BCD to 7-Segment Display Decoder

As we saw in the previous tutorial, a Digital Decoder IC, is a device which converts one digital format into another and one of the most commonly used device for doing this is called the Binary Coded Decimal (BCD) to 7-Segment Display Decoder. 7-segment LED (Light Emitting Diode) or LCD (Liquid Crystal Display) type displays, provide a very convenient way of displaying information or digital data in the form of numbers, letters or even alpha-numerical characters.

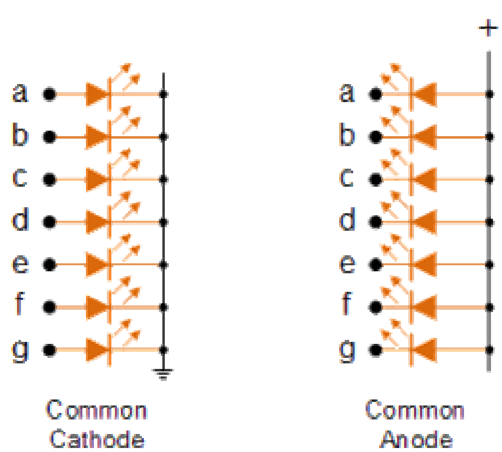
Typically **7-segment displays** consist of seven individual coloured LED's (called the segments), within one single display package. In order to produce the required numbers or HEX characters from 0 to 9 and A to F respectively, on the display the correct combination of LED segments need to be illuminated and **BCD to 7-segment Display Decoders** such as the 74LS47 do just that.

A standard 7-segment LED display generally has 8 input connections, one for each LED segment and one that acts as a common terminal or connection for all the internal display segments. Some single displays have also have an additional input pin to display a decimal point in their lower right or left hand corner.

In electronics there are two important types of 7-segment LED digital display.

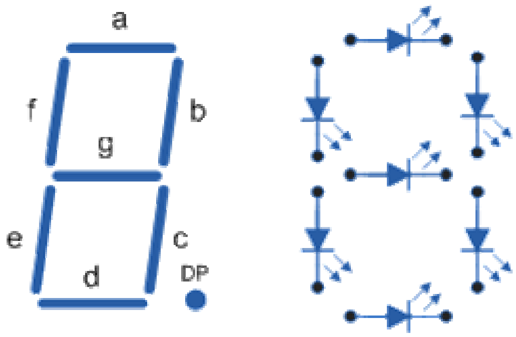
- 1. The Common Cathode Display (CCD) – In the common cathode display, all the cathode connections of the LED's are joined together to logic "0" or ground. The individual segments are illuminated by application of a "HIGH", logic "1" signal to the individual Anode terminals.
- 2. The Common Anode Display (CAD) – In the common anode display, all the anode connections of the LED's are joined together to logic "1" and the individual segments are illuminated by connecting the individual Cathode terminals to a "LOW", logic "0" signal.

Common Cathode and Common Anode Format



Electrical connection of the individual diodes for a common cathode display and a common anode display and by illuminating each light emitting diode individually, they can be made to display a variety of numbers or characters.

7-Segment Display Format

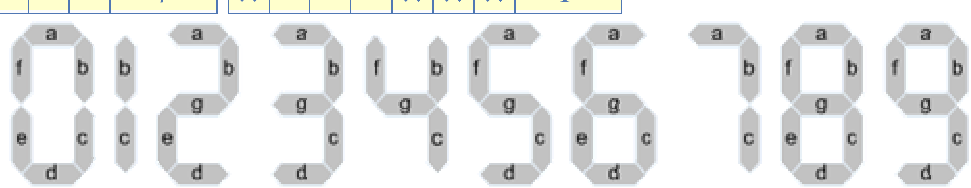


So in order to display the number 3 for example, segments a, b, c, d and g would need to be illuminated. If we wanted to display a different number or letter then a different set of segments would need to be illuminated. Then for a 7-segment display, we can produce a truth table giving the segments that need to be illuminated in order to produce the required character as shown below.

Truth Table for a 7-segment display

Individual Segments							Display
a	b	c	d	e	f	g	
×	×	×	×	×	×		0
	×	×					1
×	×		×	×		×	2
×	×	×	×			×	3
	×	×			×	×	4
×		×	×		×	×	5
×		×	×	×	×	×	6
×	×	×					7

Individual Segments							Display
a	b	c	d	e	f	g	
×	×	×	×	×	×	×	8
×	×	×	×		×	×	9
×	×	×		×	×	×	A
		×	×	×	×	×	b
×			×	×	×		C
	×	×	×	×		×	d
×			×	×	×	×	E
×				×	×	×	F



7-Segment Display Elements for all Numbers.

It can be seen that to display any single digit number from 0 to 9 in binary or letters from A to F in hexadecimal, we would require 7 separate segment connections plus one additional connection for the LED's "common" connection. Also as the segments are basically a standard light emitting diode, the driving circuit would need to produce up to 20mA of current to illuminate each individual segment and to display the number 8, all 7 segments would need to be lit resulting a total current of nearly 140mA, (8 x 20mA).

Obviously, the use of so many connections and power consumption is impractical for some electronic or microprocessor based circuits and so in order to reduce the number of signal lines required to drive just one single display, display decoders such as the BCD to 7-Segment Display Decoder and Driver IC's are used instead.

Binary Coded Decimal

Binary Coded Decimal (BCD or "8421" BCD) numbers are made up using just 4 data bits (a nibble or half a byte) similar to the **Hexadecimal** numbers we saw in the binary tutorial, but unlike hexadecimal numbers that range in full from 0 through to F, BCD numbers only range from 0 to 9, with the binary number patterns of 1010 through to 1111 (A to F) being invalid inputs for this type of display and so are not used as shown below.

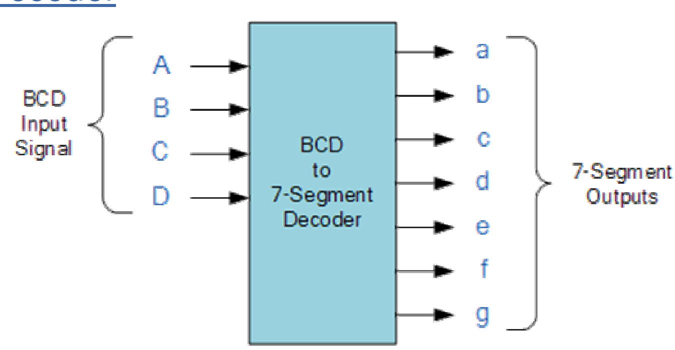
Decimal	Binary Pattern				BCD
	8	4	2	1	
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7

Decimal	Binary Pattern				BCD
	8	4	2	1	
8	1	0	0	0	8
9	1	0	0	1	9
10	1	0	1	0	Invalid
11	1	0	1	1	Invalid
12	1	1	0	0	Invalid
13	1	1	0	1	Invalid
14	1	1	1	0	Invalid
15	1	1	1	1	Invalid

BCD to 7-Segment Display Decoders

A binary coded decimal (BCD) to 7-segment display decoder such as the TTL 74LS47 or 74LS48, have 4 BCD inputs and 7 output lines, one for each LED segment. This allows a smaller 4-bit binary number (half a byte) to be used to display all the denary numbers from 0 to 9 and by adding two displays together, a full range of numbers from 00 to 99 can be displayed with just a single byte of 8 data bits.

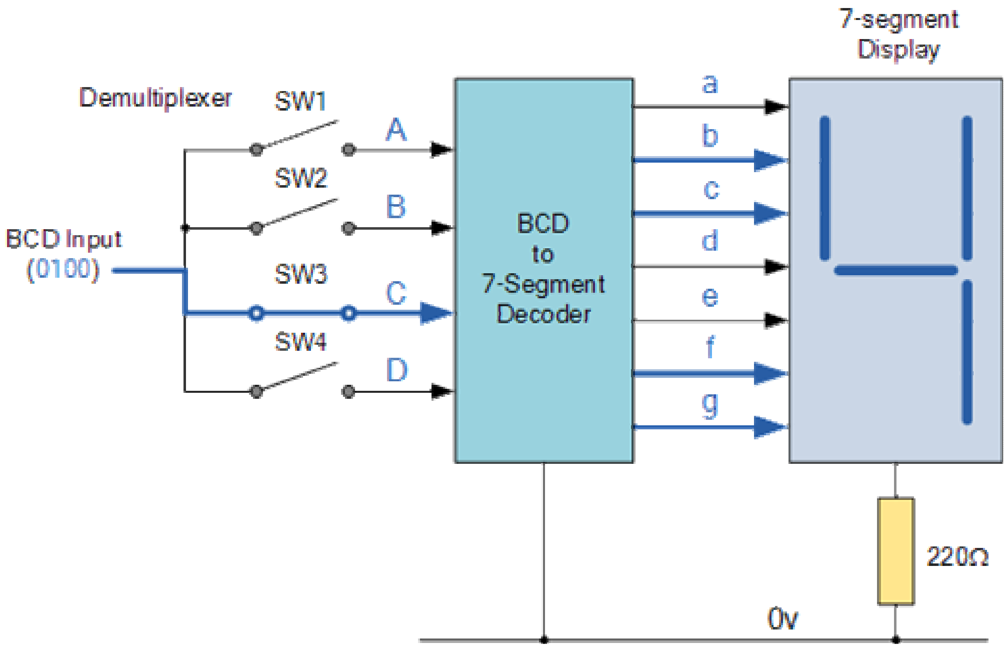
BCD to 7-Segment Decoder



The use of **packed** BCD allows two BCD digits to be stored within a single byte (8-bits) of data, allowing a single data byte to hold a BCD number in the range of 00 to 99.

An example of the 4-bit BCD input (0100) representing the number 4 is given below.

Display Decoder Example No1



In practice current limiting resistors of about 150Ω to 220Ω would be connected in series between the decoder/driver chip and each LED display segment to limit the maximum current flow. Different display decoders or drivers are available for the different types of display available, e.g. 74LS48 for common-cathode LED types, 74LS47 for common-anode LED types, or the CMOS CD4543 for liquid crystal display (LCD) types.

Liquid crystal displays (LCD´s) have one major advantage over similar LED types in that they consume much less power and nowadays, both LCD and LED displays are combined together to form larger Dot-Matrix Alphanumeric type displays which can show letters and characters as well as numbers in standard Red or Tri-colour outputs.

BCD Counter

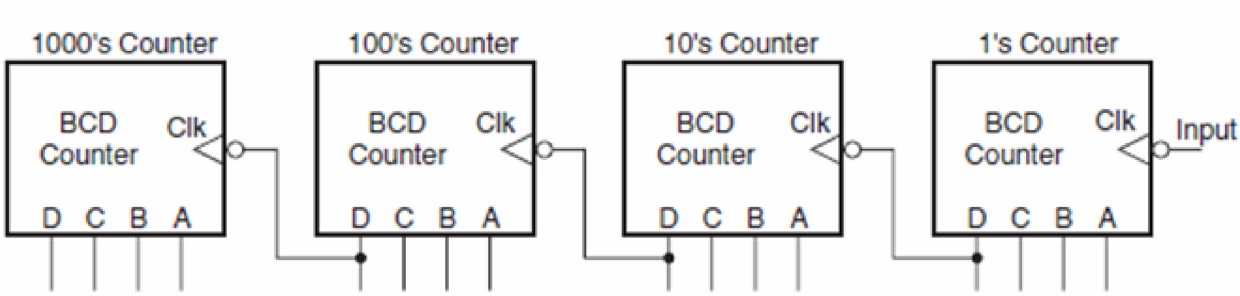
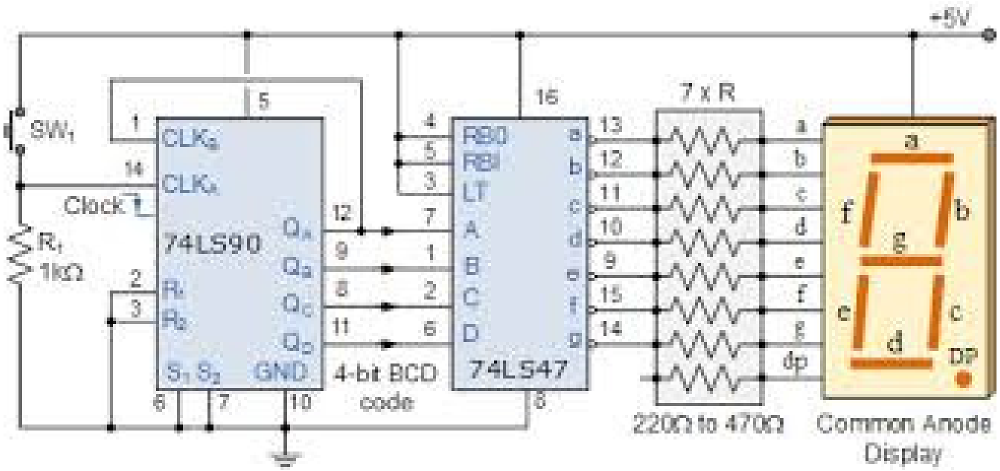
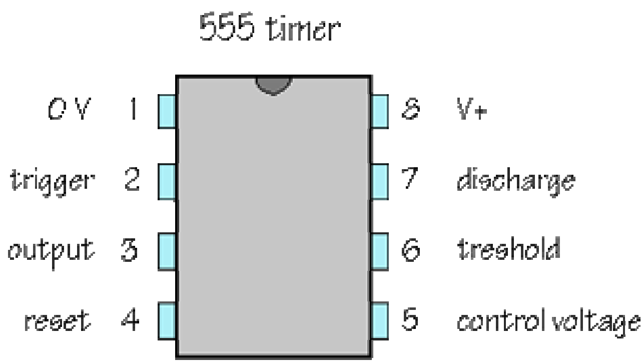
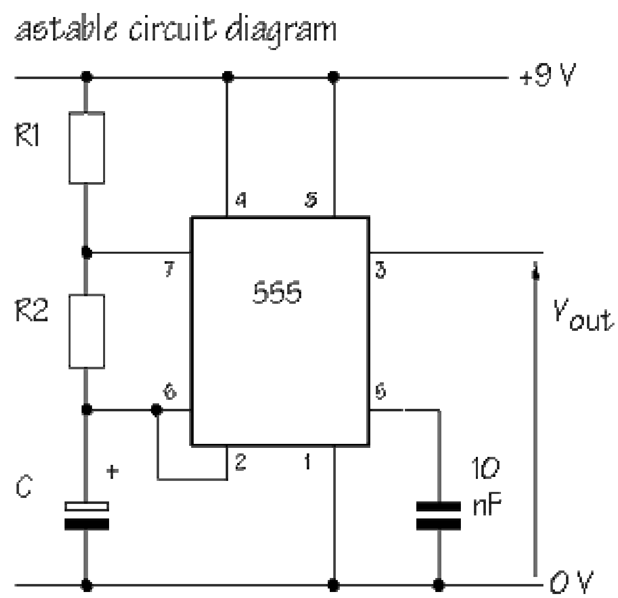


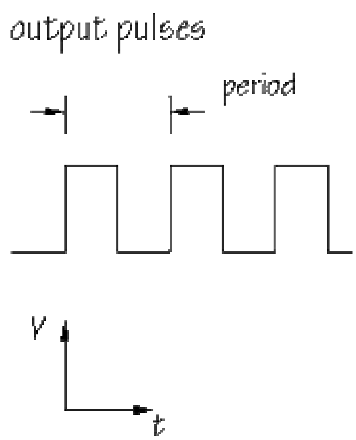
Figure 11.21 Cascading BCD counters.



555 Timer Pulse Cct.







The frequency of the output pulses is given by:

$$f = \frac{1.44}{(R1 + 2R2) \times C}$$

The **design formula** is just what you need if you know the values of *R*1, *R*2 and *C* and want to calculate the frequency of the output pulses which these values will give.

In this case, you know that the frequency you want is 0.33 Hz, one pulse every 3 seconds, but you *don't* know what values of *R*1, *R*2 and *C* you should use.

You can approach this problem by rearranging the design formula:

$$(R1 + 2R2) \times C = \frac{1.44}{f}$$