ASYNCHRONOUS COUNTER WITH SEVEN SEGMENT DISPLAY

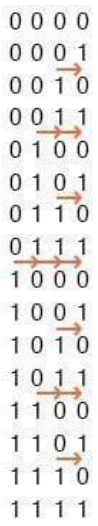
Binary count sequences follow a pattern of octave (factor of 2) frequency division, and that J-K flipflop multivibrators set up for the “toggle” mode are capable of performing this type of frequency division, we can envision a circuit made up of several J-K flipflops, cascaded to produce four bits of output.

The main problem facing us is to determine how to connect these flip-flops together so that they toggle at the right times to produce the proper binary sequence.

Examine the following binary count sequence, paying attention to patterns preceding the “toggling” of a bit between 0 and 1:

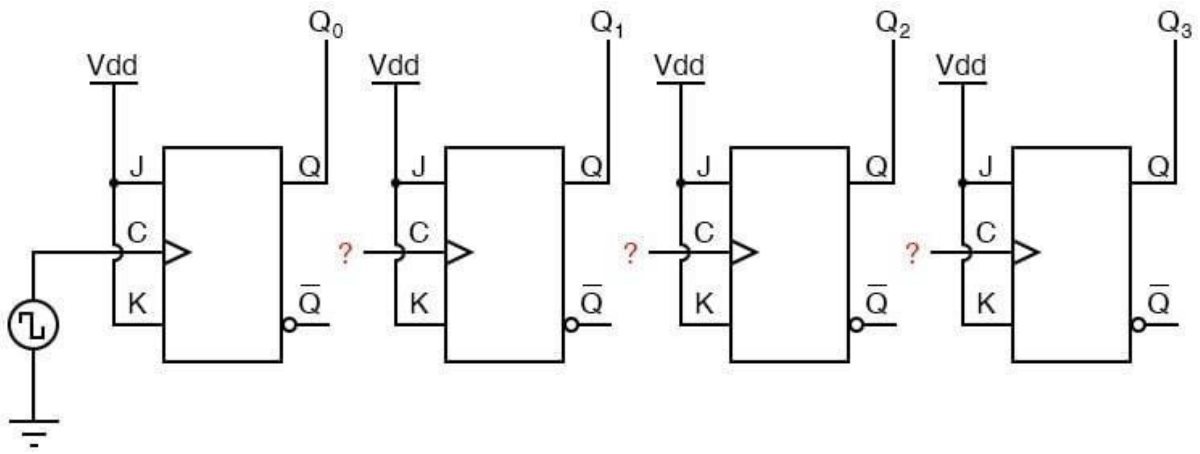
```
0 0 0 0
0 0 0 1
0 0 1 0
0 0 1 1
0 1 0 0
0 1 0 1
0 1 1 0
0 1 1 1
1 0 0 0
1 0 0 1
1 0 1 0
1 0 1 1
1 1 0 0
1 1 0 1
1 1 1 0
1 1 1 1
```

Note that each bit in this four-bit sequence toggles when the bit before it (the bit having a lesser significance, or place-weight), toggles in a particular direction: from 1 to 0. Small arrows indicate those points in the sequence where a bit toggles, the head of the arrow pointing to the previous bit transitioning from a “high” (1) state to a “low” (0) state:



Starting with four J-K flip-flops connected in such a way to always be in the "toggle" mode, we need to determine how to connect the clock inputs in such a way so that each succeeding bit toggles when the bit before it transitions from 1 to 0.

The Q outputs of each flip-flop will serve as the respective binary bits of the final, fourbit count:



If we used flip-flops with negative-edge triggering (bubble symbols on the clock inputs), we could simply connect the clock input of each flip-flop to the Q output of the flip-flop before it, so that when the bit before it changes from a 1 to a 0, the "falling edge" of that signal would "clock" the next flipflop to toggle the next bit:

Asynchronous counter interface with seven segment display using IC 7447

Nowadays it is very easy to display numbers and letters across multiple LED displays using microcontrollers, such as the Arduino or Raspberry Pi, along with a small bit of software related code to display the required digits. But sometimes as an Electronics student or hobbyist we want to display two or more numbers or digits as part of our project or digital logic circuit. So how can we do this.

7-segment displays provide a convenient way of displaying numerical information from zero to nine as they basically consist of a load of light emitting diodes connected together within a single indicator package.

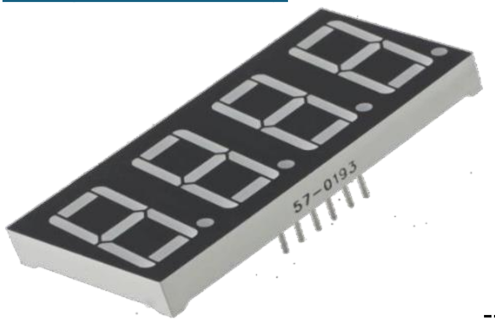
Each light emitting diode (called a segment) is illuminated using an electrical current, and by illuminating various combinations of segments so that some segments will be turned 'ON' and emitting light while others will be turned 'OFF' we can display individual characters or numbers.

As we saw in our tutorial about the Light Emitting Diode, LEDs are just like normal diodes, in that they only allow current to flow in one direction. This difference between the two is that an LED emits light energy from its PN-junction when an electrical current passes through it.

This electroluminescence action occurs whenever the Anode (A) terminal of the LED is more positive than its Cathode (K) terminal by approximately 2 volts. The typical supply current required to illuminate an LED junction ranges from between about 6mA to 20mA and whose value is commonly controlled using a resistor in series with the LED.

So by forward-biasing any one of the displays LED segments so that the anode terminal is towards the supply (positive) and the cathode terminal is towards ground (negative), we can produce a set of randomly lit segments or a decimal number from 0 to 9 providing a visual output for our project.

7-segment display



A name suggests, a 7-segment display consists of seven segments, meaning it consists of the demitting diodes or LED's, which together can be used to form one complete digit on display.

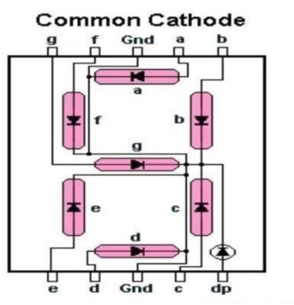
Actually, most 7-segment displays contain eight internal LED's as the eighth one is used for a decimal point, usually in one of the bottom corners of the display.

So if a 7-segment display consists of seven LED's (ignoring the decimal point for now), one for each segment, and an LED has two terminals, an Anode and a Cathode, does that mean that each single 7segment display will have 14 connecting pins or terminals. Well the answer is. No.

While an LED segment can be illuminated individually as required, one terminal of each internal LED is connected to a common point or node. Thus instead of having 14 connecting pins for the display we will only have only eight (7+1) pins, one each for the seven individual LEDs plus a common pin, and it is this 'common pin' which identifies the type and name of 7-segment display.

When the cathode terminals of all the LEDs used in the display are shorted together, the display is referred to as a Common-cathode, (CC) display. Likewise, when all the anode terminals of the LEDs used in the display are shorted together, the display is referred to as a Common-anode, (CA) display.

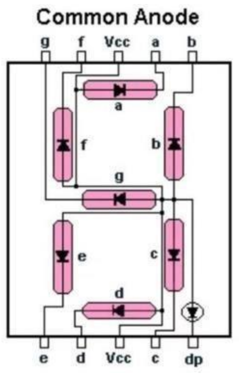
Thus a 7-segment display can be either a Common Cathode (CC) or a Common Anode (CA) type display.



Common Cathode (CC) Configuration

The Common Cathode (CC) Display - In the common cathode display, all the cathode (K) connections of the LED segments are tied together and connected to ground or zero volts. The individual segments are illuminated by the application of a suitable electric current to forward bias the individual Anode terminals (a to g). Thus common cathode display requires a driving circuit that can source a current.

Common Anode (CA) Configuration



The 74LS47 Decoder

The Common Anode (CA) Display - In the common anode display, all the anode (A) connections of the LED segments are joined together to a positive voltage supply.

The connection between the 74LS47 decoder/driver and the common anode display, requires seven resistors (eight if the decimal point is included) to limit current flow. For each LED segment of the display to light properly, the flow of current through each segment needs to be carefully controlled.

The best method of limiting the current through a displays segment is to use a current limiting resistor in series with each of the seven LED segments as shown. If we do not use a series connected resistor, maximum current would flow and the LED would be very bright for a short period of time, before becoming permanently destroyed.

As each LED segment of a typical 7-segment LED display is rated to operate at between 6 to 20mA offering a voltage drop across the LED's diode junction of about 1.8 volts for normal brightness. We can calculate the value of the current limiting resistor needed to produce the required current per LED segment.

TTL Decoder IC's

74LS48- Common Cathode
74LS247-Common Anode

CMOS Decoder IC's

74HC4511 Common Cathode

CD4513 Common Cathode

The TTL 74LS47 is the most popular 7-segment decoder IC by far and which is capable of driving common anode (CA) displays. The TTL 74LS47 has a 4-bit BCD input and seven individual active 'LOW' outputs for driving each of the seven LED segments.

Active 'LOW' means the output pin switches to ground (0V) to light an LED segment, while a 'HIGH' output will turn the LED segment 'OFF'. The HDSP series of displays is a good starting point but any standard common anode display will do, (and there are plenty to choose from).

With the aid of four switches, a 4-bit binary number is applied to the BCD inputs A, B, C and D of the 74LS47 decoder to produce the output signals a, b, c, d, e, f and g used to drive the 7-segment display generating the required numbers from 0 to 9 as shown.

7-segment Display Numbers

7-Segment Display Elements for all Ten Number Digits

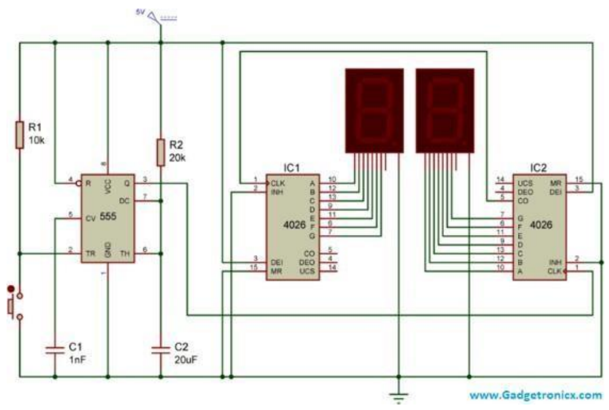
While the operation of the four SPST switches will cause the corresponding numbers or random characters to display, it can be a bit tedious operating the four switches at a time. So it would be better if we had a single IC chip that was able to generate the 4line binary information without using the four switches, and there is, the 74LS90 BCD Counter.

The 74LS90 integrated circuit that can be configured as a MOD-10 decade (divide-by10) counter to produce a BCD output code, counting from 0000 to 1001 and then resets itself back to 0000. By using this asynchronous decade counter/divider IC, we can increment the digits on the 7-segment display using just one single switch as shown.

Single Digit 7-segment Display Counter

We can now increment the numbers on the display from 0 to 9 by simply pressing one pushbutton switch, SW, ten times. By changing the position of the pushbutton and 1KΩ resistor, the count can be made to change on either the activation or the release of the push button, SW.

Our simple circuit shows how we can produce a 0 to 9 digital counter using a 74LS90 BCD Counter and a 74LS47 7-segment display driver. But this single-digit 0 to 9 counter can be extended with the addition of a second counter stage to make a two-digit 00 to 99 counter.



Two Digit 7-segment Display Counter

So how does this 2-digit 7-segment display counter work. The first half of the digital counter circuit works the same as before except that the activation of the pushbutton Sw, increments the 'one's' (also called 'units') LED display.

The first 74LS90 BCD counter, U1 counts upwards from 0 to 9 (0000 to 1001) on each closing (trailingedge) of SW. However, when the counting sequence reaches '8' (1000) on the one's display, pin-11 of U1 corresponding to output 'D' goes 'HIGH' and stays HIGH until U1 resets itself back to zero on the 10th count at which time pin-11 of U1 goes 'LOW' again.

As output pin-11 (BCD pin D) of U1 is connected to the clock A (CLK) input pin-14 of the second 74LS90 BCD counter U3, each successive HIGH/LOW switching action of pin-11 (output D) of U1 increments the second LED display for the ten's digit. Thus causing the two LED displays when placed side-by-side to count upwards from 00 to 99 before resetting back to 00 again for the next count.