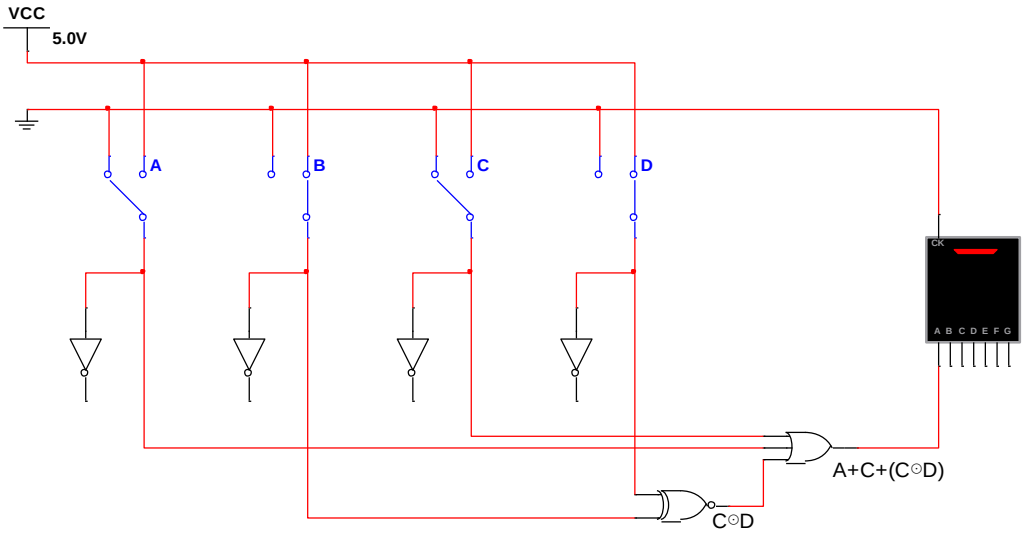


- **Logic Expression**:

$$a = C + A + \bar{C}\bar{D} + CD$$

$$a = A + C + (C \odot D)$$

- **Circuit Diagram**:

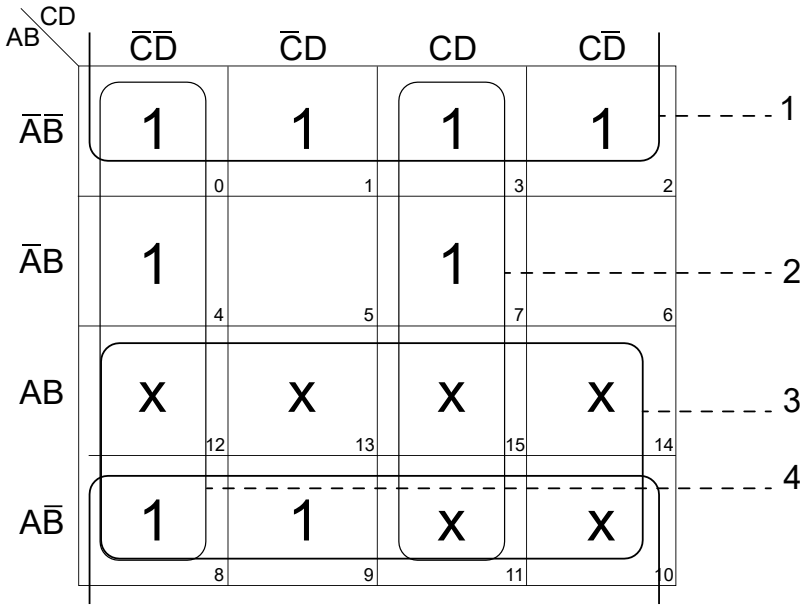


- **For Segment b:**

To derive logic expression for segment **"b"** we need construct K-Map using truth table we constructed above and then we can easily extract logic expression.

Using the truth table, we will map our High (1) values for segment **"b"** and after that we will make groups on K-Map and then extract a SOP expression to construct a logic circuit for the specific segment.

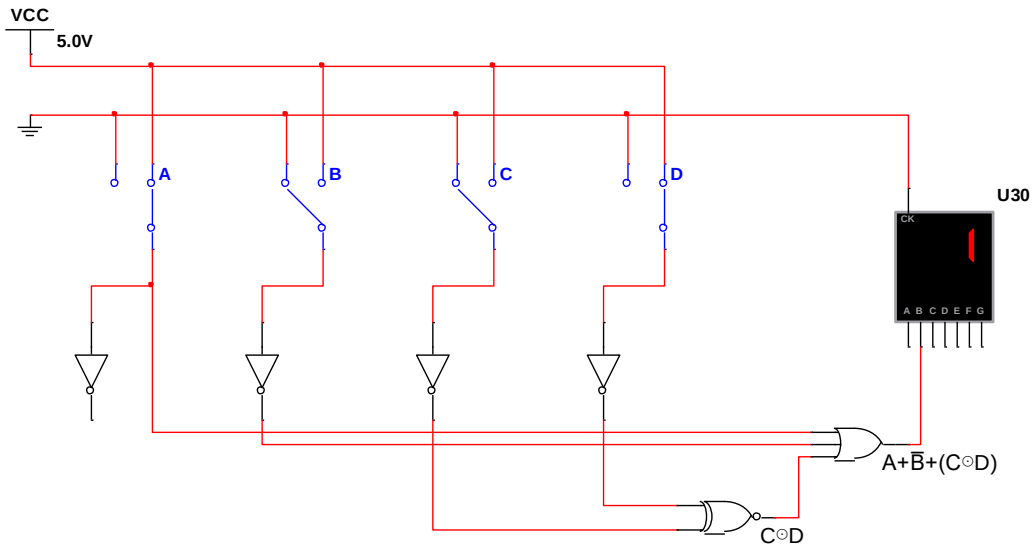
- **K-Map**:



- **Logic Expression**:

$$b = A + \bar{B} + \bar{C}\bar{D} + CD$$
$$b = A + \bar{B} + (C \odot D)$$

- **Circuit Diagram**:

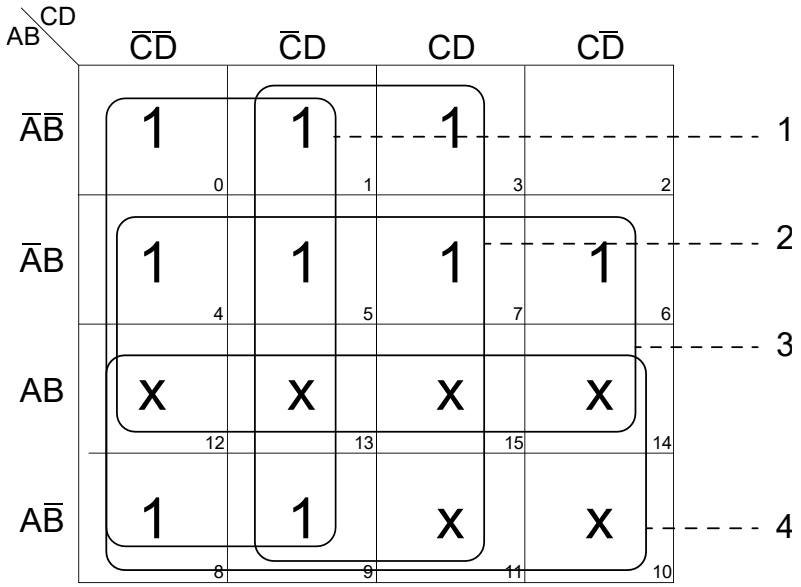


- **For Segment c:**

To derive logic expression for segment **"c"** we need construct K-Map using truth table we constructed above and then we can easily extract logic expression.

Using the truth table, we will map our High (1) values for segment **"c"** and after that we will make groups on K-Map and then extract a SOP expression to construct a logic circuit for the specific segment.

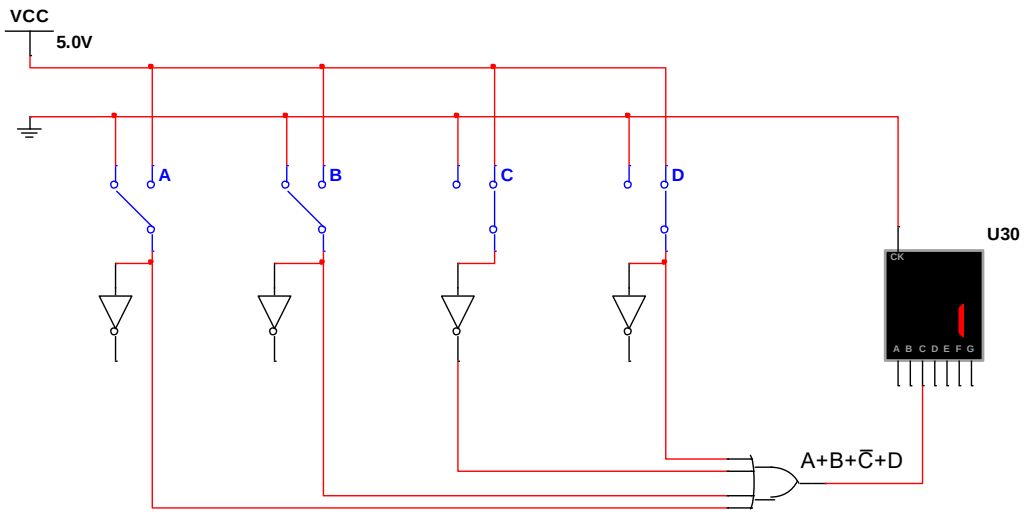
- **K-Map**:



- **Logic Expression**:

$$c = A + B + \bar{C} + D$$

- **Circuit Diagram**:

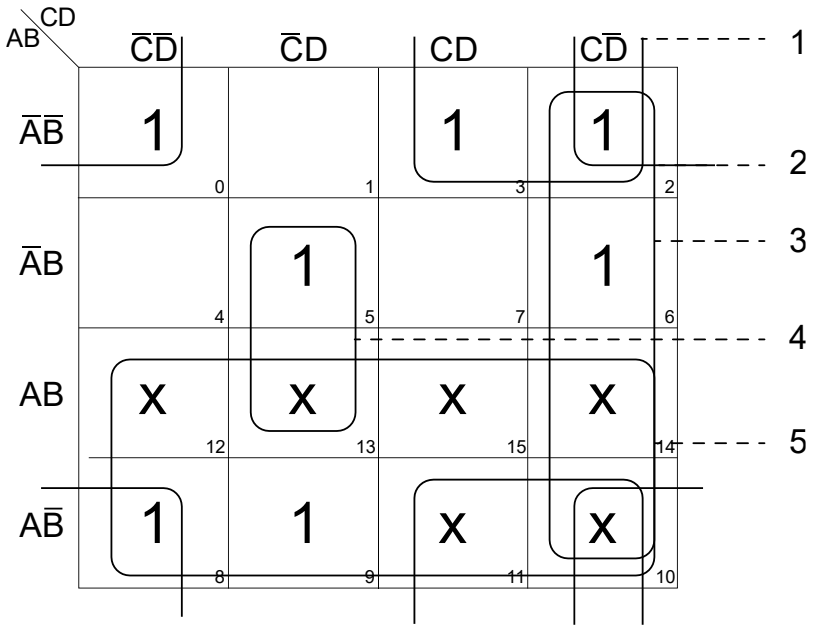


- **For Segment d:**

To derive logic expression for segment **"d"** we need construct K-Map using truth table we constructed above and then we can easily extract logic expression.

Using the truth table, we will map our High (1) values for segment **"d"** and after that we will make groups on K-Map and then extract a SOP expression to construct a logic circuit for the specific segment.

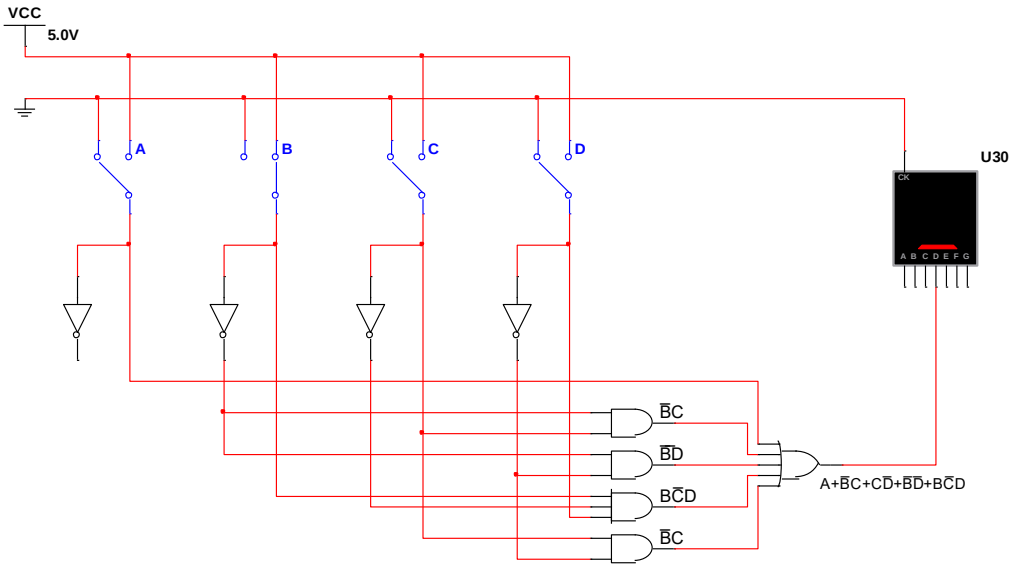
- **K-Map**:



- **Logic Expression**:

$$d = A + \bar{B}C + C\bar{D} + \bar{B}\bar{D} + B\bar{C}D$$

- **Circuit Diagram**:

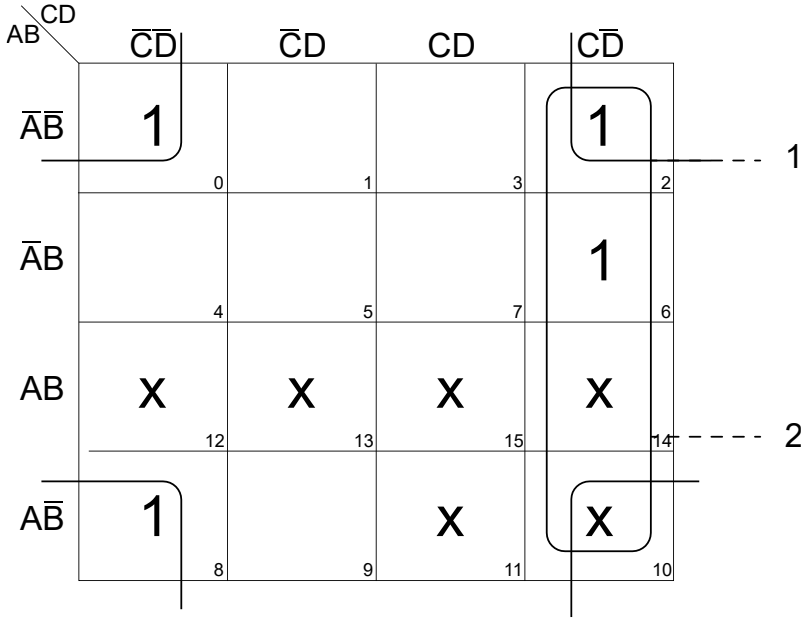


- **For Segment e:**

To derive logic expression for segment **"e"** we need construct K-Map using truth table we constructed before for all the segments and then we can easily extract logic expression.

Using the truth table, we will map our High (1) values for segment **"e"** and after that we will make groups on K-Map and then extract a SOP expression to construct a logic circuit for the specific segment.

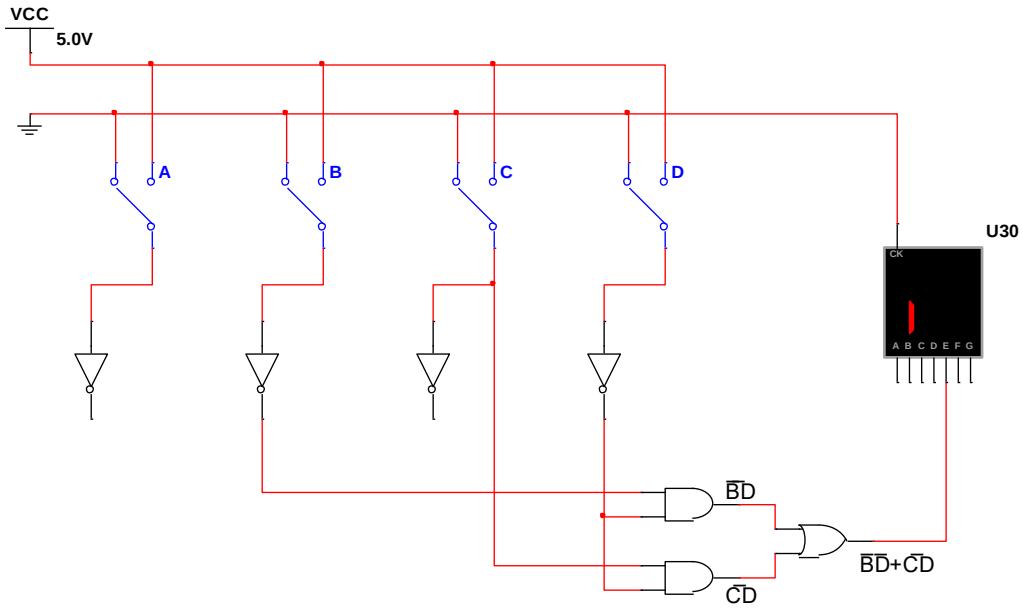
- **K-Map**:



- **Logic Expression**:

$$e = \bar{B}\bar{D} + C\bar{D}$$

- **Circuit Diagram**:

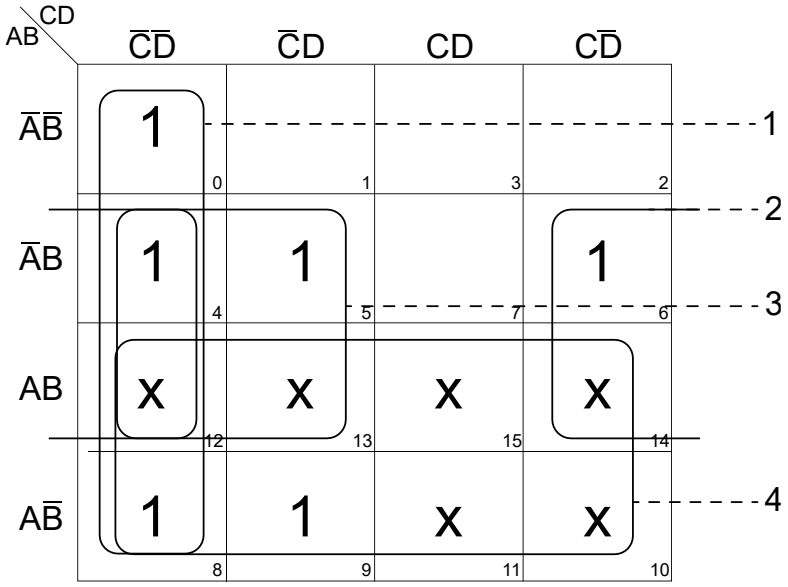


- **For Segment f:**

To derive logic expression for segment **"f"** we need construct K-Map using truth table we constructed before for all the segments and then we can easily extract logic expression.

Using the truth table, we will map our High (1) values for segment **"f"** and after that we will make groups on K-Map and then extract a SOP expression to construct a logic circuit for the specific segment.

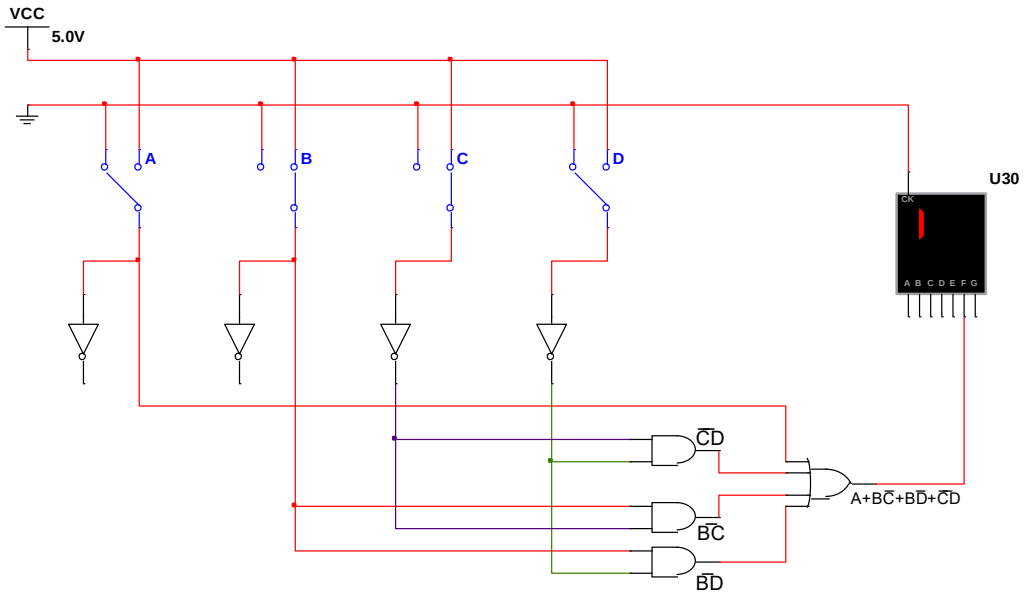
- **K-Map**:



- **Logic Expression**:

$$f = A + B\bar{C} + B\bar{D} + \bar{C}\bar{D}$$

- **Circuit Diagram**:

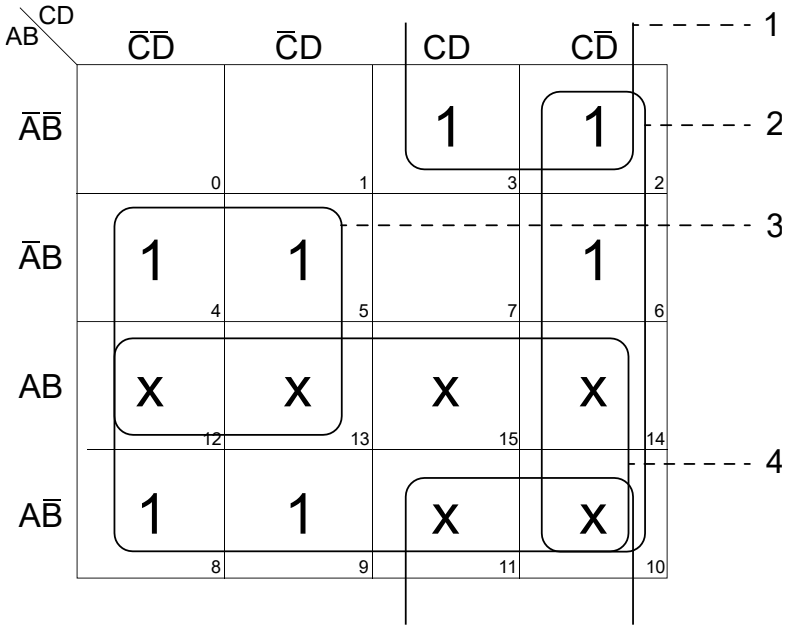


- **For Segment g:**

To derive logic expression for segment **"g"** we need construct K-Map using truth table we constructed before for all the segments and then we can easily extract logic expression.

Using the truth table, we will map our High (1) values for segment **"g"** and after that we will make groups on K-Map and then extract a SOP expression to construct a logic circuit for the specific segment.

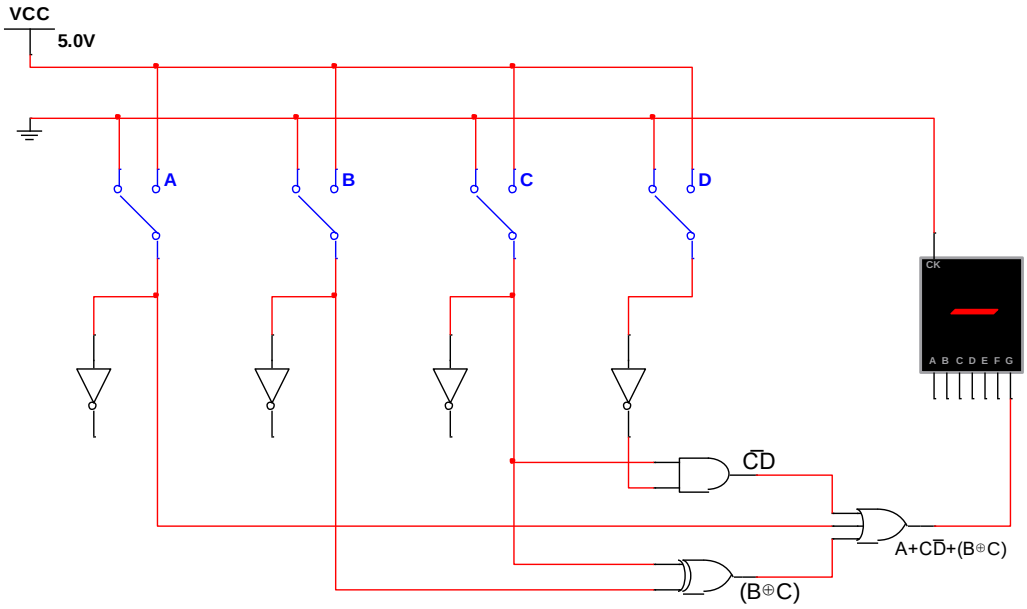
- **K-Map**:



- **Logic Expression**:

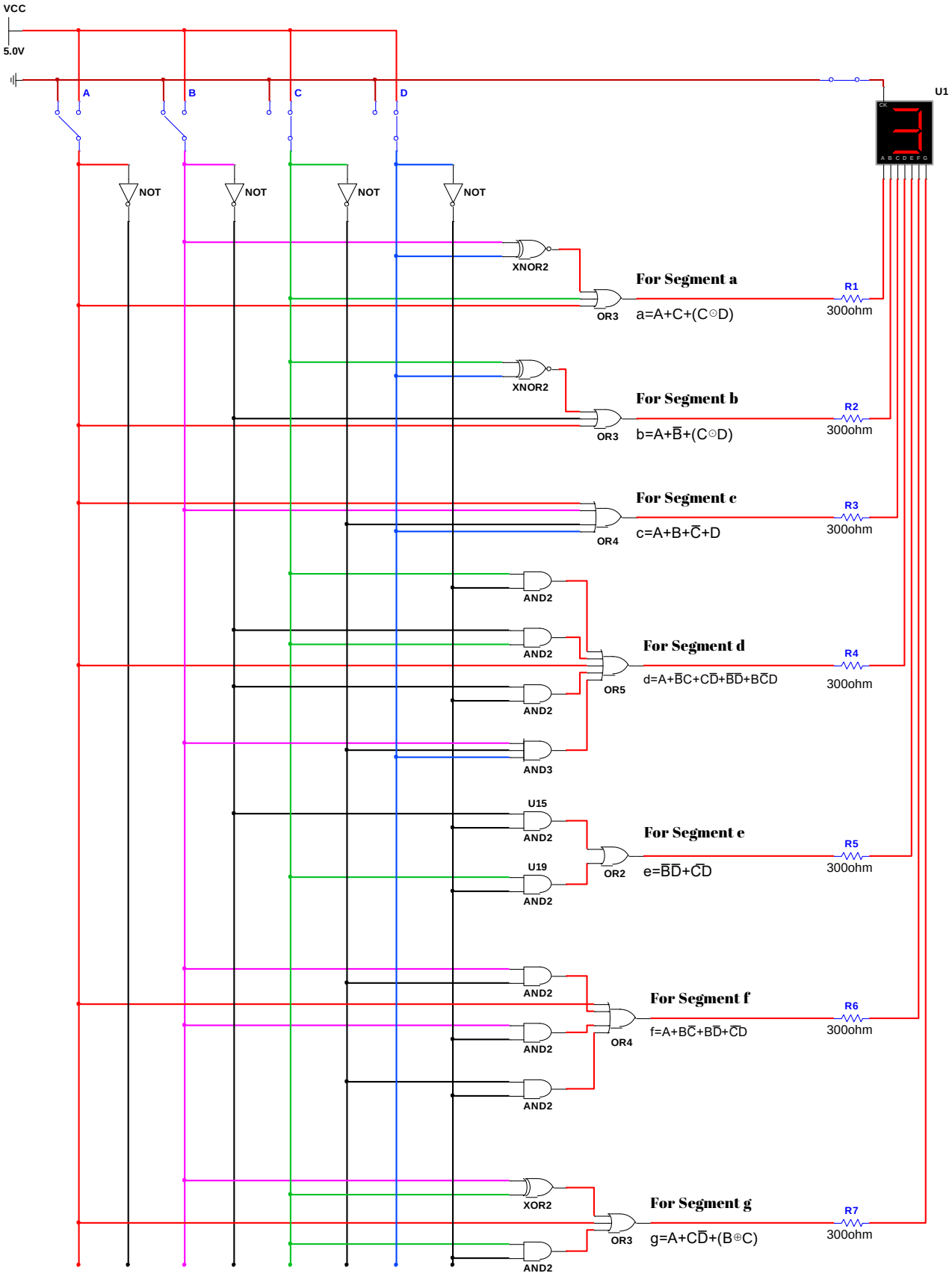
$$g = A + C\bar{D} + B\bar{C} + \bar{B}C$$
$$g = A + C\bar{D} + (B \oplus C)$$

- **Circuit Diagram**:



5. Combining All Segment Circuits.

After constructing circuits for all the seven segments we need to combine them into one circuit and each segment output should be connected to the corresponding pin of seven segment display. By changing the BCD values in the input, seven segments will display specific decimal number according their BCD code.



6. Applications.

- It used as a display for a calculator.
- We also can use it for digital watches.
- It is also utilized in motor cycle odometer.
- Using this we also can indicate frequency.
- These devices can also be used as speedometer.