ISim Timing Diagrams

By:

Sebastian Hoyos

Quick VHDL

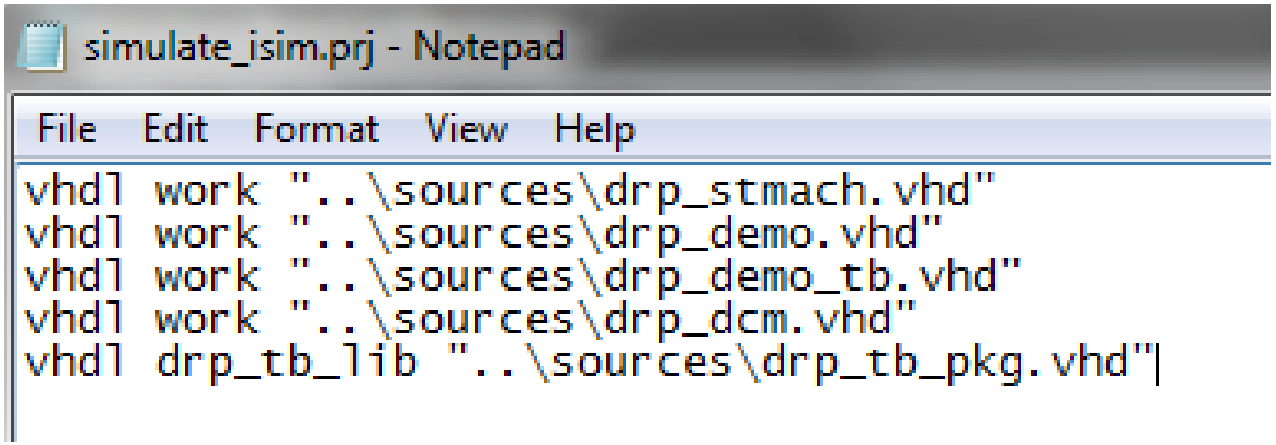
- Entity: Describes the interface
- Architecture: Actual implementation

AND Gate

```
-- (this is a VHDL comment)

-- import std_logic from the IEEE library
library IEEE;
use IEEE.std_logic_1164.all;

-- this is the entity
entity ANDGATE is
  port (
       IN1 : in std_logic;
       IN2 : in std_logic;
       OUT1: out std_logic);
end ANDGATE;

architecture RTL of ANDGATE is
begin

 OUT1 <= IN1 and IN2;

end RTL;
```

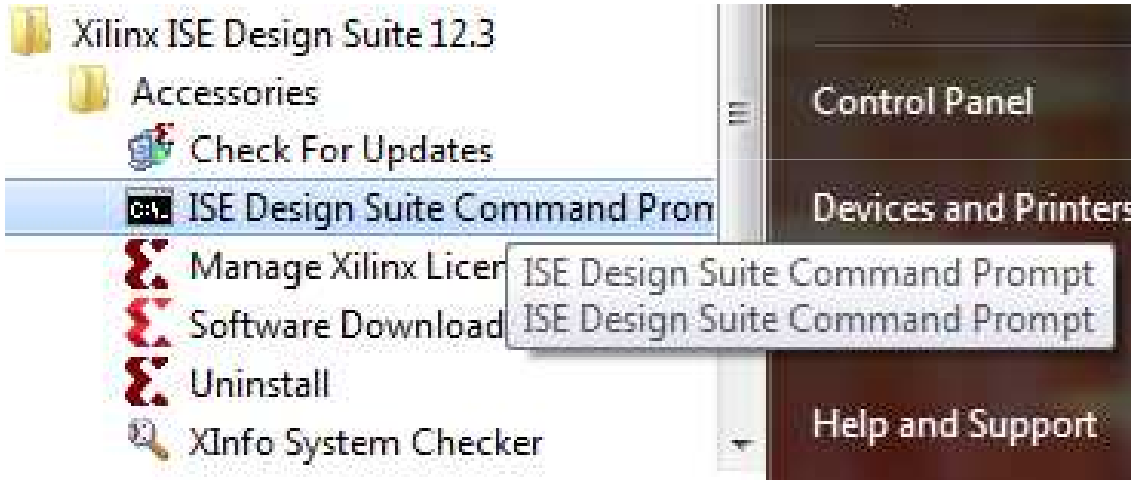
ISim Project File Syntax

- verilog|vhdl <library_name> {<file_name>.v|.vhd}
  - *verilog|vhdl* indicates that the source is a Verilog or VHDL file. Include either verilog or vhdl.
  - *<library_name>* indicates the library that a particular source on the given line should be compiled. “work” is the library.
  - *<file_name>* is the source file or files associated with the library.



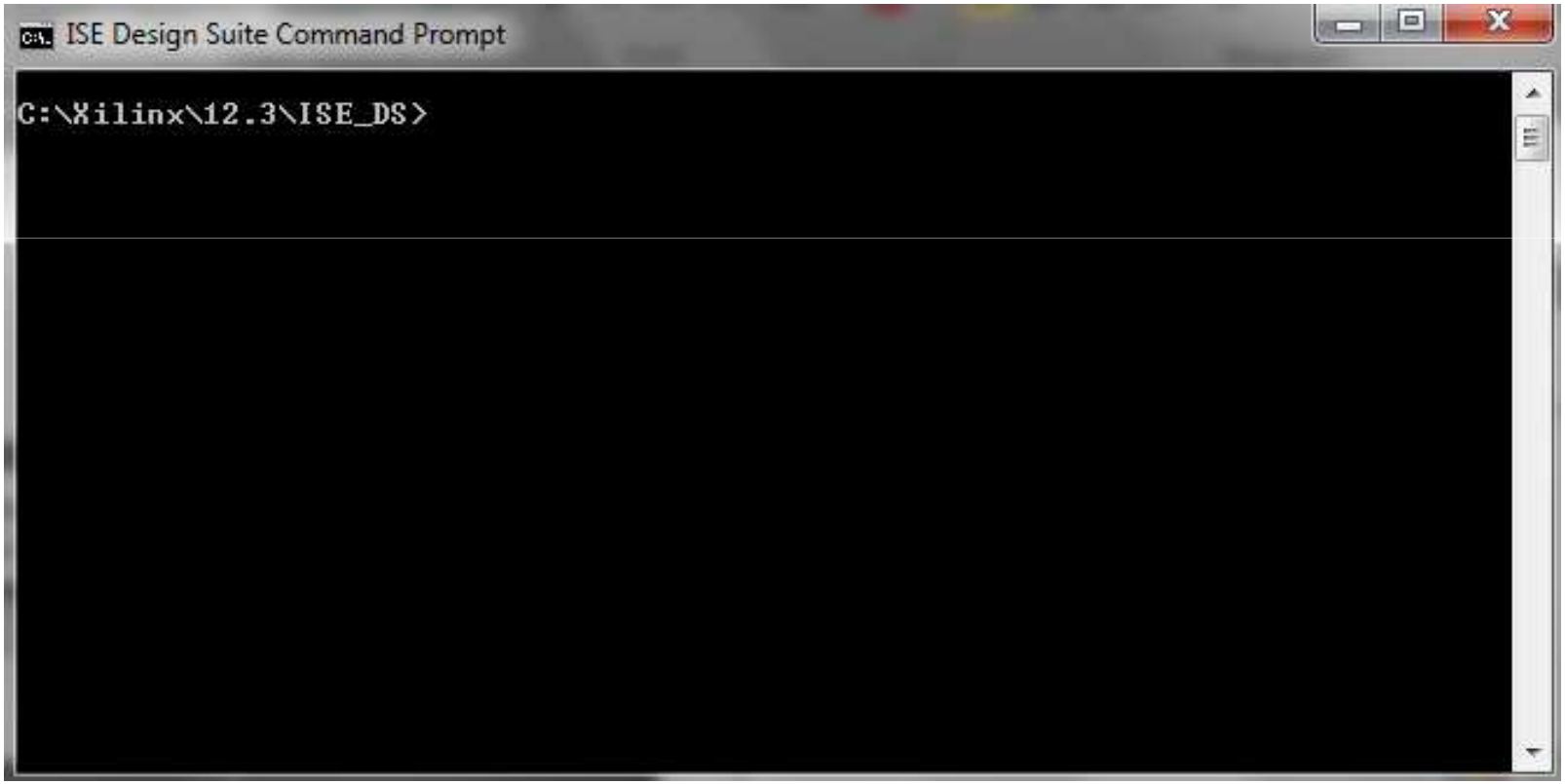
Building the Simulation .exe

- Open Command Prompt



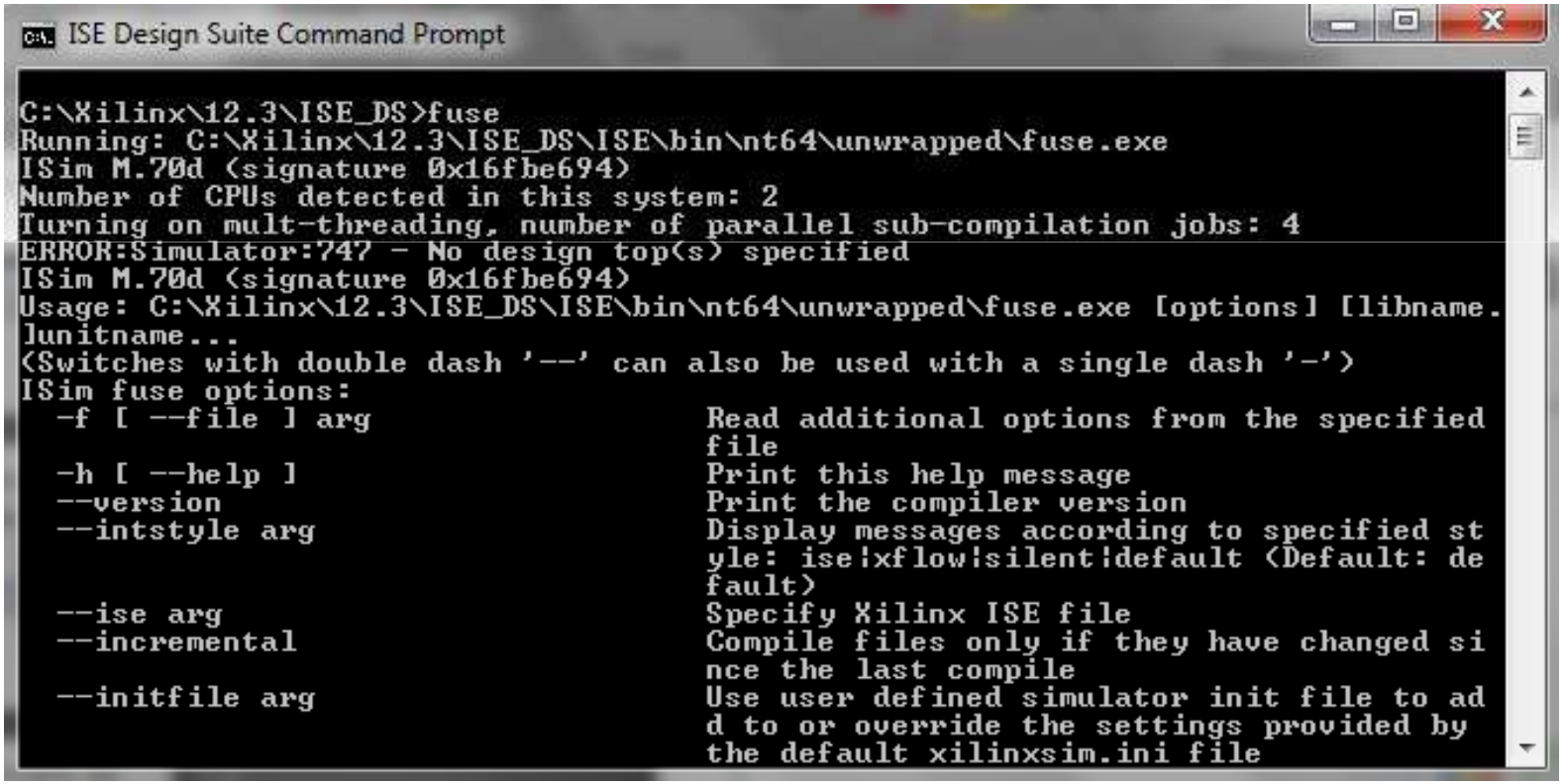
Building the Simulation .exe

- The following is opened:



Building the Simulation .exe

- Without any parameters:



Fuse.exe Syntax

- fuse –incremental –prj <project file> -o <simulation executable> <library.top_unit>
  - -incremental: requests fuse to compile only the files that have changed since the last compile
  - -prj: specifies an ISim project file to use for input
  - -o: specifies the name of the simulation executable output file

```
C:\Xilinx\12.3\ISE_DS>fuse.exe -incremental -prj simulate_isim.prj -o simulate_i
sim.exe work.drp_demo_tb
```

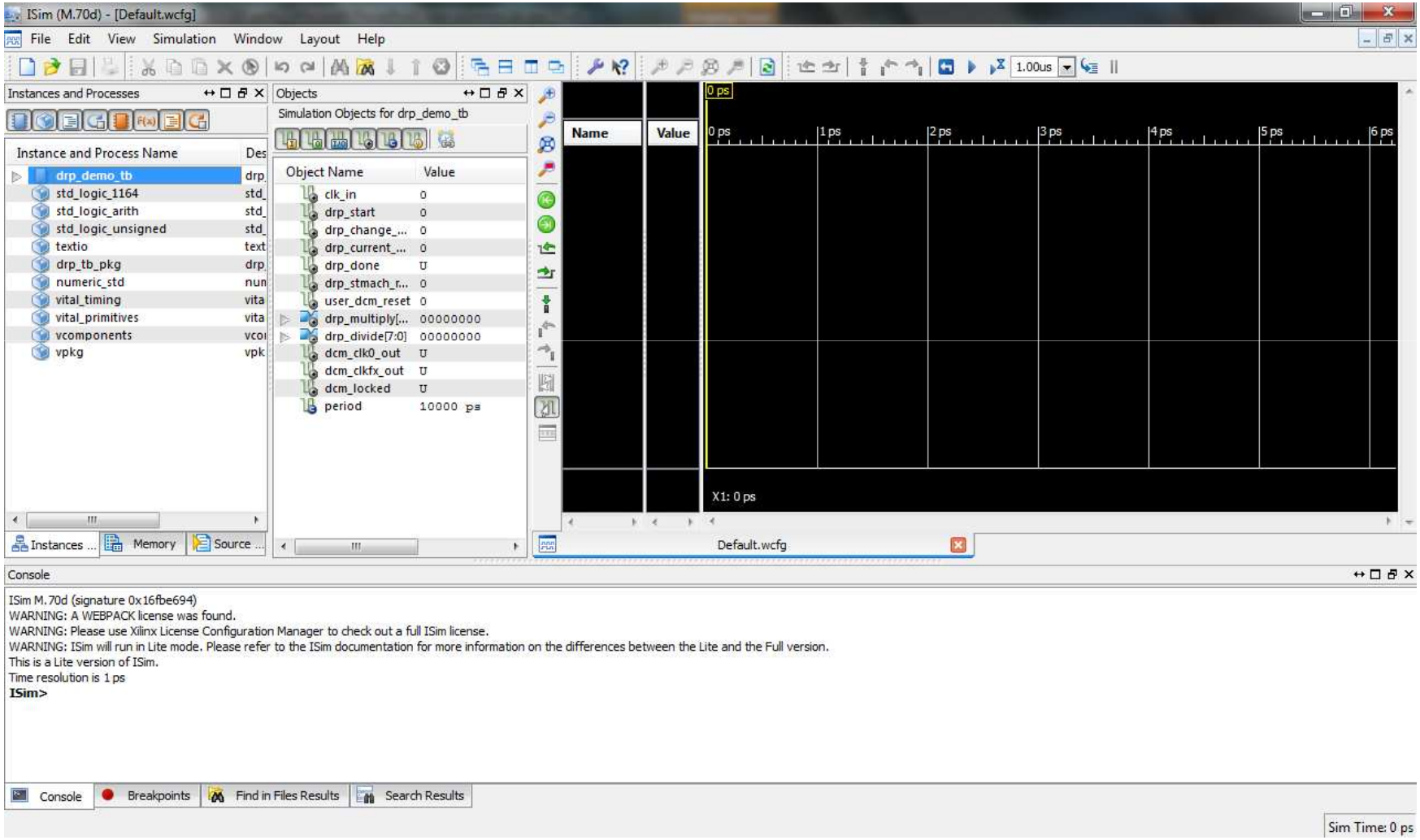
Running the Sim .exe

- Executable is created by fuse.exe
- This .exe calls the Simulator
- Double click does nothing, needs parameters
- We got to invoke the GUI mode

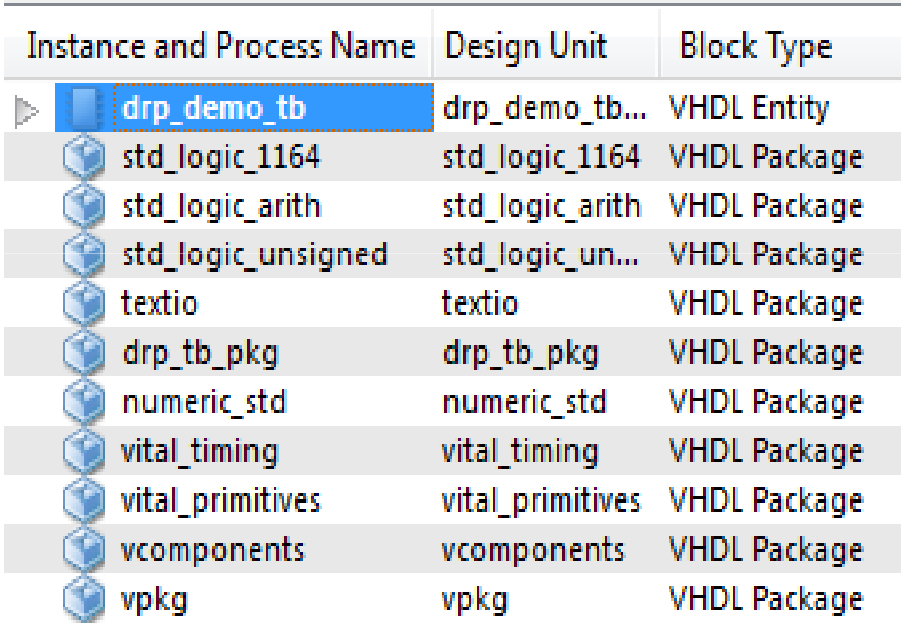
Simulation .exe Syntax

- Simulation_executable –gui -wdb <waveform_database_file>
  - -gui: launches ISim in GUI mode.
  - -wdb: specifies the file name of the simulation database output file.

Isim GUI


ISim (M.70d) - [Default.wcfg]
File Edit View Simulation Window Layout Help
Instances and Processes
Instance and Process Name
drp_demo_tb
std_logic_1164
std_logic_arith
std_logic_unsigned
textio
drp_tb_pkg
numeric_std
vital_timing
vital_primitives
vcomponents
vpkg
Objects
Simulation Objects for drp_demo_tb
Object Name
Value
clk_in
drp_start
drp_change_...
drp_current_...
drp_done
drp_stmach_r...
user_dcm_reset
drp_multiply[...
drp_divide[7:0]
dcm_clk0_out
dcm_clkfx_out
dcm_locked
period
10000 ps
Name
Value
0 ps
1 ps
2 ps
3 ps
4 ps
5 ps
6 ps
X1: 0 ps
Default.wcfg
Instances ...
Memory
Source ...
Console
ISim M.70d (signature 0x16fbe694)
WARNING: A WEBPACK license was found.
WARNING: Please use Xilinx License Configuration Manager to check out a full ISim license.
WARNING: ISim will run in Lite mode. Please refer to the ISim documentation for more information on the differences between the Lite and the Full version.
This is a Lite version of ISim.
Time resolution is 1 ps
ISim>
Console
Breakpoints
Find in Files Results
Search Results
Sim Time: 0 ps

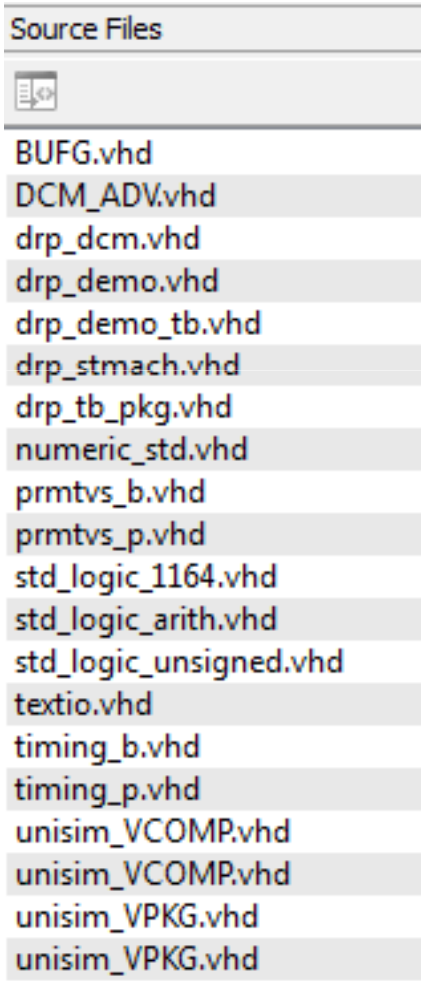
Instances and Processes



Instance and Process Name	Design Unit	Block Type
drp_demo_tb	drp_demo_tb...	VHDL Entity
std_logic_1164	std_logic_1164	VHDL Package
std_logic_arith	std_logic_arith	VHDL Package
std_logic_unsigned	std_logic_un...	VHDL Package
textio	textio	VHDL Package
drp_tb_pkg	drp_tb_pkg	VHDL Package
numeric_std	numeric_std	VHDL Package
vital_timing	vital_timing	VHDL Package
vital_primitives	vital_primitives	VHDL Package
vcomponents	vcomponents	VHDL Package
vpkg	vpkg	VHDL Package

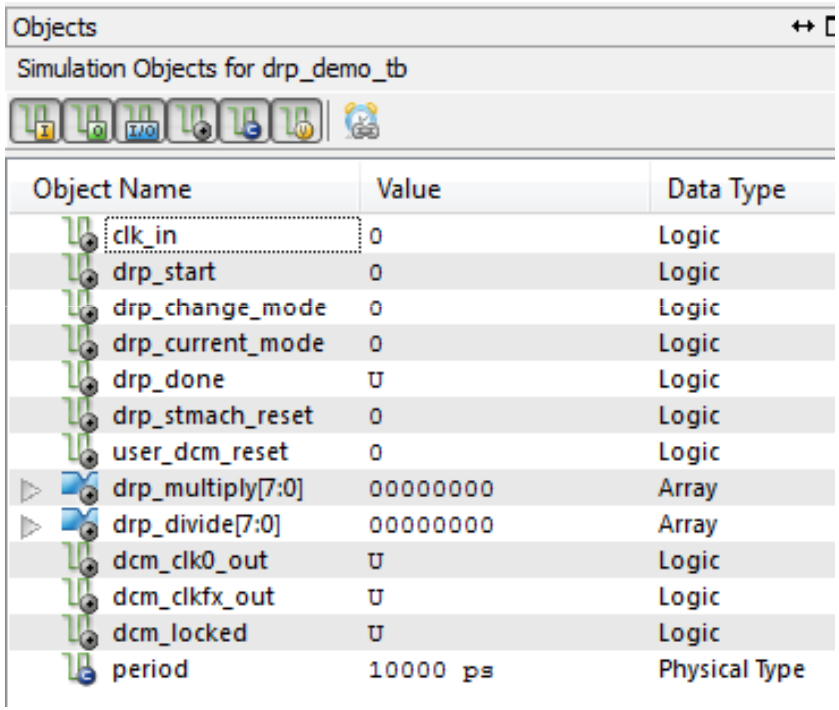
- Displays the block (instance and process) hierarchy associated with the wave configuration open in the Wave window.
- Instantiated and elaborated entities/modules are displayed in a tree structure, with entity components being ports, signals and other entities/modules.

Source Files Panel



- Displays the list of all the files associated with the design.
- The list of files is provided by the **fuse** command during design parsing and elaboration.
- The HDL source files are available for quick access to the read-only source code.

Objects Panel

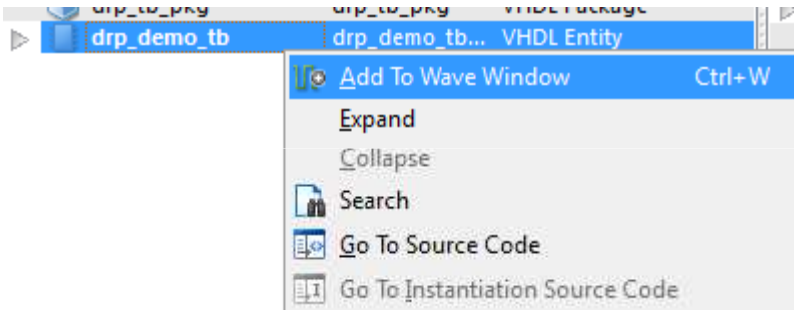


- Displays all ports and signals associated with the selected instances and processes in the Instances and Processes panel.
  - **Object Name** - Displays the name of the signal, accompanied by the symbol which represents the type of object it is.
  - **Value** - The value of the signals at the current simulation time or at the main cursor, as determined by the Sync Time toolbar button.
  - **Data Type** - Displays the data type of the corresponding simulation object, logic or an array.

Wave Window

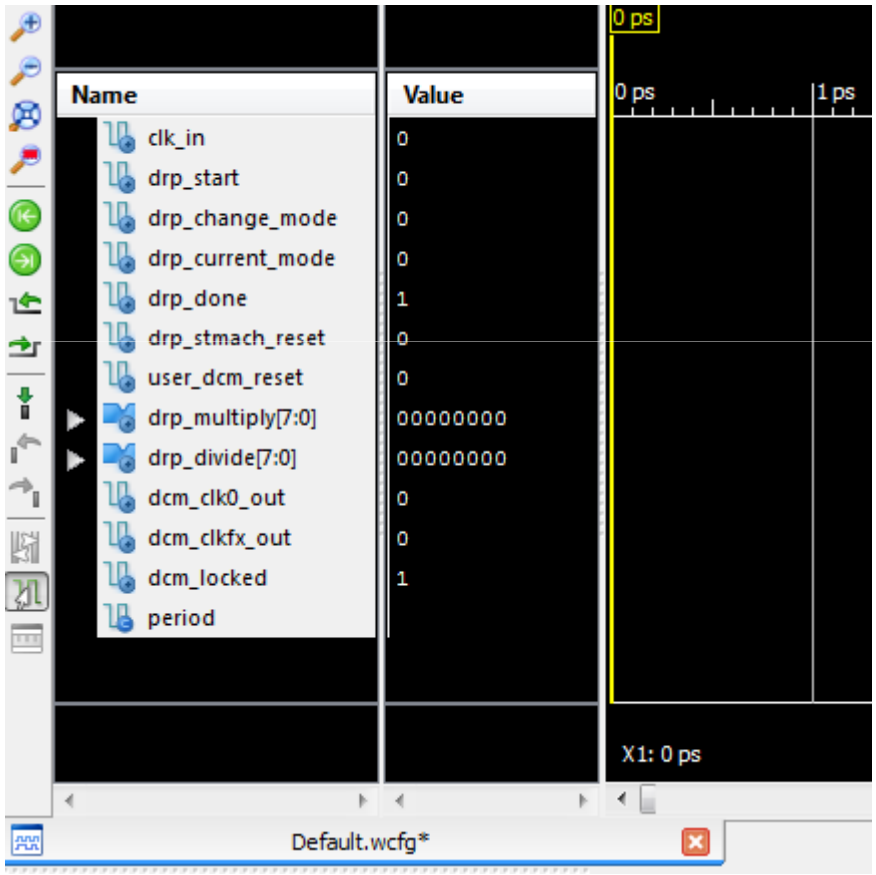
- Displays signals, buses, their properties and any other wave objects.
- The wave configuration is used to drive the simulation

Configuring the Wave Window



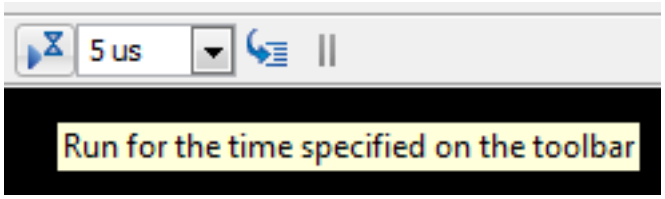
- Select the VHDL Entity and right click on it. The entity is the one with the ports, buses and signals.
- Choose “Add to Wave Window”

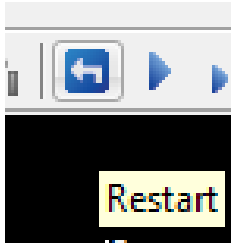
Wave Window with Signals



- Signals, ports and buses added.
- The values are the one listed in the Objects Panel

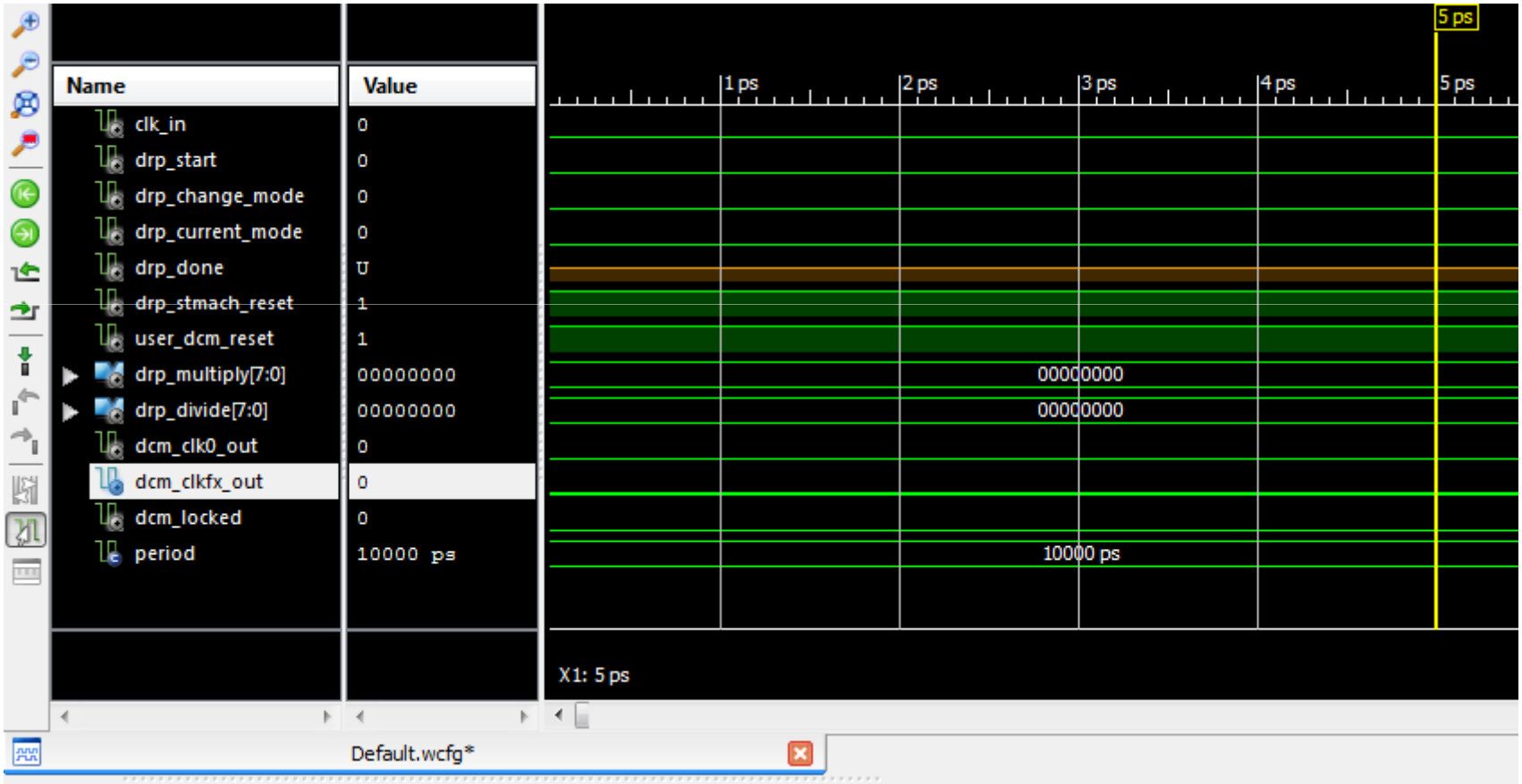
Running the Wave Sim for a Specified Time



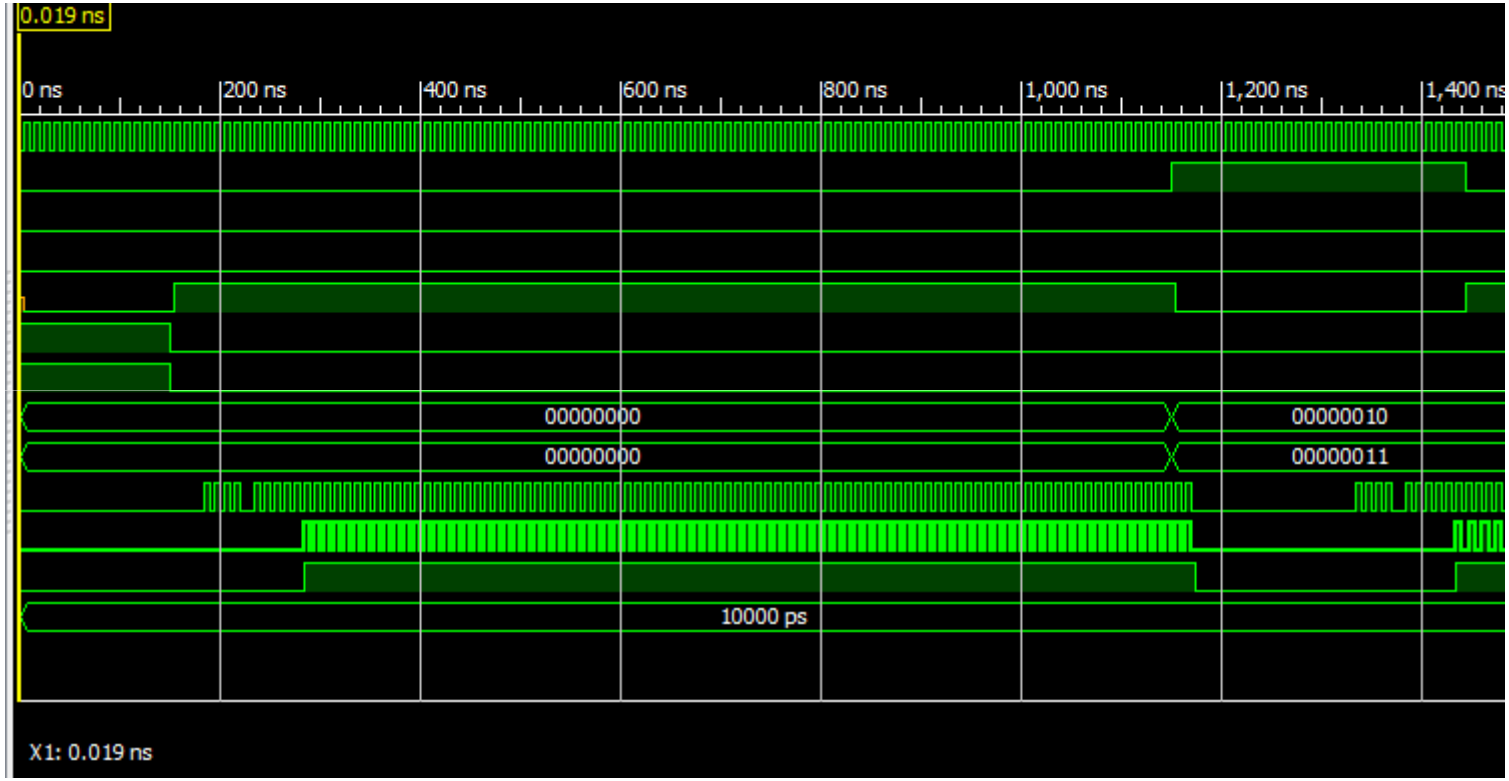


- Type **5 us** in the Simulation Time field and press Enter or click the “Run For” button
- Type **run 5 us** in the Console prompt
- If Simulation doesn’t start, press the “Restart” option

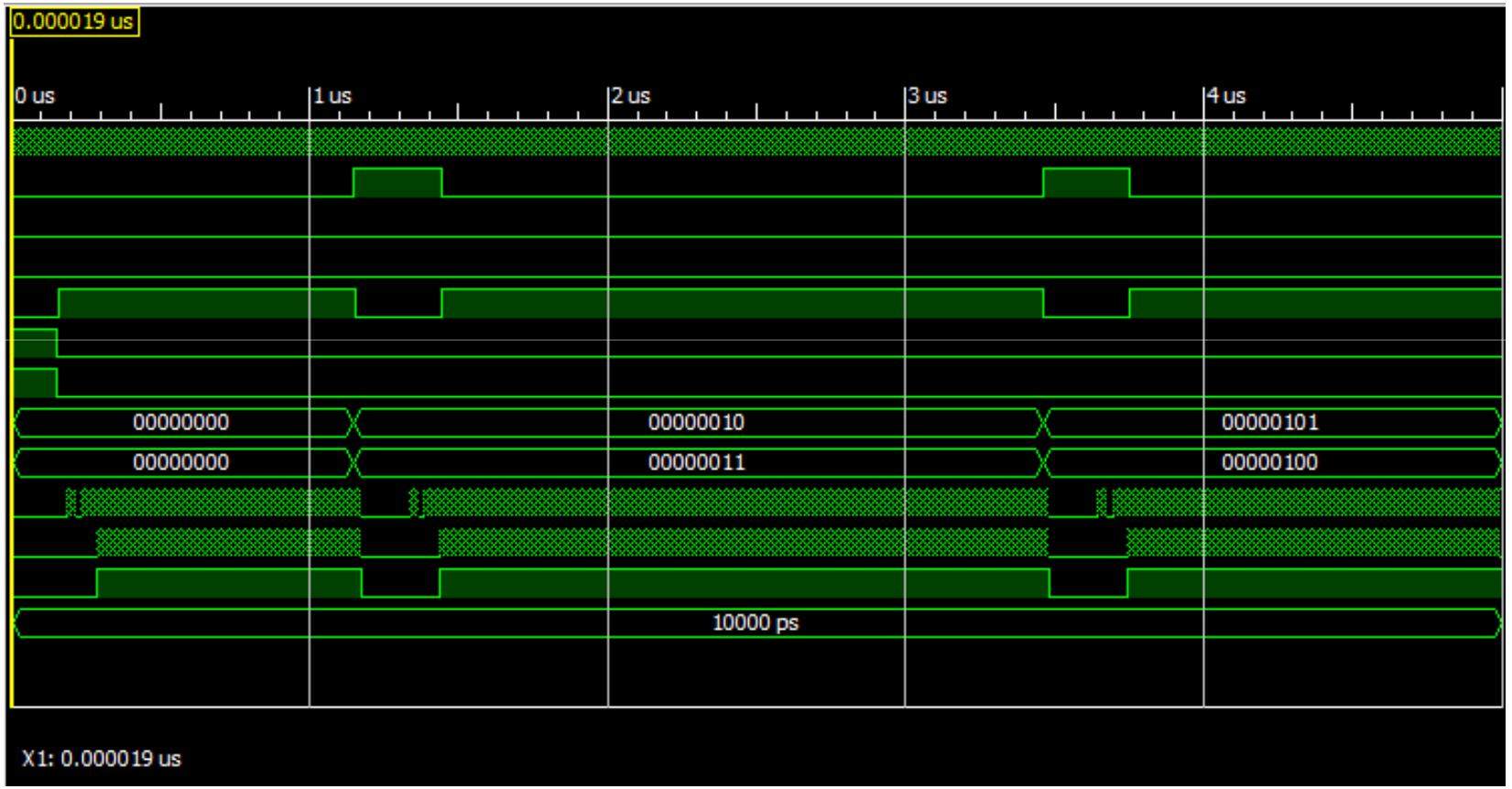
Wave Sim Results



Zoomed Out Wave Sim Results



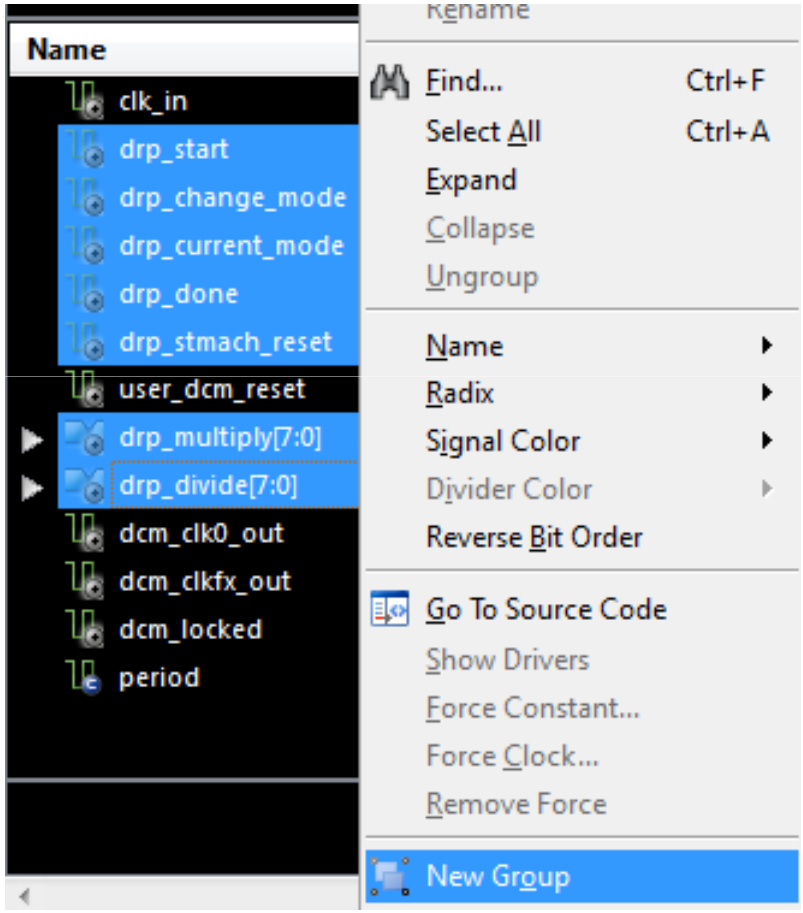
Full Simulation



Now… lets make it pretty (unlike this PowerPoint)

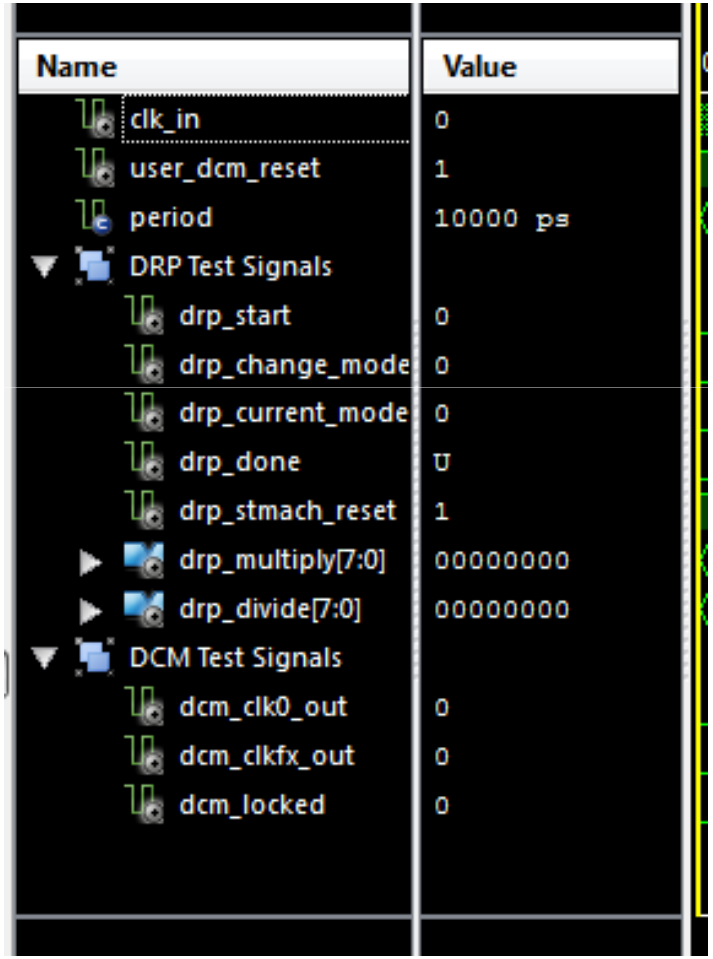
Groups

- Hold **Ctrl** and select the signals in the wave window
- Right click on any of them
- Select “New Group” in the menu
- Name the group



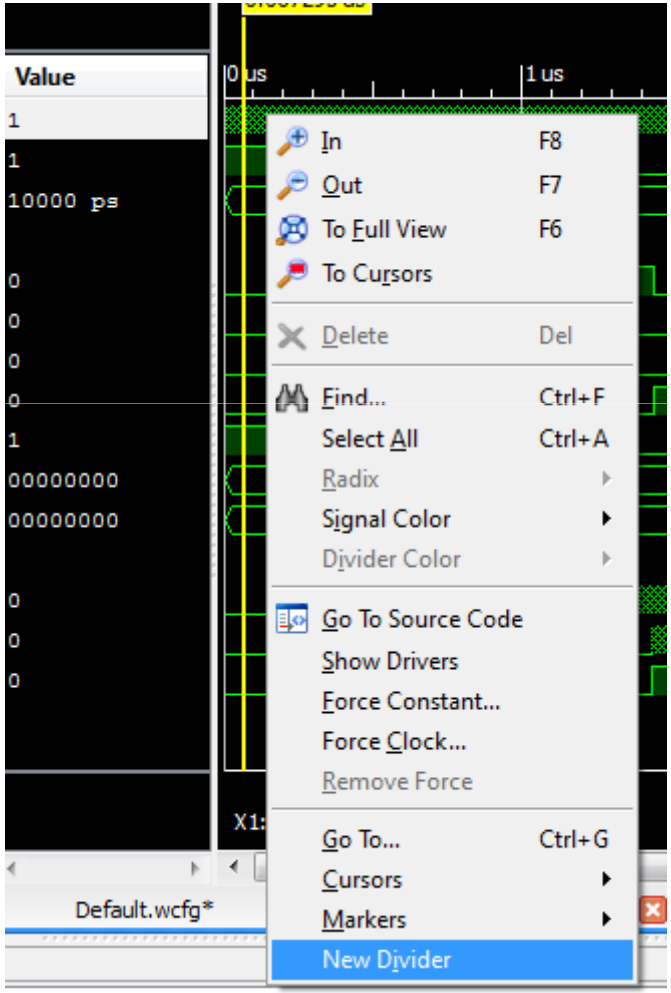
Groups

- End Result ->
- You have to expand the group to see the signals



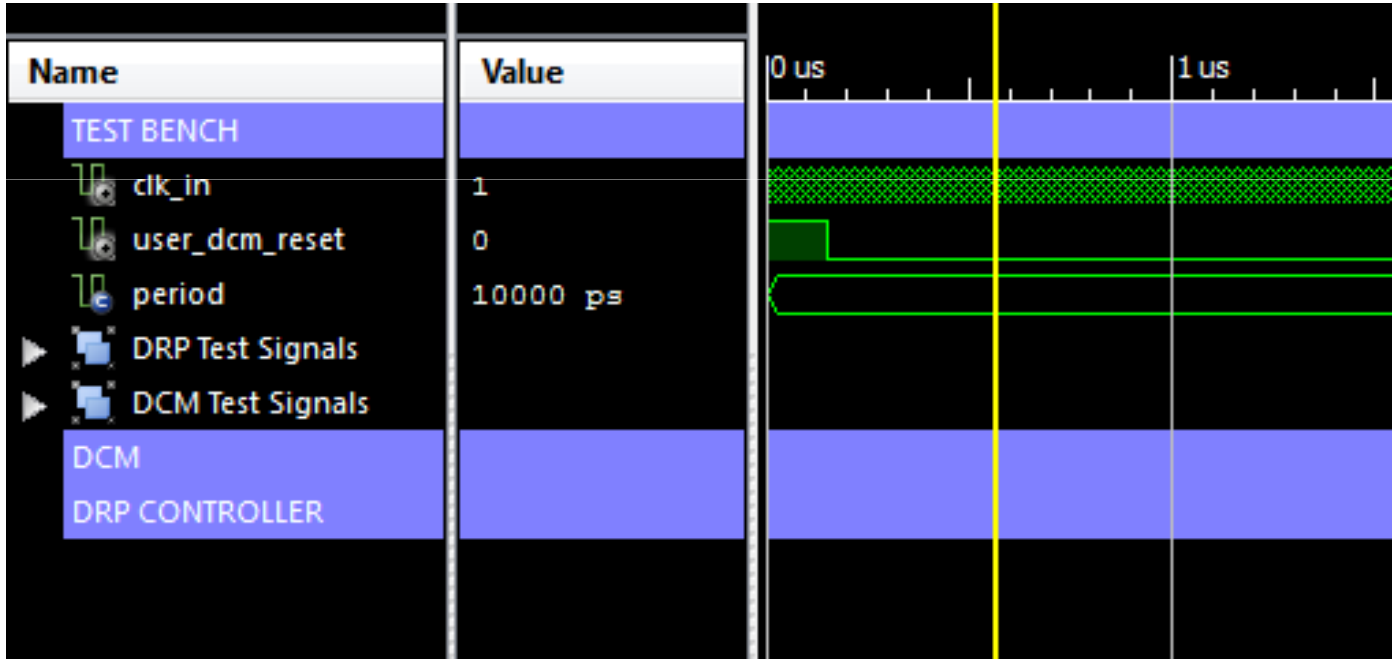
Dividers

- Right Click on any part of the Wave Window
- Select **New Divider**
- Name it
- Move it to where you want it to be
- Rename them by double clicking on them or using **F2**



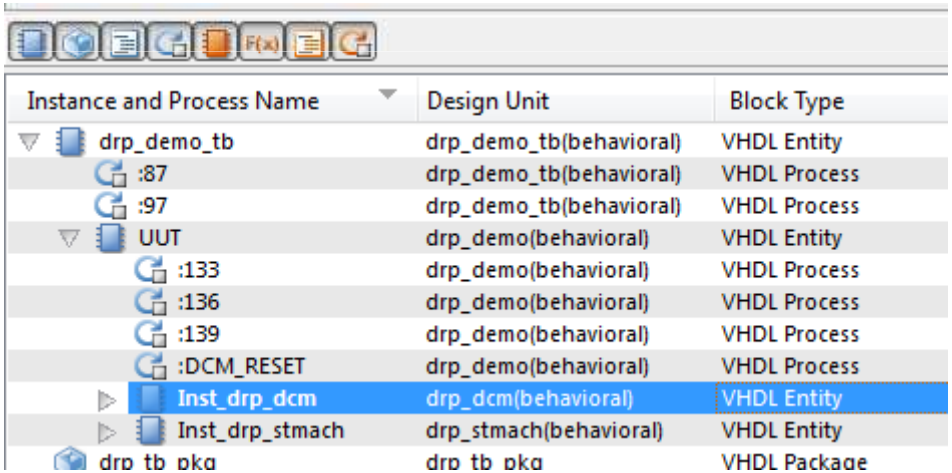
Dividers

- End Result:



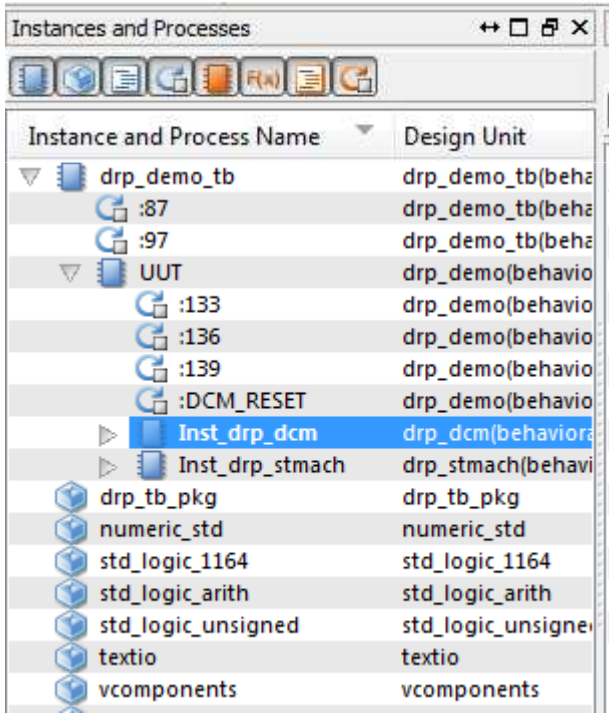
Adding Signals from Sub-Modules

- Allows you to study the interaction between the sub-modules and the test bench test signals
- This interacts with an entity that is in a deeper level
- Expand the test bench entity until you see the **instanced** entities

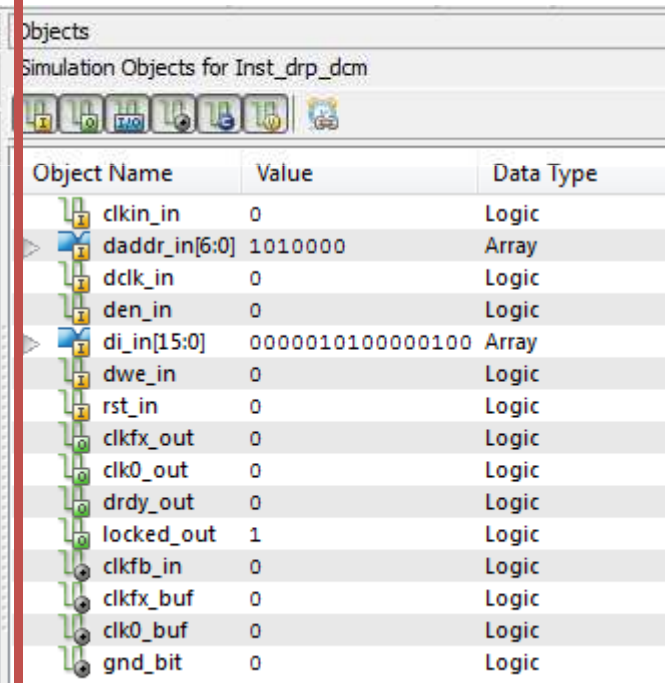


Objects from the instanced entity

- You'll get the following instances:

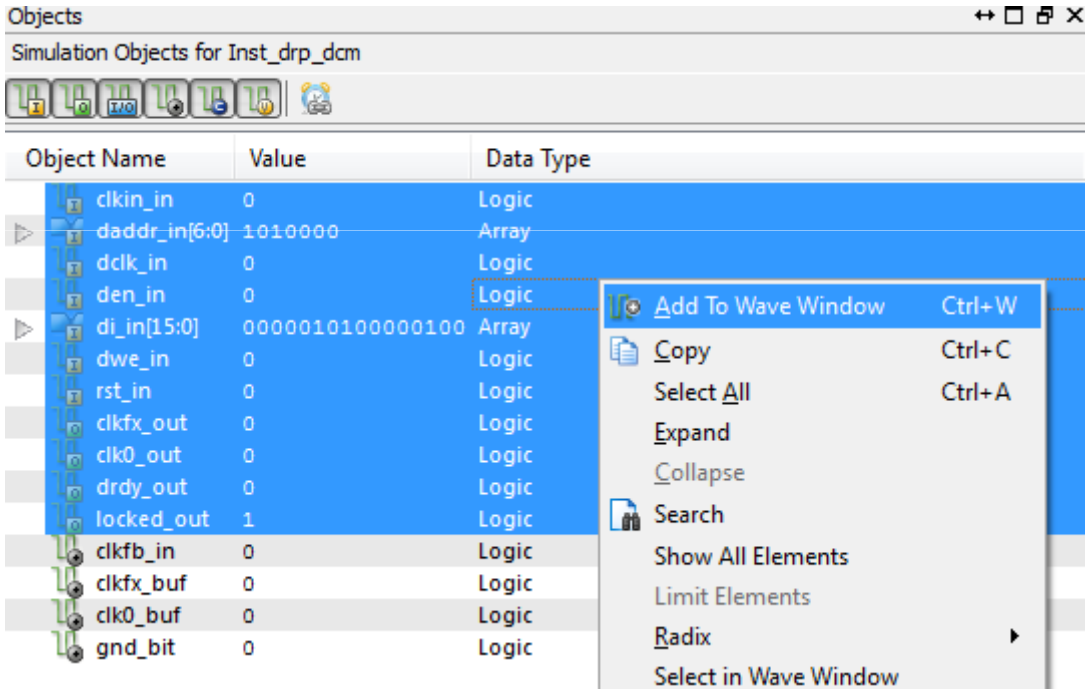


- You'll get the following signals:

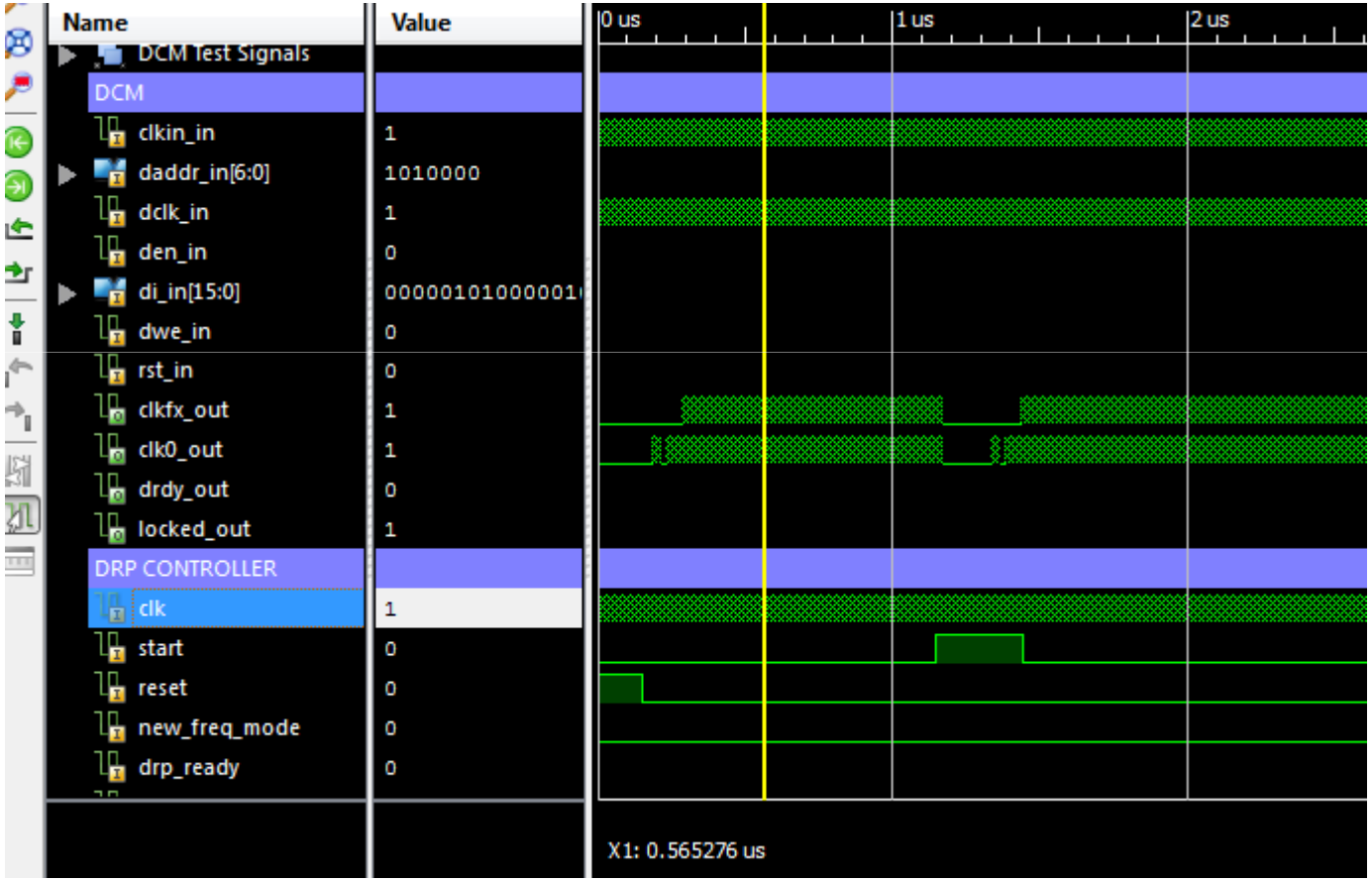


Adding the Signals

- Select the signals
- Right Click
- Add to Wave Window

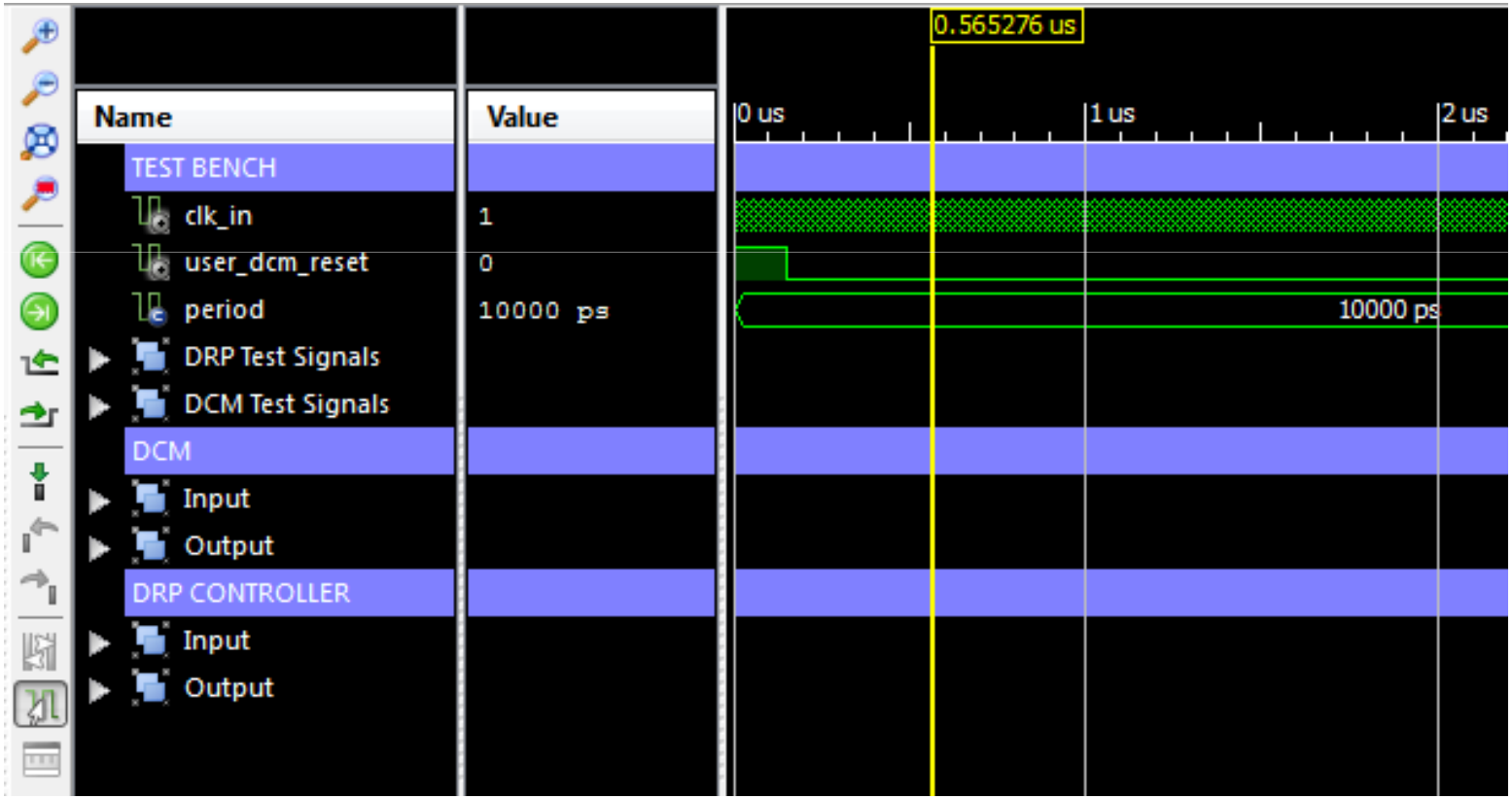


Added the Signals



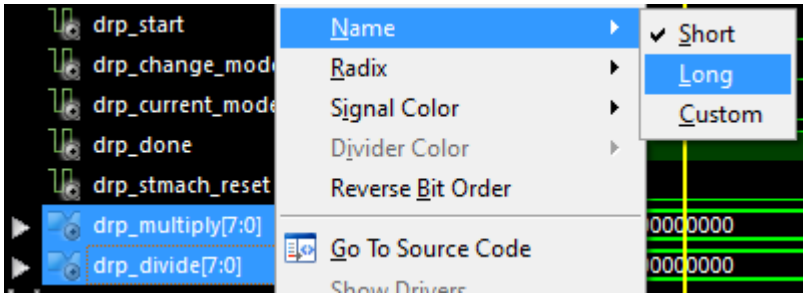
Organized by Groups

- The signals are organized in groups

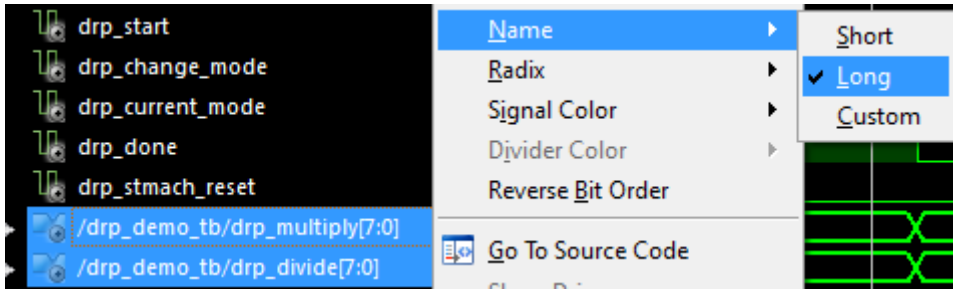


Naming the Signals

- Name:



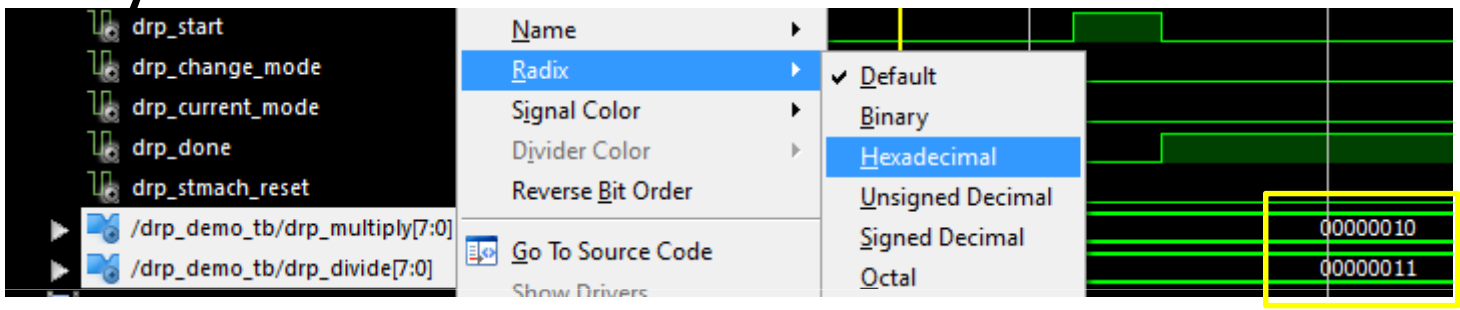
  – Select **Long**



- Use **Custom** for setting a desired name

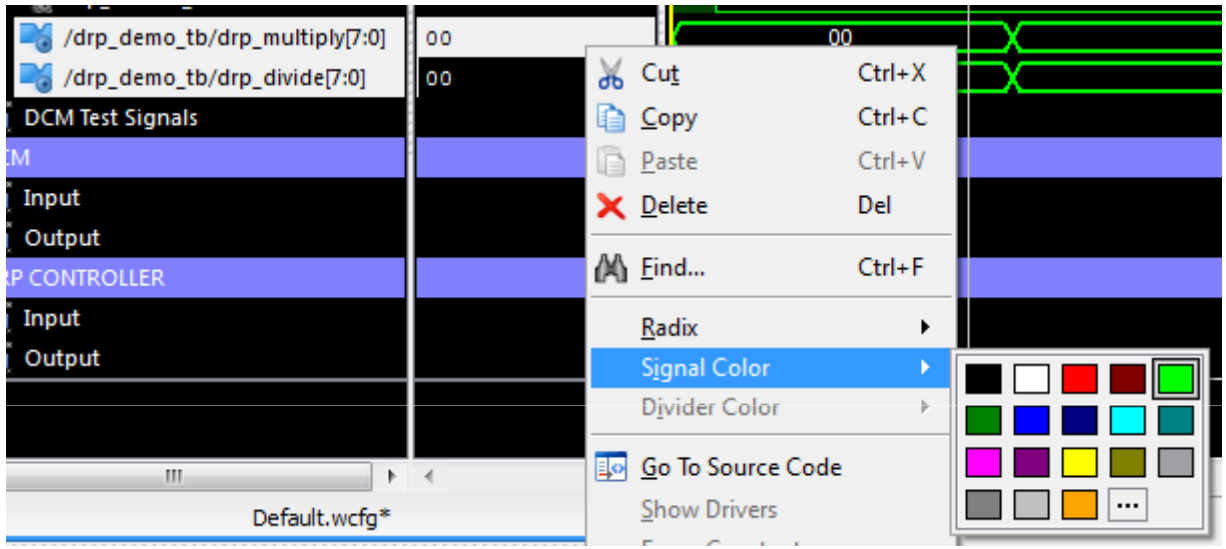
Radix

- Binary:



- Select Hexadecimal:

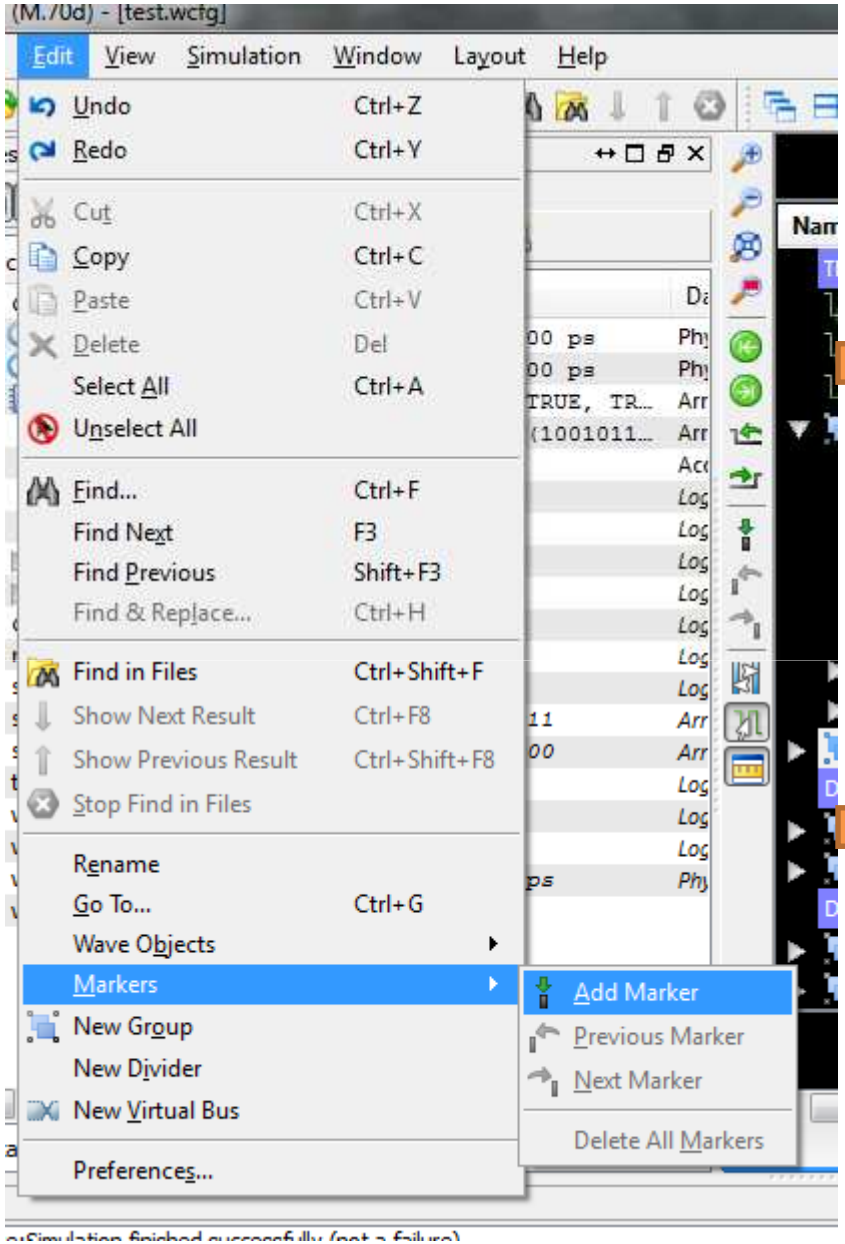
Crayola Effect

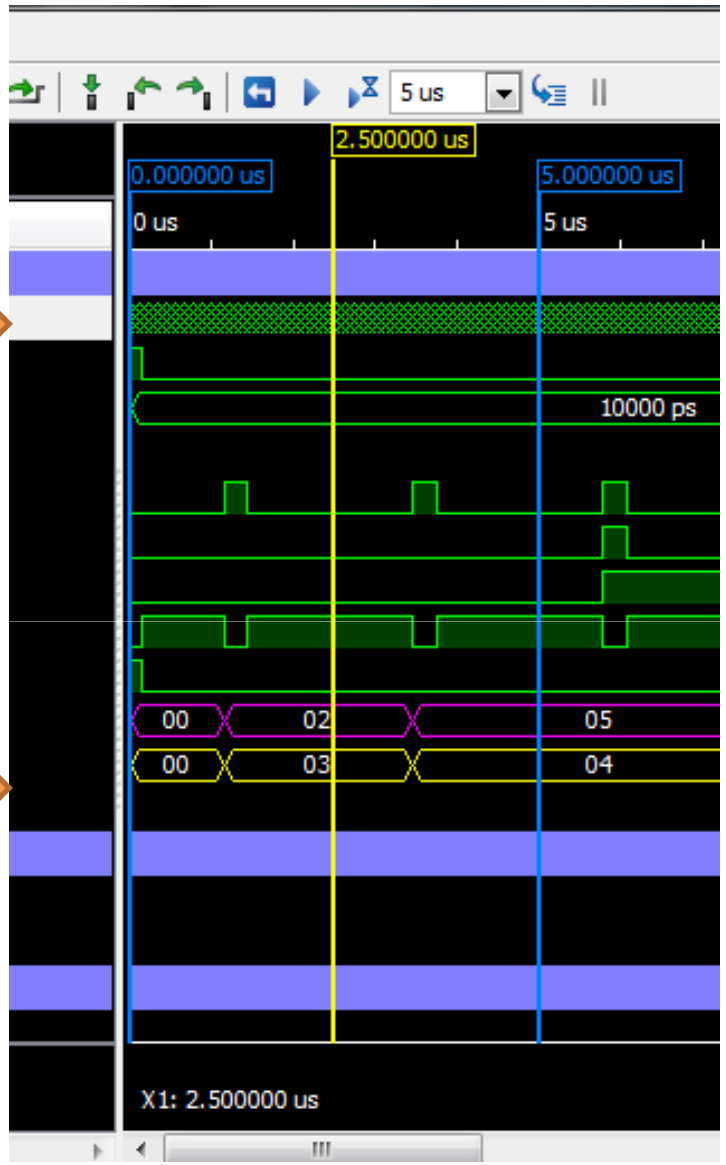


- Changing the Color:

Markers

- Useful to identify times in the diagram
- Most used to determine when a test starts
- The tests are defined in the test bench file

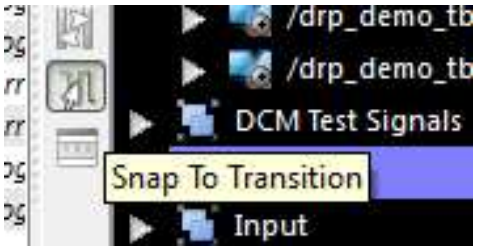

(M.70d) - [test.wcfg]
Edit
View
Simulation
Window
Layout
Help
Undo Ctrl+Z
Redo Ctrl+Y
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Select All Ctrl+A
Unselect All
Find... Ctrl+F
Find Next F3
Find Previous Shift+F3
Find & Replace... Ctrl+H
Find in Files Ctrl+Shift+F
Show Next Result Ctrl+F8
Show Previous Result Ctrl+Shift+F8
Stop Find in Files
Rename
Go To... Ctrl+G
Wave Objects
Markers
New Group
New Divider
New Virtual Bus
Preferences...
Add Marker
Previous Marker
Next Marker
Delete All Markers


5 us
2.500000 us
0.000000 us
5.000000 us
0 us
5 us
10000 ps
00
02
05
00
03
04
X1: 2.500000 us

Measuring Time

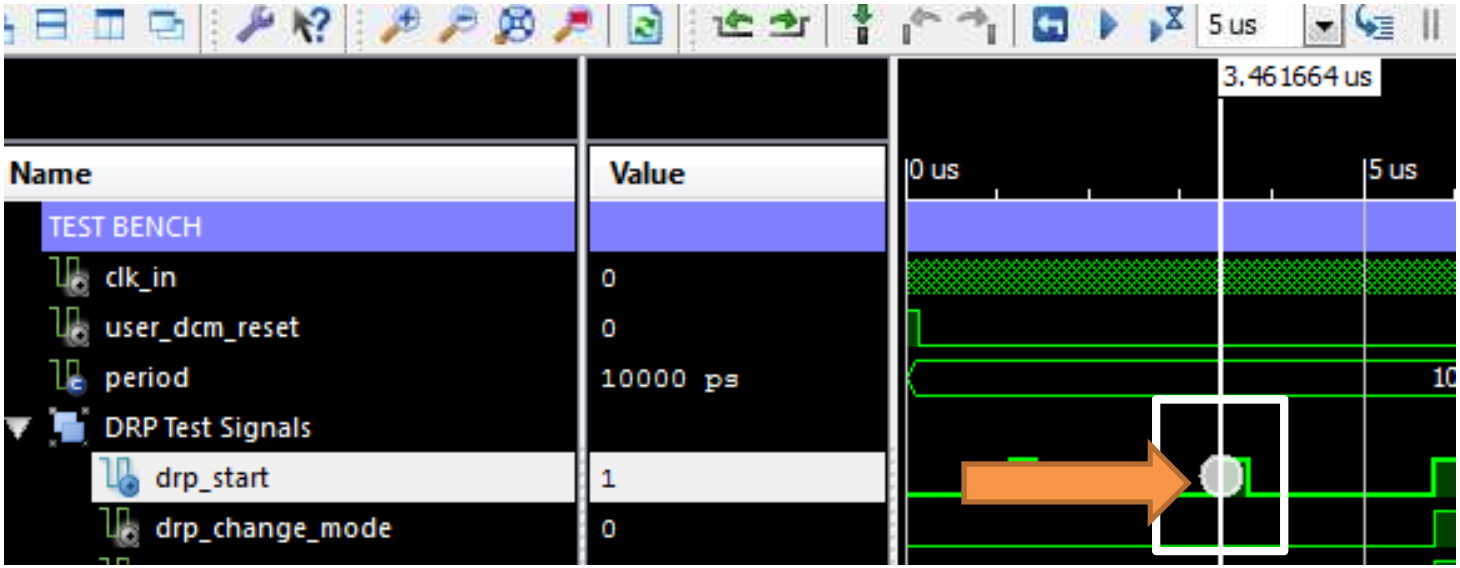
- Measuring time between 2 endpoints
- Useful for calculating the frequency of a signal

- 1st: Click on the Snap to Transition toggle button



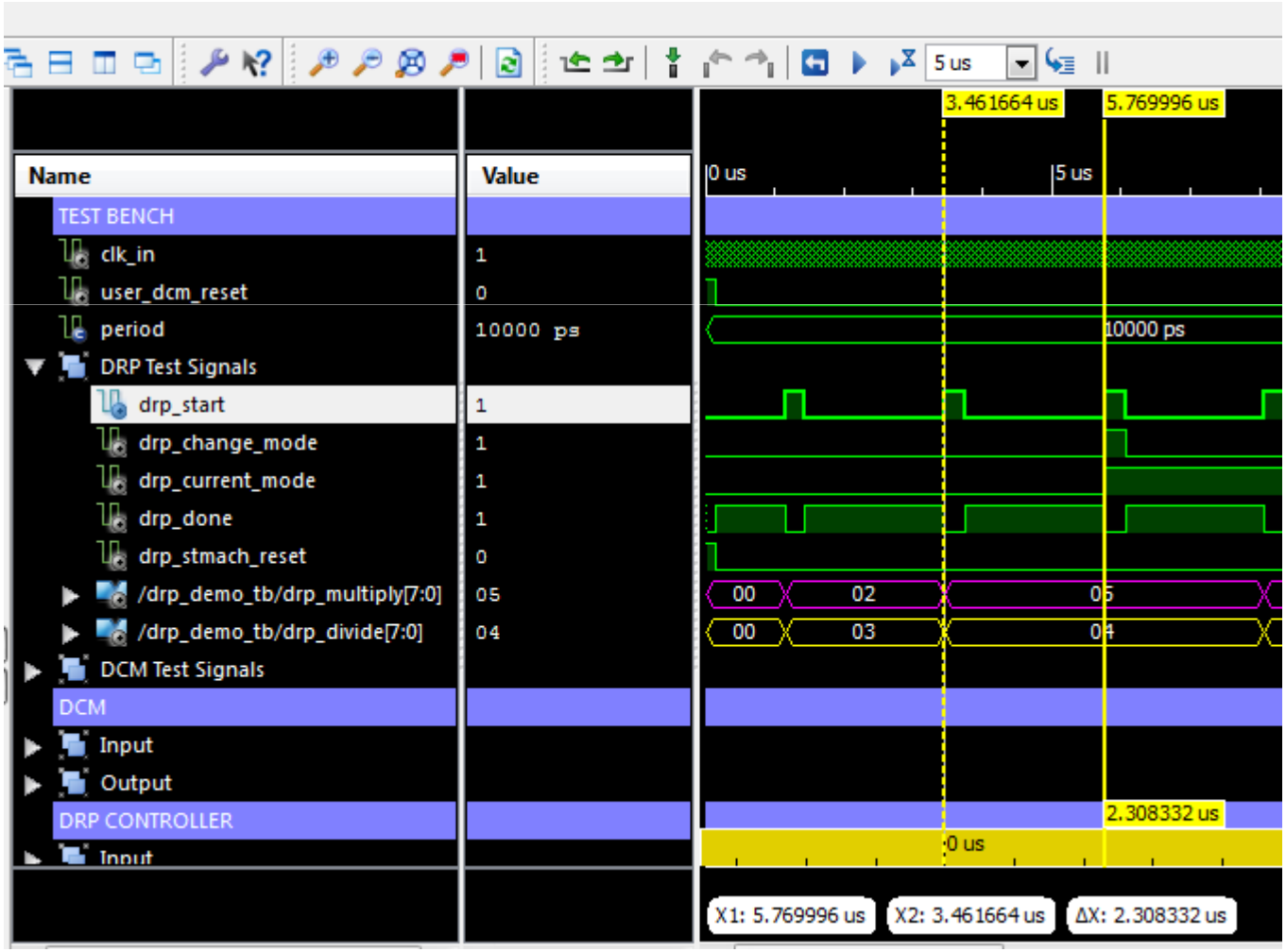
Measuring Time

- Drag the cursor over a signal until it snaps on a rising edge



Measuring Time

- Repeat for a second rising edge



Measuring Time

- The time difference is

$$\Delta X = 2.308332\ \mu s$$

$$f = \frac{1}{\Delta X} = 433.213\ KHz$$