CE 1911

Dr. Johnson

Introduction

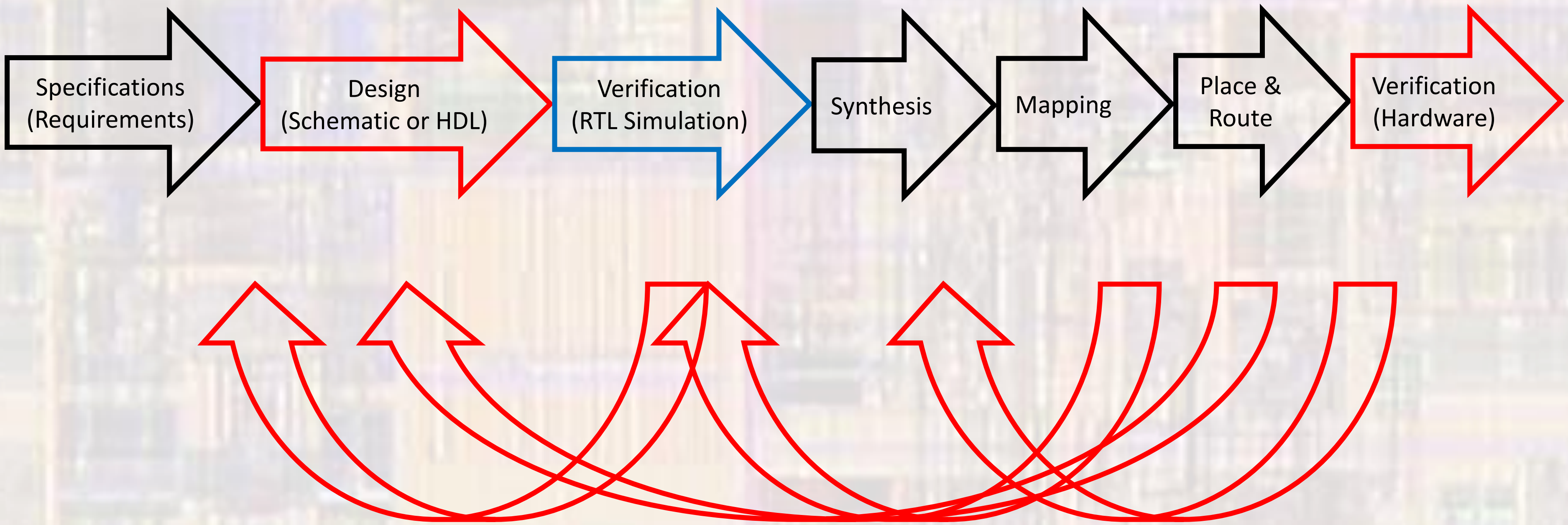
Last updated 3/5/21

CE1911 Intro

- Major topics
  - Synchronous logic design
    - Gate Level Design
    - HDL Design
  - Digital design verification
    - Test benches
    - FPGA implementation
  - Digital system design
    - Finite State Machines
    - Data Paths
    - Single Cycle Processor

CE1911 Intro

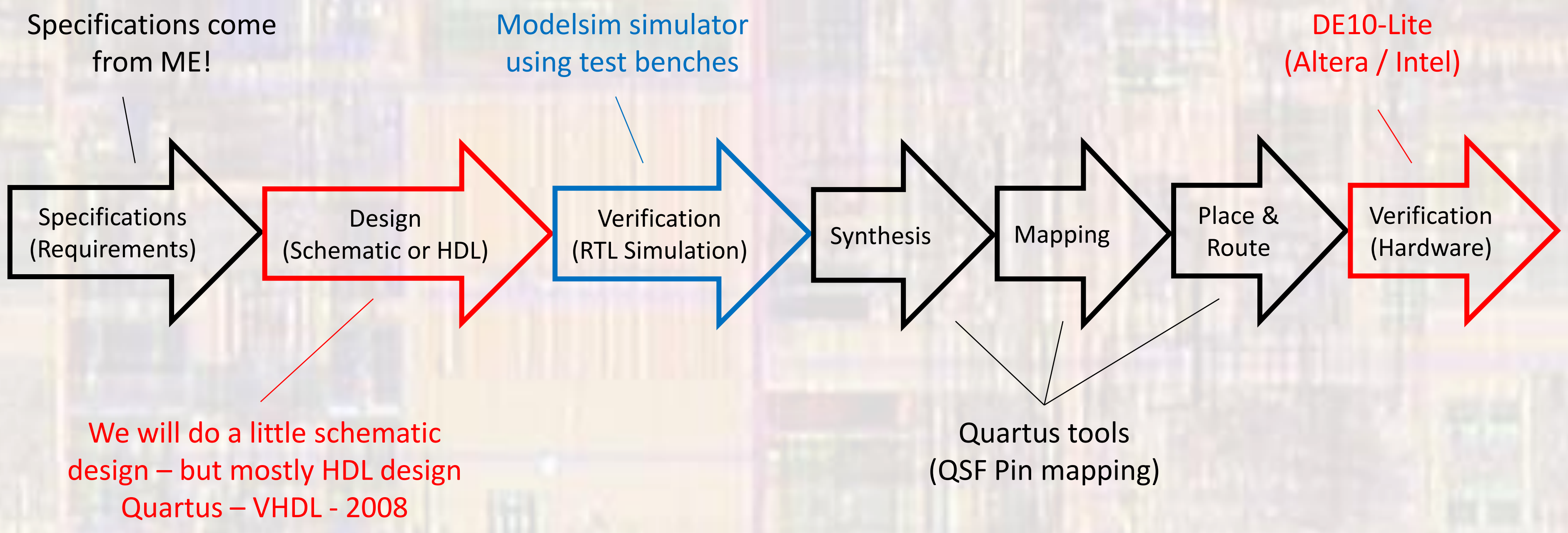
- Digital Design Flow


Specifications (Requirements)
Design (Schematic or HDL)
Verification (RTL Simulation)
Synthesis
Mapping
Place & Route
Verification (Hardware)

Feedback all through the process

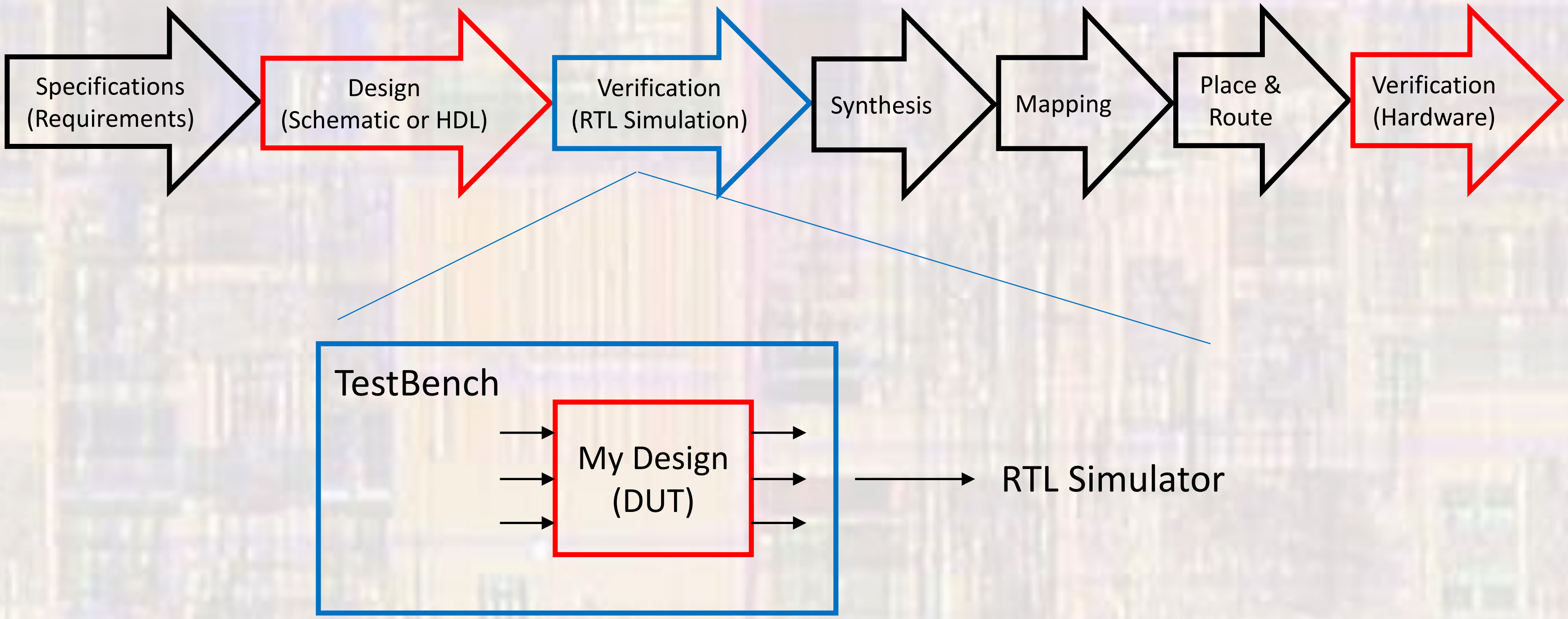
CE1911 Intro

- Digital Design Flow


Specifications come from ME!
Modelsim simulator using test benches
DE10-Lite (Altera / Intel)
Specifications (Requirements)
Design (Schematic or HDL)
Verification (RTL Simulation)
Synthesis
Mapping
Place & Route
Verification (Hardware)
We will do a little schematic design – but mostly HDL design Quartus – VHDL - 2008
Quartus tools (QSF Pin mapping)

Test Bench Flow

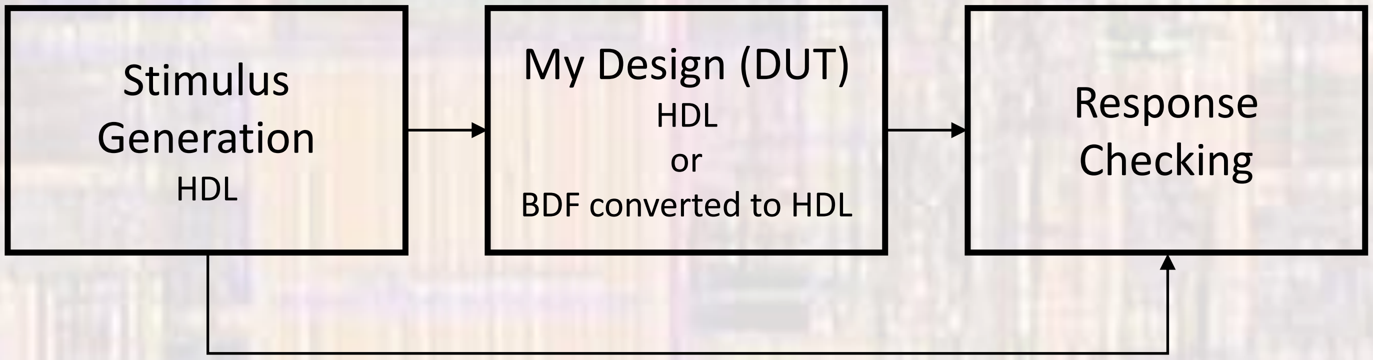
- Design Flow



DUT = Device Under Test

Test Bench Flow

- Test Bench Concept

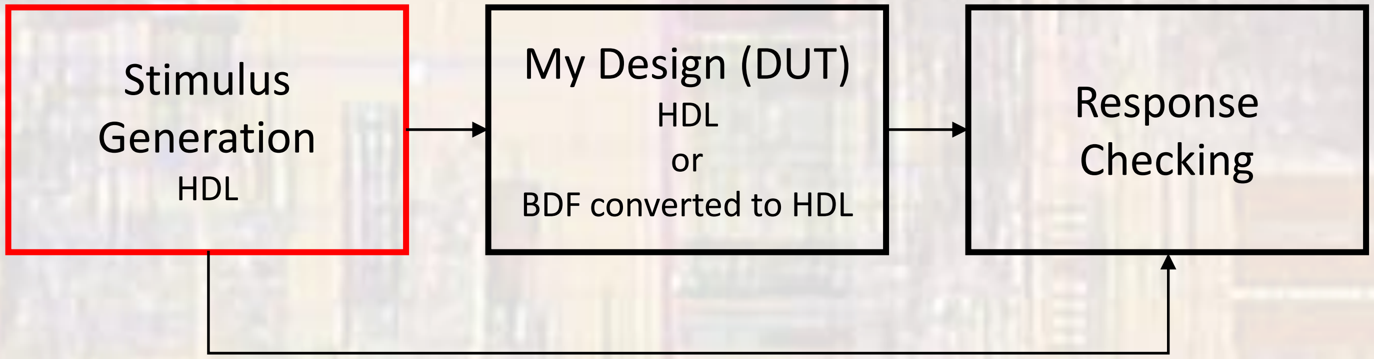


Test Bench Flow

- Test Bench Concept

  - Stimulus Generation
    - Brute Force Stimulus – Enter combinations by hand
    - Automated Stimulus – Create combinations via code

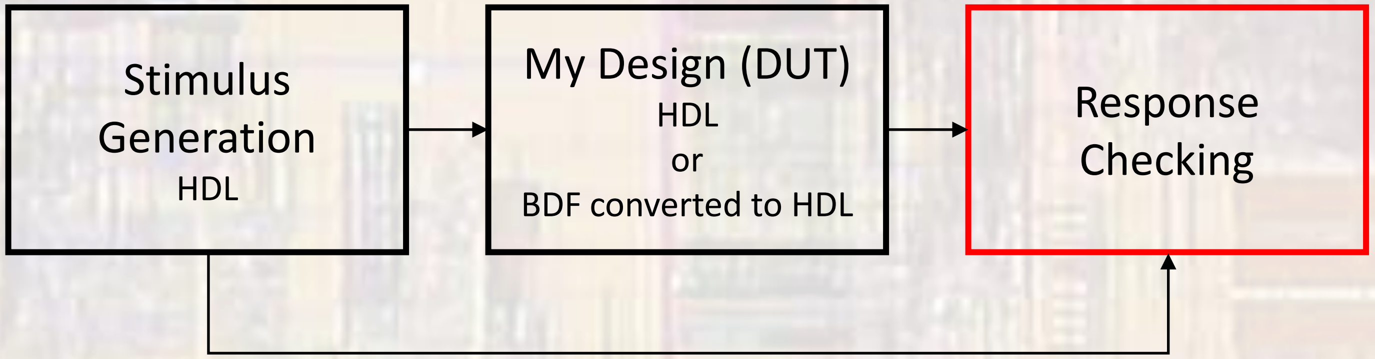
    - Exhaustive – All possible input combinations tested
    - Directed Stimulus – Test only specific input combinations



Test Bench Flow

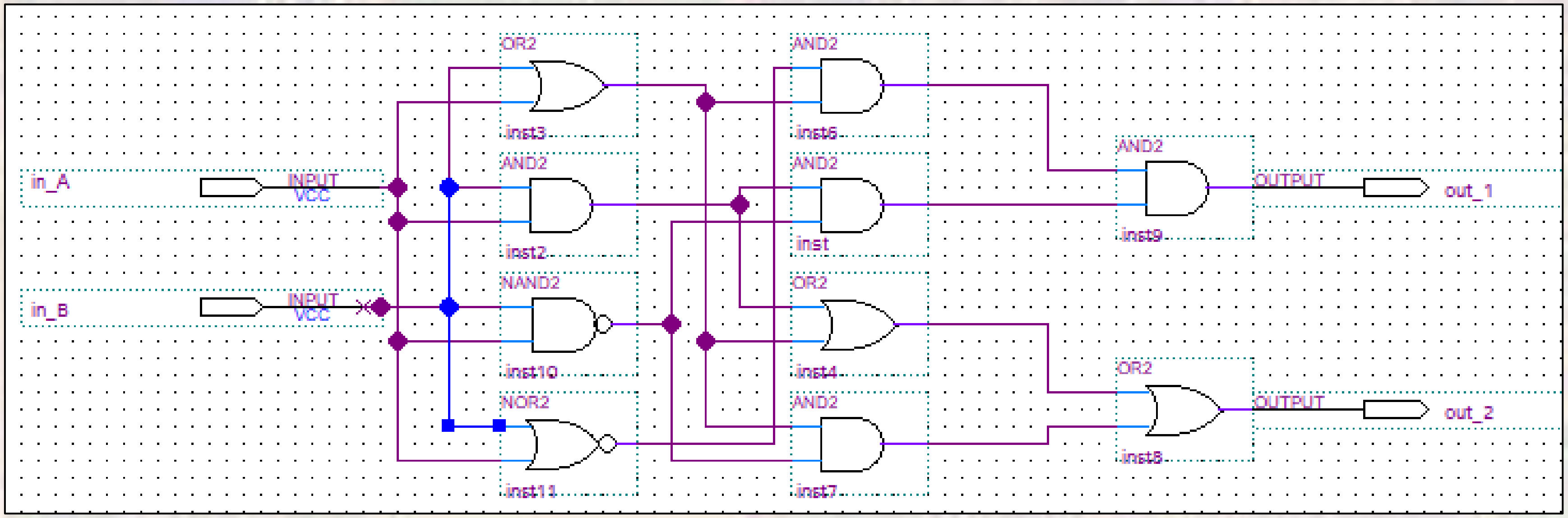
- Test Bench Concept

  - Response Checking
    - Waveform Interpretation
    - Simulation Checking
      - Compare expected outputs to actual outputs
      - Can be brute force or automated
    - Decoupled Simulation Checking
      - Compare actual results to independently created expected results



Test Bench Flow

- Example: random logic
    - Create schematic (bdf)



We will only create schematics for educational purposes
Real world digital design is done almost entirely via an HDL

Test Bench Flow

- Example: random logic
  - Create a VHDL file for your schematic
  - Select File → Create/Update → Create HDL Design File from Current File

```
-- Copyright (C) 2017  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 FPGA IP 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 17.1.0 Build 590 10/25/2017 SJ Lite Edition"
-- CREATED		"Fri Feb 23 15:52:47 2018"

LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY work;

ENTITY MyFirstLogic IS
	PORT
	(
		in_A :  IN  STD_LOGIC;
		in_B :  IN  STD_LOGIC;
		out_1 :  OUT  STD_LOGIC;
		out_2 :  OUT  STD_LOGIC
	);
END MyFirstLogic;

ARCHITECTURE bdf_type OF MyFirstLogic IS

SIGNAL	SYNTHESIZED_WIRE_12 :  STD_LOGIC;
SIGNAL	SYNTHESIZED_WIRE_13 :  STD_LOGIC;
SIGNAL	SYNTHESIZED_WIRE_14 :  STD_LOGIC;
SIGNAL	SYNTHESIZED_WIRE_4 :  STD_LOGIC;
SIGNAL	SYNTHESIZED_WIRE_8 :  STD_LOGIC;
SIGNAL	SYNTHESIZED_WIRE_9 :  STD_LOGIC;
SIGNAL	SYNTHESIZED_WIRE_10 :  STD_LOGIC;
SIGNAL	SYNTHESIZED_WIRE_11 :  STD_LOGIC;

BEGIN 

SYNTHESIZED_WIRE_11 <= SYNTHESIZED_WIRE_12 AND SYNTHESIZED_WIRE_13;

SYNTHESIZED_WIRE_13 <= NOT(in_A AND in_B);

SYNTHESIZED_WIRE_4 <= NOT(in_A OR in_B);

SYNTHESIZED_WIRE_12 <= in_B AND in_A;

SYNTHESIZED_WIRE_14 <= in_A OR in_B;

SYNTHESIZED_WIRE_9 <= SYNTHESIZED_WIRE_14 OR SYNTHESIZED_WIRE_12;

SYNTHESIZED_WIRE_10 <= SYNTHESIZED_WIRE_4 AND SYNTHESIZED_WIRE_14;

SYNTHESIZED_WIRE_8 <= SYNTHESIZED_WIRE_14 AND SYNTHESIZED_WIRE_13;

out_2 <= SYNTHESIZED_WIRE_8 OR SYNTHESIZED_WIRE_9;

out_1 <= SYNTHESIZED_WIRE_10 AND SYNTHESIZED_WIRE_11;

END bdf_type;
```

Test Bench Flow

- Example: random logic
  - Create a component template for your design (DUT)
    - Select 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
-- 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.


-- Generated by Quartus Prime Version 16.1 (Build Build 196 10/24/2016)
-- Created on Wed Feb 21 11:07:19 2018
```

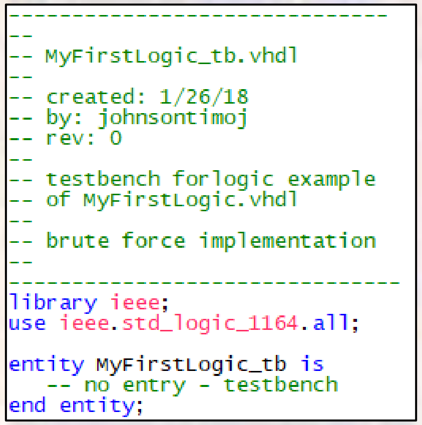
```
COMPONENT logic_schematic_example
    PORT
    (
        in_A        :    IN STD_LOGIC;
        in_B        :    IN STD_LOGIC;
        out_1       :    OUT STD_LOGIC;
        out_2       :    OUT STD_LOGIC
    );
END COMPONENT;
```

Test Bench Flow

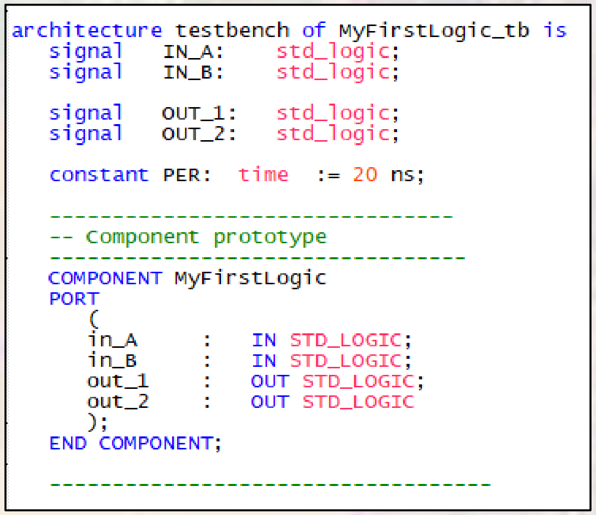
- Example: random logic
  - Prepare to use the VHDL file
    - Remove the BDF file from the project
      - In project Navigator – Files, Right click the BDF file and select: Remove file from project
    - Add the created VHDL file to the project
      - Select Project → Add/remove files in project → 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

Test Bench Flow

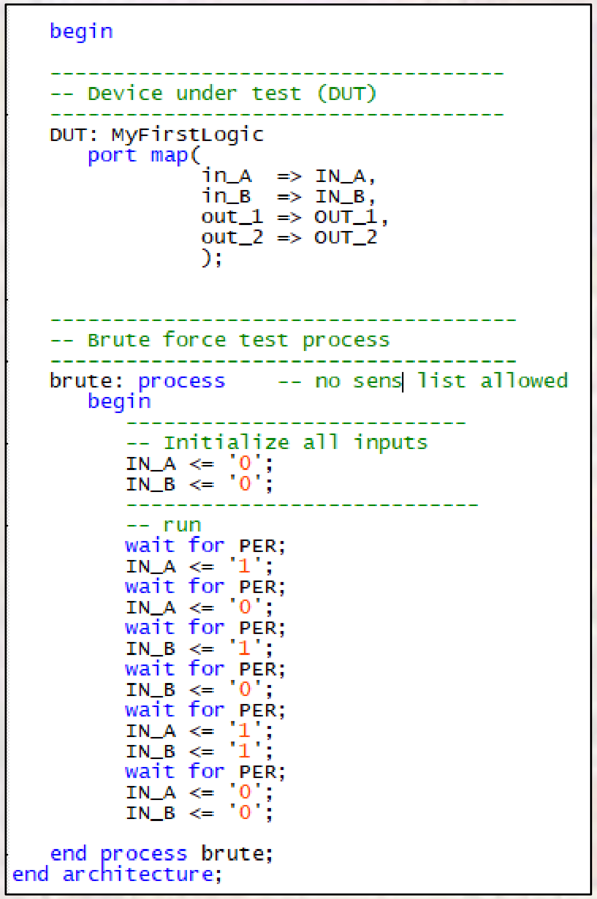
- Example: random logic
  - Create a test bench



```
-------------------------------
--
-- MyFirstLogic_tb.vhdl
--
-- created: 1/26/18
-- by: johnsontimoj
-- rev: 0
--
-- testbench forlogic example
-- of MyFirstLogic.vhdl
--
-- brute force implementation
--
--------------------------------
library ieee;
use ieee.std_logic_1164.all;

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



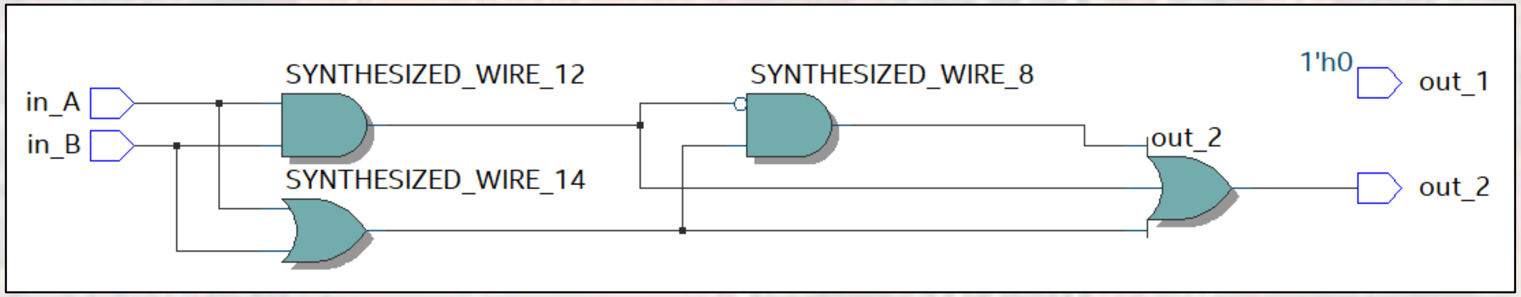
```
architecture testbench of MyFirstLogic_tb is
   signal    IN_A:     std_logic;
   signal    IN_B:     std_logic;

   signal    OUT_1:    std_logic;
   signal    OUT_2:    std_logic;

   constant PER:   time   := 20 ns;

   ---------------------------------
   -- Component prototype
   ----------------------------------
   COMPONENT MyFirstLogic
   PORT
      (
      in_A      :    IN STD_LOGIC;
      in_B      :    IN STD_LOGIC;
      out_1     :    OUT STD_LOGIC;
      out_2     :    OUT STD_LOGIC
      );
   END COMPONENT;

   ------------------------------------
```



```
   begin

   ------------------------------------
   -- Device under test (DUT)
   ------------------------------------
   DUT: MyFirstLogic
      port map(
            in_A   => IN_A,
            in_B   => IN_B,
            out_1  => OUT_1,
            out_2  => OUT_2
            );


   -------------------------------------
   -- Brute force test process
   -------------------------------------
   brute: process     -- no sens list allowed
      begin
         --------------------------
         -- Initialize all inputs
         IN_A <= '0';
         IN_B <= '0';
         ---------------------------
         -- run
         wait for PER;
         IN_A <= '1';
         wait for PER;
         IN_A <= '0';
         wait for PER;
         IN_B <= '1';
         wait for PER;
         IN_B <= '0';
         wait for PER;
         IN_A <= '1';
         IN_B <= '1';
         wait for PER;
         IN_A <= '0';
         IN_B <= '0';

   end process brute;
end architecture;
```

Test Bench Flow

- Example: random logic
  - Elaborate the design (create RTL)
    - Processing → Start → Start Analysis and Elaboration
  - Check the RTL
    - Tools → Netlist Viewers → RTL Viewer

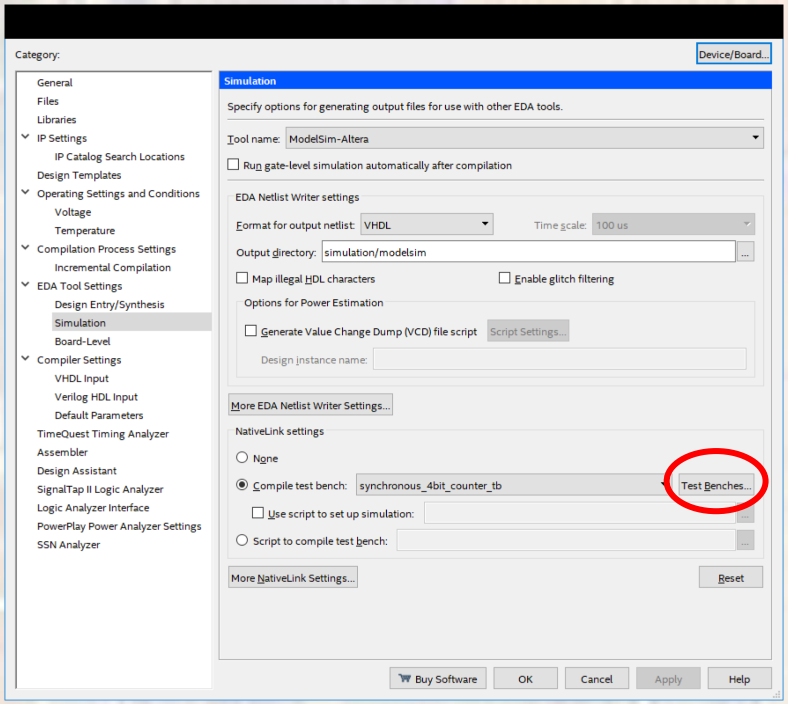


Test Bench Flow

- Example: random logic
  - Setup the test bench
    - Select 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



Category:

- 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

Design instance name:

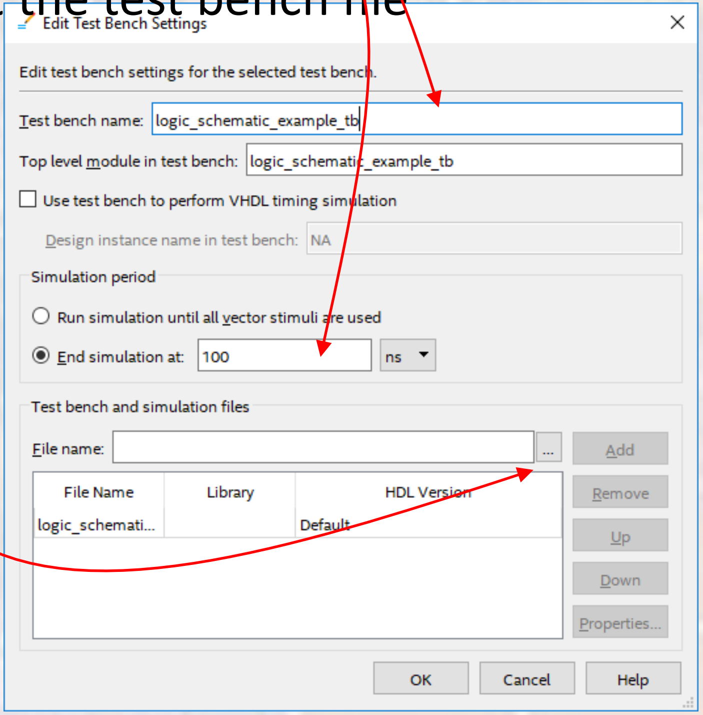
NativeLink settings

○ None

● Compile test bench: synchronous_4bit_counter_tb

☐ Use script to set up simulation:

○ Script to compile test bench:



Edit Test Bench Settings

Edit test bench settings for the selected test bench.

Test bench name: logic_schematic_example_tb

Top level module in test bench: logic_schematic_example_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: 100 ns

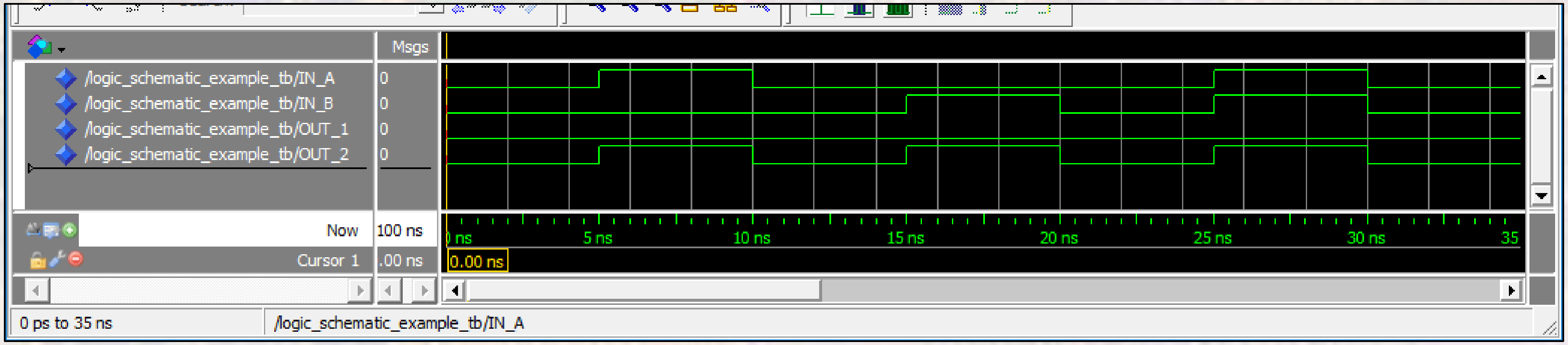
Test bench and simulation files

File name:

File Name	Library	HDL Version
logic_schemati...		Default

Test Bench Flow

- Example: random logic
  - Run the simulation
    - Select Tools → Run Simulation Tool → RTL Simulation



	Msgs
/logic_schematic_example_tb/IN_A	0
/logic_schematic_example_tb/IN_B	0
/logic_schematic_example_tb/OUT_1	0
/logic_schematic_example_tb/OUT_2	0
Now	100 ns
Cursor 1	.00 ns

$\underbrace{\hspace{10em}}_{\text{1 full cycle}}$

    - Verify the waveforms