بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

# Integer Clock Divider

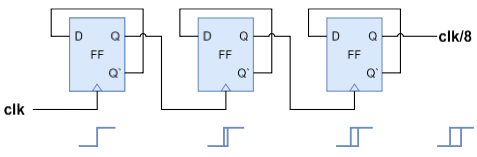
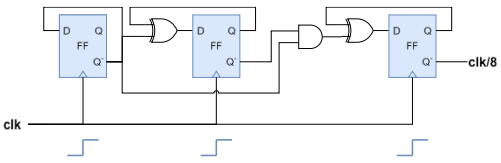
## Motivation

From its name, an integer clock divider is a digital circuit that divides the frequency of a clock by an integer number.

For example, if I have a digital system that contains a CPU and a UART module, it is known that the CPU operates at a higher frequency than the UART. So, if the frequency of the system clock must be divided to a suitable value that the UART can operate at.

Another aspect that should be considered is power consumption. Dynamic power depends on the frequency of the clock signal, $P_{clk} = \frac{1}{2}\alpha C_L {V_{DD}}^2 f$, and if there are digital blocks with low performance, then we should divide its clock to reduce the power consumption.

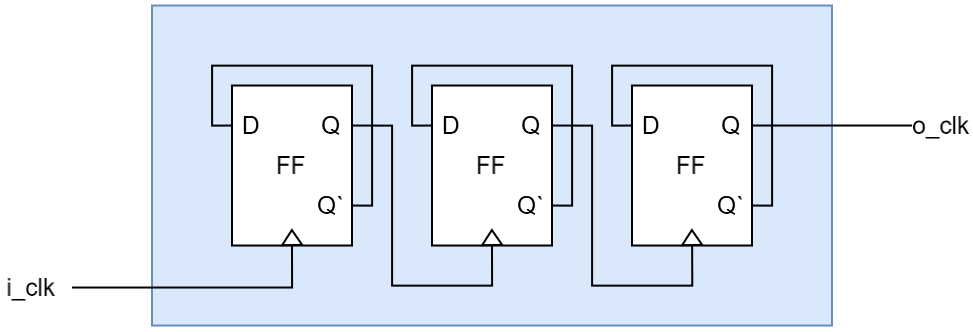
There are two types of clock divider which are: synchronous and asynchronous. These types have opposite advantages and disadvantages which are summarized in the table below.

| | Asynchronous CLK Divider | Synchronous CLK Divider |
|---|---|---|
| Advantages | • Each FF stage runs at lower clock frequence and hence reduces power consumption.<br>• Reduced loading on high frequency clock. | • Reduced clock jitter |
| Disadvantages | • Clock jitter accumulation | • Each FF stage runs at higher clock frequence and hence increases power consumption.<br>• Large loading on high frequency clock. |
| Example |  |  |

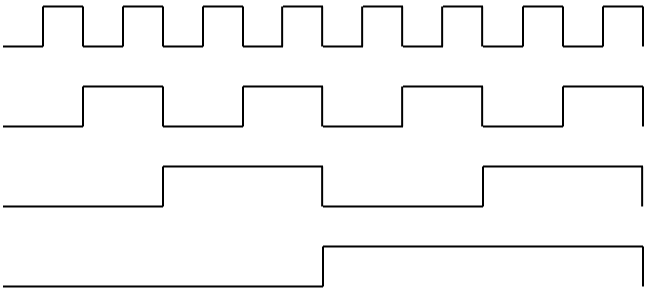
# Asynchronous Clock Divider

## Clock Divider by 8

### Block Diagram




### Waveform



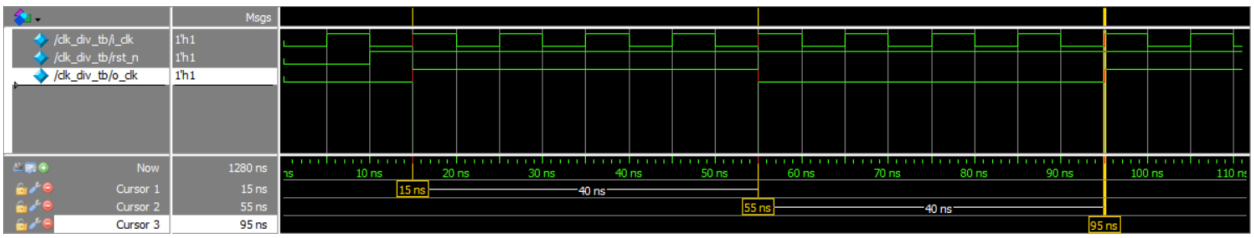
### RTL

GitHub

### TB

GitHub

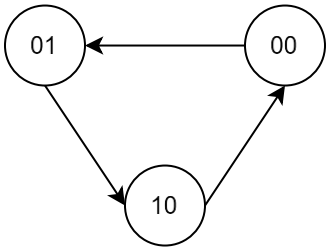
### Simulation



# Synchronous Clock Divider

## Using Mod-N Counter: Clock Divider by 3

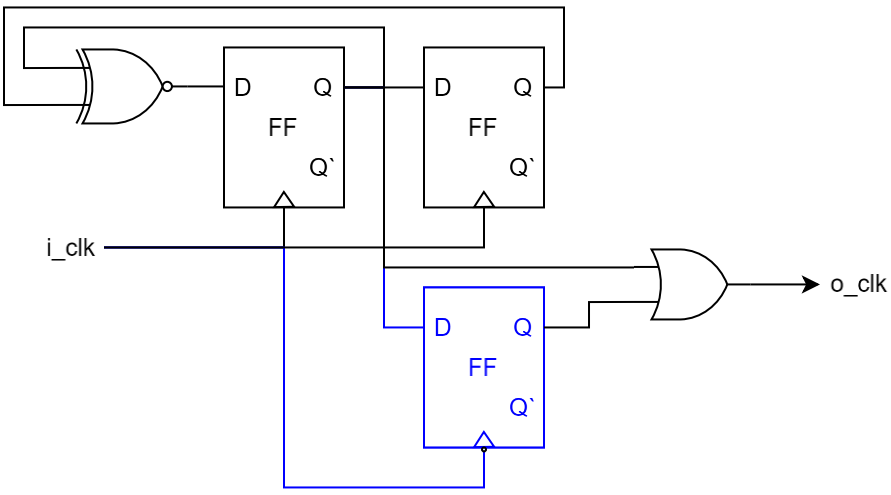
### State Diagram



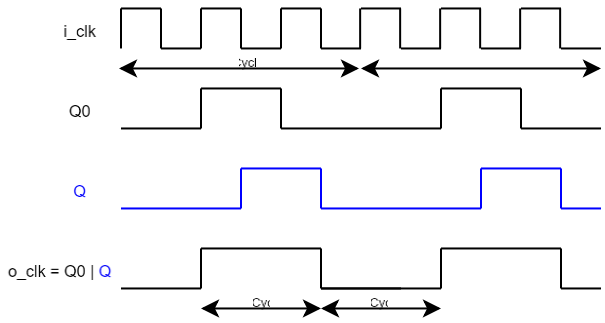

### State Table

| Current State | | Next State | | | |
|---|---|---|---|---|---|
| Q1 | Q0 | Q1 | Q0 | D1 | D0 |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |

### Block Diagram



## Waveform



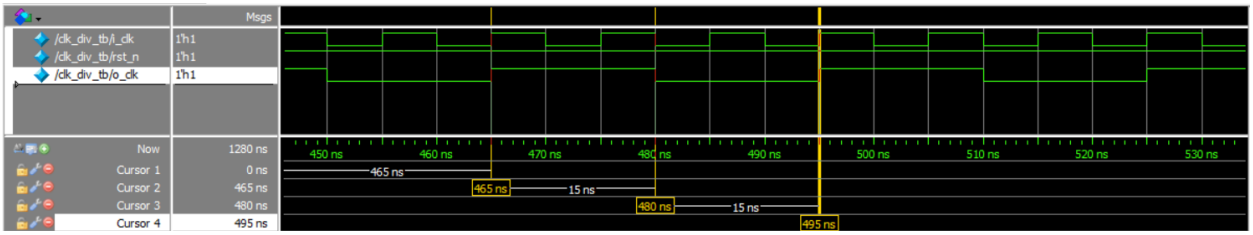

## RTL

GitHub

## TB

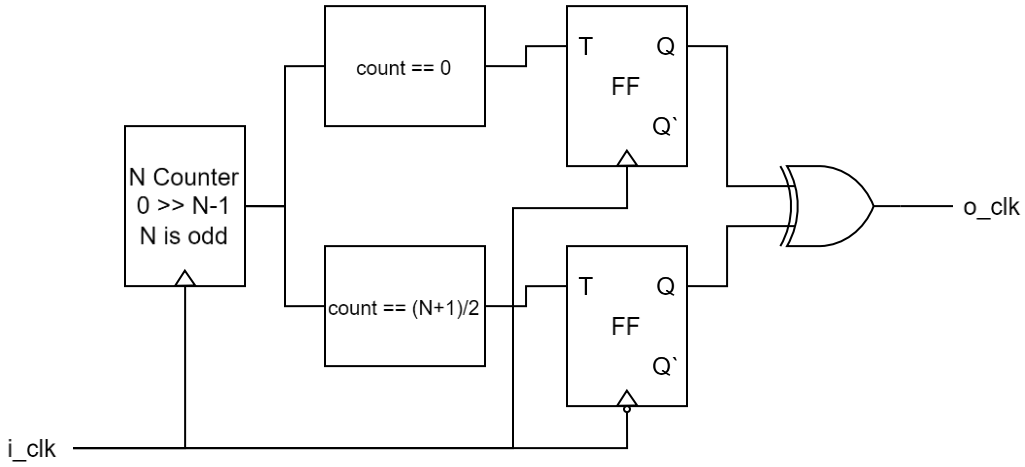
GitHub

## Simulation

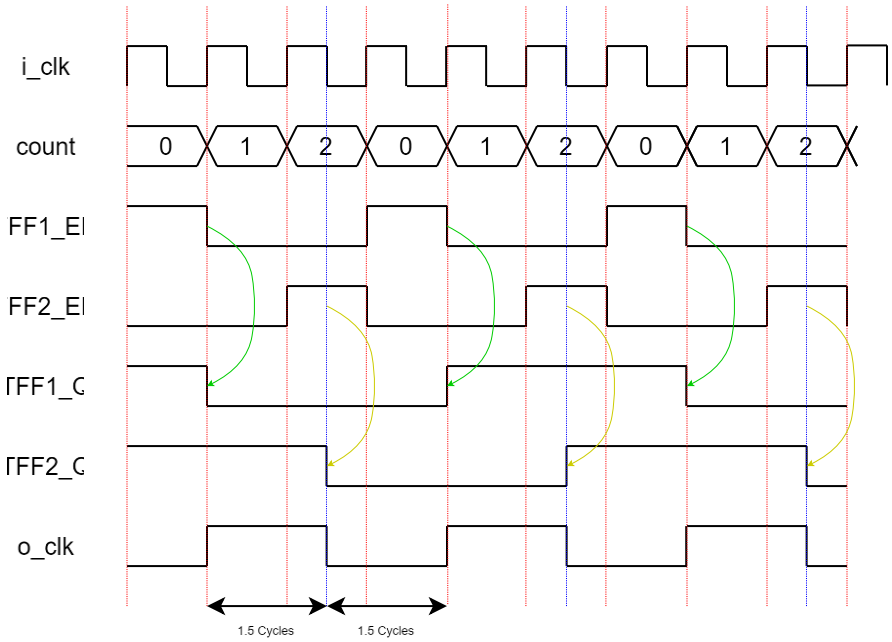


# Using T-FF: Odd Clock Divider

## Block Diagram




## Waveform



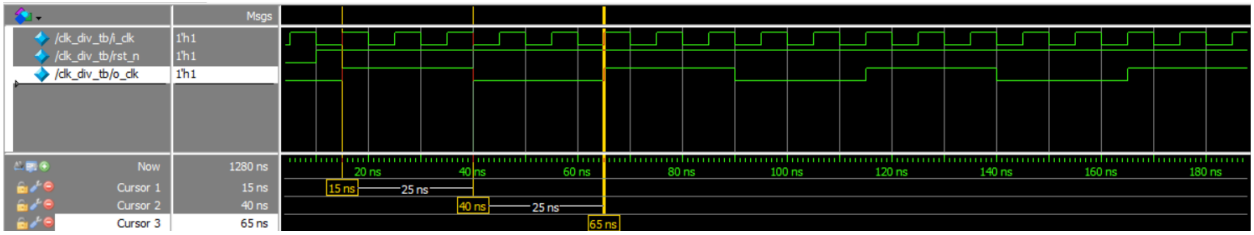

## RTL

GitHub

## TB

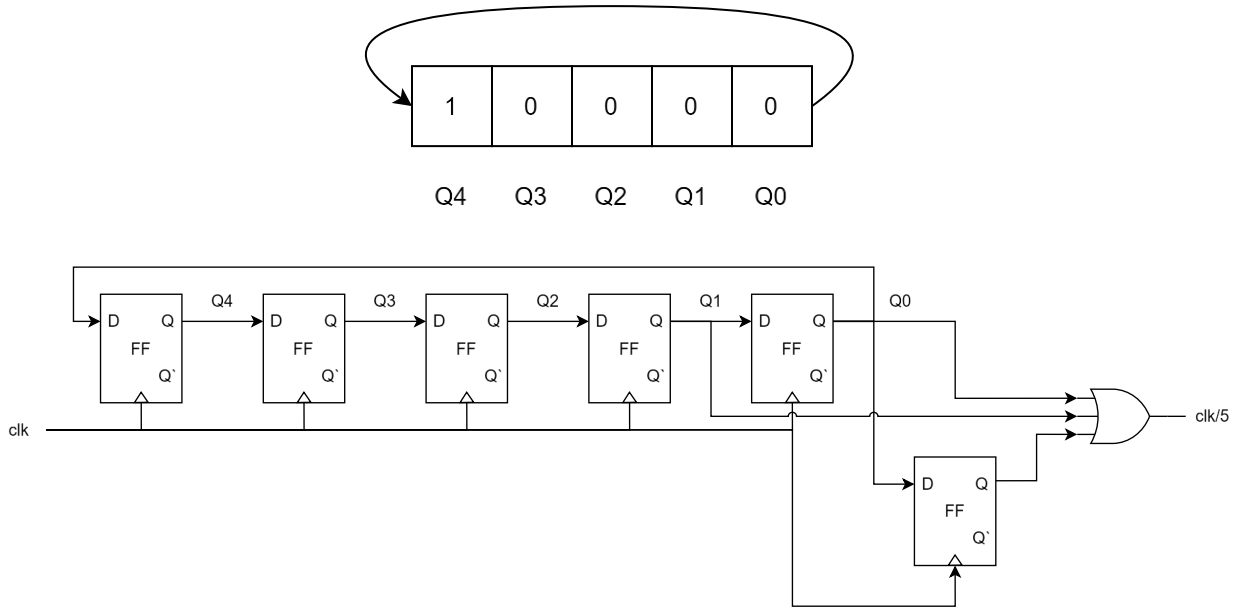
GitHub

## Simulation

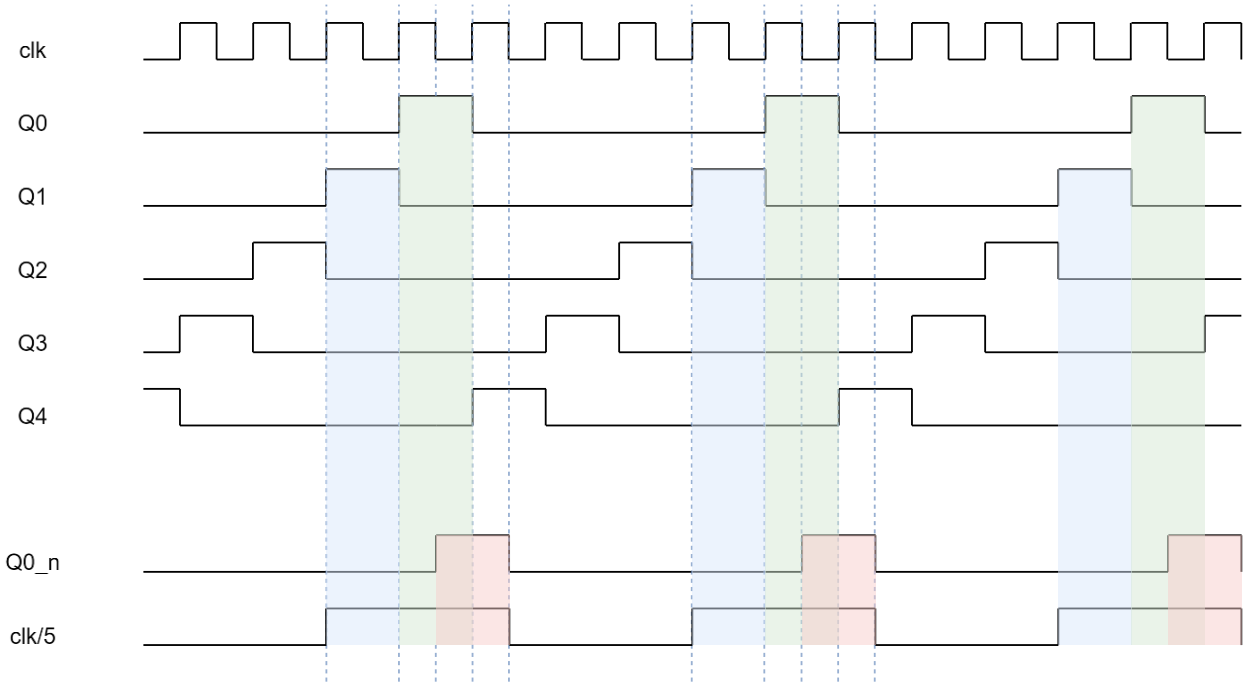

Msgs
/clk_div_tb/i_clk
/clk_div_tb/rst_n
/clk_div_tb/o_clk
1'h1
1'h1
1'h1
Now
1280 ns
Cursor 1
15 ns
Cursor 2
40 ns
Cursor 3
65 ns
20 ns
40 ns
60 ns
80 ns
100 ns
120 ns
140 ns
160 ns
180 ns
15 ns
25 ns
40 ns
25 ns
65 ns

# Using Ring Counter: Odd Clock Divider

## Block Diagram




## Waveform



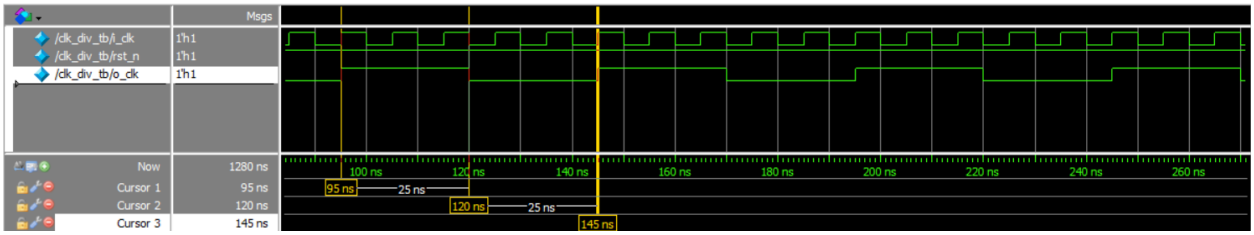

## RTL

[GitHub]

## TB

[GitHub]

## Simulation



# Behavioral: Even Clock Divider

## Idea

We use a counter to count to half the value of the divide ratio then we reset this counter.

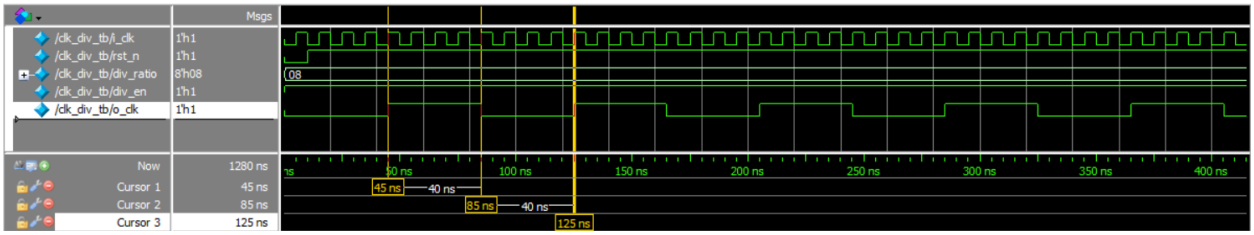
## RTL

GitHub

## TB

GitHub

## Simulation



# General Clock Divider

## Idea

We use two counters; one counts with positive edge of the clock and the other counts with negative edge of the clock and each one counts to the division ration – 1.

If the value of the counter is greater than division ratio divided by two, a pulse is generated.

Then to generate the divided clock, we OR the pulse signals generated from the two counters if the division ratio is odd number and if it is an even number, then the divided clock is the same as the pulse generated from the positive edge counter.

## RTL

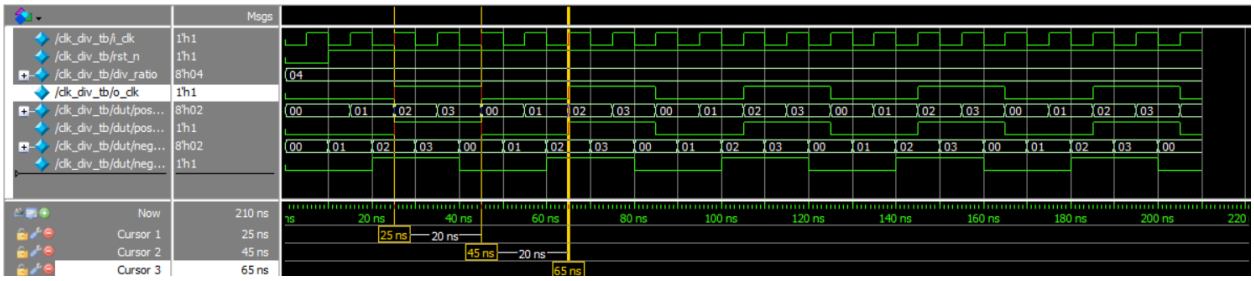
GitHub

## TB

GitHub

## Simulation

Even Ratio = 4




Odd Ratio = 7