Last updated 1/19/21

These slides describe how to create testbench code

Upon completion: You should be able to write testbench code that allows visual verification of circuit operation via waveforms

- Testbench
 - VHDL file used to run simulations on a block(s)
 - Testbenches have:
 - One or more blocks instantiated in it
 - Signals to drive the block's inputs
 - Signals to sense the block's outputs
 - Additional simulation related code
 - Generate input signals (with timing) to drive the block
 - Note this makes the testbench un-synthesizable

- Testbench process
 - Create simulatable VHDL files (testbench) to drive the inputs of our design
 - Note: these are not synthesizable
 - Instantiate our design in the testbench file
 - Simulate the design
 - ModelSim is our simulation tool
 - View the resulting output waveforms and check for correctness

© tj

- Create Design
 - Arbitrary logic example
 - Set to top level, Elaborate and check for errors

```
_____
-- testbench_ex_hdl.vhdl
-- created 2/17/21
-- tj
-- rev 0
-- Simple design to show testbench operation
-- Inputs: rstb, clk, a, b, c
-- Outputs: q, foo
library ieee;
use ieee.std_logic_1164.all;
entity testbench_ex_hdl is
   port(
         i_clk:
                  in std_logic;
         i_rstb: in std_logic;
                  in std_logic;
         i_a:
         i_b:
                  in std_logic;
                 in std_logic;
         i_c:
                  out std_logic;
         o_q:
         o_foo: out std_logic
);
end entity;
```

```
rchitecture behavioral of testbench_ex_hdl is
   -- create a couple of itermediate signals
   signal one: std_logic;
   signal two: std_logic;
begin
   -- input logic processes
   logic1: process(i_a, i_b)
   begin
      if(i_a = '0' and i_b = '0') then
one <= '1';
      else
          one <= '0':
      end if;
   end process;
   logic2: process(i_c)
   begin
      case i_c is
    when '0' => two <= '1';
    when '1' => two <= '0';
    when others => two <= '0';</pre>
      end case:
   end process:
   -- flipflop process
   process(i_clk, i_rstb)
   begin
      if(i_rstb = '0') then
          o_q <= '0';
      elsif(rising_edge(i_clk)) then
          o_q <= two;
      end if:
   end process;
   -- simple output mapping
   o_foo <= one;
```



end architecture;

- Create component prototype
 - Rt-click the design file and select Create VHDL Component... .cmp file created in the p

🕤 Quartus Prime Lite Edi	ition - Z:/msoe_curren	t/21_Q3_CE1911,	/Projects/0	Class_Projects/G	ieneral_HDL/	/general_hdl - g	eneral_hdl
File Edit View Proje	ect Assignments	Processing To	ools Wir	ndow Help			
- 🖪 🔁	- 🗋 💼	って	general_h	dl	•	2 🎸	Š
Project Navigator	Files	▼ Q 📮 🗗 ×	👇 Con	npilation Report	t - general_h	dl 🗵 🛭 💠	testbench_e
reg_64_bit_tb.vhdl	l	^	7	1	• •	E 🖪 🖪	<mark>} </mark>
reg_64_bit.vhdl			1 F				
abo WHD reg_8_bit_en.vhdl			2		ch ox hd	1 vbd1	
abe reg_8_bit.vhdl			4	Lestber	icn_ex_nu	1. VIIUT	
abc counter_8bit_unsig	gned_tb.vhdl		5	created tj	2/17/21		
abo UHD counter_8bit_unsig	gned.vhdl		7	 rev 0			
counter_8bit_unsig	gned.cmp		9				
abo testbench (-		10			p show tes	tbench op
	Open						
Tasks	Remove File from Pro	ject				lk.a.b.	c
	Set as Top-Level Entit	у		Ctr	l+Shift+V	, u, o, .	-
	Create AHDL Include F	iles for Current	File				
Con	Create Symbol Files f	or Current File				_1164.all;	
· · · ·	Create Verilog Instant	iation Template I	Files for C	urrent File		k_hdl is	
	Create VHDL Compor	ent Declaration I	Files for Ci	urrent File		in std_log	ic;
						in std_log	ic;
<	Properties						
	🔽 📐 🔽 < <filte< th=""><th>si.>></th><th></th><th></th><th>💏 Fi</th><th>nd 💏 Fin</th><th>d Next</th></filte<>	si.>>			💏 Fi	nd 💏 Fin	d Next
	essage						
> 0 RI	unning Quartus	Prime Netli	ist Vie	vers Prepro	cess	-14	
	umber of proces	_npp genera sors bas no	u_ndi ·	-c general_ specified	_nuine which ma	tilst_type	=sgate erloading
	uartus Prime Ne	tlist Viewe	ers Pre	process was	success	ful. 0 err	ors, 1 wa
8							
System (1) Proce	essing (31)						
Create VHD&L Component	t Declaration Files for	Current File					
the struger as							

.cmp file created in the project directory Block_name.cmp \rightarrow component prototype

Copyright Your use o and other functions, (including associated to the ter Subscripti the Intel agreement, the sole p Intel and refer to t	t View Help (C) 2018 Inte f Intel Corpor software and t and any outpu device progra documentation ms and conditi on Agreement, FPGA IP Licens including, wi urpose of prog sold by Intel he applicable	el Corporat ration's de cools, and ut files fr mming or s n or inform tons of the the Intel ce Agreemen thout limi gramming lo or its aut agreement	tion. All rights reserved. esign tools, logic functions its AMPP partner logic rom any of the foregoing simulation files), and any nation are expressly subject e Intel Program License Quartus Prime License Agreement, it, or other applicable license itation, that your use is for ogic devices manufactured by chorized distributors. Please for further details.
Generated	by Quartus Pri Wed Feb 17 09	ime Version	n 18.1 (Build Build 625 09/12/2018
Generated Created on COMPONENT tes PORT (by Quartus Pri Wed Feb 17 09 tbench_ex_hdl	ime Version 0:45:47 202	n 18.1 (Build Build 625 09/12/2018 21
Generated Created on COMPONENT tes PORT (by Quartus Pri Wed Feb 17 09 tbench_ex_hdl i_clk	ime Version 0:45:47 202 :	IN STD_LOGIC;
Generated Created on COMPONENT tes PORT (by Quartus Pri Wed Feb 17 09 tbench_ex_hdl i_clk i_rstb i a	ime Version 0:45:47 202 : : :	IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC;
Generated Created on COMPONENT tes PORT (by Quartus Pri Wed Feb 17 09 tbench_ex_hdl i_clk i_rstb i_a i_b	ime Version 0:45:47 202 : : : : :	IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC;
Generated Created on COMPONENT tes PORT (by Quartus Pri Wed Feb 17 09 tbench_ex_hdl i_clk i_rstb i_a i_b i_c	ime Version 0:45:47 202 : : : : : : : : : : :	IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC;
Generated Created on COMPONENT tes PORT (by Quartus Pri Wed Feb 17 09 tbench_ex_hdl i_clk i_rstb i_a i_b i_c o_q	ime Version 0:45:47 202 : : : : : : : : : :	IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; OUT STD_LOGIC;
Generated Created on COMPONENT tes PORT (by Quartus Pri Wed Feb 17 09 tbench_ex_hdl i_clk i_rstb i_a i_b i_c o_q o_foo	ime Version 0:45:47 202 : : : : : : : : : :	IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; OUT STD_LOGIC; OUT STD_LOGIC;
Generated Created on COMPONENT tes PORT ();	by Quartus Pri Wed Feb 17 05 tbench_ex_hdl i_clk i_rstb i_a i_b i_c o_q o_foo	ime Version 0:45:47 202 : : : : : : : : : :	IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; IN STD_LOGIC; OUT STD_LOGIC; OUT STD_LOGIC;

Create Testbench Code

- Name your testbench the same as your design file but add _tb to the end
 - counter_8bit.vhdl → counter_8bit_tb.vhdl
 - shift_reg_nbit.vhdl → shift_reg_nbit_tb.vhdl
- When we start using the DE10 we will do the same thing
 - Name your DE10 implementation the same as your design file but add _de10 to the end
 - counter_8bit.vhdl → counter_8bit_de10.vhdl
 - shift_reg_nbit.vhdl → shift_reg_nbit_de10.vhdl
- In both cases we never change the base design file
 - Ensures what we designed is what we simulate, and what we simulated is what we build



- Create Testbench Code
 - Description and entity



Description

Note – entity statement required nothing in it

testbench name

Create Testbench Code

- Testbench signals
- Device to test



testbench is the type (can call it anything you want) testbench name

> Names used in the testbench for all inputs and outputs to your design – I use all caps and remove the o_ and i_

The period of 1 clock cycle on the DE10 (used later to make things easier)

Prototype for the component – type it in or let the tool generate it for you

Instantiation of the component – type it in (connect DUT wires to testbench wires)

DUT stands for device under test – you can label it anything you want

);

component signals

testbench signals

- Create Testbench Code
 - Test processes



Process for the clock signal Runs continuously

Process for the reset_bar signal Runs once then waits forever

Run process – cycles through a and c then waits forever

Run process – cycles b and runs forever

CE 1911

- Create Testbench Code
 - The wait command



wait for xx causes the process block or construct to continue running (looping)

Wait with no time causes the process block or construct to remain in its current state - signals no longer change

- Quartus Interaction and ModelSim
 - Test benches can contain un-synthesizable code
 - Quartus will generate errors on compilation if you set the testbench to the top level entity
 - Use the block under test as the top level entity

- Elaborate your design
 - Run Analysis and Elaboration
 - Do not do a full compile, it is a waste of time
 - Make sure your design (NOT the testbench) is selected as the top level block)
 - Checks the design and testbench for errors (checks everything in the project)
 - Creates the mathematical models of your design used in simulation



- Prepare for simulation
 - Provides ModelSim with the required information to run your testbench
 - Assignments \rightarrow Settings \rightarrow EDA Tool Settings \rightarrow Simulation

General Simulation Files Specify option: for some time output files for use with other EDA tools. Libraries Specify option: for some time output files for use with other EDA tools. IP Catalog Search Locations Tool Inne: ModelSim-Altera Design Templates Image tellever with the entional cally after compilation Votage EDA Netlist Writer settings Temperature Output girectory: simulation/methic * Compilation Process Settings Output girectory: simulation/methic * DA Tool Strings Output girectory: simulation/methic * De Tool Strings Map illegal HDL characters	Gread File Ubdated Set Value Posting Staff Industors P Set Ings Posting Staff Industors Set Value Posting Staff Industors Operating Staff Industors Time goale Dougs Operating Staff Industors Diago Dougs Tom Posters Staff Industors Diago Dougs Tom Posters Del Tod Staffing Not Industors Diago Dougs Tom Posters Staff Industors Staff Industors Staff Industors Staff Industors
Device transferies Simulation Board-Level Complex Strings VHD. Input Design Instance name Design Instance name More EDA Netlist Writer Settings Signal Tap Logic Analyzer Logic Analyzer SSN Analyzer More NativeLink Settings Source Test Descenter Settings Source Test Descenter Settings Settings Settings Settings Settings Settings Settings Settings Settings Reset	

14

- Prepare for simulation
 - Provides ModelSim with the required information to run your testbench

isting test be	nch settings:			<u>N</u> ew	
Name	Top Level Module Design Ir	nstance Run For	Test Bench File(s)	<u>E</u> dit	
				<u>D</u> elete	
			OK Cance		
			OK Cance	.::	

Prepare for simulation

 Provides ModelSim with the required information to run your testbench

Image: Mew Test Bench Settings X	
Create new test bench settings.	Type in name of your testbench file
Test bench name: testbench_ex_hdl_tb	no .vhdl extension
Top level <u>m</u> odule in test bench: testbench_ex_hdl_tb	
Use test bench to perform VHDL timing simulation	Automatically filled in
Design instance name in test bench: NA	
Simulation period	
\bigcirc Run simulation until all <u>v</u> ector stimuli are used	
● <u>E</u> nd simulation at: 500 ns ▼	Select End simulation at button
Test bench and simulation files	Enter how long you want the simulation to run
Eile name:Add	
File Name Library HDL Version Remove	Select
Цр	
Down	
Properties	
OK Cancel Help	
OK Cancel Hetp	

- Prepare for simulation
 - Provides ModelSim with the required information to run your testbench

Select File	>	
Look in:	Z:\msoe_current\21_Q3_CE1lass_Projects\General_HDL	Select your testbench file Be careful not to get the .bak version
File <u>n</u> ame: te	stbench_ex_hdl_tb.vhdl	Select Open
Files of type: Al	l Files (*) 🔻 Cancel	

- Prepare for simulation
 - Provides ModelSim with the required information to run your testbench

- New Test Bench Settings X	
Create new test bench settings.	
Test bench name: testbench_ex_hdl_tb	and the second se
Top level <u>m</u> odule in test bench: testbench_ex_hdl_tb	and a state of the
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: 500 ns ▼ 	Select Add
Eile name: testbench_ex_hdl_tb.vhdl Add File Name Library HDL Version Remove	
<u>⊔</u> p Down	
Properties	
OK Cancel Help	

- Prepare for simulation
 - Provides ModelSim with the required information to run your testbench

	Specify settings fo	r each test bench.	2		
	Existing test bench	n settings:			<u>N</u> ew
🛃 New Test Bench Settings 💦 🔪	< Name	Top Level Module Design Instance	Run For	Test Bench File(s)	<u>E</u> dit
Create new test bench settings.	mod_10_count	mod_10_count NA	1000 ns	mod_10_counterA_tb.vhd	Delete
[est bench name: testbench ex hdl tb	shift_reg 8 bit	testbench_ex_h NA	500 ns	testbench_ex_hdl_tb.ybc	a di seconda
op level <u>m</u> odule in test bench: testbench_ex_hdl_tb					
Use test bench to perform VHDL timing simulation					
Design instance name in test bench: NA				OK Cancel	Help
Simulation period					
O Run simulation until all vector stimuli are used	Select (C Settings - general_h	dl		- • ×
● <u>E</u> nd simulation at: 500 ns ▼		Category: General	Simula	ion	Device/Board
Test bench and simulation files		Files Libraries	Specify	options for generating output files for u	able testbenches are be
File parer		✓ IP Settings IP Catalog Sea Decime Templater	rch Locations	gate-level simulation automatically after Select th	ne one you want to run
Add		Voperating Setting: Voltage	s and Conditions EDA	Vetlist Writer settings	
Eile Hame Library HDL Version Remove		Temperature Compilation Proce	Eorn ess Settings Outp	at for output netlist: VHDL Time scale: Ut directory: simulation/modelsim	100 us *
testbench_ex_h Default Up		Incremental C EDA Tool Settings	ompilation 🗆 M	ap illegal <u>H</u> DL characters <u>E</u> nable glitch filt	tering
Down		Sign Entry/Simeration	synthesis Op	Generate Value Change Dump (VCD) file script Script Settings	
		 Compiler Settings VHDL Input 		Design instance name:	
Properties		Verilog HDL In Default Param	put <u>M</u> ore	DA Netlist Writer Settings	
OK Const. Usin		3 Timing Analyzer Assembler	Nativ	eLink settings	
OK Cancel Help		Design Assistant Signal Tap Logic A	nalyzer 🔍 🤅	omplinest bench: testbench_ex_hdl_tb	▼ Test Benches
The second secon		Logic Analyzer Int Power Analyzer Se	erface E	Use script to st up simulation:	
		SSN Analyzer	Os	cript to compile test on ch:	
			More	lativeLink Settings	Reset
	10				
	19				C

- Start ModelSim
 - Tools → Run Simulation Tool → RTL Simulation



- Start ModelSim
 - ModelSim will start in a new window
 - Often is starts minimized so you need to select it from the toolbar

ModelSim - INTEL FPGA STARTER EDITION 10.5b	– 🗆 X
<u>File E</u> dit <u>V</u> iew <u>C</u> ompile <u>S</u> imulate A <u>d</u> d W <u>a</u> ve T <u>o</u> ols Layo <u>u</u> t Boo <u>k</u> marks <u>W</u> indow <u>H</u> elp	
│ B - ☞ 🖬 ☜ 🚭 ¾ 🖻 ♀ ♀ ◎ - 桷 目: 兩 │ ଅ 🛱 繰 🖄 │ 🗣 含 🖛 → B 🔽 ○ ps 🗄 L B: L B: S () 🖄 🖄 🖑 │ 🕇 🏞 🕇 ∦	1 - 🔐 🏦
Layout Simulate 🔽 🛛 ColumnLayout AllColumns 💌 🛛 🖏 - 🖏 - 🦃 🕋 - 🥵 🗍 I 💿 IKO 🛐 🚛 🎉	N 🖪 🕸 💷 💷 🖪
╡ <u> <u> </u> </u>	
🌠 sim - Default : 🖞 🖻 🔟 🔁 Objects : ::::::::::::::::::::::::::::::::::	:
Tinstance Design unit Design unit type Top Category Visibility Total coverage Visibility Total coverage Manage Manag	s
<pre>comparison testbench Architecture DU Instance +acc= dut testbench Architecture DU Instance +acc= dock testbench Process - +acc= dock testbench Process - +acc= ac testbench Process - +acc= bonly testbench Process - +acc= standard standard Package Package +acc= stal_logic_1164 std_logic_1 Package Package +acc= std_logic_1164 std_logic_1 Package Package +acc= foo year and y</pre>	499.2ns
In Iranscript	<u>+ er</u> ×
<pre># view signals # .main_pane.objects.interior.cs.body.tree # run 500 ns VSIM 2></pre>	- -
Now: 500 ns Delta: 2 sim:/testbench_ex_hdl_tb	499050 ps to 499268 ps

21

- Simulation Results
 - Resize windows as desired
 - Rt-click and select Zoom Full on the waveform window

ModelSim - INTEL FPGA STARTER EDITION 10.5b	– 🗆 X
<u>File E</u> dit <u>V</u> iew <u>C</u> ompile <u>S</u> imulate A <u>d</u> d W <u>a</u> ve T <u>o</u> ols Layo <u>u</u> t Boo <u>k</u> marks <u>W</u> indow <u>H</u> elp	
│ B - ☞ 🖬 🗇 🚭 │ X 🖻 🛍 그 그 │ ○ - 桷 🗄 🗰 🖯 🕸 🖄 🖉 🖉 🖾 🛛 🕞 🎓 🛊 │ 🚱 🛊 🖛 ⇒ │ BT 💿 ps 🛨 🖳 B≱ 🔀 @ │ 🎬 💁 🚸 │ 🕇 🏞 🛊 │ 🛣	t - 😤 🏦
Layout Simulate 🗾 ColumnLayout AllColumns 💌 🛛 🐔 - 🖓 🖆 - 🦃 🗍 I 💿 TIO 🗔 🚛 🎾	
灶 ≟ Ҽ Ҽ ҄ ҍ ヱ Ӻ 垂 Э+ - +€ - Э- Search: 📃 🚽 @ @ @ @ @ @ @ @ @ @ @ @ @ @ [및 ■ ■ ■ ■ ■ ■	
②[t::::::::::::::::::::::::::::::::::::	
Instruction Instruction Instruction Instruction Instruction Image: Instruction Image: Instruction Image: Instruction Image: Instruction Image: Instruction Image: Instruction Image: Instruction Image	400 ns 450 ns 500 ns
Transcript Transcript Start time: 10:37:03 on Feb 17,2021 Loading std.textio(body) Loading idee.std_logic_1164(body)	<u> </u>
[*] Loading work.testbench_ex_hdl_tb(t [*] Loading work.testbench_ex_hdl(beha Make sure you can see	
<pre># add wave * # view structure # .main_pane.structure.interior.cs.b the signal names # .main_pane.objects.interior.cs.body.tree # run 500 ns</pre>	
VSIM 2>	-
Now: 500 ns Delta: 2 C	0 ps to 525 ns
22	

- Simulation Results
 - Move around in the simulation window



- Verify Design
 - Check critical sections to prove design works



Run Process Examples

Up/dn counter – DIR changes the direction



Run Process Examples

8 bit shift register (L/R based on DIR) (shift/no-shift based on SHIFT)

```
run: process
                   -- no sensitivity list
   begin
      -- Initialize inputs
      D <= '0';
SHIFT <= '0';
                                  Initialize the input signals
      DIR <= '0':
      wait for 2*PER; -- wait for reset | Wait for reset to complete
      -- verify no shift
      D <= '1'; wait for PER;
                                       No-shift
      D <= '0': wait for PER;
      -- verify shift lt
      SHIFT <= '1':
      D <= '1'; wait for 8*PER;</pre>
                                       Shift 1 way – enough times to test all bits
      D \le '0'; wait for 8*PER:
      -- verify shift rt
      DIR <= '1
      D <= '1'; wait for 8*PER;
                                    Shift the other way – enough times to test all bits
      D <= '0'; wait for 8*PER;
end process run;
                                    No 'wait' at the end – repeats the whole process
-- End test processes
```

Run Process Examples

Multiple RUN process – used for independent signals

```
-- Run Processes
n_s: process
begin
   TNS <= '0';
  wait for 10*PER;
  TNS <= '1';
   wait for 15*PER;
   TNS <= '0';
  wait for 15*PER;
end process n_s;
ew: process
begin
   TEW <= '0';
  wait for 13*PER;
   TEW <= '1';
   wait for 7*PER;
end process ew;
-- End test processes
```

```
Independent N/S signal
```

```
Independent E/W signal
```