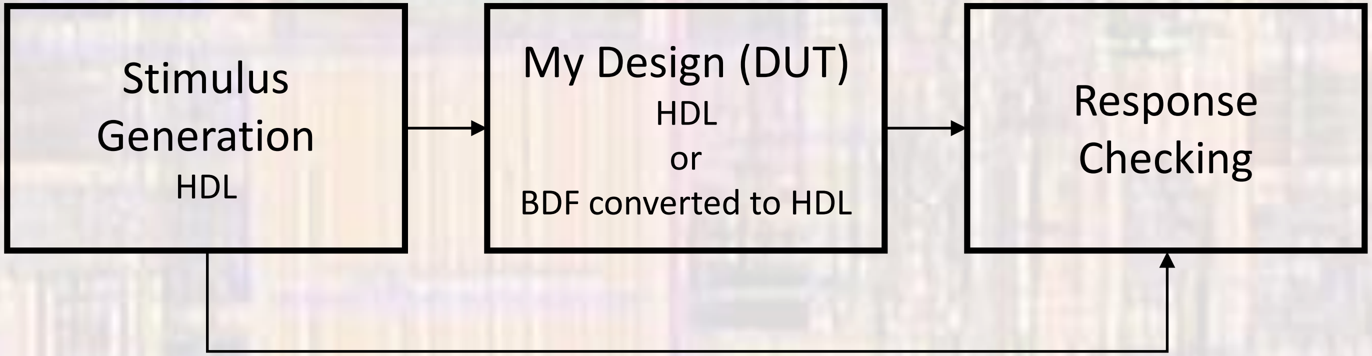
ModelSim Testbench
(Schematic)

Last updated 2/20/19

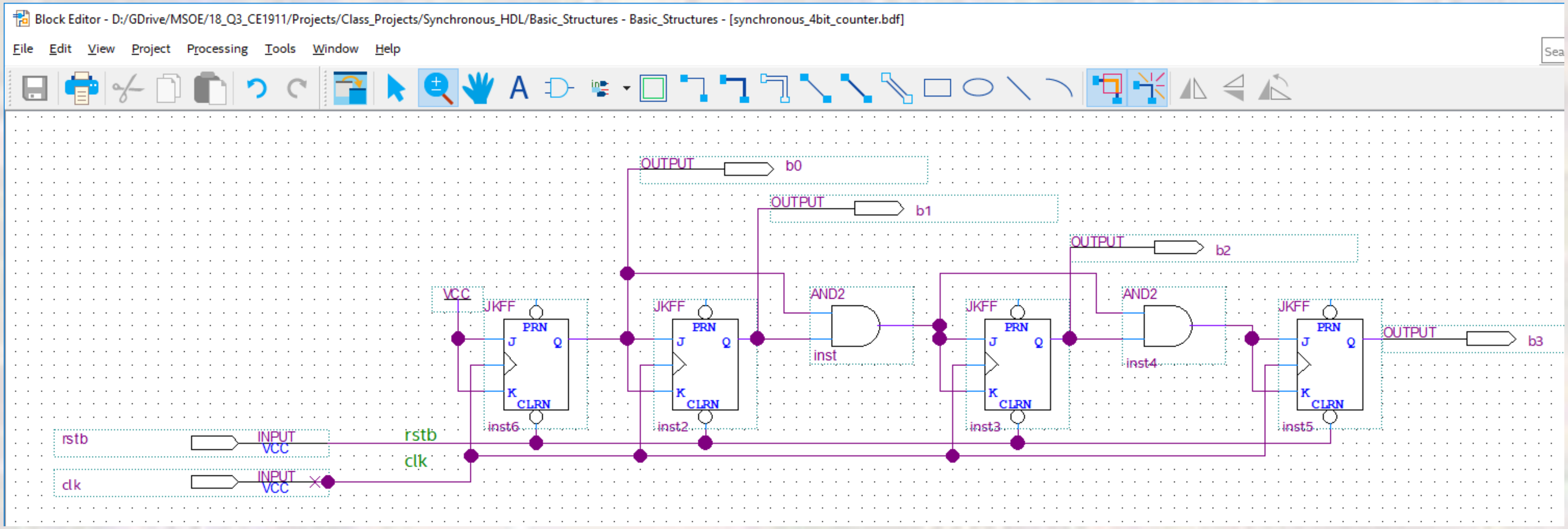
ModelSim Testbench (schematic)

- Test Bench Concept



ModelSim Testbench (schematic)

- Example: Counter – 4 bit
  - Create schematic (bdf)



Block Editor - D:/GDrive/MSOE/18_Q3_CE1911/Projects/Class_Projects/Synchronous_HDL/Basic_Structures - Basic_Structures - [synchronous_4bit_counter.bdf]

ModelSim Testbench (schematic)

- Example: Counter – 4 bit
  - Create a vhdl file of your schematic
    - File → Create/Update → Create HDL Design File from Current File

```
-- Copyright (C) 2016  Intel Corporation. All rights reserved.
-- Your use of Intel Corporation's design tools, logic functions
-- and other software and tools, and its AMPP partner logic
-- functions, and any output files from any of the foregoing
-- (including device programming or simulation files), and any
-- associated documentation or information are expressly subject
-- to the terms and conditions of the Intel Program License
-- Subscription Agreement, the Intel Quartus Prime License Agreement,
-- the Intel MegaCore Function License Agreement, or other
-- applicable license agreement, including, without limitation,
-- that your use is for the sole purpose of programming logic
-- devices manufactured by Intel and sold by Intel or its
-- authorized distributors.  Please refer to the applicable
-- agreement for further details.

-- PROGRAM       "Quartus Prime"
-- VERSION       "Version 16.1.0 Build 196 10/24/2016 SJ Lite Edition"
-- CREATED       "Mon Feb 12 11:52:17 2018"

LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY work;

ENTITY synchronous_4bit_counter IS
    PORT
    (
        rstb :  IN  STD_LOGIC;
        clk :  IN  STD_LOGIC;
        b0 :  OUT  STD_LOGIC;
        b1 :  OUT  STD_LOGIC;
        b2 :  OUT  STD_LOGIC;
        b3 :  OUT  STD_LOGIC
    );
END synchronous_4bit_counter;

ARCHITECTURE bdf_type OF synchronous_4bit_counter IS

SIGNAL    SYNTHESIZED_WIRE_7 :  STD_LOGIC;
SIGNAL    JKFF_inst2 :  STD_LOGIC;
SIGNAL    SYNTHESIZED_WIRE_8 :  STD_LOGIC;
SIGNAL    JKFF_inst3 :  STD_LOGIC;
SIGNAL    SYNTHESIZED_WIRE_9 :  STD_LOGIC;
SIGNAL    SYNTHESIZED_WIRE_10 :  STD_LOGIC;


BEGIN
b0 <= SYNTHESIZED_WIRE_7;
b1 <= JKFF_inst2;
b2 <= JKFF_inst3;
SYNTHESIZED_WIRE_10 <= '1';
```

```
SYNTHESIZED_WIRE_8 <= SYNTHESIZED_WIRE_7 AND JKFF_inst2;

PROCESS(clk,rstb)
VARIABLE synthesized_var_for_JKFF_inst2 : STD_LOGIC;
BEGIN
IF (rstb = '0') THEN
    synthesized_var_for_JKFF_inst2 := '0';
ELSIF (RISING_EDGE(clk)) THEN
    synthesized_var_for_JKFF_inst2 := (NOT(synthesized_var_for_JKFF_inst2) AND SYNTHESIZED_WIRE_7) OR (synthesized_var_for_JKFF_inst2 AND
(NOT(SYNTHESIZED_WIRE_7)));
END IF;
    JKFF_inst2 <= synthesized_var_for_JKFF_inst2;
END PROCESS;


PROCESS(clk,rstb)
VARIABLE synthesized_var_for_JKFF_inst3 : STD_LOGIC;
BEGIN
IF (rstb = '0') THEN
    synthesized_var_for_JKFF_inst3 := '0';
ELSIF (RISING_EDGE(clk)) THEN
    synthesized_var_for_JKFF_inst3 := (NOT(synthesized_var_for_JKFF_inst3) AND SYNTHESIZED_WIRE_8) OR (synthesized_var_for_JKFF_inst3 AND
(NOT(SYNTHESIZED_WIRE_8)));
END IF;
    JKFF_inst3 <= synthesized_var_for_JKFF_inst3;
END PROCESS;


SYNTHESIZED_WIRE_9 <= SYNTHESIZED_WIRE_8 AND JKFF_inst3;


PROCESS(clk,rstb)
VARIABLE synthesized_var_for_b3 : STD_LOGIC;
BEGIN
IF (rstb = '0') THEN
    synthesized_var_for_b3 := '0';
ELSIF (RISING_EDGE(clk)) THEN
    synthesized_var_for_b3 := (NOT(synthesized_var_for_b3) AND SYNTHESIZED_WIRE_9) OR (synthesized_var_for_b3 AND (NOT(SYNTHESIZED_WIRE_9)));
END IF;
    b3 <= synthesized_var_for_b3;
END PROCESS;


PROCESS(clk,rstb)
VARIABLE synthesized_var_for_SYNTHESIZED_WIRE_7 : STD_LOGIC;
BEGIN
IF (rstb = '0') THEN
    synthesized_var_for_SYNTHESIZED_WIRE_7 := '0';
ELSIF (RISING_EDGE(clk)) THEN
    synthesized_var_for_SYNTHESIZED_WIRE_7 := (NOT(synthesized_var_for_SYNTHESIZED_WIRE_7) AND SYNTHESIZED_WIRE_10) OR
(synthesized_var_for_SYNTHESIZED_WIRE_7 AND (NOT(SYNTHESIZED_WIRE_10)));
END IF;
    SYNTHESIZED_WIRE_7 <= synthesized_var_for_SYNTHESIZED_WIRE_7;
END PROCESS;


END bdf_type;
```

ModelSim Testbench (schematic)

- Example: Counter – 4 bit
    - Create a component template for your design (DUT)
        - File → Create/Update → Create VHDL Component Declaration Files from Current File

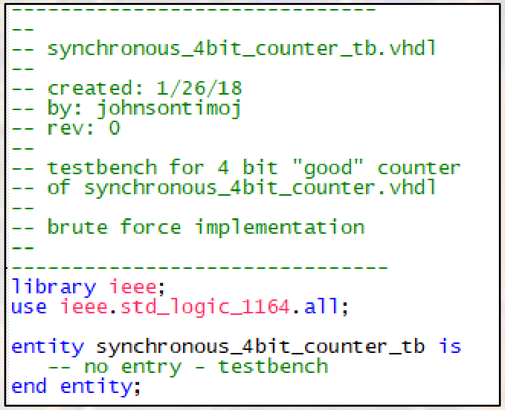
```
-- Copyright (C) 2016  Intel Corporation. All rights reserved.
-- Your use of Intel Corporation's design tools, logic functions
-- …
-- of programming logic
-- devices manufactured by Intel and sold by Intel or its
-- authorized distributors.  Please refer to the applicable
-- agreement for further details.


-- Generated by Quartus Prime Version 16.1 (Build Build 196 10/24/2016)
-- Created on Mon Feb 12 12:36:19 2018

COMPONENT synchronous_4bit_counter
    PORT
    (
        rstb        :     IN STD_LOGIC;
        clk     :     IN STD_LOGIC;
        b0      :     OUT STD_LOGIC;
        b1      :     OUT STD_LOGIC;
        b2      :     OUT STD_LOGIC;
        b3      :     OUT STD_LOGIC
    );
END COMPONENT;
```

ModelSim Testbench (schematic)

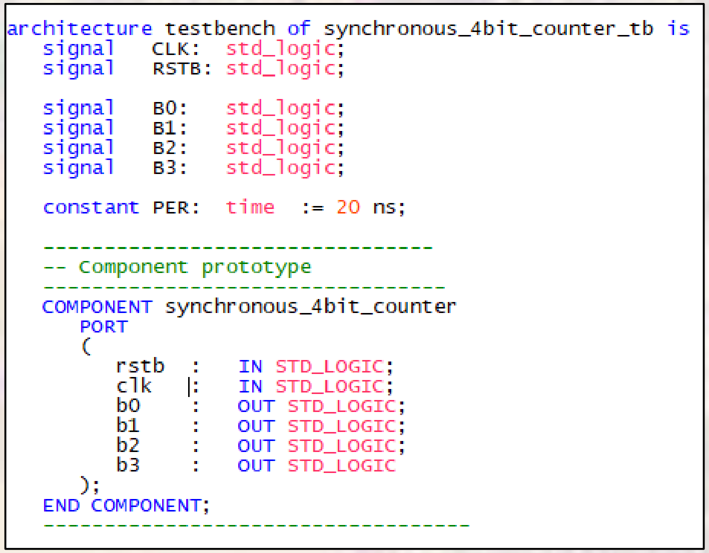
- Example: Counter – 4 bit
  - Prepare to use the VHDL file
    - Remove the BDF file from the project
      - In project Navigator – Files, Right click the BDF file and select remove
    - Add the created VHDL file to the project
      - Project → add/remove files in project → file name …
      - Select the file to add
    - Set the VHDL file as the Top-Level Entity
      - In project Navigator – Files, Right click the VHDL file and select: Set as Top-Level Entity

ModelSim Testbench (schematic)

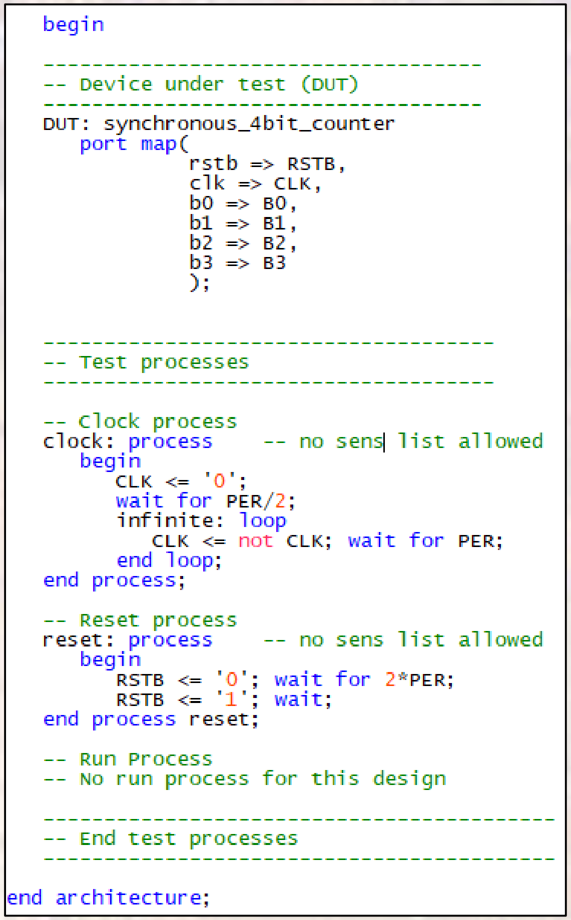
- Example: Counter – 4bit
  - Create a test bench



```
------------------------------
--
-- synchronous_4bit_counter_tb.vhdl
--
-- created: 1/26/18
-- by: johnsontimoj
-- rev: 0
--
-- testbench for 4 bit "good" counter
-- of synchronous_4bit_counter.vhdl
--
-- brute force implementation
--
-------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity synchronous_4bit_counter_tb is
    -- no entry - testbench
end entity;
```



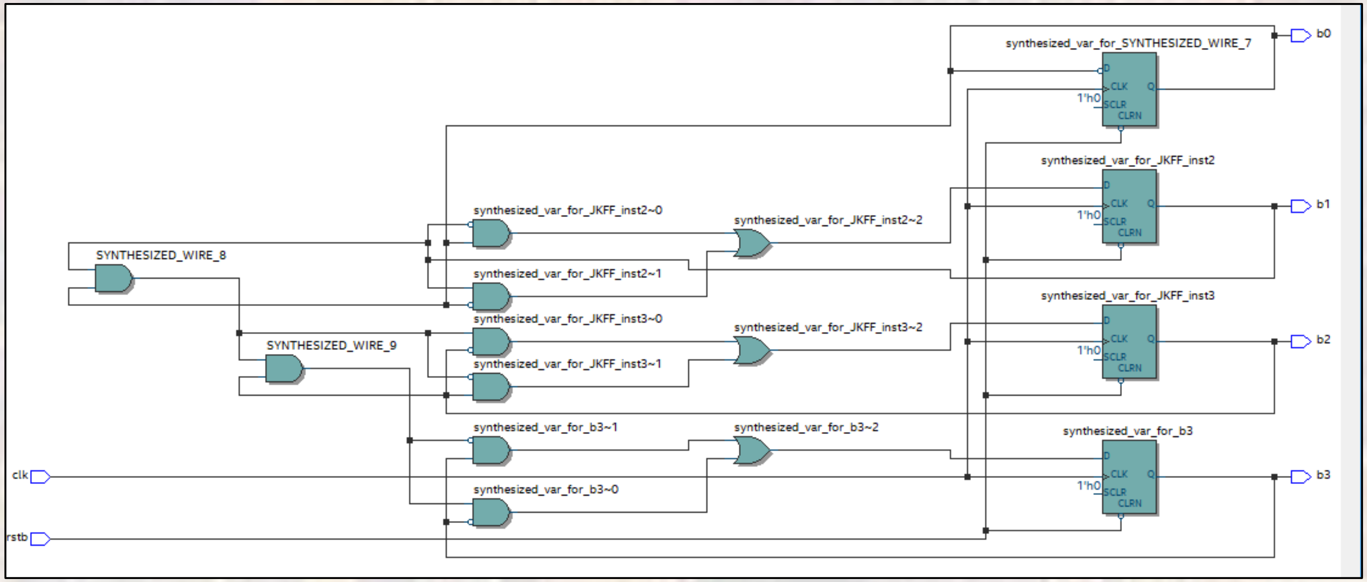
```
architecture testbench of synchronous_4bit_counter_tb is
   signal   CLK:  std_logic;
   signal   RSTB: std_logic;

   signal   B0:   std_logic;
   signal   B1:   std_logic;
   signal   B2:   std_logic;
   signal   B3:   std_logic;

   constant PER:  time   := 20 ns;

   ----------------------------------
   -- Component prototype
   -----------------------------------
   COMPONENT synchronous_4bit_counter
      PORT
      (
         rstb   :   IN STD_LOGIC;
         clk    :   IN STD_LOGIC;
         b0     :   OUT STD_LOGIC;
         b1     :   OUT STD_LOGIC;
         b2     :   OUT STD_LOGIC;
         b3     :   OUT STD_LOGIC
      );
   END COMPONENT;
   ------------------------------------
```



```
   begin

   ------------------------------------
   -- Device under test (DUT)
   ------------------------------------
   DUT: synchronous_4bit_counter
      port map(
               rstb => RSTB,
               clk => CLK,
               b0 => B0,
               b1 => B1,
               b2 => B2,
               b3 => B3
               );


   -------------------------------------
   -- Test processes
   -------------------------------------

   -- Clock process
   clock: process      -- no sens list allowed
      begin
         CLK <= '0';
         wait for PER/2;
         infinite: loop
            CLK <= not CLK; wait for PER;
         end loop;
   end process;

   -- Reset process
   reset: process      -- no sens list allowed
      begin
         RSTB <= '0'; wait for 2*PER;
         RSTB <= '1'; wait;
   end process reset;

   -- Run Process
   -- No run process for this design

   ------------------------------------------
   -- End test processes
   ------------------------------------------

end architecture;
```

ModelSim Testbench (schematic)

- Example: Counter – 4 bit
  - Elaborate the design (create RTL)
    - Processing → Start → Start Analysis and Elaboration
  - Check the RTL
    - Tools → Netlist Viewers → RTL Viewer


synthesized_var_for_SYNTHESIZED_WIRE_7
b0
D
CLK
Q
1'h0
SCLR
CLRN
synthesized_var_for_JKFF_inst2
b1
synthesized_var_for_JKFF_inst2~0
synthesized_var_for_JKFF_inst2~2
SYNTHESIZED_WIRE_8
synthesized_var_for_JKFF_inst2~1
synthesized_var_for_JKFF_inst3
synthesized_var_for_JKFF_inst3~0
synthesized_var_for_JKFF_inst3~2
SYNTHESIZED_WIRE_9
synthesized_var_for_JKFF_inst3~1
b2
synthesized_var_for_b3~1
synthesized_var_for_b3~2
synthesized_var_for_b3
clk
b3
synthesized_var_for_b3~0
rstb

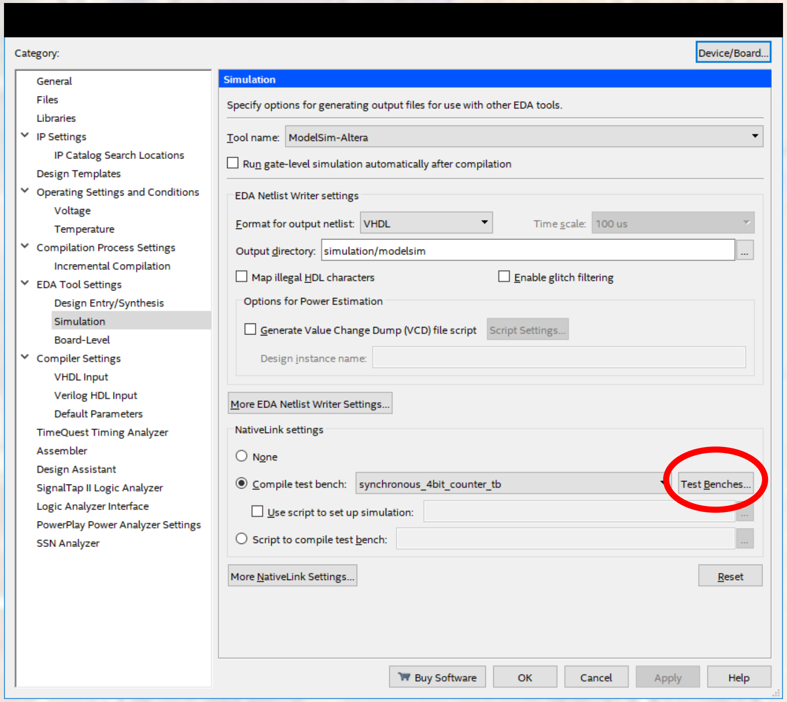
Note: JK Flip-Flops are generated from D Flip-Flops

ModelSim Testbench (schematic)

- Example: Counter – 4 bit
  - Setup the test bench
    - Assignments → Settings → EDA Tool Settings → Simulation → Test Benches : enter the test bench file

      : select the end simulation time

      : select File name … and select the test bench file



[redacted]

Category:

Device/Board...

- General
- Files
- Libraries
- IP Settings
  - IP Catalog Search Locations
- Design Templates
- Operating Settings and Conditions
  - Voltage
  - Temperature
- Compilation Process Settings
  - Incremental Compilation
- EDA Tool Settings
  - Design Entry/Synthesis
  - Simulation
  - Board-Level
- Compiler Settings
  - VHDL Input
  - Verilog HDL Input
  - Default Parameters
- TimeQuest Timing Analyzer
- Assembler
- Design Assistant
- SignalTap II Logic Analyzer
- Logic Analyzer Interface
- PowerPlay Power Analyzer Settings
- SSN Analyzer

Simulation

Specify options for generating output files for use with other EDA tools.

Tool name: ModelSim-Altera

□ Run gate-level simulation automatically after compilation

EDA Netlist Writer settings

Format for output netlist: VHDL   Time scale: 100 us

Output directory: simulation/modelsim

□ Map illegal HDL characters   □ Enable glitch filtering

Options for Power Estimation

□ Generate Value Change Dump (VCD) file script   Script Settings...

Design instance name:

More EDA Netlist Writer Settings...

NativeLink settings

○ None

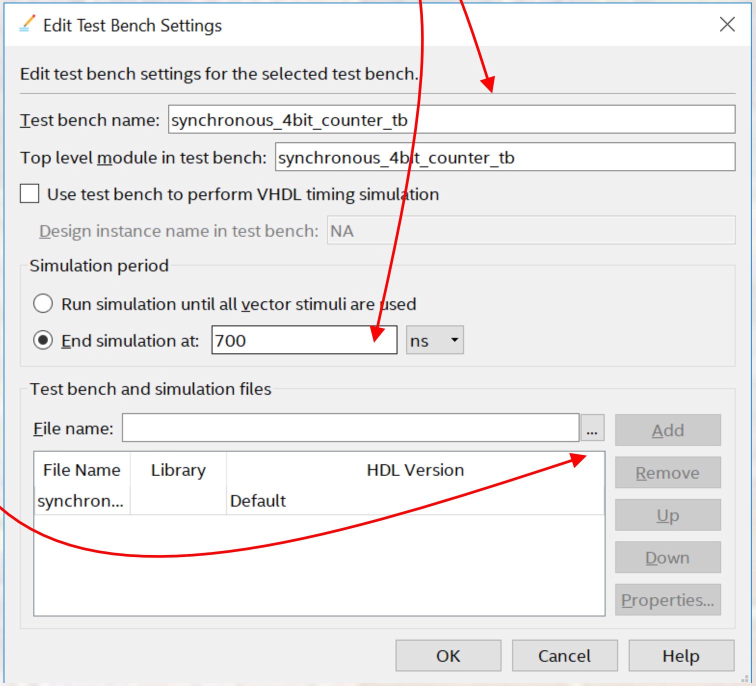
● Compile test bench: synchronous_4bit_counter_tb   Test Benches...

□ Use script to set up simulation:

○ Script to compile test bench:

More NativeLink Settings...   Reset

Buy Software   OK   Cancel   Apply   Help



Edit Test Bench Settings

Edit test bench settings for the selected test bench.

Test bench name: synchronous_4bit_counter_tb

Top level module in test bench: synchronous_4bit_counter_tb

□ Use test bench to perform VHDL timing simulation

Design instance name in test bench: NA

Simulation period

○ Run simulation until all vector stimuli are used

● End simulation at: 700 ns

Test bench and simulation files

File name: ...   Add

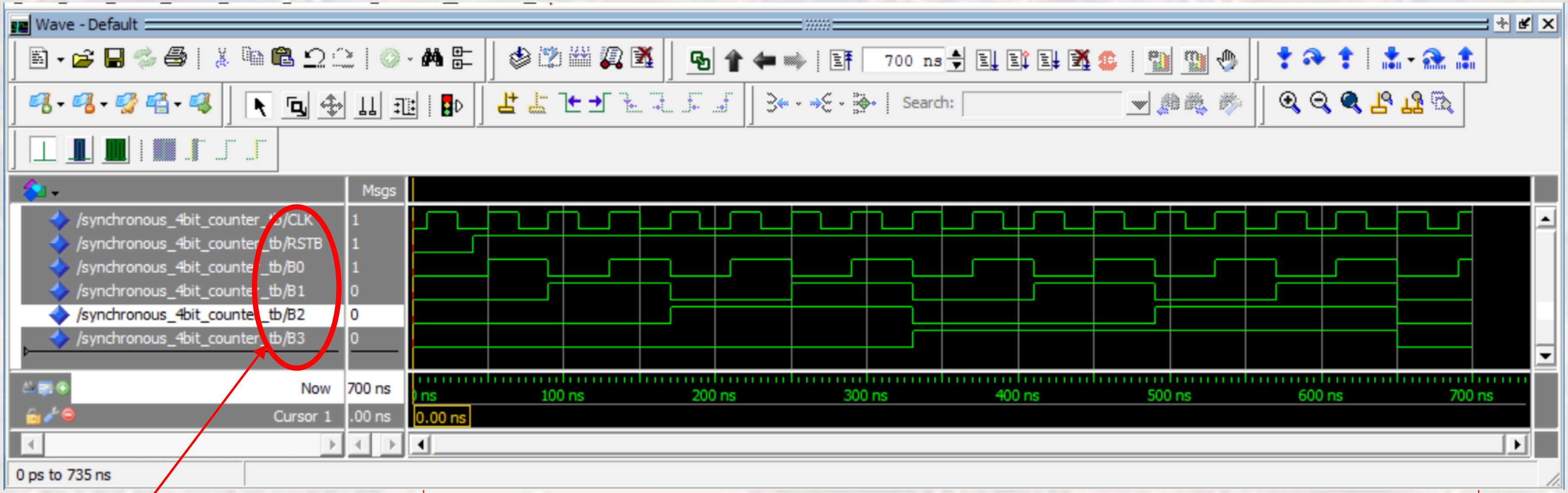
File Name	Library	HDL Version
synchron...		Default

Remove   Up   Down   Properties...

OK   Cancel   Help

ModelSim Testbench (schematic)

- Example: Counter – 4 bit
  - Run the simulation
    - Tools → Run Simulation Tool → RTL Simulation


Wave - Default
Search:
Msgs
/synchronous_4bit_counter_tb/CLK 1
/synchronous_4bit_counter_tb/RSTB 1
/synchronous_4bit_counter_tb/B0 1
/synchronous_4bit_counter_tb/B1 0
/synchronous_4bit_counter_tb/B2 0
/synchronous_4bit_counter_tb/B3 0
Now 700 ns
Cursor 1 .00 ns
0.00 ns
ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
0 ps to 735 ns

Signal names visible

Full simulation cycle plus restart visible

- Verify the waveforms