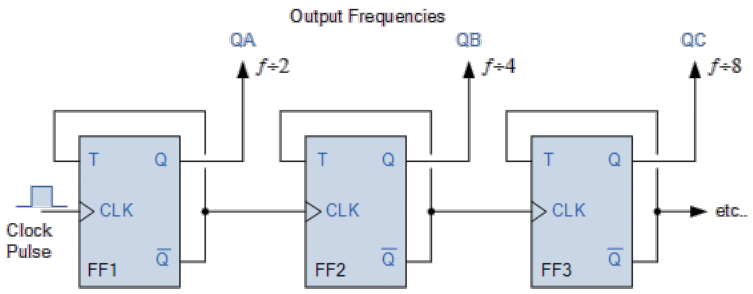


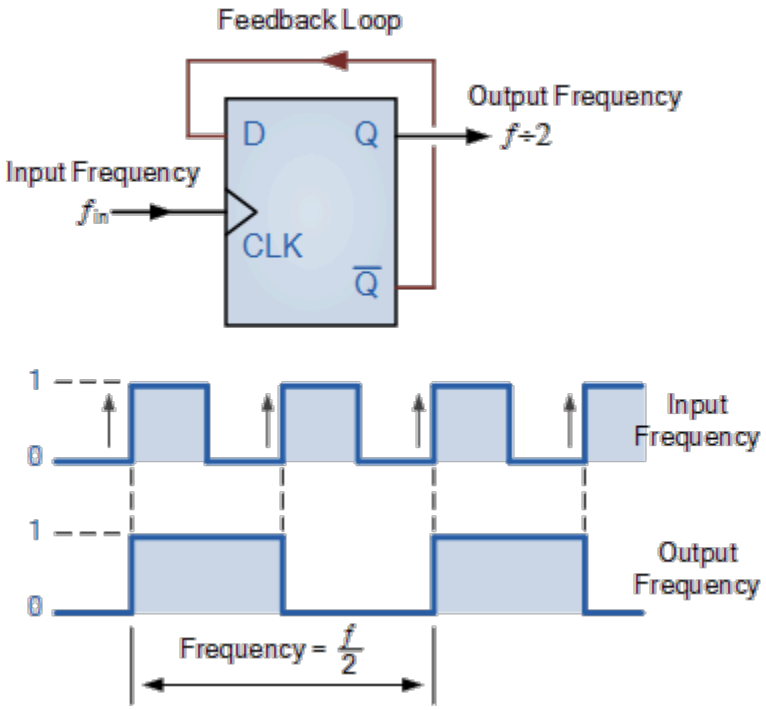
Frequency Division

Frequency Division uses divide-by-2 toggle flip-flops as binary counters to reduce the frequency of the input clock signal

In the Sequential Logic tutorials we saw how D-type Flip-Flop´s work and how they can be connected together to form a Data Latch.

Another useful feature of the D-type Flip-Flop is as a binary divider, for **Frequency Division** or as a “divide-by-2” counter. Here the inverted output terminal $\bar{Q}$ (NOT-Q) is connected directly back to the Data input terminal D giving the device “feedback” as shown below.

Divide-by-2 Counter



It can be seen from the frequency waveforms above, that by “feeding back” the output from $\bar{Q}$ to the input terminal D, the output pulses at Q have a frequency that are exactly one half (ƒ ÷ 2) that of the input clock frequency. In other words the circuit produces **Frequency Division** as it now divides the input frequency by a factor of two (an octave).

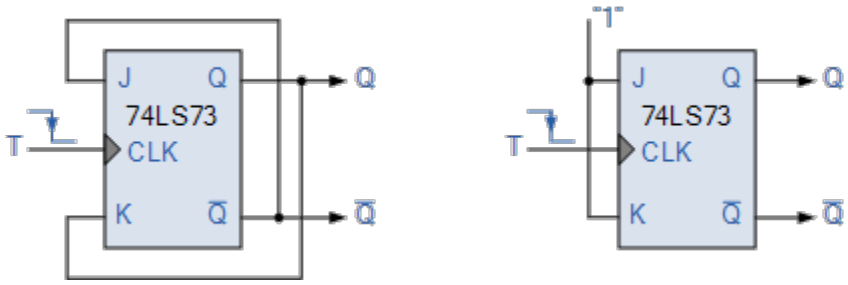
This then produces a type of counter called a “ripple counter” and in ripple counters, the clock pulse triggers the first flip-flop whose output triggers the second flip-flop, which in turn triggers the third flip-flop and so on through the chain producing a rippling effect (hence their name) of the timing signal as it passes through the chain.

The Toggle Flip-Flop

Another type of digital device that can be used for frequency division is the T-type or Toggle flip-flop. With a slight modification to a standard JK flip-flop, we can construct a new type of flip-flop called a **Toggle flip-flop**.

Toggle flip flops can be made from D-type flip-flops as shown above, or from standard JK flip-flops such as the 74LS73. The result is a device with only two inputs, the “Toggle” input itself and the negative controlling “Clock” input as shown.

74LS73 Toggle Flip Flop



A “Toggle flip-flop” gets its name from the fact that the flip-flop has the ability to toggle or switch between its two different states, the “toggle state” and the “memory state”. Since there are only two states, a T-type flip-flop is ideal for use in frequency division and binary counter design.

Binary ripple counters can be built using “Toggle” or “T-type flip-flops” by connecting the output of one to the clock input of the next. Toggle flip-flops are ideal for building ripple counters as it toggles from one state to the next, (HIGH to LOW or LOW to HIGH) at every clock cycle so simple frequency divider and ripple counter circuits can easily be constructed using standard T-type flip-flop circuits.

If we connect together in series, two T-type flip-flops the initial input frequency will be "divided-by-two" by the first flip-flop (ƒ ÷ 2) and then "divided-by-two" again by the second flip-flop (ƒ ÷ 2) ÷ 2, giving an output frequency which has effectively been divided four times, then its output frequency becomes one quarter value (25%) of the original clock frequency, (ƒ ÷ 4).

Each time we add another toggle or "T-type" flip-flop to the chain, the output clock frequency is halved or divided-by-2 again and so on, giving an output frequency of 2^n where "n" is the number of flip-flops used in the sequence.

Then the Toggle or T-type flip-flop is an edge triggered divide-by-2 device based upon the standard JK-type flip flop and which is triggered on the rising edge of the clock signal. The result is that each bit moves right by one flip-flop. All the flip-flops can be asynchronously reset and can be triggered to switch on either the leading or trailing edge of the input clock signal making it ideal for **Frequency Division**.

This type of counter circuit used for frequency division is commonly known as an **Asynchronous 3-bit Binary Counter** as the output on QA to QC, which is 3 bits wide, is a binary count from 0 to 7 for each clock pulse.

In an asynchronous counter, the clock is applied only to the first stage with the output of one flip-flop stage providing the clocking signal for the next flip-flop stage and subsequent stages derive the clock from the previous stage with the clock pulse being halved by each stage.

This arrangement is commonly known as *Asynchronous* as each clocking event occurs independently as all the bits in the counter do not all change at the same time. As the counter counts sequentially in an upwards direction from 0 to 7. This type of counter is also known as an "up" or "forward" counter (**CTU**) or a **"3-bit Asynchronous Up Counter"**. The three-bit asynchronous counter shown is typical and uses flip-flops in the toggle mode. Asynchronous "Down" counters (**CTD**) are also available.

Truth Table for a 3-bit Asynchronous Up Counter

Clock Cycle	Output Bit Pattern		
	QC	QB	QA
0	0	0	0
1	0	0	1

2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Therefore we can see that the output from the D-type flip-flop is at half the frequency of the input, in other words it counts in 2's. By cascading together more D-type or Toggle Flip-Flops, we can produce a divide-by-2, divide-by-4, divide-by-8, etc. circuit which will divide the input clock frequency by 2, 4 or 8 times, in fact any value to the power-of-2 we want making a binary counter circuit.

Binary Counters

Thus we can see that a counter is nothing more than a specialised register or pattern generator that produces a specified output pattern or sequence of binary values (or states) upon the application of an input pulse signal called the "Clock".

The clock is actually used for data transfer in these applications. Typically, counters are logic circuits that can increment or decrement a count by one but when used as asynchronous divide-by-n counters they are able to divide these input pulses producing a clock division signal.

Counters are formed by connecting flip-flops together and any number of flip-flops can be connected or "cascaded" together to form a "divide-by-n" binary counter where "n" is the number of counter stages used and which is called the **Modulus**. The modulus or simply "MOD" of a counter is the number of output states the counter goes through before returning itself back to zero, i.e, one complete cycle.

Then a counter with three flip-flops like the circuit above will count from 0 to 7 ie, 2^n-1. It has eight different output states representing the decimal numbers 0 to 7 and is called a **Modulo-8** or **MOD-8** counter. A counter with four flip-flops will count from 0 to 15 and is therefore called a **Modulo-16** counter and so on.

An example of this is given as.

3-bit Binary Counter = 2^3 = 8 (modulo-8 or MOD-8)

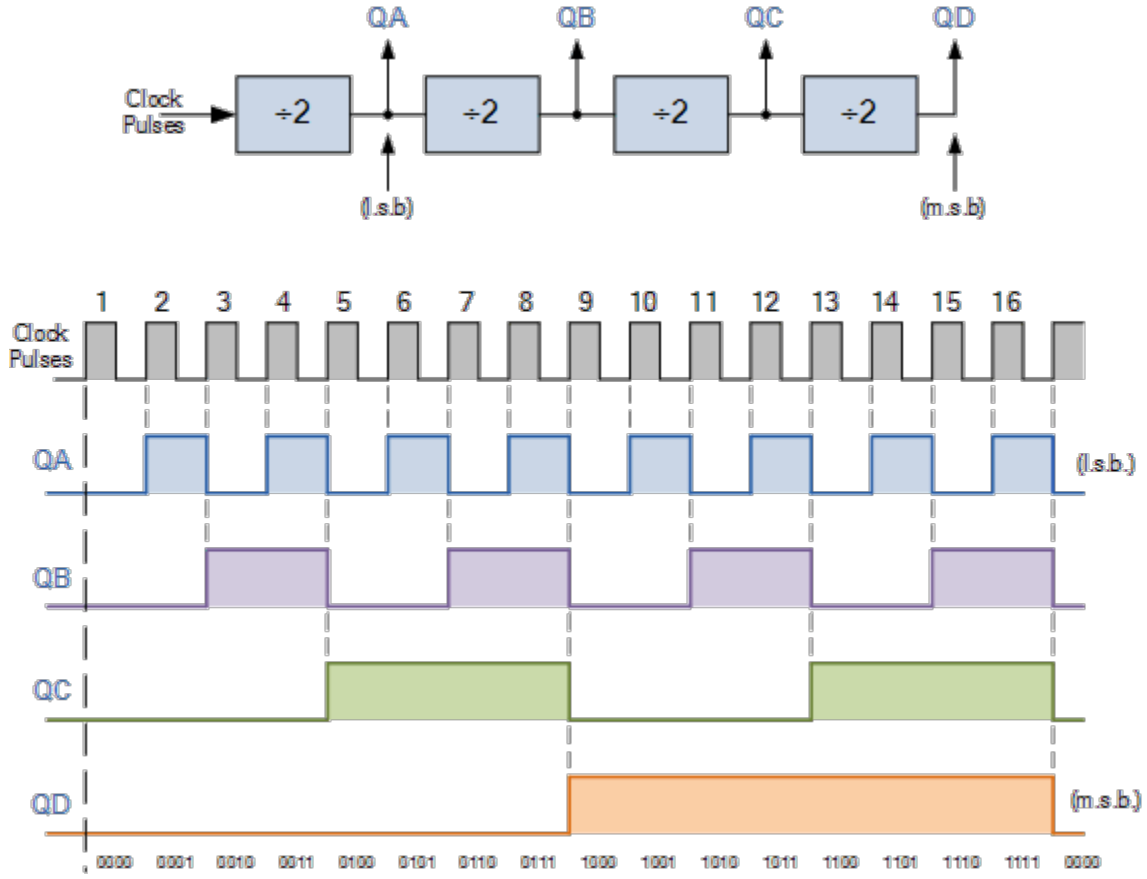
4-bit Binary Counter = 2^4 = 16 (modulo-16 or MOD-16)

8-bit Binary Counter = 2^8 = 256 (modulo-256 or MOD-256)

and so on..

The Modulo number can be increased by adding more flip-flops to the counter and cascading is a method of achieving higher modulus counters. Then the modulo or MOD number can simply be written as: MOD number = 2^n

4-bit Modulo-16 Counter



Multi-bit asynchronous counters connected in this manner are also called **“Ripple Counters”** or ripple dividers because the change of state at each stage appears to “ripple” itself through the counter from the LSB output to its MSB output connection. Ripple counters are available in standard IC form, from the 74LS393 Dual 4-bit counter to the 74HC4060, which is a 14-bit ripple counter with its own built in clock oscillator and produce excellent frequency division of the fundamental frequency.

Frequency Division Summary

For **frequency division**, toggle mode flip-flops are used in a chain as a divide by two counter. One flip-flop will divide the clock, f_{IN} by 2, two flip-flops will divide f_{IN} by 4 (and so on). One benefit of using toggle flip-flops for frequency division is that the output at any point has an exact 50% duty cycle.

The final output clock signal will have a frequency value equal to the input clock frequency divided by the MOD number of the counter. Such circuits are known as “divide-by-n” counters. Counters can be formed by connecting individual flip-flops together and are classified according to the way they are clocked.

In *Asynchronous counters*, (ripple counter) the first flip-flop is clocked by the external clock pulse and then each successive flip-flop is clocked by the output of the preceding flip-flop. In *Synchronous counters*, the clock input is connected to all of the flip-flop so that they are clocked simultaneously.

In the next tutorial we will look at Asynchronous counters, and see that the main characteristic of an asynchronous counter is that each flip-flop in the chain derives its own clock from the previous flip-flop and is therefore independent of the input clock.

54 Comments

Write your comment here

Notify me of follow-up comments by email.

SUBMIT

Yscz Uddin

A modulo 8 Gray code sequence synchronous counter using JK flip flop is to be designed with one input terminal which receives pulse signals and one output terminal. It should be capable of counting in gray code number system and producing an output pulse for every eight input pulse.

Can any one slove this

Posted on November 06th 2020 | 9:00 am

Reply

---

amir imran

hi sir what is mean by frequency divide by 8 devices is it 2^3 or 2^8????????

Posted on June 19th 2020 | 6:31 am

Reply

Vinita jat

i can help u sir for exam ISRo

Posted on January 07th 2020 | 10:49 am

Reply

YORAM

NICE WEBSITE .

Posted on October 12th 2019 | 9:39 am

Reply

Sagar s Sagar s

I want design circuit and verilog code for counter that counts 16 clk cycle and produces a control signal

Posted on September 10th 2019 | 5:59 pm

Reply

Emmanuel John Paghi

It's great and interesting.
Please help me with circuits and truth table of mod 2-12 .

Posted on May 05th 2019 | 2:12 am

Reply

Sumit kumar

Wait for all digital techniques topics

Posted on December 10th 2018 | 8:03 am

Reply

Shivankar tiwari

Is it possible to make different frequencies of 2 source same by any method??

Posted on October 12th 2018 | 3:50 pm

Reply

Stephanie

I dont understand my homework that they gaved me so can i get help it's about a frequency table about division

Posted on September 13th 2018 | 9:17 pm

Reply

---

girish

frequency division which 50mhz is reduced to 33mhz by using counter..

Posted on September 09th 2018 | 7:10 am

Reply

Ravindra Kurandle

I am interested

Posted on August 26th 2018 | 7:41 am

Reply

Assaf Meiron

Nice article, helpful.
Thank you

Posted on August 09th 2018 | 3:02 pm

Reply

KHyogananda

Nice information.

Sir, can you update on how to design frequency division circuit.
For example div / 3, div / 5 circuits.

Thank you.

Posted on May 04th 2018 | 10:56 am

Reply

Stephanie

Well my homework has 1,4,7,5,7,3,1 as the numbers and it told me too add then divide so that's what i don't get

Posted on September 13th 2018 | 9:19 pm

Reply

Sachin

Give us more detail information

Posted on February 25th 2018 | 11:54 am

Reply

Marian Woelf

Hey there, I dunno if there is someone familiar with this topic but I am trying to find a solution for my

(currently) not working rpm meter. I hoocked it up to the pickup coil of my 4-cylinder engine which gives an output current of ~0.24 V at idle speed. The Puls from the coil picks up both Ignition coil signals resulting in double rpm on meter. So the question would be if a cirquit with a D-Type Flip-Flop would solve my problem and take out every second pulse created by the generator? Unfortunately I have no idea about the ampere from this signal. I just know that it is very small as I wasn't able to measure it with my multimeter even in the smallest ratio it would only sho '001' on display. Would be huge if anyone could help me.

Posted on January 30th 2018 | 7:22 am Reply

---

conforter

pi need the formulae for frequency division

Posted on January 11th 2018 | 6:59 pm Reply

Himanshu

How to convert 33 percent duty cycle to 50 percent duty cycle and also the reverse using clock divider?

Posted on November 25th 2017 | 3:36 am Reply

Rakib Hasan

Frequency Division = f/2^n
n= Number of FFs .

Posted on September 18th 2017 | 4:14 am Reply

ABHAY PRATAP SINGH

Sir,
please will you explain me non inter frequency division circuit using finite sate machine.

Posted on July 01st 2017 | 1:25 pm Reply

Theo Aftinos

Because i d'nt know very good English , i would like to be all text into to my language ! that mean Greek language ! please it is possible ? thaks very much !

Posted on February 21st 2017 | 9:07 am Reply

Wayne Storr

Hello Theo, I am sorry but we do not provide translated tutorials. Just copy and past the text into Google Translation (https://www.google.gr/#q=translate)

Posted on February 21st 2017 | 10:10 am Reply

View More