

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;

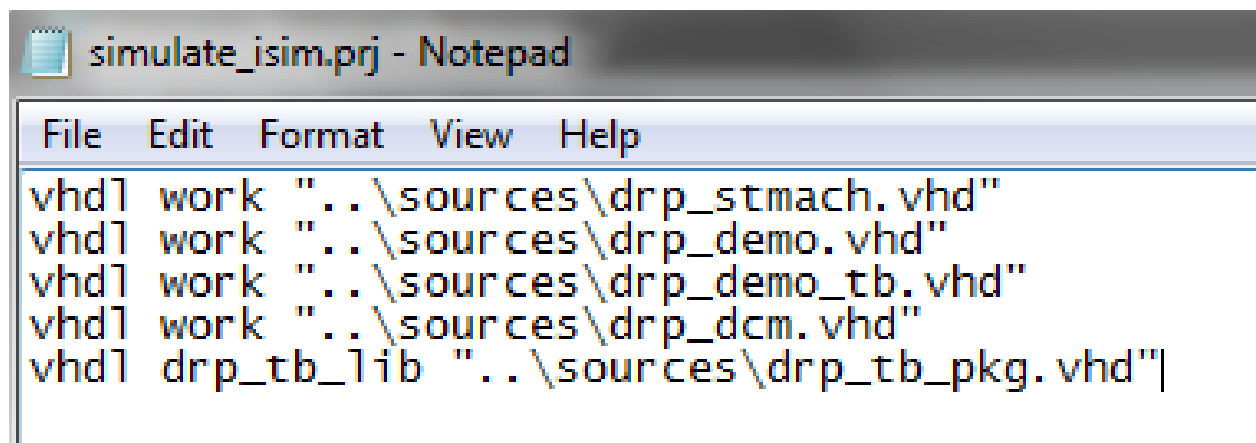
architecture RTL of ANDGATE is
begin

  OUT1 <= IN1 and IN2;

end RTL;
```

ISim Project File Syntax

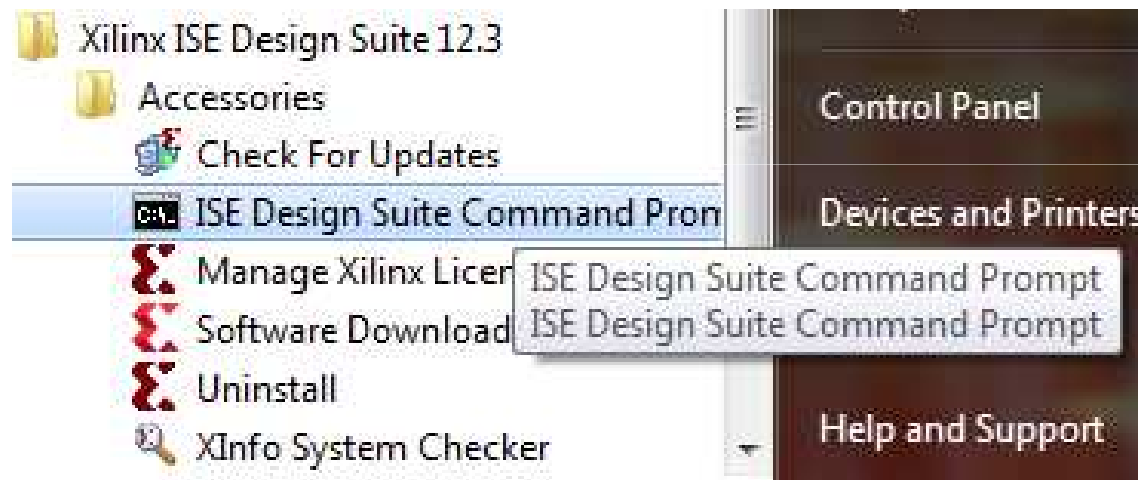
- `verilog|vhdl <library_name> {<file_name>.v|.vhd}`
 - *verilog/vhdl* indicates that the source is a Verilog or VHDL file. Include either verilog or vhdl.
 - *<library_name>* indicates the library that a particular source on the given line should be compiled. “work” is the library.
 - *<file_name>*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



```
simulate_isim.prj - Notepad
File Edit Format View Help
vhdl work "..\sources\drp_stmach.vhd"
vhdl work "..\sources\drp_demo.vhd"
vhdl work "..\sources\drp_demo_tb.vhd"
vhdl work "..\sources\drp_dcm.vhd"
vhdl drp_tb_lib "..\sources\drp_tb_pkg.vhd"
```

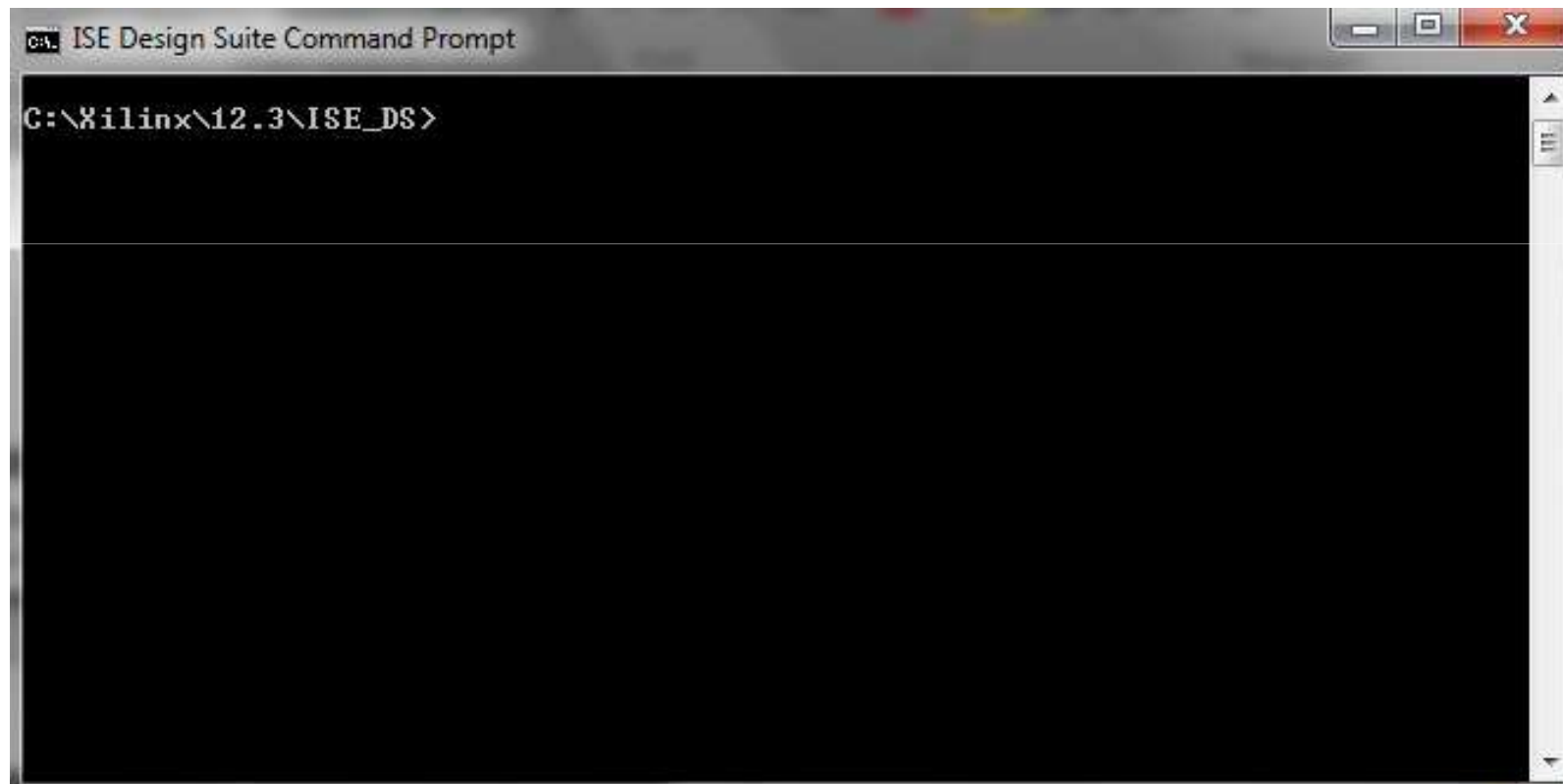
Building the Simulation .exe

- Open Command Prompt



Building the Simulation .exe

- The following is opened:



```
C:\Xilinx\12.3\ISE_DS>
```

The image shows a screenshot of a Windows command prompt window titled "ISE Design Suite Command Prompt". The window has a dark background and a light-colored title bar with standard Windows window controls (minimize, maximize, close). The current directory path is displayed as "C:\Xilinx\12.3\ISE_DS>".

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.

```
C:\Users\Sebastian\Documents\Xilinx\completed\scripts>simulate_isim.exe -gui -wdb simulate_isim.wdb_
```

Isim GUI

The screenshot displays the ISim GUI interface. The main window is titled "ISim (M.70d) - [Default.wcfg]". The menu bar includes File, Edit, View, Simulation, Window, Layout, and Help. The toolbar contains various icons for file operations and simulation control. The main workspace is divided into several panes:

- Instances and Processes:** A tree view showing the simulation hierarchy. The root is "drp_demo_tb", which contains several sub-entities like "std_logic_1164", "std_logic_arith", "textio", "drp_tb_pkg", "numeric_std", "vital_timing", "vital_primitives", "vcomponents", and "vpkg".
- Objects:** A table listing simulation objects for "drp_demo_tb".
- Waveform Viewer:** A large area showing a signal waveform. The x-axis represents time in picoseconds (ps), ranging from 0 ps to 6 ps. The y-axis represents the value of the signal. The signal is currently at 0 ps.
- Console:** A text area at the bottom showing simulation output. It includes warnings about the license and the version of ISim.

Object Name	Value
clk_in	0
drp_start	0
drp_change_...	0
drp_current_...	0
drp_done	U
drp_stmach_r...	0
user_dcm_reset	0
drp_multiply[...	00000000
drp_divide[7:0]	00000000
dcm_clk0_out	U
dcm_clkfx_out	U
dcm_locked	U
period	10000 ps

Console Output:

```
ISim M.70d (signature 0x16fbc694)
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>
```

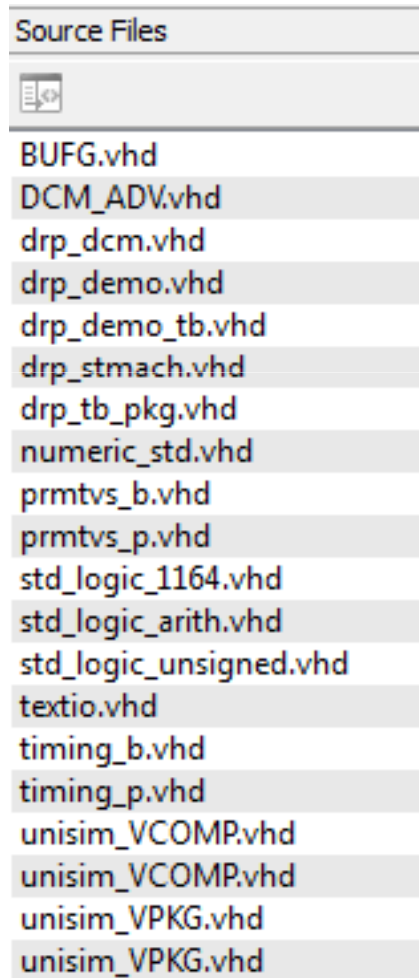
Simulation Time: 0 ps

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

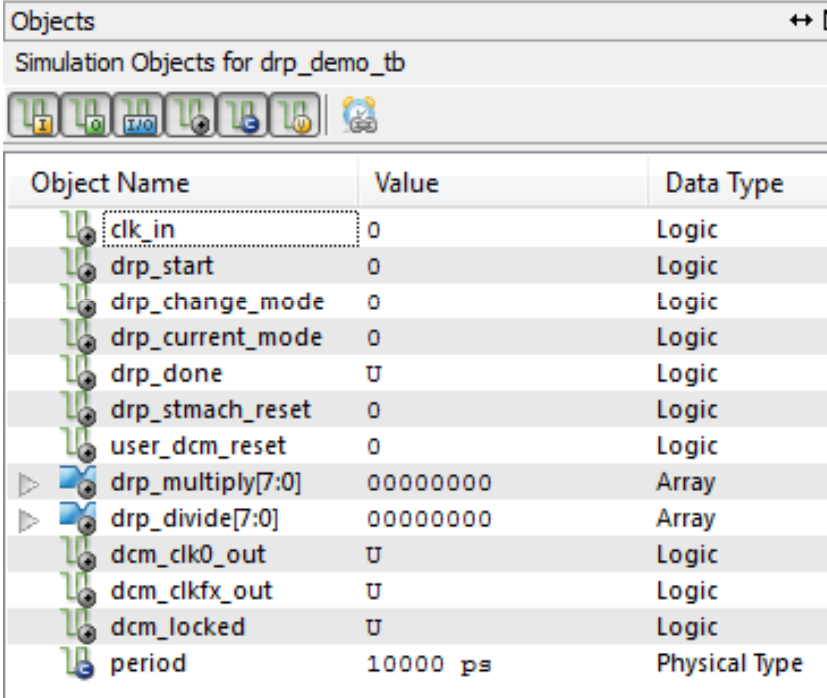
- Displays the block (instance and process) hierarchy associated with the wave configuration open in the Wave window.
- Instantiated and elaborated entities/modules are displayed in a tree structure, with entity components being ports, signals and other entities/modules.

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.

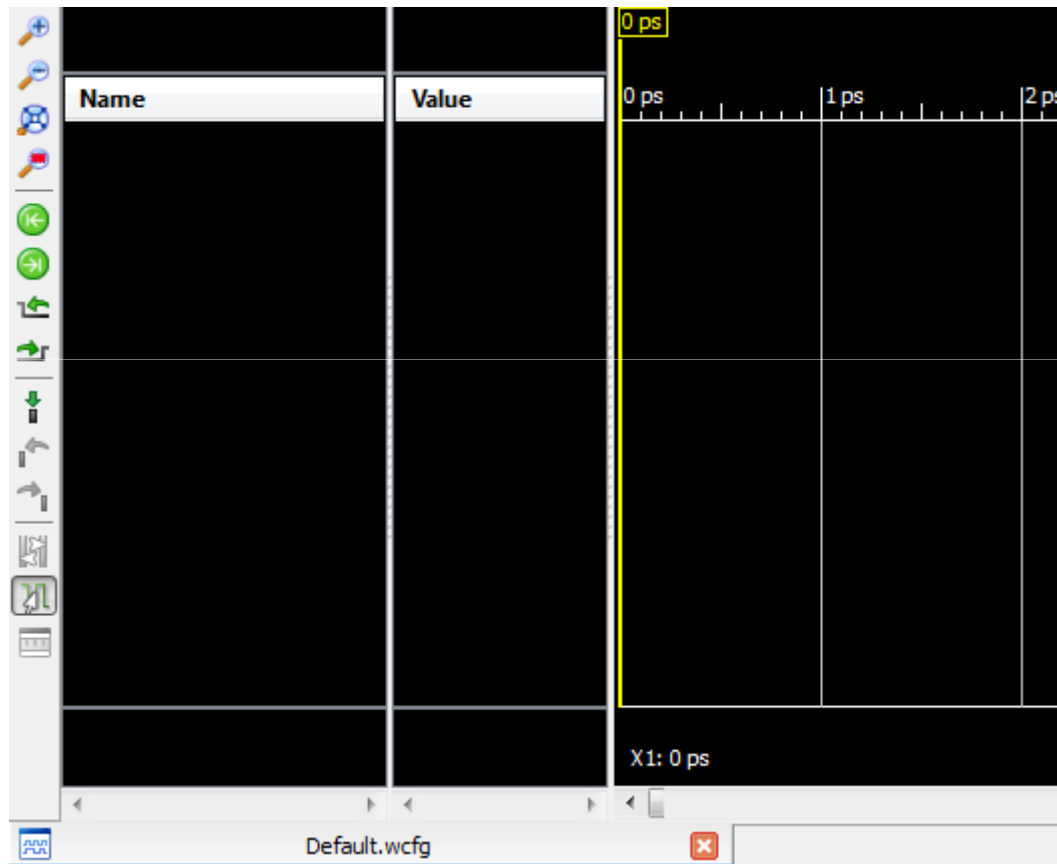
Objects Panel



Object Name	Value	Data Type
clk_in	0	Logic
drp_start	0	Logic
drp_change_mode	0	Logic
drp_current_mode	0	Logic
drp_done	U	Logic
drp_stmach_reset	0	Logic
user_dcm_reset	0	Logic
drp_multiply[7:0]	00000000	Array
drp_divide[7:0]	00000000	Array
dcm_clk0_out	U	Logic
dcm_clkfx_out	U	Logic
dcm_locked	U	Logic
period	10000 ps	Physical Type

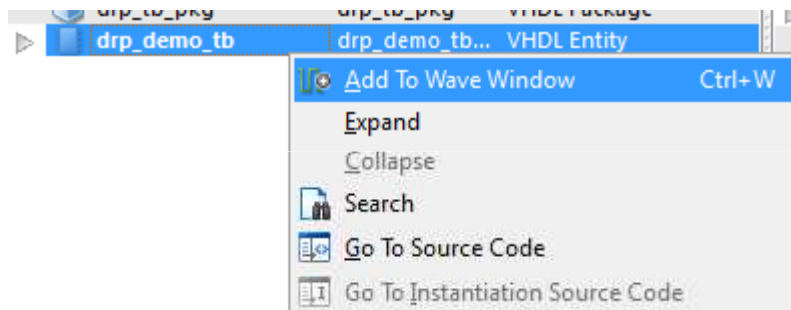
- Displays all ports and signals associated with the selected instances and processes in the Instances and Processes panel.
 - **Object Name** - Displays the name of the signal, accompanied by the symbol which represents the type of object it is.
 - **Value** - The value of the signals at the current 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



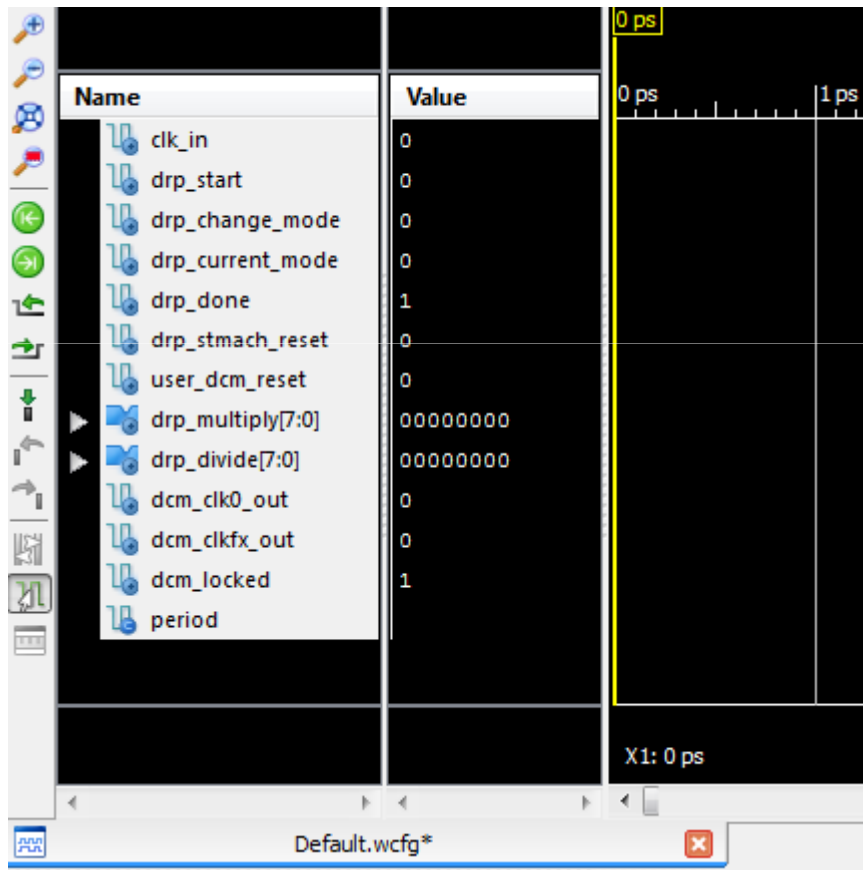
- Displays signals, buses, their properties and any other wave objects.
- The wave configuration is used to drive the simulation

Configuring the Wave Window



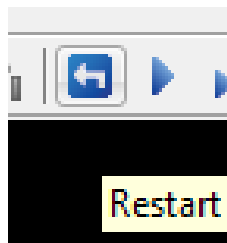
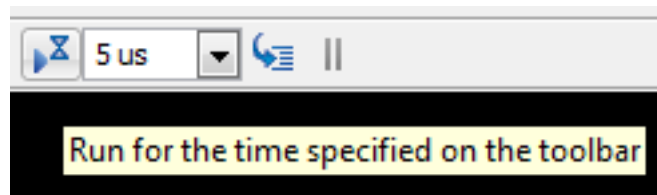
- Select the VHDL Entity and right click on it. The entity is the one with the ports, buses and signals.
- Choose “Add to Wave Window”

Wave Window with Signals



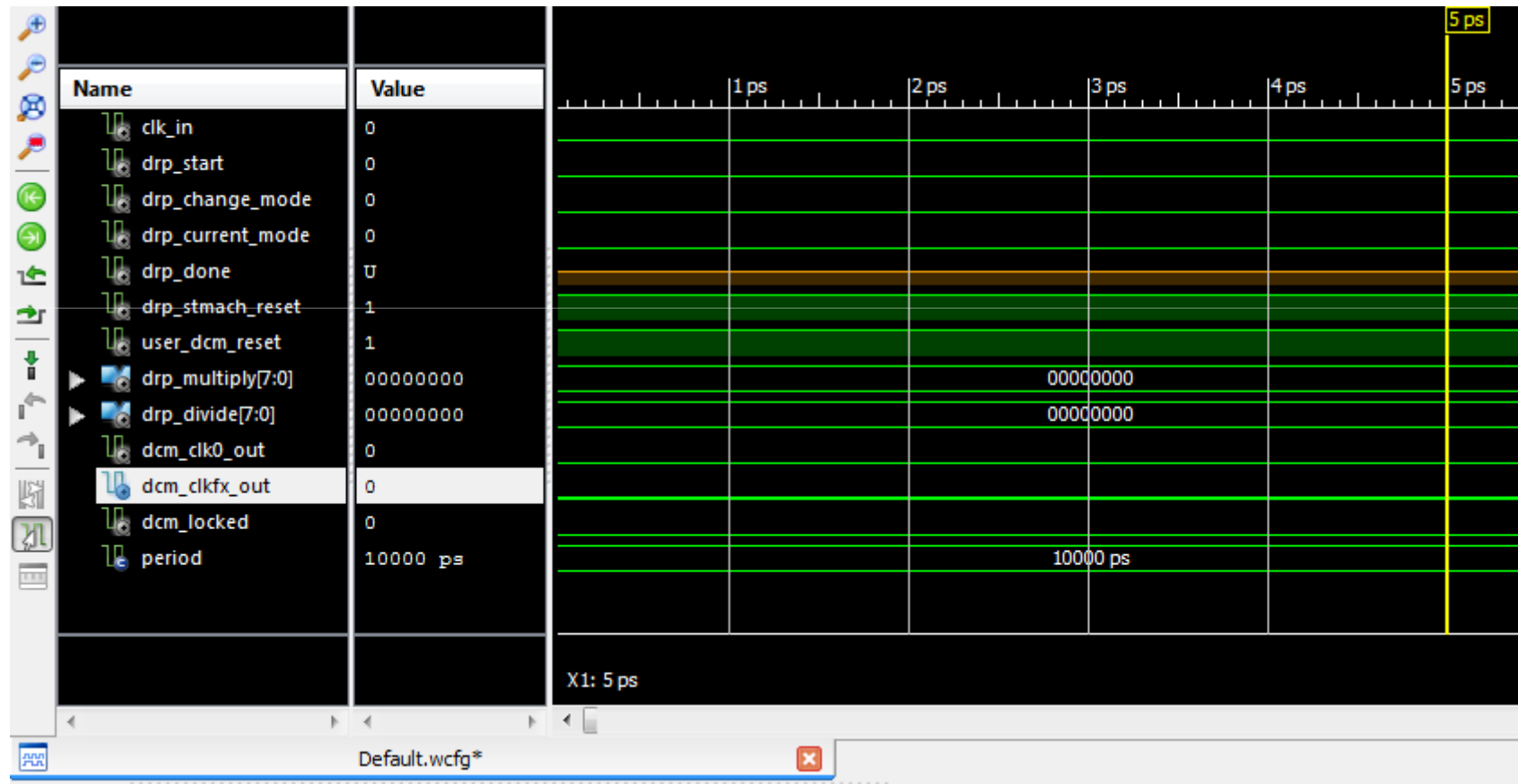
- Signals, ports and buses added.
- The values are the one listed in the Objects Panel

Running the Wave Sim for a Specified Time

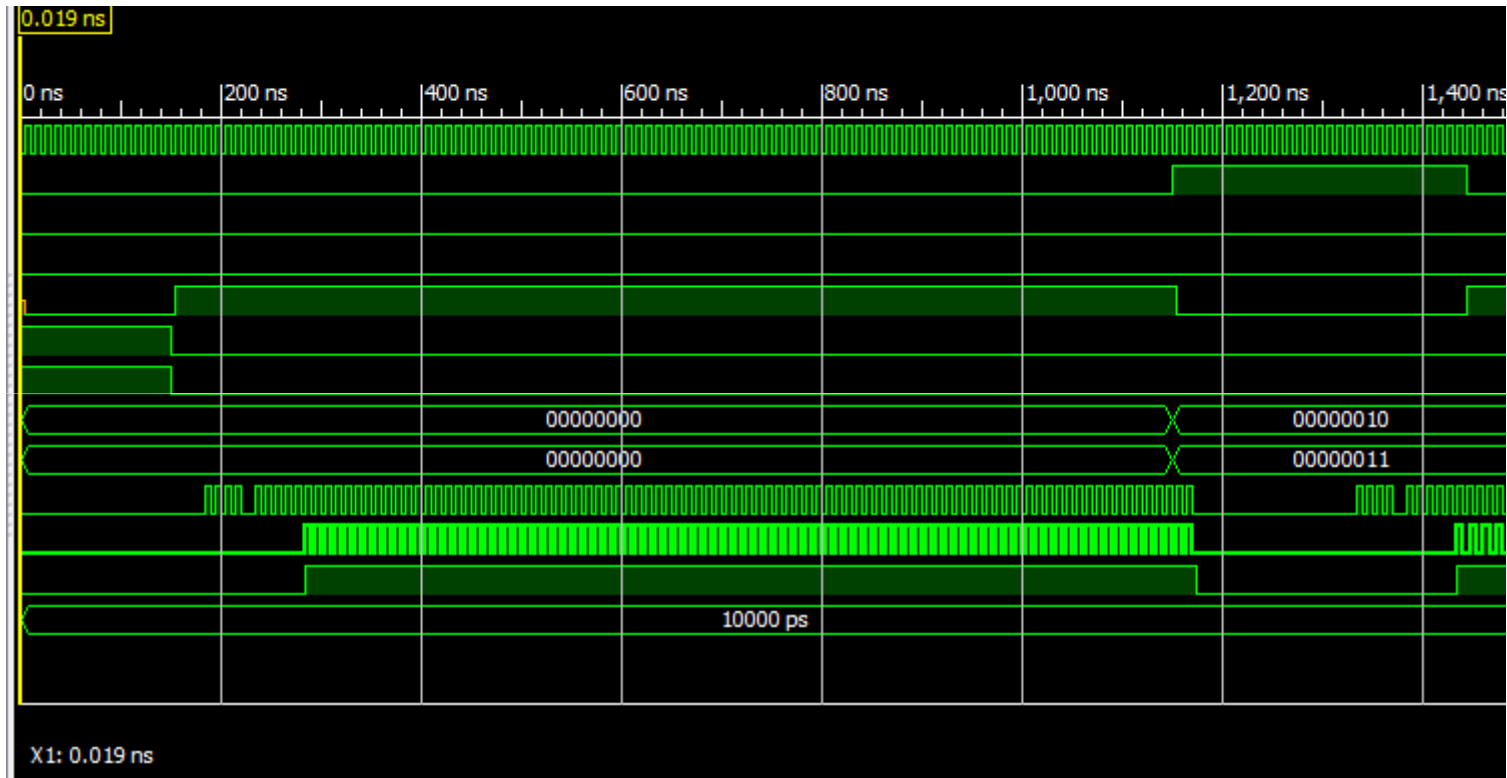


- Type **5 us** in the Simulation Time field and press Enter or click the “Run For” button
- Type **run 5 us** in the Console prompt
- If Simulation doesn’t start, press the “Restart” option

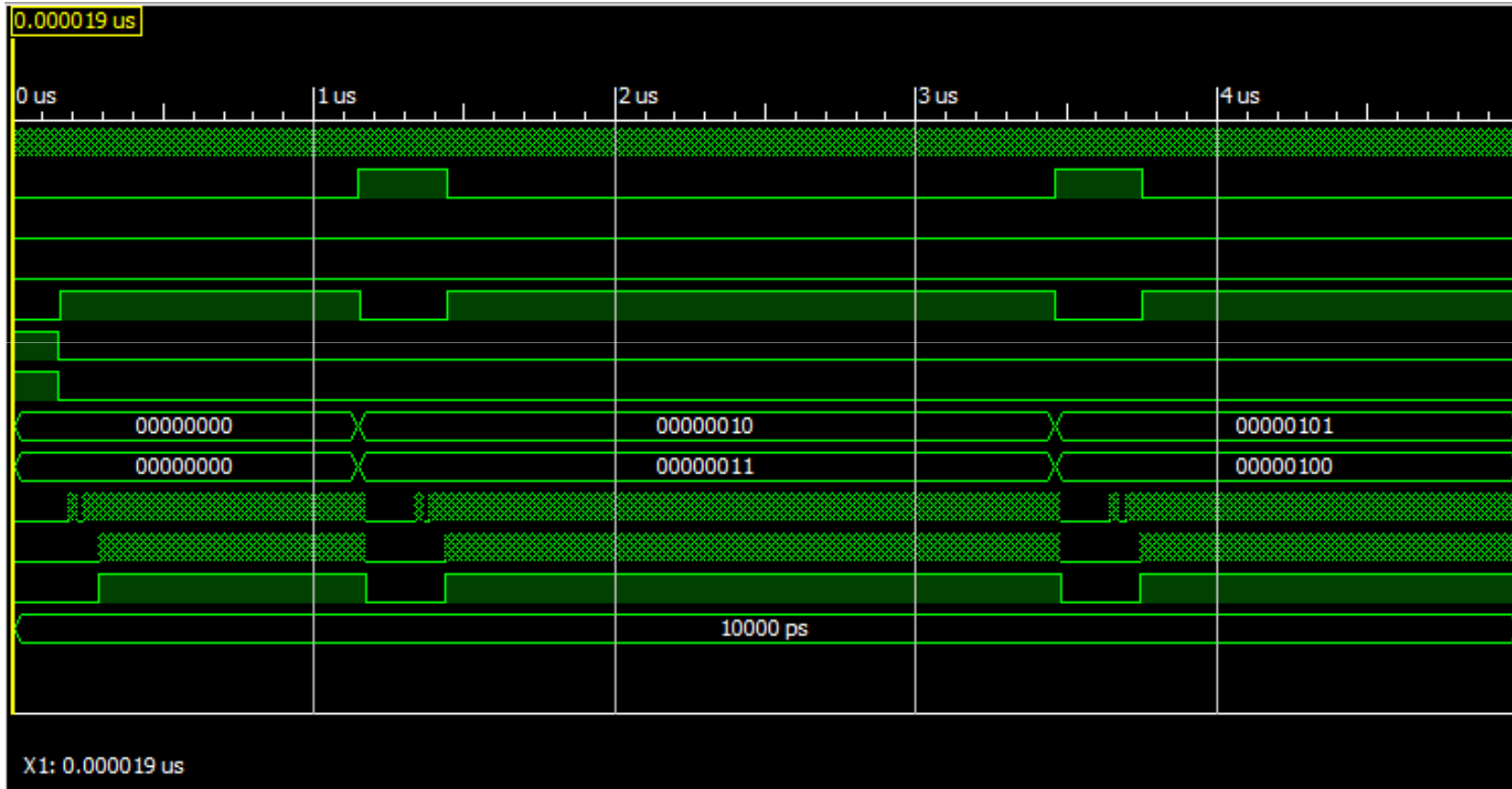
Wave Sim Results



Zoomed Out Wave Sim Results



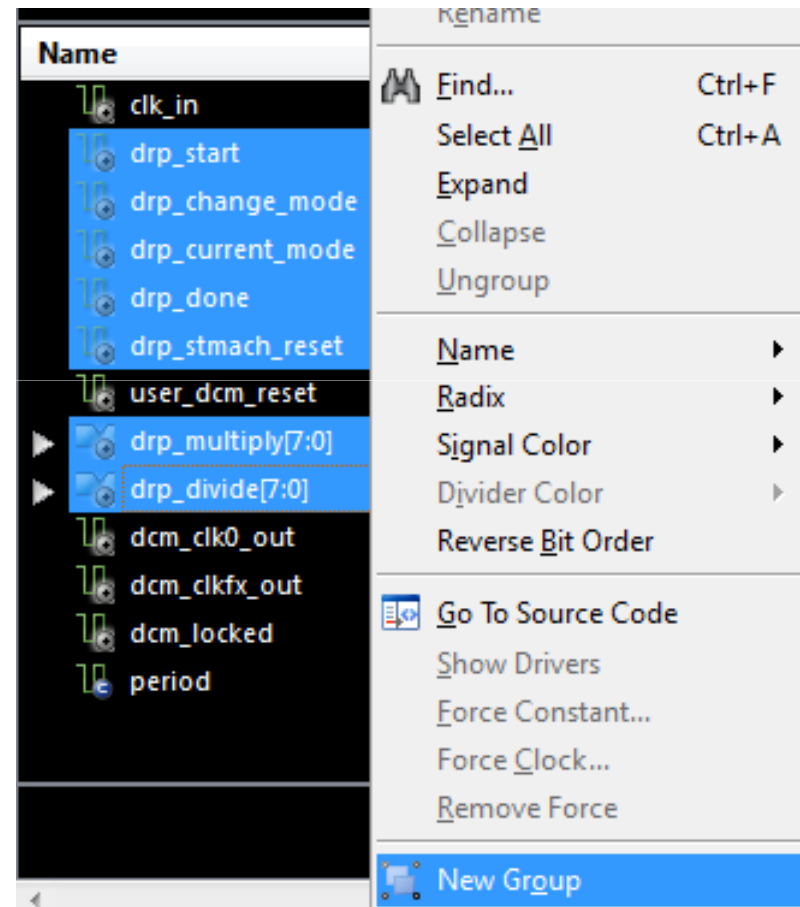
Full Simulation



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



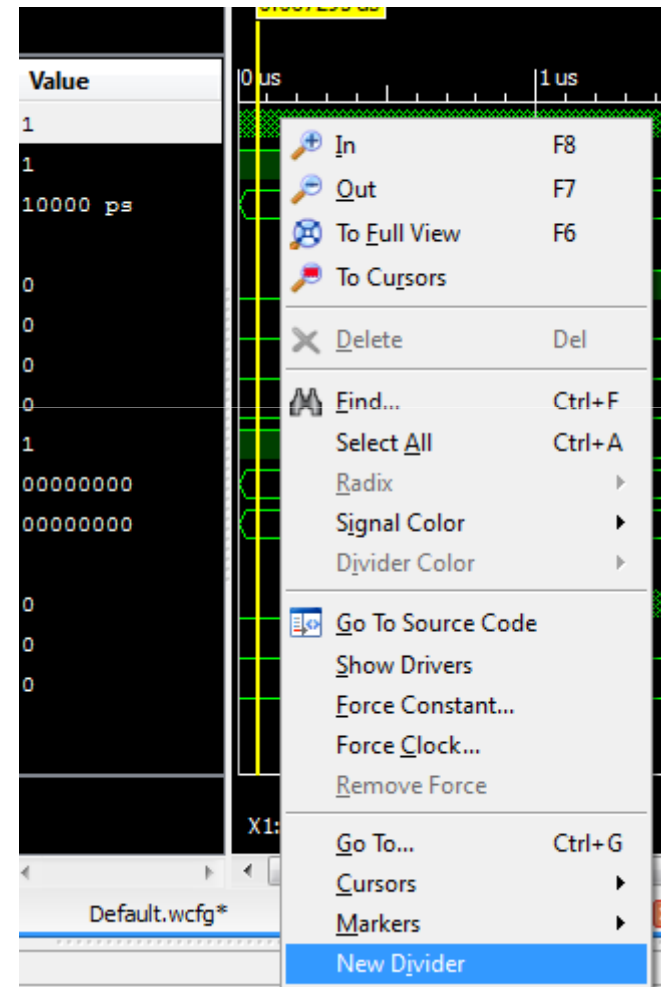
Groups

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

Name	Value
clk_in	0
user_dcm_reset	1
period	10000 ps
▼ DRP Test Signals	
drp_start	0
drp_change_mode	0
drp_current_mode	0
drp_done	0
drp_stmach_reset	1
▶ drp_multiply[7:0]	00000000
▶ drp_divide[7:0]	00000000
▼ DCM Test Signals	
dcm_clk0_out	0
dcm_clkfx_out	0
dcm_locked	0

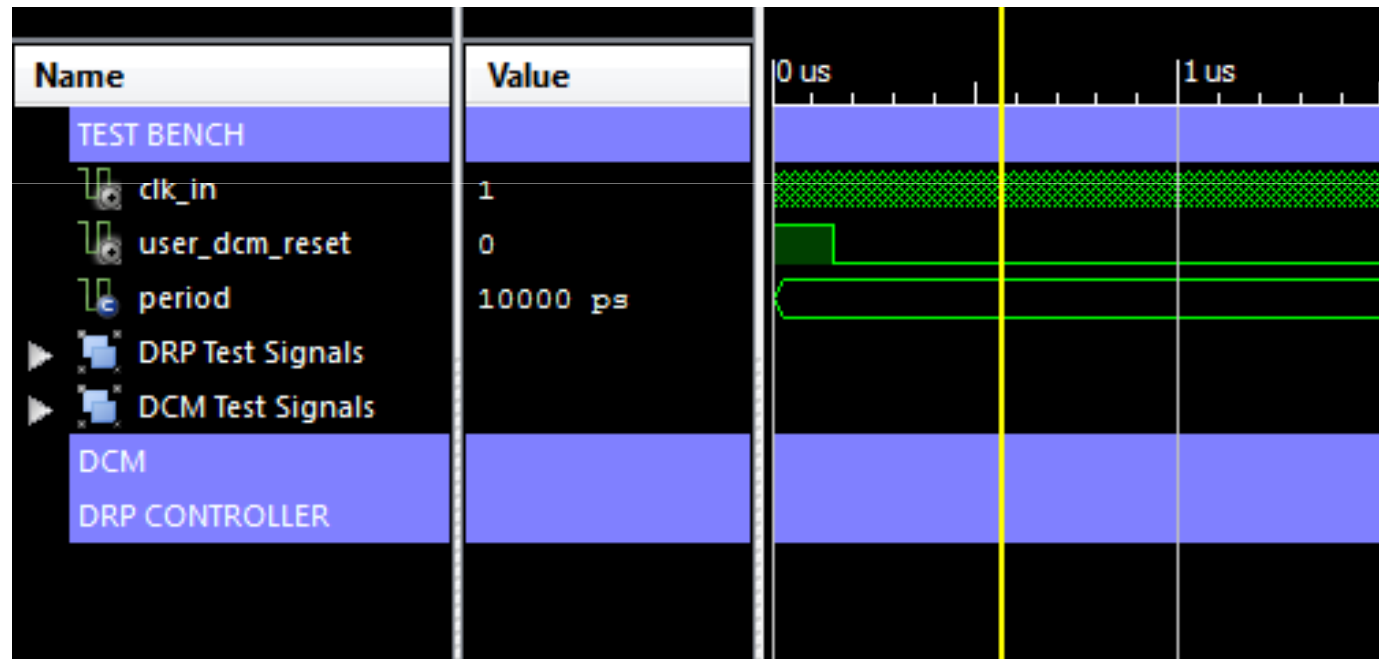
Dividers

- 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**



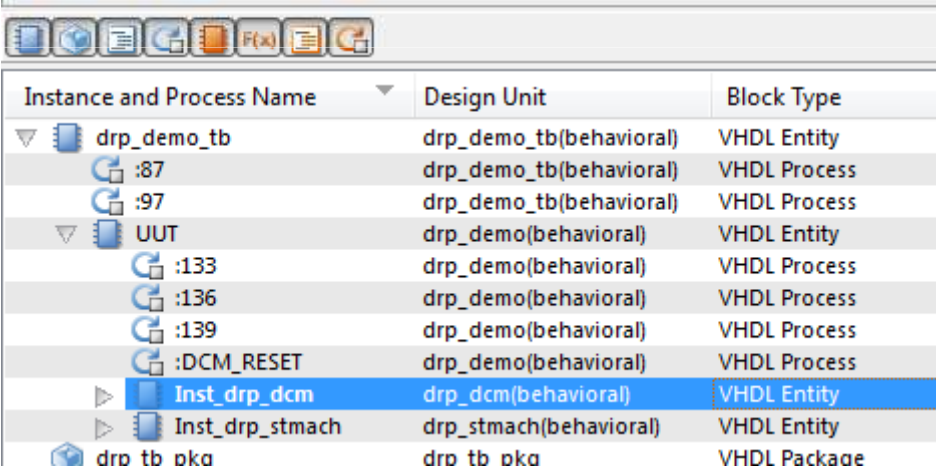
Dividers

- 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



Instance and Process Name	Design Unit	Block Type
drp_demo_tb	drp_demo_tb(behavioral)	VHDL Entity
:87	drp_demo_tb(behavioral)	VHDL Process
:97	drp_demo_tb(behavioral)	VHDL Process
UUT	drp_demo(behavioral)	VHDL Entity
:133	drp_demo(behavioral)	VHDL Process
:136	drp_demo(behavioral)	VHDL Process
:139	drp_demo(behavioral)	VHDL Process
:DCM_RESET	drp_demo(behavioral)	VHDL Process
Inst_drp_dcm	drp_dcm(behavioral)	VHDL Entity
Inst_drp_stmach	drp_stmach(behavioral)	VHDL Entity
drp_tb_pka	drp_tb_pka	VHDL Package

Objects from the instanced entity

- You'll get the following instances:
- You'll get the following signals:

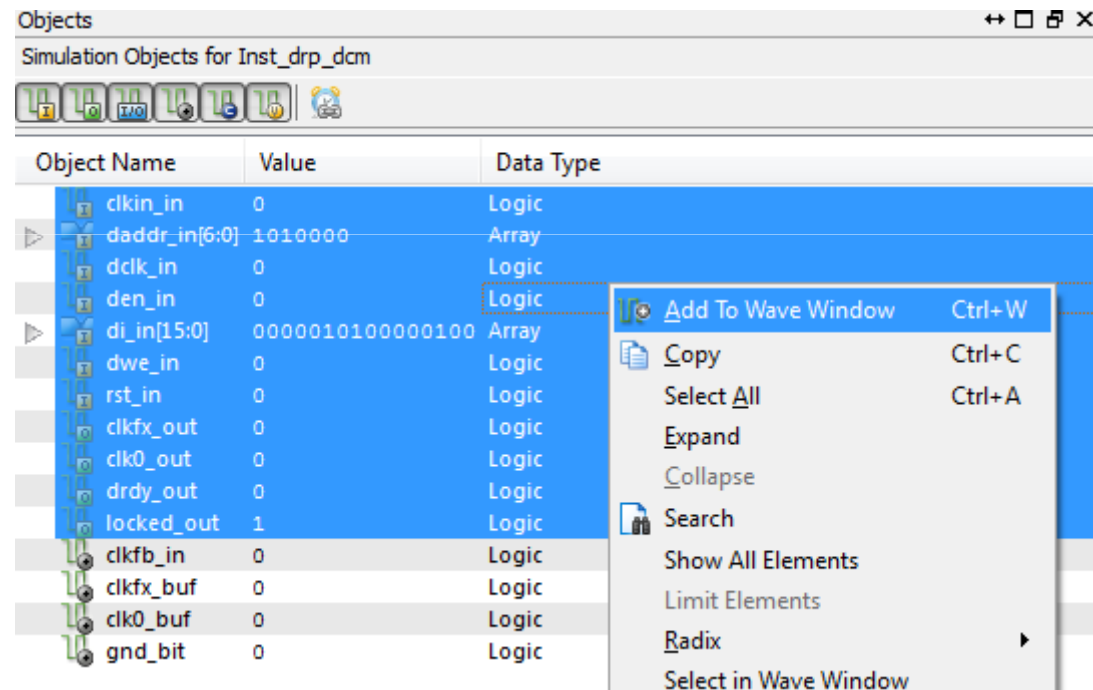
The screenshot displays two windows from the Xilinx Vivado IDE. The left window, titled "Instances and Processes", shows a tree view of the design hierarchy. The right window, titled "Objects", shows a table of simulation objects for the selected instance.

Instance and Process Name	Design Unit
drp_demo_tb	drp_demo_tb(behavioral)
:87	drp_demo_tb(behavioral)
:97	drp_demo_tb(behavioral)
UUT	drp_demo(behavioral)
:133	drp_demo(behavioral)
:136	drp_demo(behavioral)
:139	drp_demo(behavioral)
:DCM_RESET	drp_demo(behavioral)
Inst_drp_dcm	drp_dcm(behavioral)
Inst_drp_stmach	drp_stmach(behavioral)
drp_tb_pkg	drp_tb_pkg
numeric_std	numeric_std
std_logic_1164	std_logic_1164
std_logic_arith	std_logic_arith
std_logic_unsigned	std_logic_unsigned
textio	textio
vcomponents	vcomponents

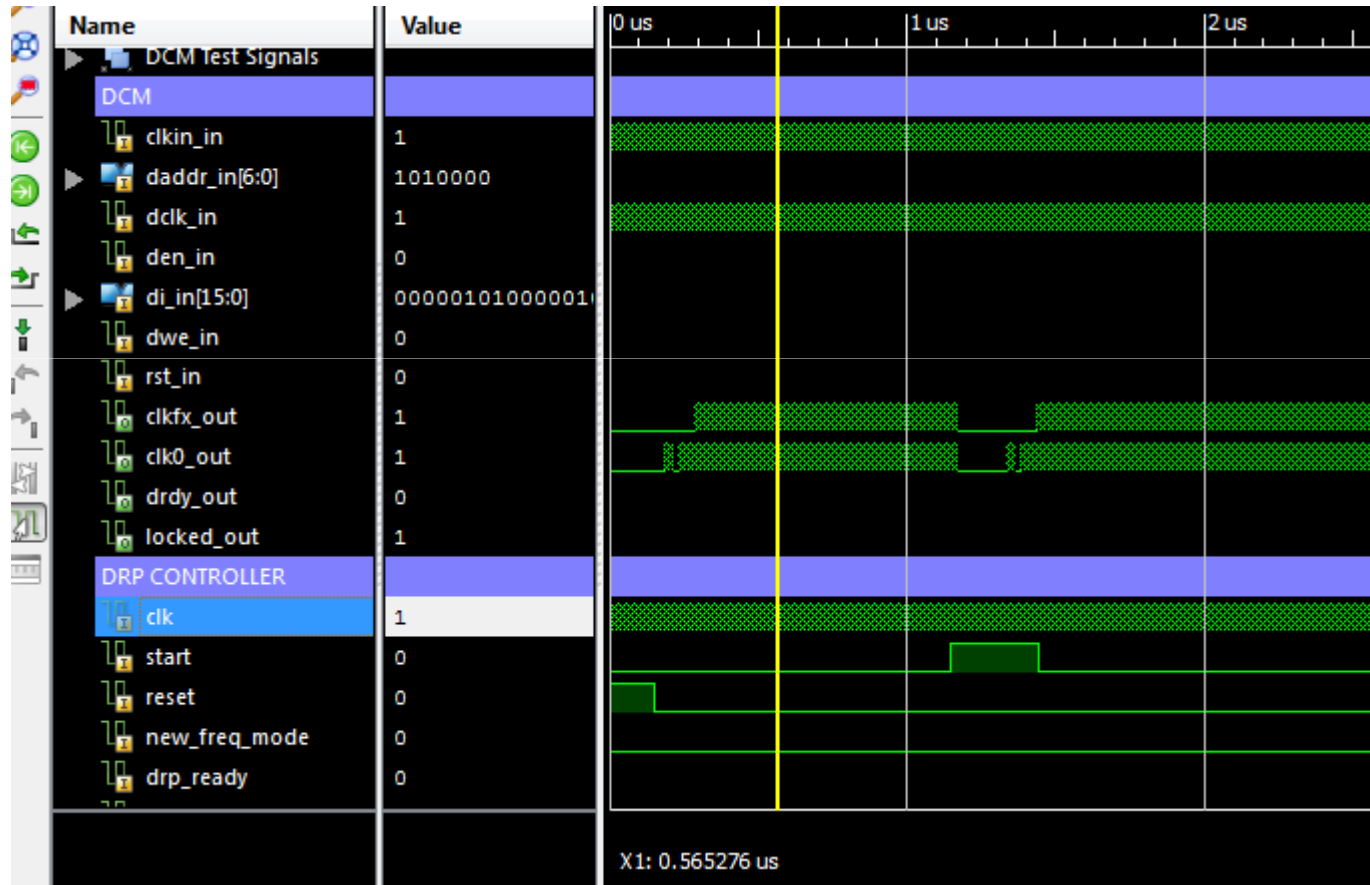
Object Name	Value	Data Type
clk_in	0	Logic
daddr_in[6:0]	1010000	Array
dclk_in	0	Logic
den_in	0	Logic
di_in[15:0]	0000010100000100	Array
dwe_in	0	Logic
rst_in	0	Logic
clkfx_out	0	Logic
clk0_out	0	Logic
drdy_out	0	Logic
locked_out	1	Logic
clkfb_in	0	Logic
clkfx_buf	0	Logic
clk0_buf	0	Logic
gnd_bit	0	Logic

Adding the Signals

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

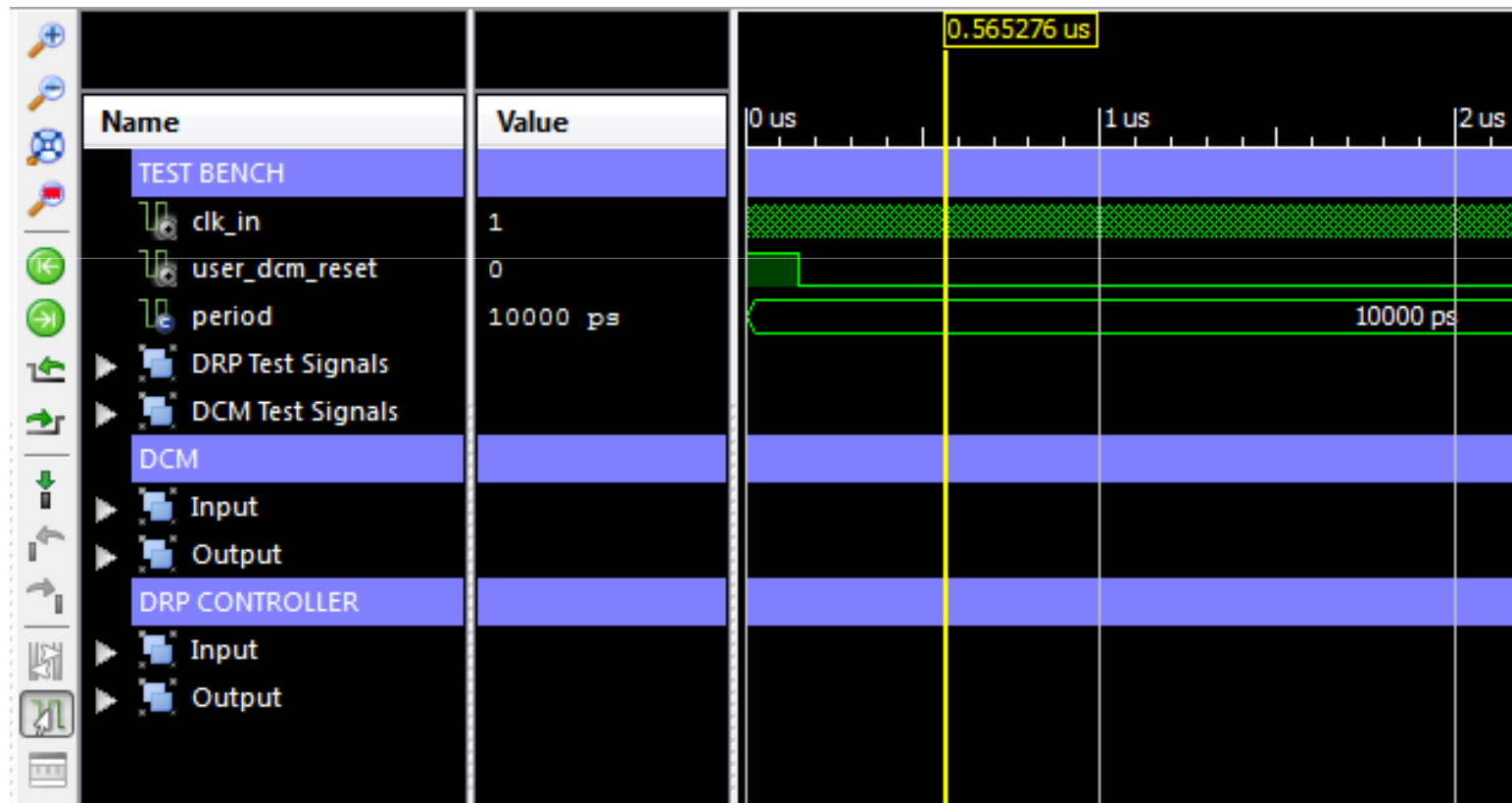


Added the Signals



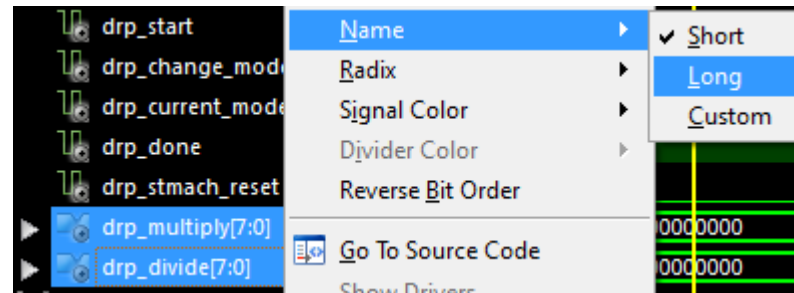
Organized by Groups

- The signals are organized in groups

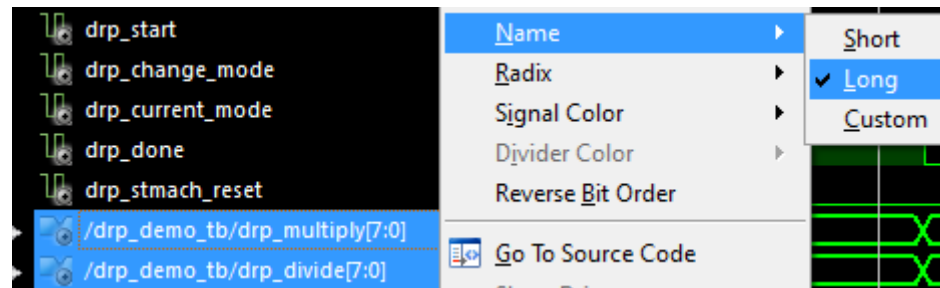


Naming the Signals

- Name:



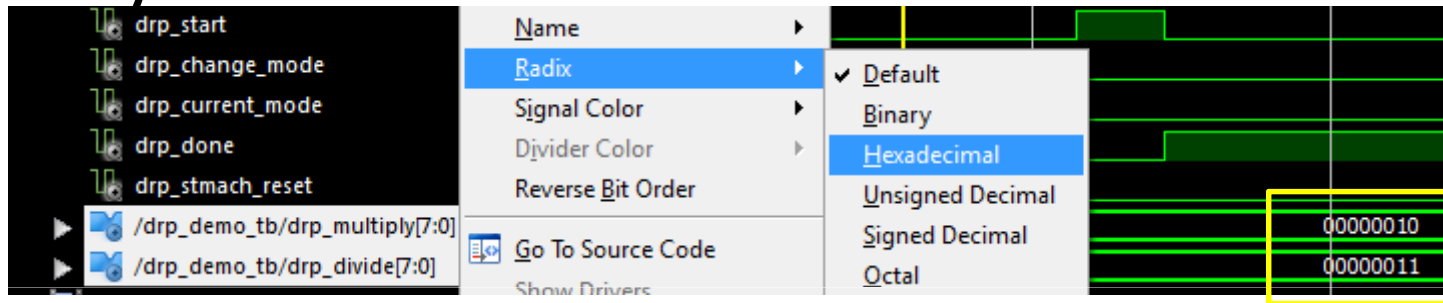
– Select **Long**



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

Radix

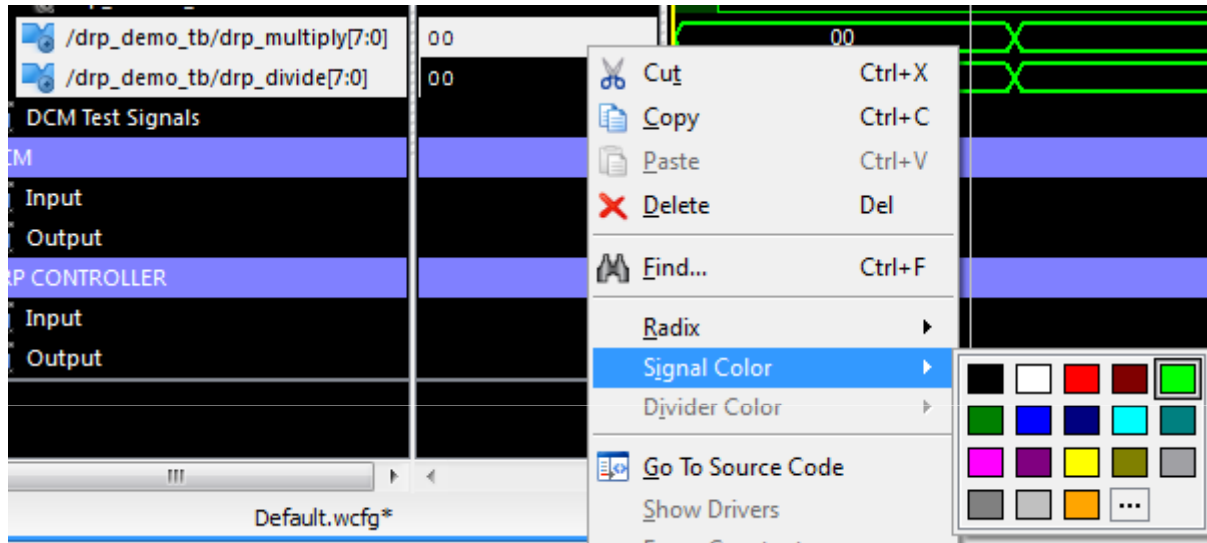
- Binary:



- Select Hexadecimal:



Crayola Effect

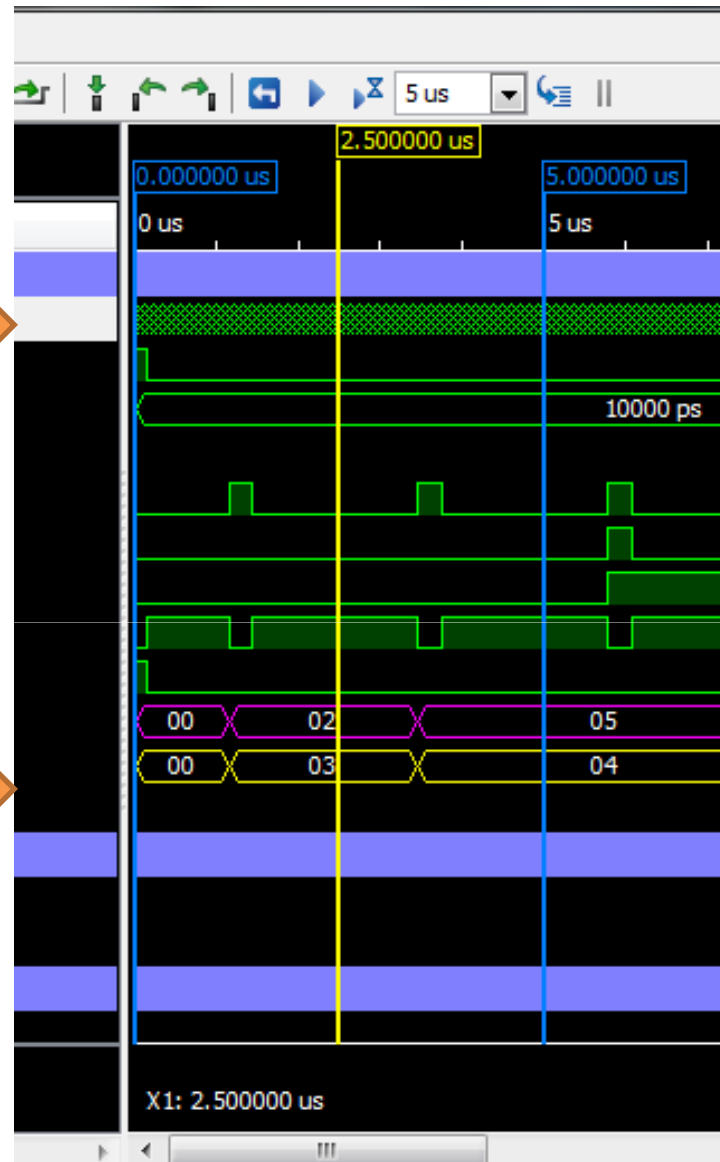
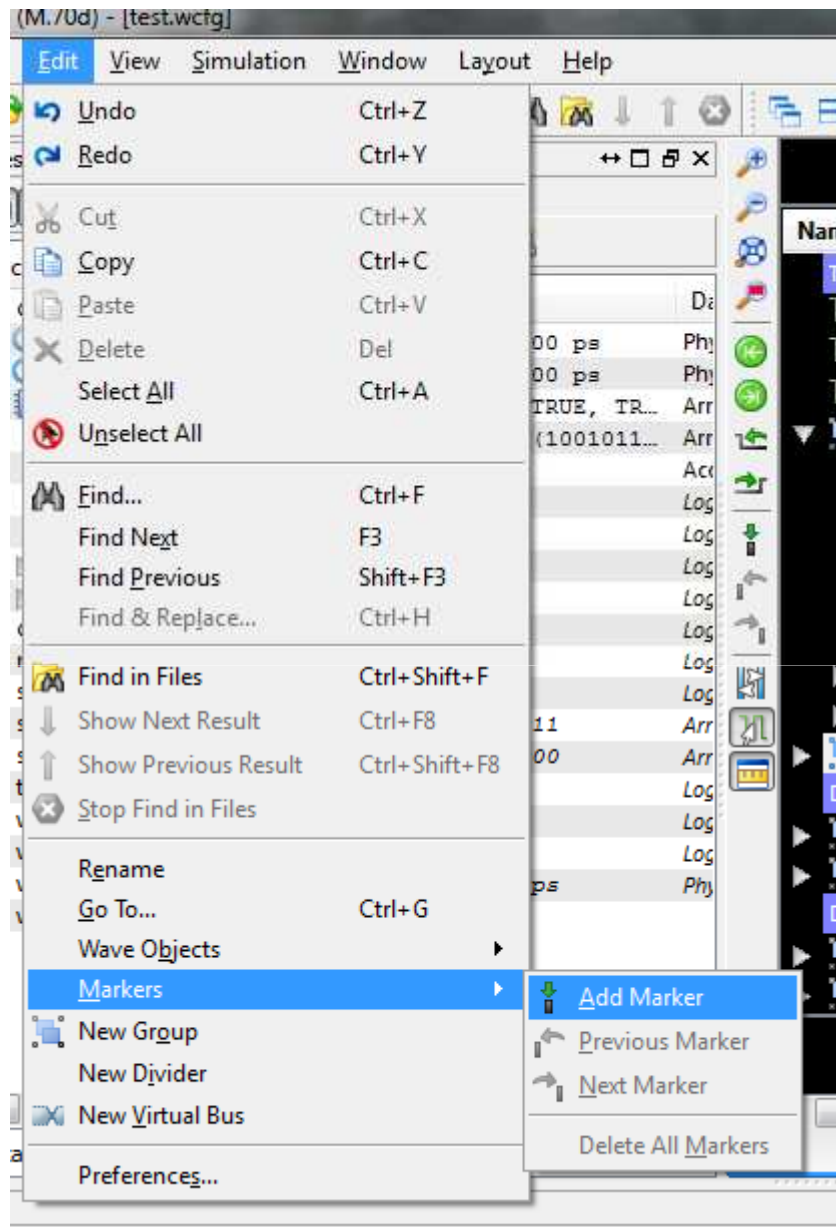


- Changing the Color:



Markers

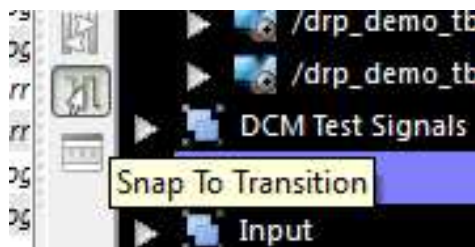
- Useful to identify times in the diagram
- Most used to determine when a test starts
- The tests are defined in the test bench file



Simulation finished successfully (not a failure)

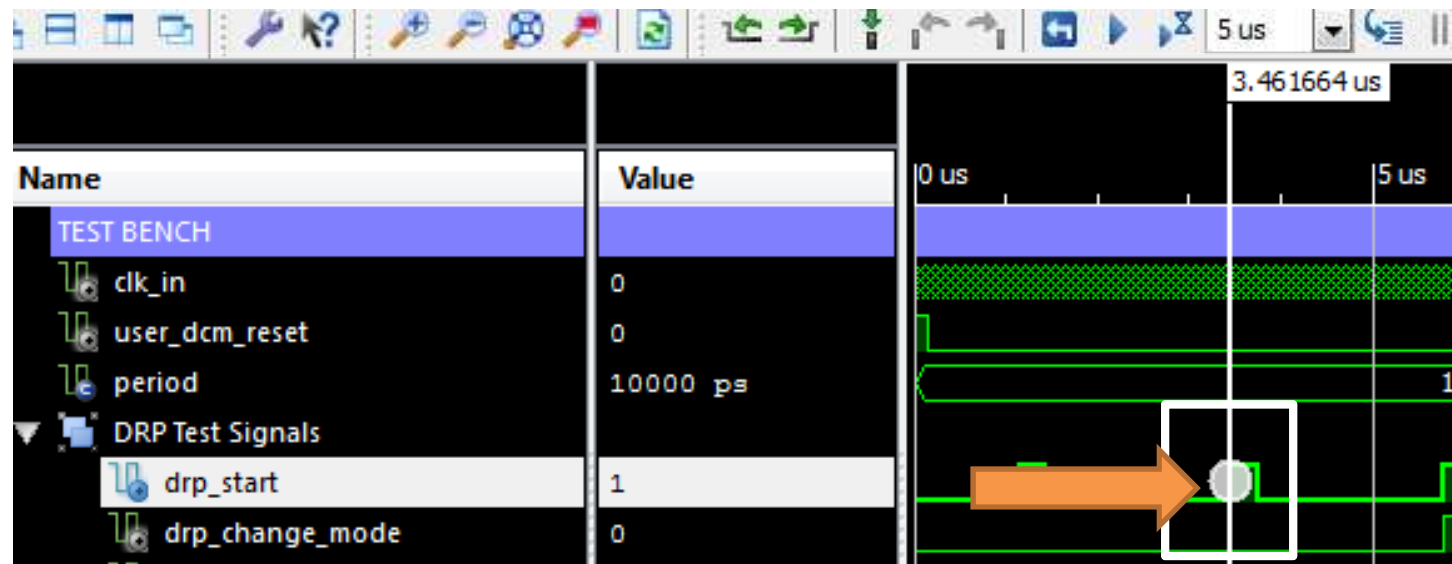
Measuring Time

- Measuring time between 2 endpoints
- Useful for calculating the frequency of a signal
- 1st: Click on the Snap to Transition toggle button



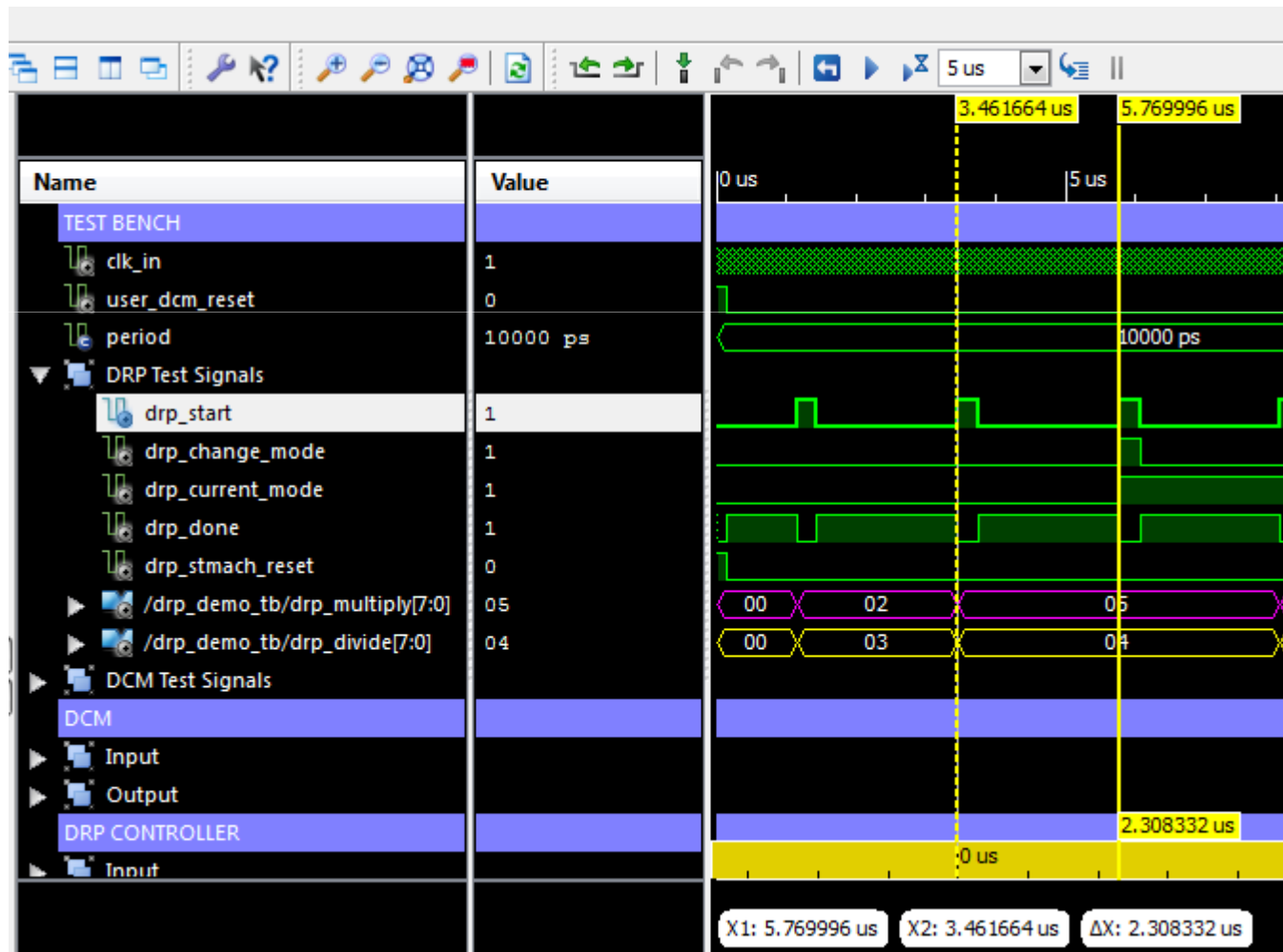
Measuring Time

- 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

$$\Delta X = 2.308332 \mu\text{s}$$

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