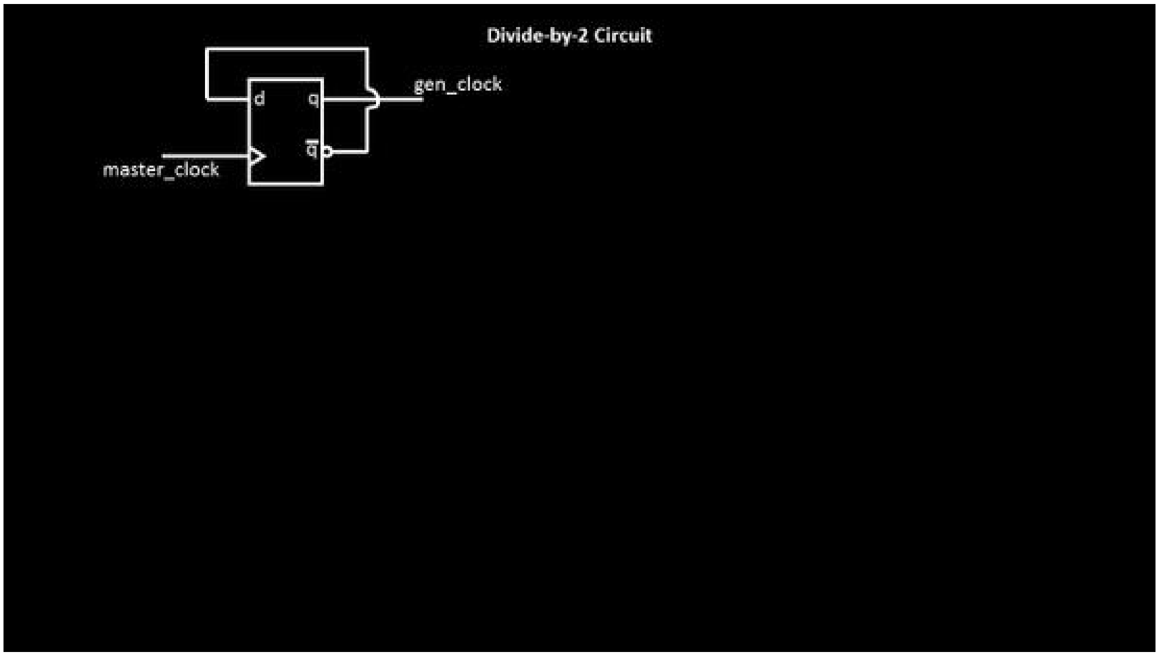
Generated Clock & master clock

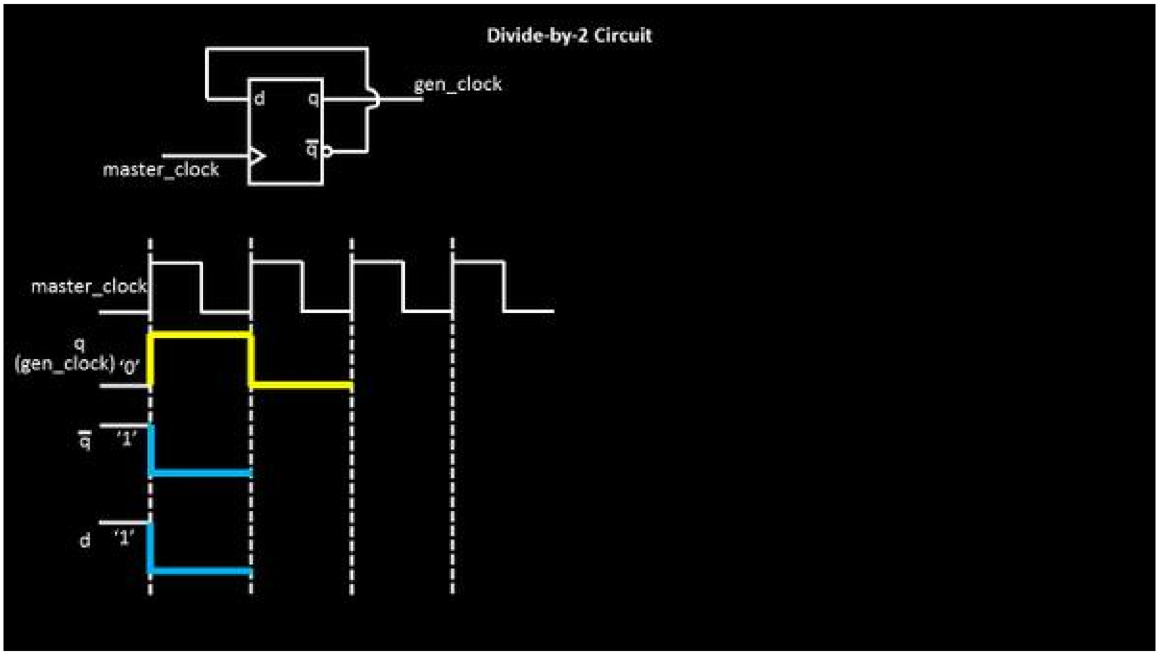
LET’S MAKE IT SIMPLE

Kunal Ghosh

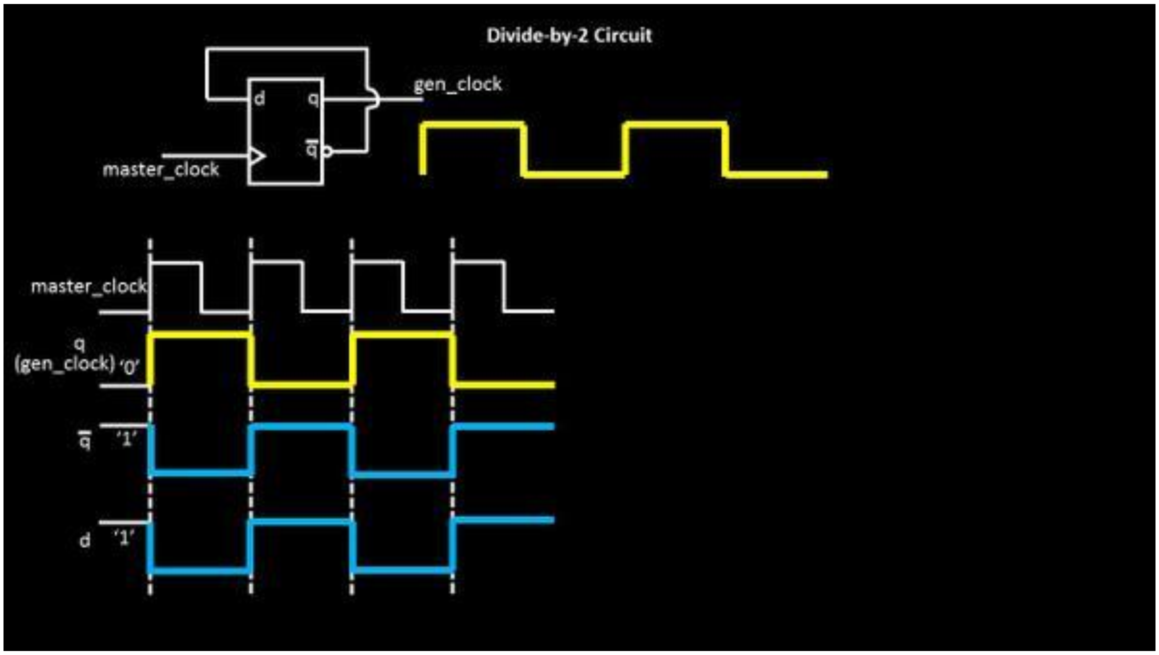
I get this one occasionally … not particularly about the concept, but about the ways we can create a generated clock definition. Too many options make it too complex… But you know what, with a handful of images, this topic becomes way too simple… Let me show how? Start with below simple divide-by-2 circuit and try to define it in all ways possible



I believe, best way to understand any topic is the 'graphical way'. So, let me show you how the input and output wave-forms look like, in below image



If you start with master clock of say 1ns period, the output is expected of 2ns period (frequency divide-by-2). So at the first rising edge of clock (assuming the initial state of circuit is 'q' = 0 and 'd' = 1), the data at 'd' pin will be propagated at 'q' output (since it's a rise-edge triggered flop), and 'q' becomes '1', 'q_bar' becomes '0', and 'd' is at '0' (‘q_bar’ is connected to ‘d’). This is shown in above image. While, we keep doing this, the state of the circuit changes at every rise edge of master_clock, and below is what you get



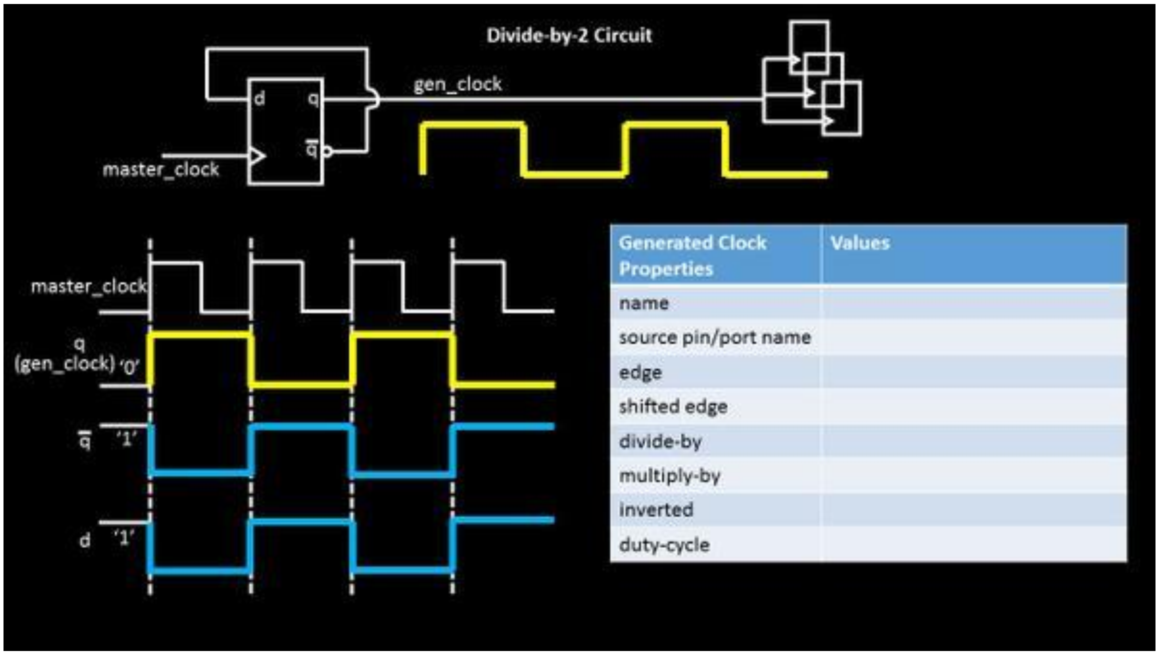
Now that was the digital hardware part of it. Let me tell you a secret of ‘static timing analysis’ of ASIC…We are ‘static’. We need definitions. We understand software language. And we will not change J . You are right in thinking.. We are adamant on this J

Designer – Ok, so let me speak to you in your language. Let me write a define this new ‘gen_clock’ for you so you understand it and use it

STA – you better do !!

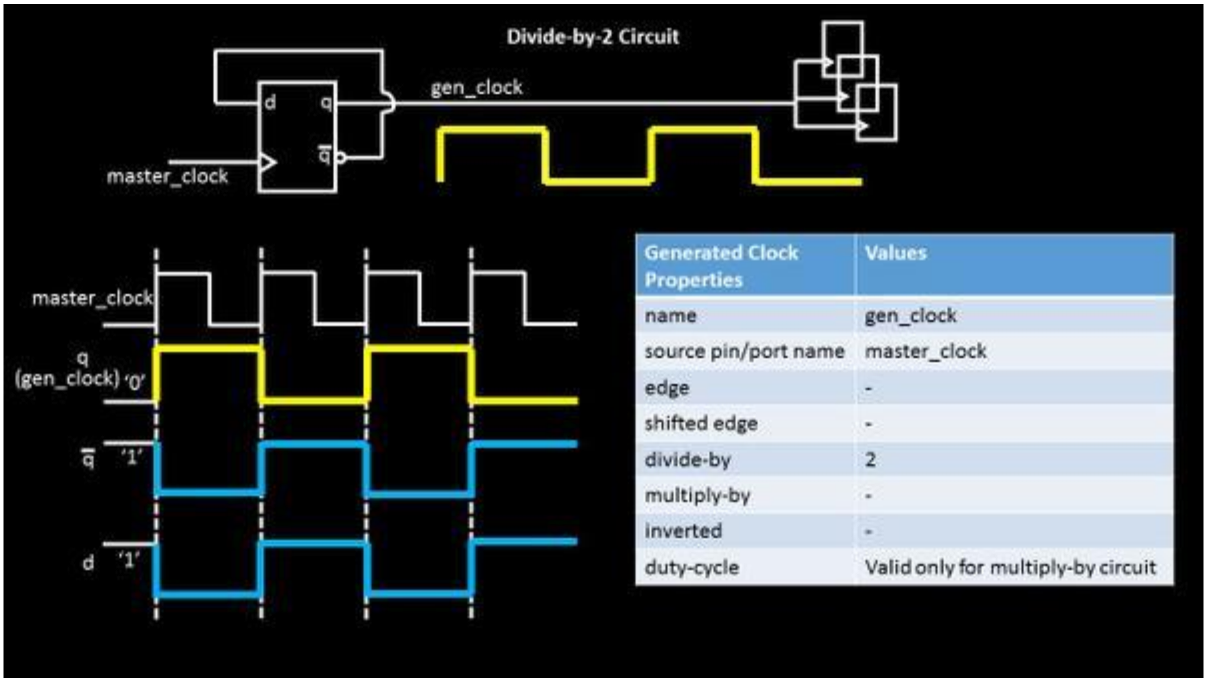
Seriously, I have worked ~6 years in this area, and believe me, these analysis is very rude

Anyways, coming back, we need to fill up below table so that the analysis engine understands it and passes this information to further flops clock pins, it might be driving



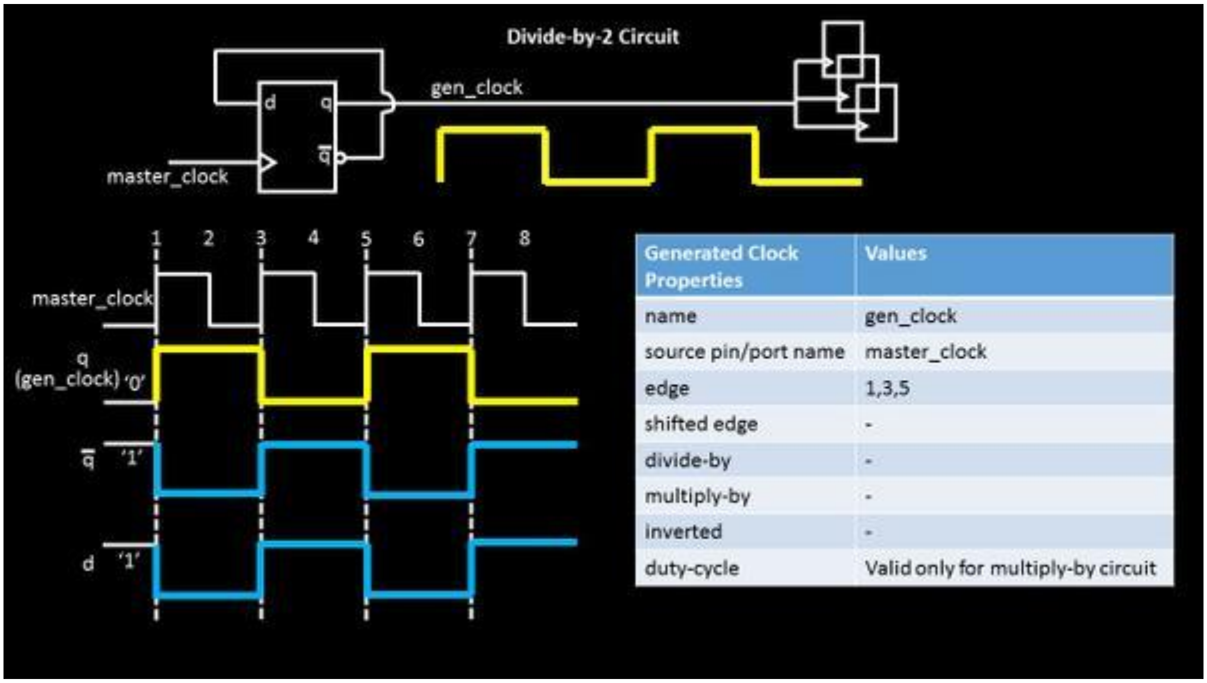
Generated Clock Properties	Values
name	
source pin/port name	
edge	
shifted edge	
divide-by	
multiply-by	
inverted	
duty-cycle	

By the look of it, below seems to be most appropriate and valid values for 'gen_clock'



Generated Clock Properties	Values
name	gen_clock
source pin/port name	master_clock
edge	-
shifted edge	-
divide-by	2
multiply-by	-
inverted	-
duty-cycle	Valid only for multiply-by circuit

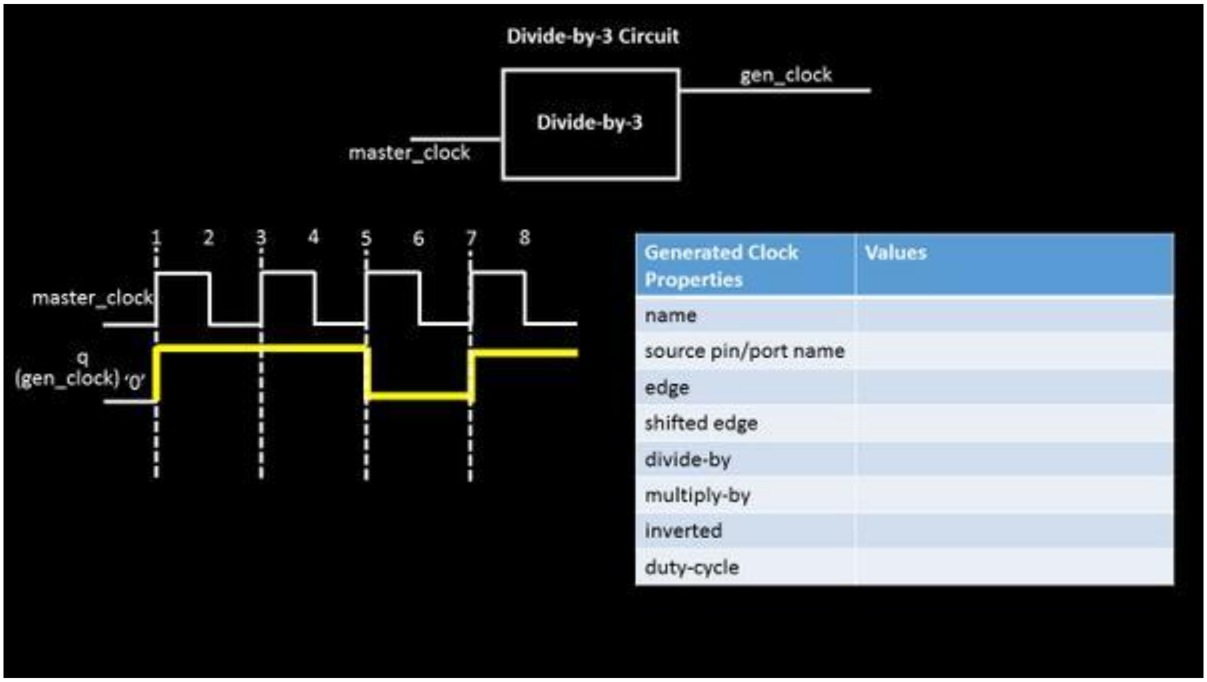
Though this looks the easiest, there are other ways to define this (not recommended for this simple one). So if we were to define the gen_clock based on the edges of master clock, below how it will like. We remove the 'divide-by' option and use the edge values of 1,3,5 to define the new clock. This says, that at '1' edge of master_clock, the first rise edge of gen_clock arrives. At '2' edge of master_clock, the first fall edge of gen_clock arrives, and at '3' edge of 'master_clock', the second rise edge arrives.



Generated Clock Properties	Values
name	gen_clock
source pin/port name	master_clock
edge	1,3,5
shifted edge	-
divide-by	-
multiply-by	-
inverted	-
duty-cycle	Valid only for multiply-by circuit

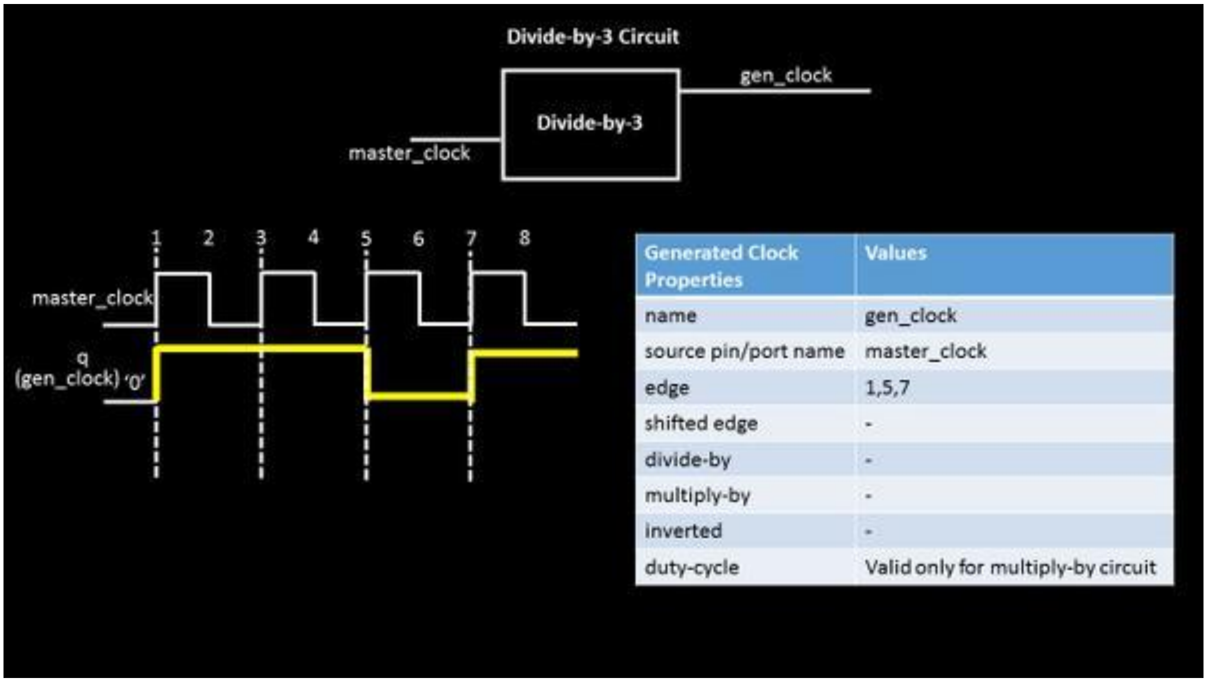
Isn't that simple…wait a sec, it was simple because I took a simple example. Let me come up with a more complex circuit, break it down to little pieces and explain how to fill up this table.

Let's take a divide-by-3 circuit and below is how its waveform at output will look like (assuming a non-50% duty cycle). The output clock period is 3 times the input clock period and hence, frequency is divided-by-3. (It would be a great idea to show the interiors of divide-by-3 circuit. Stay with me and I will do that in following posts)



Generated Clock Properties	Values
name	
source pin/port name	
edge	
shifted edge	
divide-by	
multiply-by	
inverted	
duty-cycle	

So, now, if we need to write this gen_clock definition, its simple. Just fill out the table and below is what we get



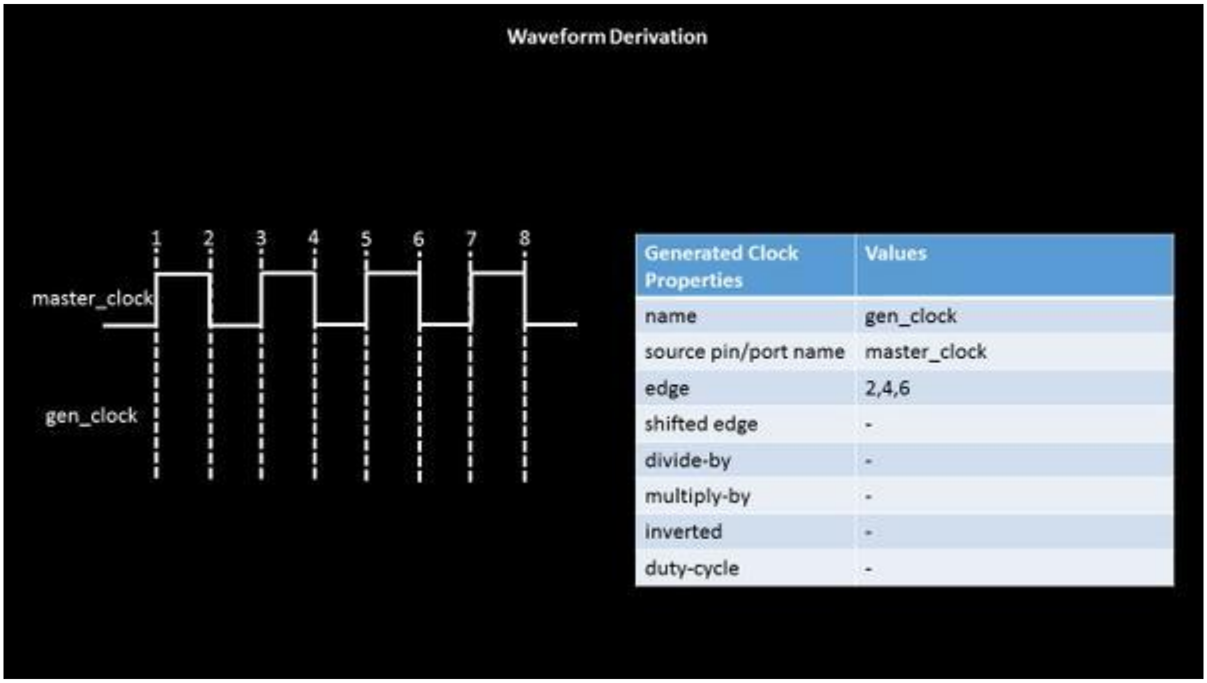
Generated Clock Properties	Values
name	gen_clock
source pin/port name	master_clock
edge	1,5,7
shifted edge	-
divide-by	-
multiply-by	-
inverted	-
duty-cycle	Valid only for multiply-by circuit

This shows a perfect usage of ‘edge’ option. For circuits producing a clock waveform which has non-50% duty cycle, the edge option can be very well used to define an output clock and can also be propagated throughout the circuit. It just says, that 1st rise edge of gen_clock arrives at 1st edge of master_clock, 1st fall edge of get_clock arrives at 5th edge of master_clock and 2nd rise edge (which completes 1 clock cycle) arrives at 7th edge of master_clock

Pay attention to the words above. I am using ‘rise’ and ‘fall’ words for defining gen_clock (as these words are the one’s that completes one clock cycle) and I am using just the word ‘edge’ for master_clock and ‘edge’ can be either rise or fall, doesn’t matter from gen_clock definition perspective

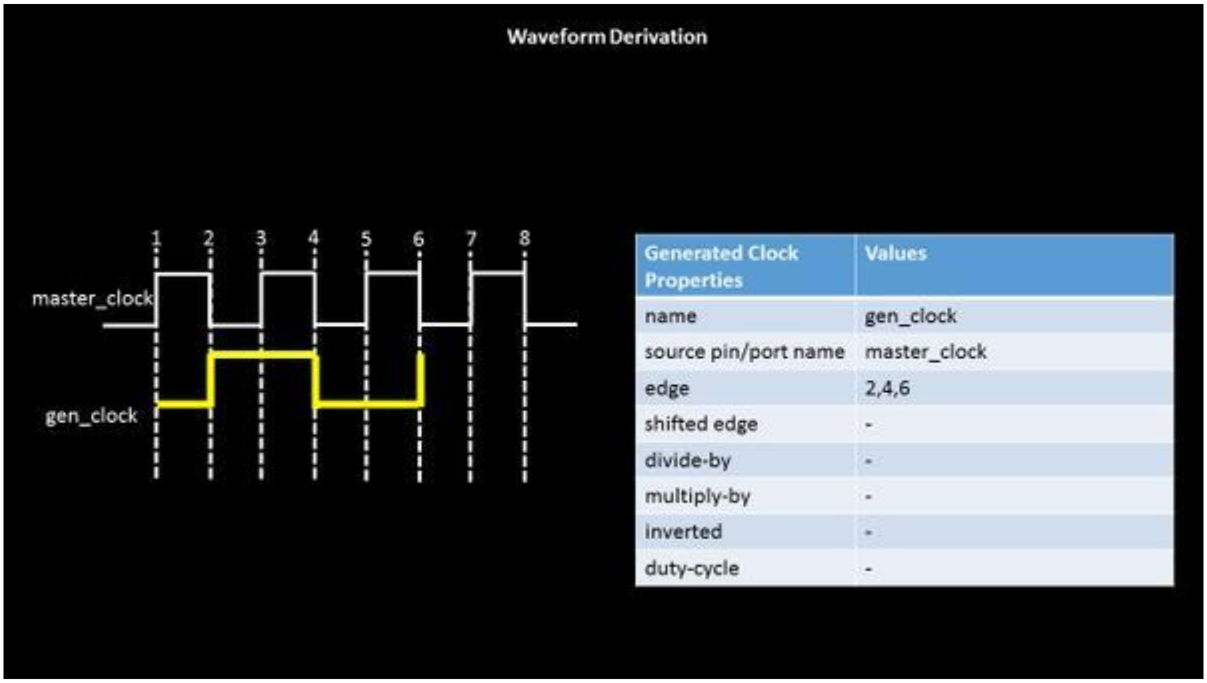
Now, to explore other options of this table, let’s consider another circuit specification, where output is inverted and divide-by-2 of input waveform.

Let’s assume that, and begin with below simple example of waveform table (by the way, this is a very classic interview question in semiconductor industries. Also, this technique is used to debug complex clock constraints in industries)


Waveform Derivation

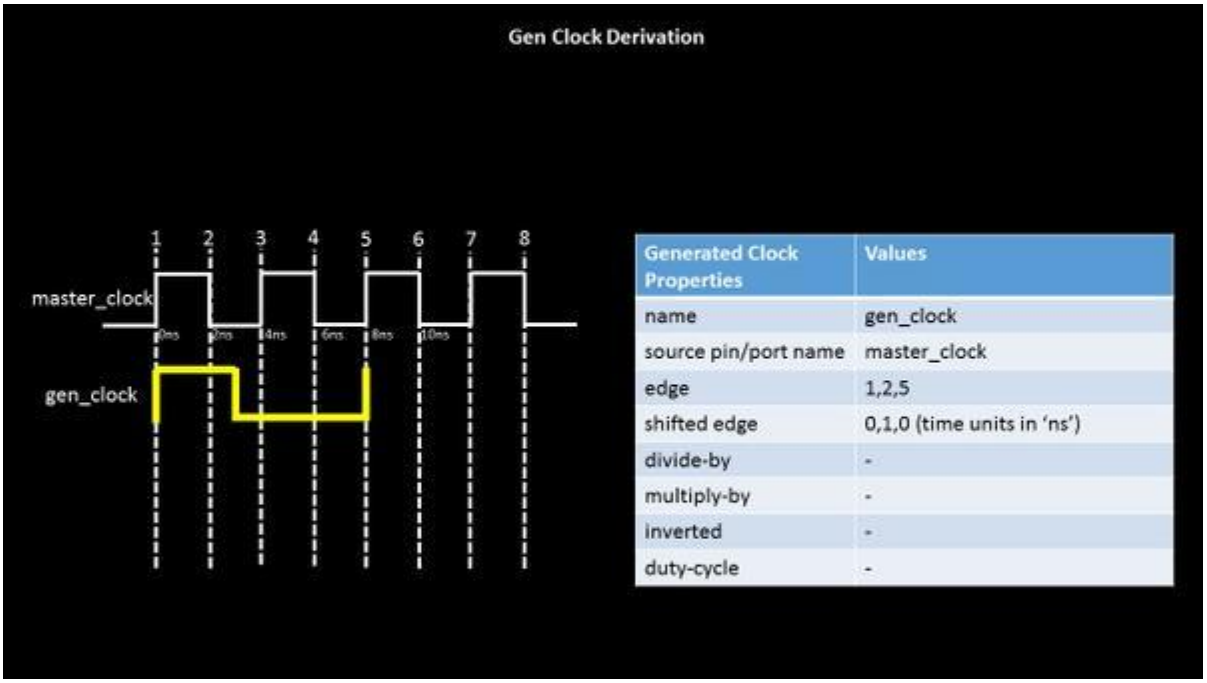
Generated Clock Properties	Values
name	gen_clock
source pin/port name	master_clock
edge	2,4,6
shifted edge	-
divide-by	-
multiply-by	-
inverted	-
duty-cycle	-

So, the above table, says, the first rise edge of the generated clock arrives at 2nd edge of master clock (remember, it doesn't matter whether 2nd edge of master clock is a rise or fall), the first fall edge of generated clock arrives at 4th edge of master clock and second rise edge comes at 6th edge of master clock and below is how your generated clock will look like


Waveform Derivation

Generated Clock Properties	Values
name	gen_clock
source pin/port name	master_clock
edge	2,4,6
shifted edge	-
divide-by	-
multiply-by	-
inverted	-
duty-cycle	-

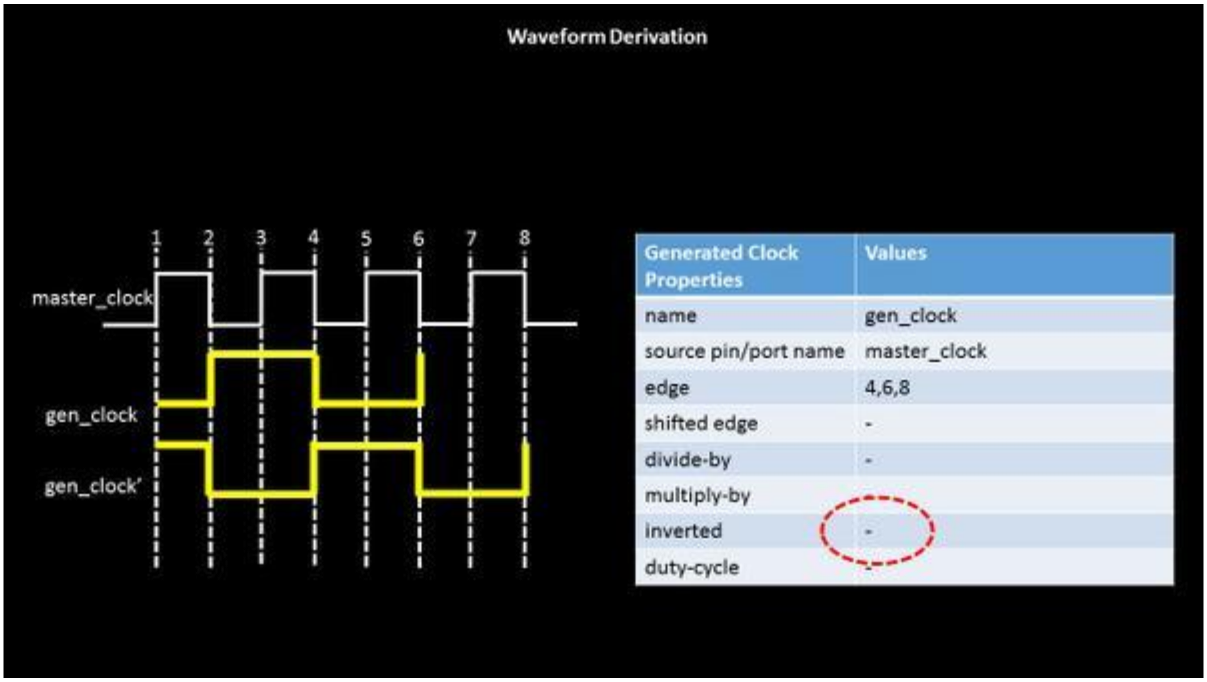
Let's put this to test. How about everything else in above table remains the same, but the also the inverted is 'ticked', then the same gen clock will look something like below. Also, there is a trick that can be done to define the below gen clock. i.e…….(sentence continued after below image)



Gen Clock Derivation

Generated Clock Properties	Values
name	gen_clock
source pin/port name	master_clock
edge	1,2,5
shifted edge	0,1,0 (time units in 'ns')
divide-by	-
multiply-by	-
inverted	-
duty-cycle	-

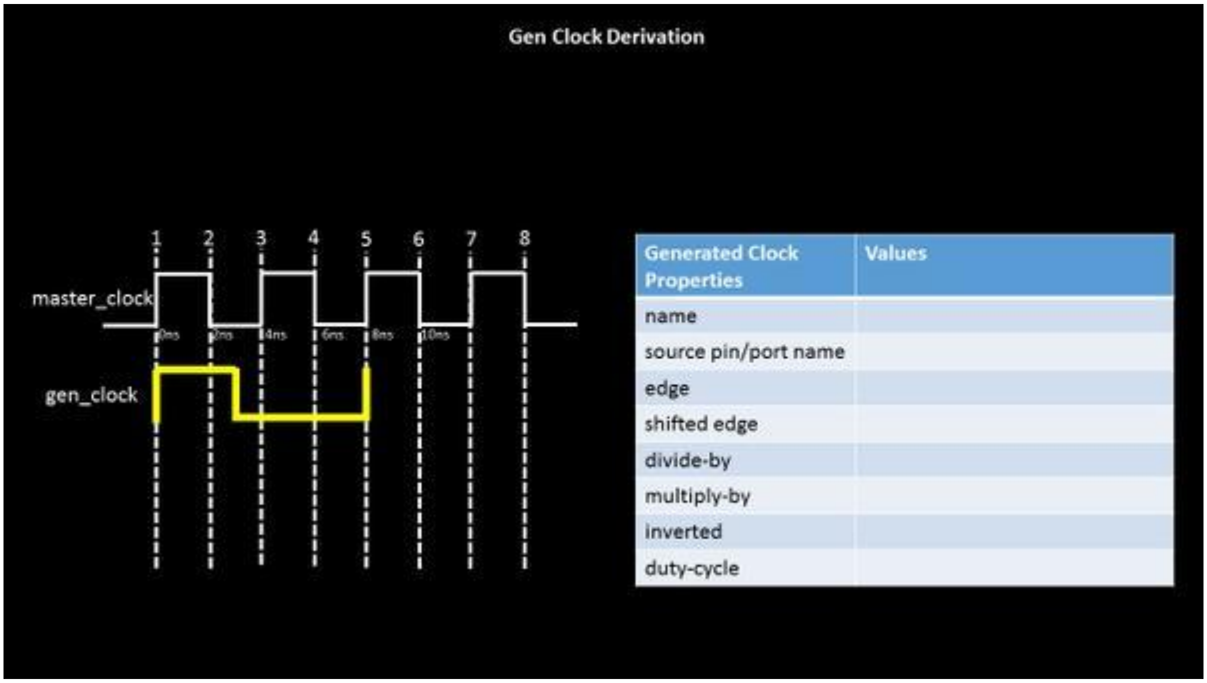
(…sentence continued from above image) the same gen_clock' can be defined using just the edges, and removing options of the table. And below is how the generated clock definition will look like for gen_clock'



Waveform Derivation

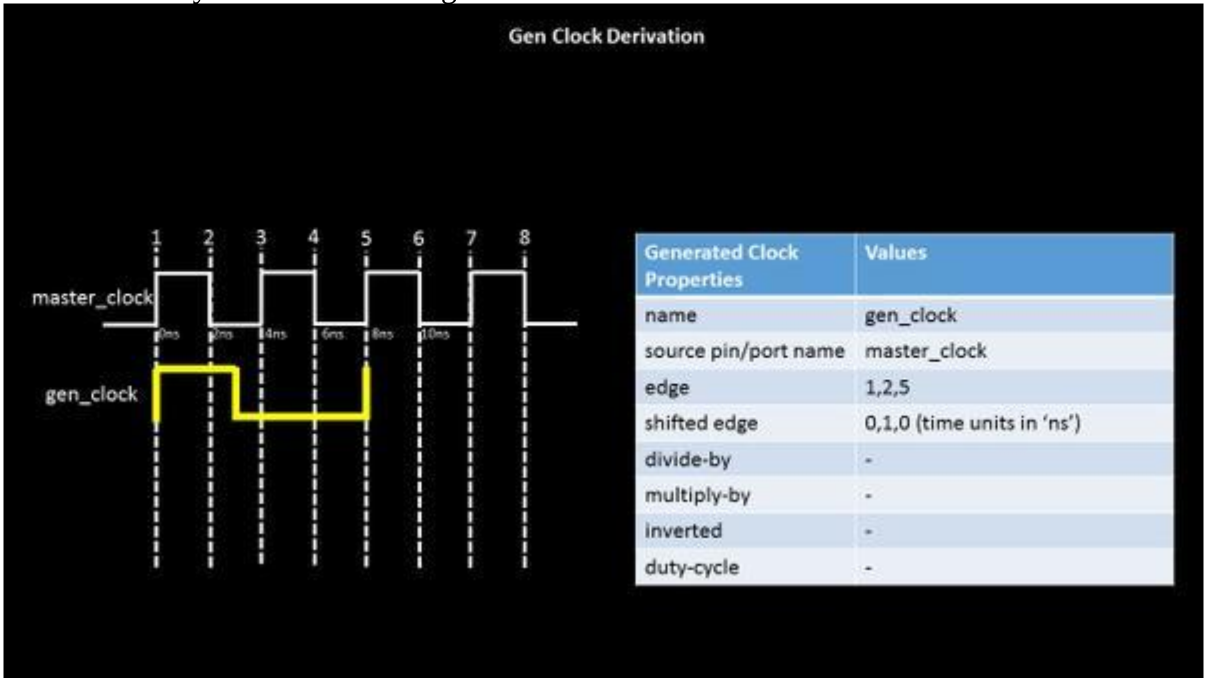
Generated Clock Properties	Values
name	gen_clock
source pin/port name	master_clock
edge	4,6,8
shifted edge	-
divide-by	-
multiply-by	
inverted	-
duty-cycle	-

And finally, let's consider a more complex table like below, where we have the generated clock edges coming in middle of clock edges at some 't' time units


Gen Clock Derivation

Generated Clock Properties	Values
name	
source pin/port name	
edge	
shifted edge	
divide-by	
multiply-by	
inverted	
duty-cycle	

This is a classic example of how ‘edge’ and ‘shifted edge’ option of generated clock can be used in conjunction. Let’s the first fall edge is in middle of 2ns and 4ns i.e. at 3ns
So this is how you will define the generated clocks


Gen Clock Derivation

Generated Clock Properties	Values
name	gen_clock
source pin/port name	master_clock
edge	1,2,5
shifted edge	0,1,0 (time units in ‘ns’)
divide-by	-
multiply-by	-
inverted	-
duty-cycle	-

You will say, the first clock edge of generated clock arrives at 1st edge of master clock, and shifted by 0ns from 1st edge (Hence you see the first element in ‘shifted edge’ at ‘0’). Next … (and quite important one) … the first fall edge of generated clock arrives at 2nd edge of master clock, ***but*** shifted by 1ns from 2nd edge of master clock. So 2nd edge of master clock is at 2ns and generated clock is shifted by 1ns (i.e. arrives at 3ns) from 2nd edge. Makes sense…If not, I would suggest you read the above sentence again. And the second rise edge of generated clock arrives at 5th edge of master clock. Hence you see the elements (0, 1, 0) in ns in the shifted edge row.

This marks the end of post related to ‘Generated clock and master clock – Lets make it simple’.