

Quartus Counter Example

Last updated 7/24/23

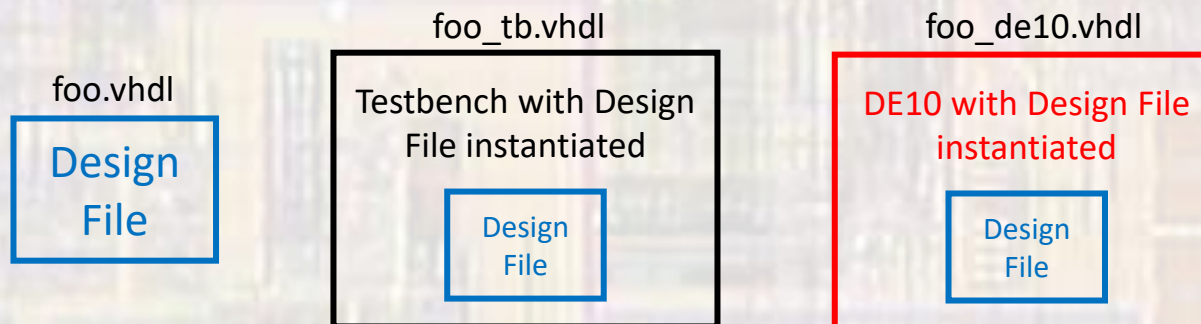
Quartus Counter Example

- Create a logic design from start to a DE10 implementation
 - This example uses “best design practices”
 - This example is not about creating HDL
 - The HDL code will be provided without explanation

Quartus Counter Example

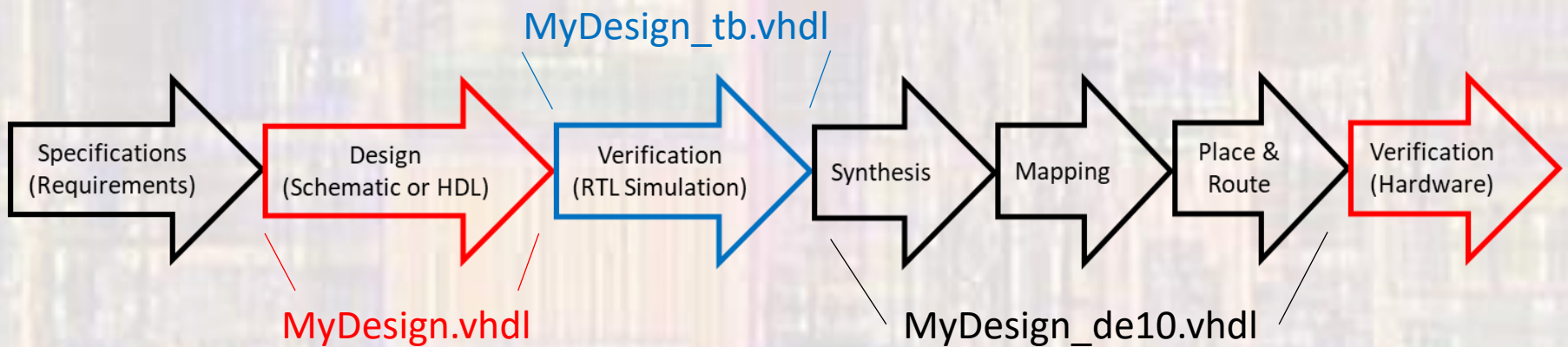
- Design Hierarchy

- 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



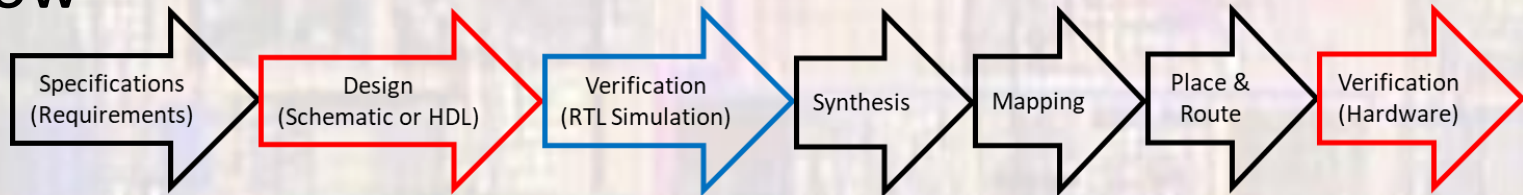
Quartus Counter Example

- Flow



Quartus Counter Example

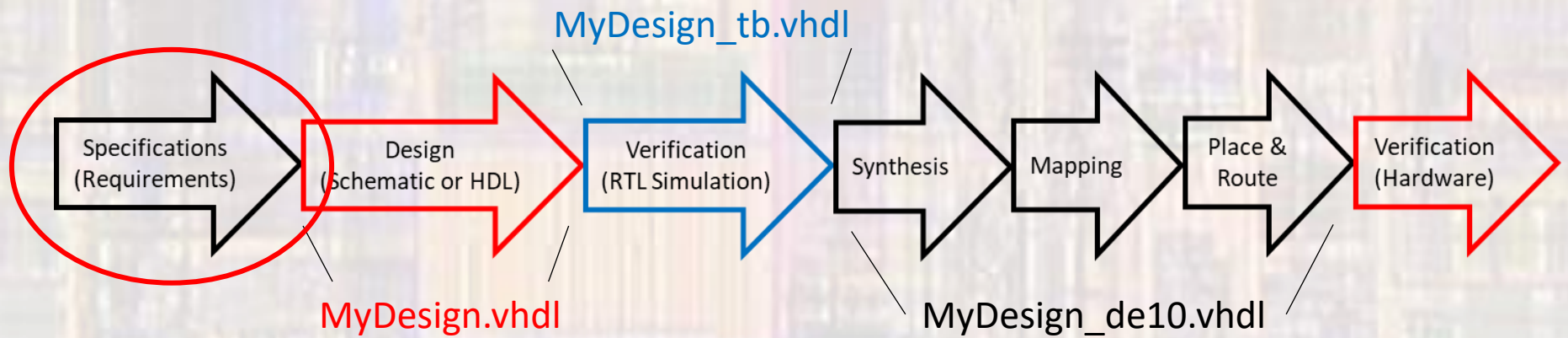
- Flow



- Create project
- Create HDL code
- Verify RTL visually
- Create testbench
- Verify operation
- Create DE10 implementation HDL
- Compile
- Verify RTL visually
- Program DE10 board
- Verify operation manually

Quartus Counter Example

- Flow

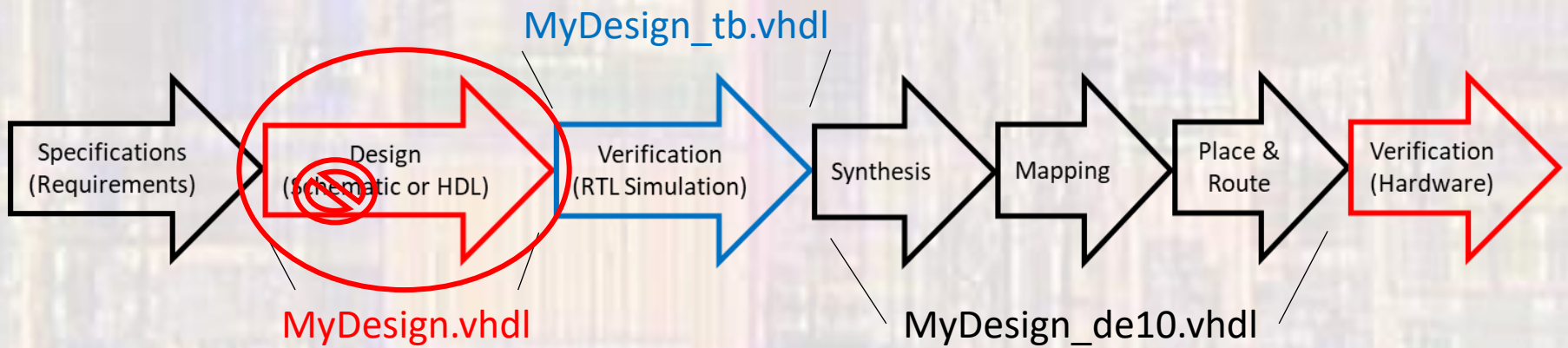


Quartus Counter Example

- Create a counter
 - Binary wrapping count
 - N bits
 - Up/down
- Test the design using ModelSim
- Implement an 8 bit version on the DE10
 - Reset and direction → switches
 - Count value → LEDs
 - Clock divider → 3Hz operation

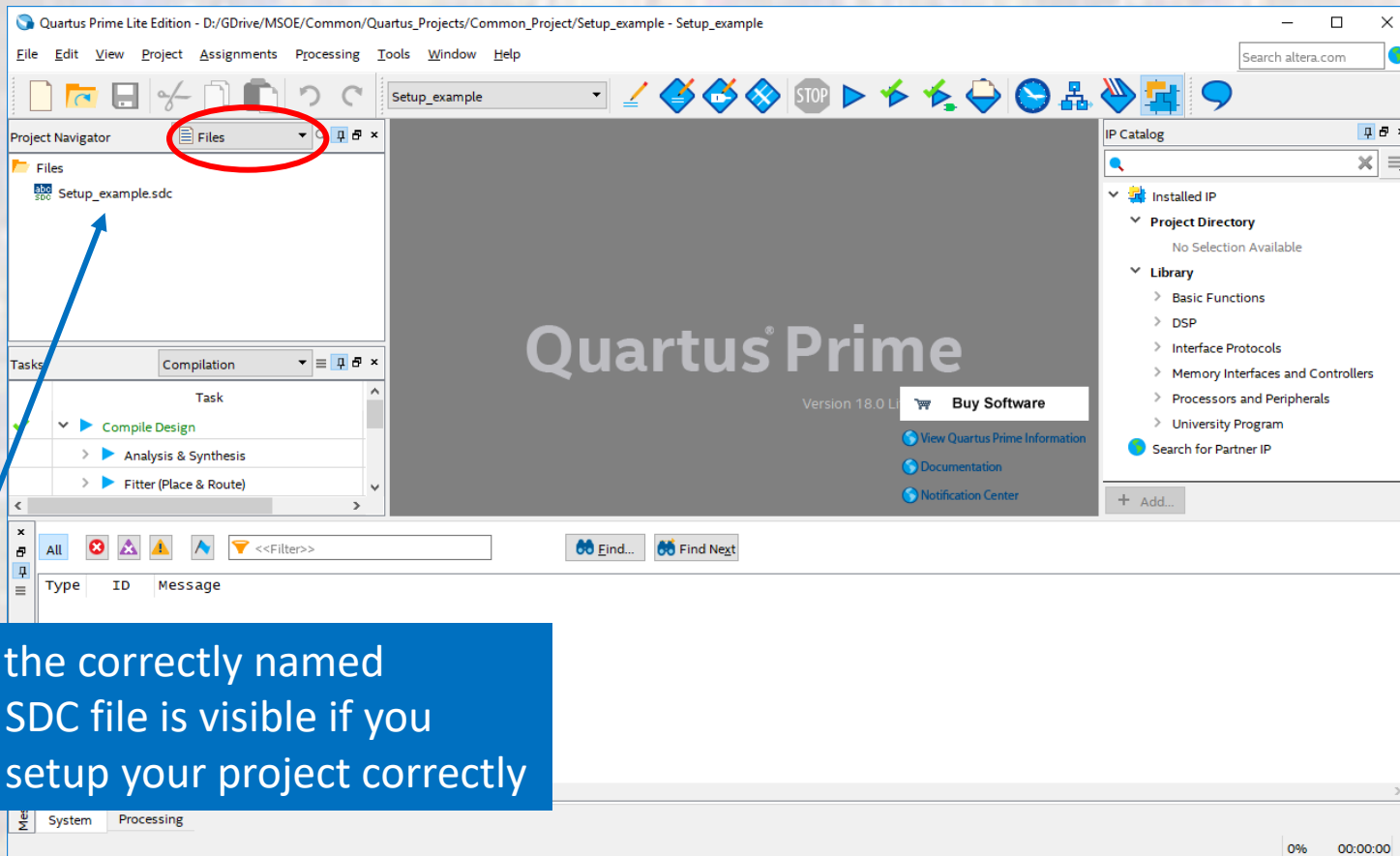
Quartus Counter Example

- Flow



Quartus Counter Example

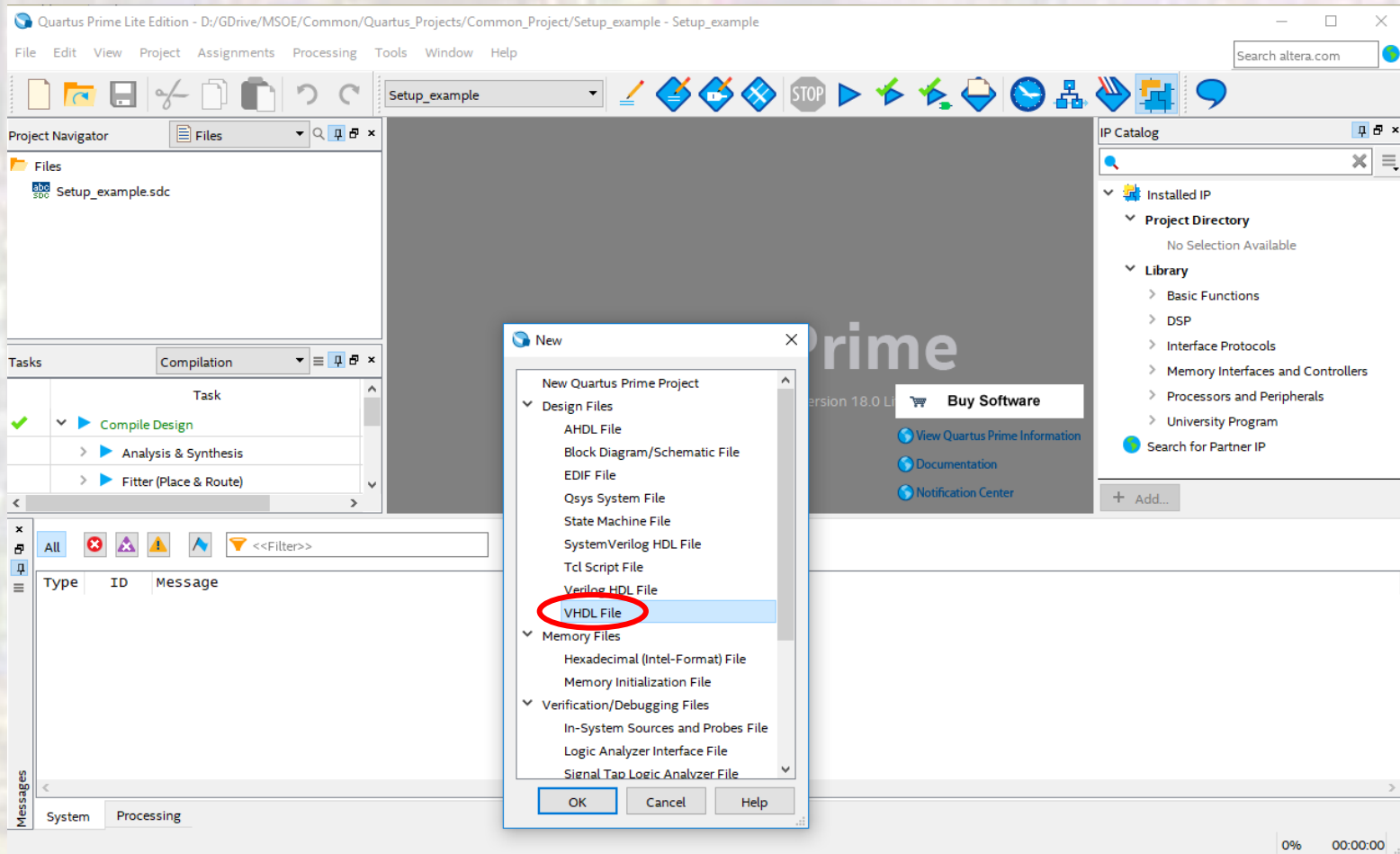
- Create a new Project
 - Use the [Quartus Project Setup](#) notes as a guide
 - Switch to [Files](#) view mode



Note: the correctly named SDC file is visible if you setup your project correctly

Quartus Counter Example

- Create a new HDL file
 - File → New → VHDL File

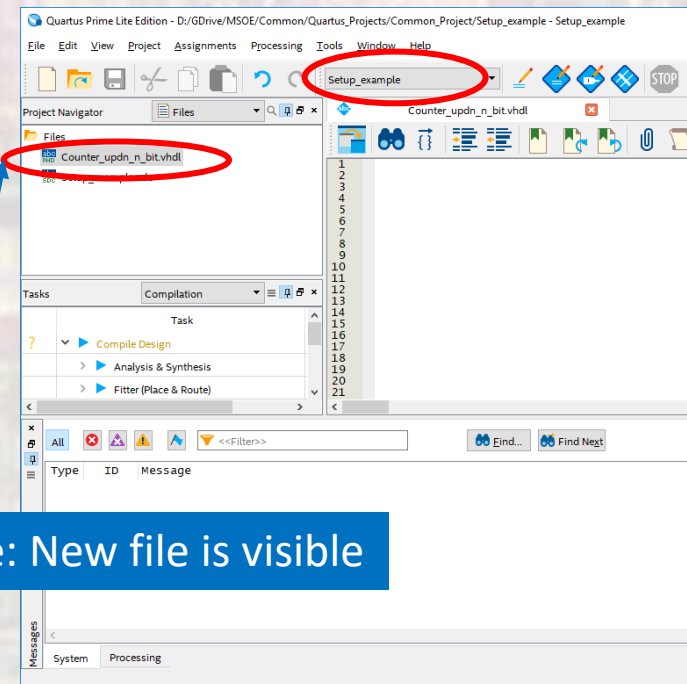
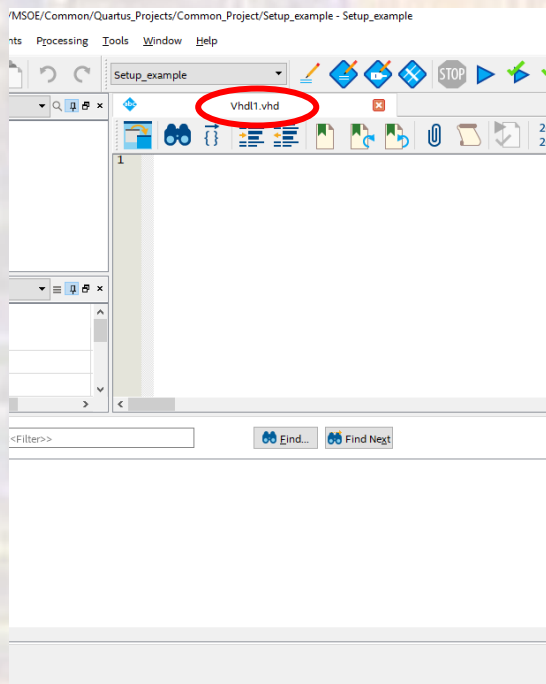


Quartus Counter Example

Use a reverse basis approach so all designs with the same basis are together in the file system making them easier to find

...
counter_mod10
counter_up_nbit
counter_updn_nbit
counter_updn_non-wrap_nbit
...

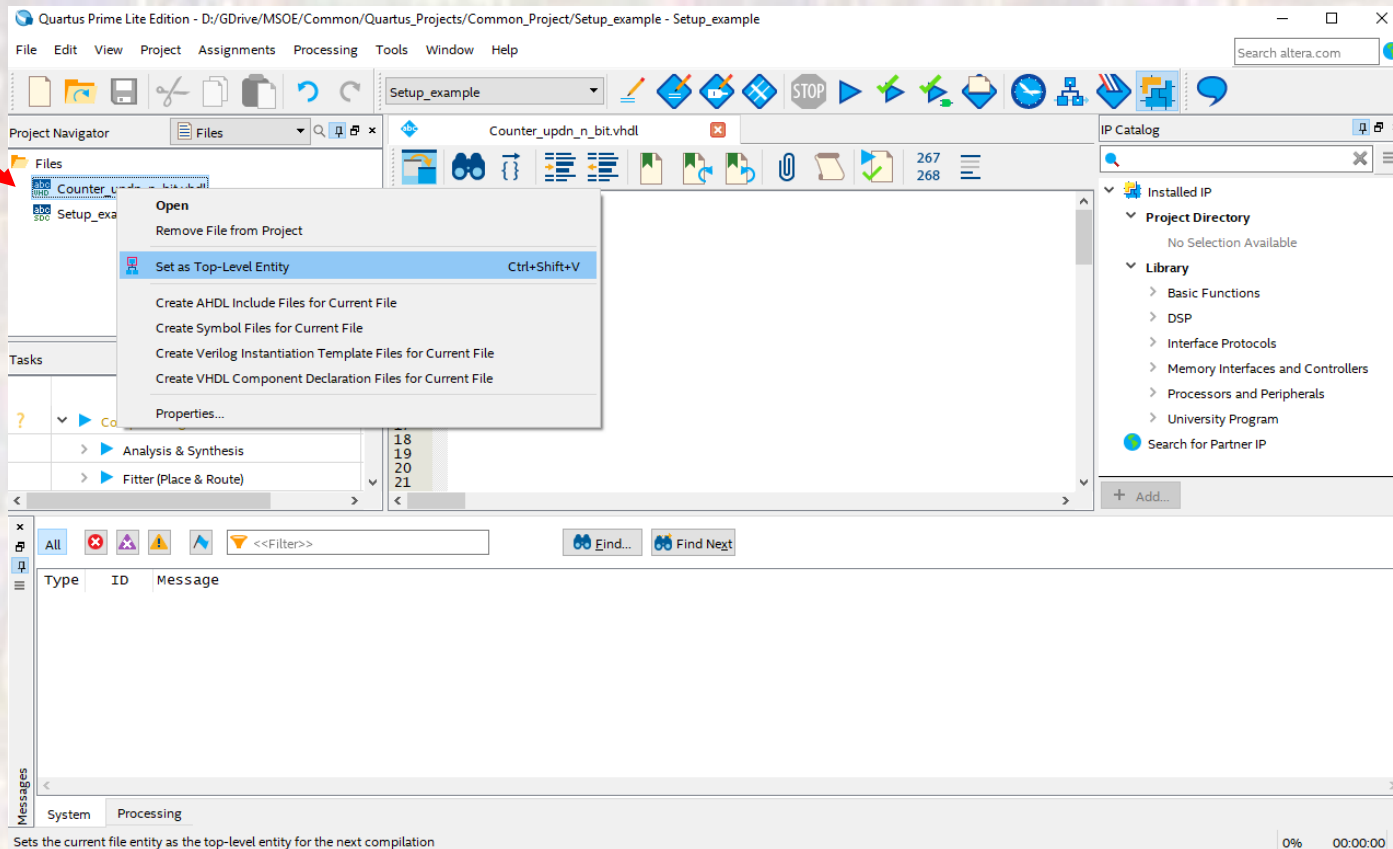
- Create a new HDL file
 - Note – the default file name is Vhdl.vhd
 - **File** → **Save As** and give it an informational file name
 - Change the file extension to vhd (for VHDL files)
 - Make sure “**Add file to current project**” is checked



Note: New file is visible

Quartus Counter Example

- Set the Top Level Entity
 - Right click on the design file in the **Files** window
 - Select **Set as Top Level Entity**



Quartus Counter Example

- Enter your design

The screenshot displays the Quartus Prime Lite Edition software interface. The main window shows the VHDL code for a counter, with the following content:

```
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
9 10
10 11
11 12
12 13
13 14
14 15
15 16
16 17
17 18
18 19
19 20
20 21
21 22
22 23
23 24
24 25
```

The code includes comments and a description of the counter's behavior:

```
-- Counter_updn_n_bit.vhdl
-- created 6/22/18
-- tj
-- rev 0
-----
-- n bit up/down counter example
-- for showing a project from start to finish
-----
--
-- Inputs: rstb, clk, dir
-- Outputs: cnt
-----
--
-- counts up when dir = 0
-- counts down when dir = 1
-----
```

The interface also shows the Project Navigator on the left, the IP Catalog on the right, and the Messages window at the bottom. The Messages window contains the following text:

```
125068 Revision "Setup_example" was previously opened in Quartus II software version 18.0.0 Lite Edition. Created Quartus
125069 Default assignment values were changed in the current version of the Quartus Prime software -- changes to default a
```

The status bar at the bottom indicates 0% progress and a time of 00:00:00.

Quartus Counter Examp

Follow the “best practices” – see web site
i_, o_ for signals in/out of the block
std logic for wires - signed, unsigned for signals
almost everything in a process
generic whenever possible
...

```
-----  
-- Counter_updn_n_bit.vhdl  
--  
-- created 6/22/18  
-- tj  
-- rev 0  
-----  
--  
-- n bit up/down counter example  
-- for showing a project from start to finish  
-----  
--  
-- Inputs: rstb, clk, dir  
-- Outputs: cnt  
-----  
--  
-- counts up when dir = 0  
-- counts down when dir = 1  
-----  
library ieee;  
use ieee.std_logic_1164.all;  
use ieee.numeric_std.all;  
  
entity Counter_updn_n_bit is  
  generic(  
    n: natural := 8  
  );  
  port (  
    i_clk:      in std_logic;  
    i_rstb:     in std_logic;  
    i_dir:      in std_logic;  
  
    o_cnt :     out std_logic_vector(n-1 downto 0)  
  );  
end entity;
```

```
architecture behavioral of Counter_updn_n_bit is  
  --  
  -- internal signals  
  --  
  signal cnt_sig:  unsigned(n-1 downto 0);  
  
begin  
  
  process(i_clk, i_rstb)  
  begin  
    -- reset  
    --  
    if (i_rstb = '0') then  
      cnt_sig <= (others => '0');  
    --  
    -- rising clk edge  
    --  
    elsif (rising_edge(i_clk)) then  
      if(i_dir = '0') then  
        cnt_sig <= cnt_sig + 1;  
      else  
        cnt_sig <= cnt_sig - 1;  
      end if;  
    end if;  
  end process;  
  
  --  
  -- Output logic  
  --  
  o_cnt <= std_logic_vector(cnt_sig);  
  
end behavioral;
```

Quartus Counter Example

- Enter your design
 - Create a component template for your design (DUT)
 - Used in your testbench and DE10 files
 - Select **File** → **Create/Update** → **Create VHDL Component Declaration Files 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.

-- Generated by Quartus Prime Version 17.1 (Build Build 590 10/25/2017)
-- Created on Fri Feb 23 09:52:29 2018

```
COMPONENT Counter_updn_n_bit
PORT
(
    i_rstb : IN STD_LOGIC;
    i_clk  : IN STD_LOGIC;
    i_dir  : IN STD_LOGIC;
    o_cnt  : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)
);
END COMPONENT;
```

Quartus Counter Example

- Elaborate
 - Check the design syntax and create RTL
 - Processing → Start → Start Analysis and Elaboration

The screenshot shows the Quartus Prime Lite Edition interface. The main window displays the 'Flow Summary' for the 'Setup_example' project. The flow status is 'Successful - Fri Jun 22 18:3'. The flow settings include: Quartus Prime Version (18.0.0 Build 614 04/24/201), Revision Name (Setup_example), Top-level Entity Name (Counter_updn_n_bit), Family (MAX 10), Device (10M50DAF484CTG), Timing Models (Final), Total logic elements (N/A until Partition Merge), Total registers (N/A until Partition Merge), Total pins (N/A until Partition Merge), Total virtual pins (N/A until Partition Merge), and Total memory bits (N/A until Partition Merge).

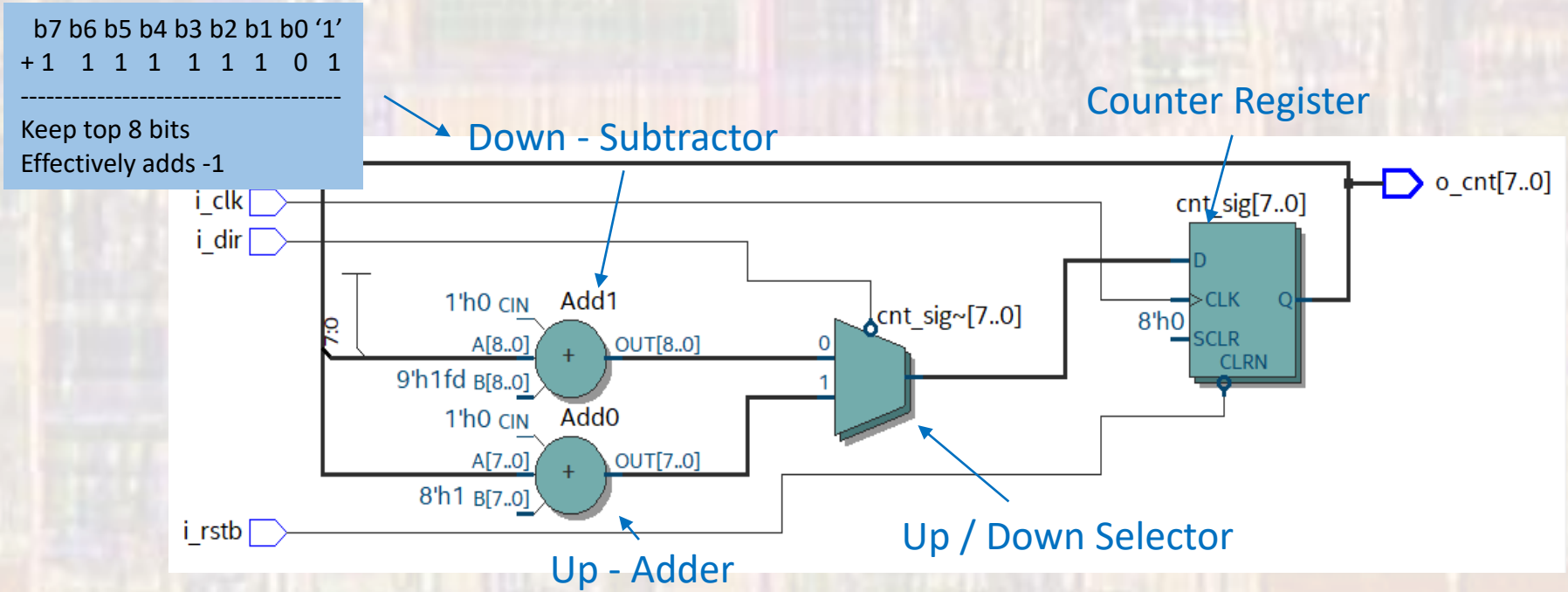
The 'Messages' window at the bottom shows the following output:

```
Running Quartus Prime Analysis & Elaboration
Command: quartus_map --read_settings_files=on --write_settings_files=off Setup_example -c Setup_example --analysis_and_elaboration
18236 Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment NUM_PARALLEL_PROCESSORS
20030 Parallel compilation is enabled and will use 2 of the 2 processors detected
12021 Found 2 design units, including 1 entities, in source file counter_updn_n_bit.vhdl
12127 Elaborating entity "counter_updn_n_bit" for the top level hierarchy
20013 Ignored 185 assignments for entity "DE10_Lite" -- entity does not exist in design
quartus Prime Analysis & Elaboration was successful. 0 errors, 187 warnings
```

A red box with the text 'Fix all Errors' is positioned over the 'Compile Design' task in the 'Tasks' pane. A yellow box with the text 'Review all Warnings' is positioned over the 'Messages' window. Another yellow box with the text 'Some of these warnings are due to the fact that we included the QSF file even though we are not using this file for the basic design' is positioned over the 'Messages' window, with an arrow pointing to the warning about the entity 'DE10_Lite'.

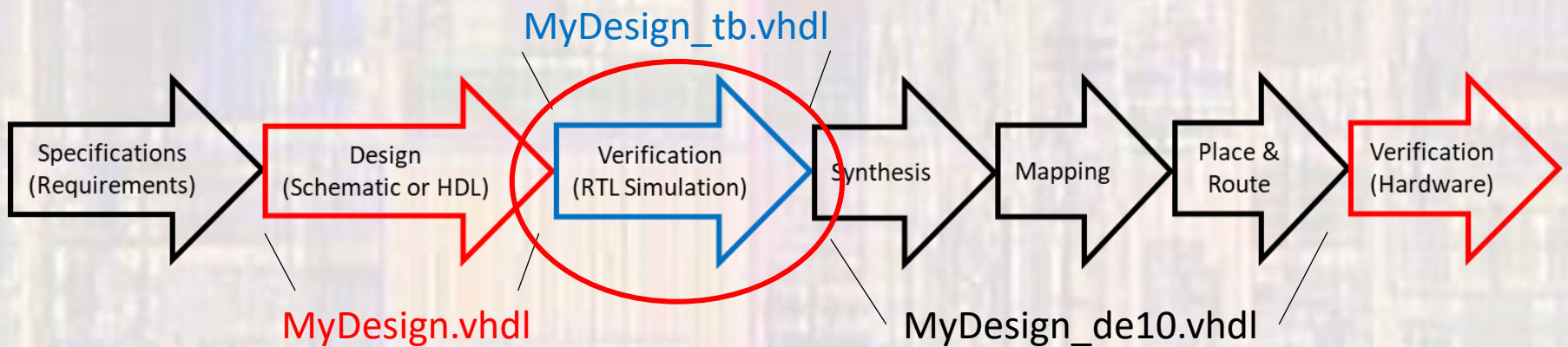
Quartus Counter Example

- Verify the RTL
 - Tools → Netlist Viewers → RTL Viewer



Quartus Counter Example

- Flow



Quartus Counter Example

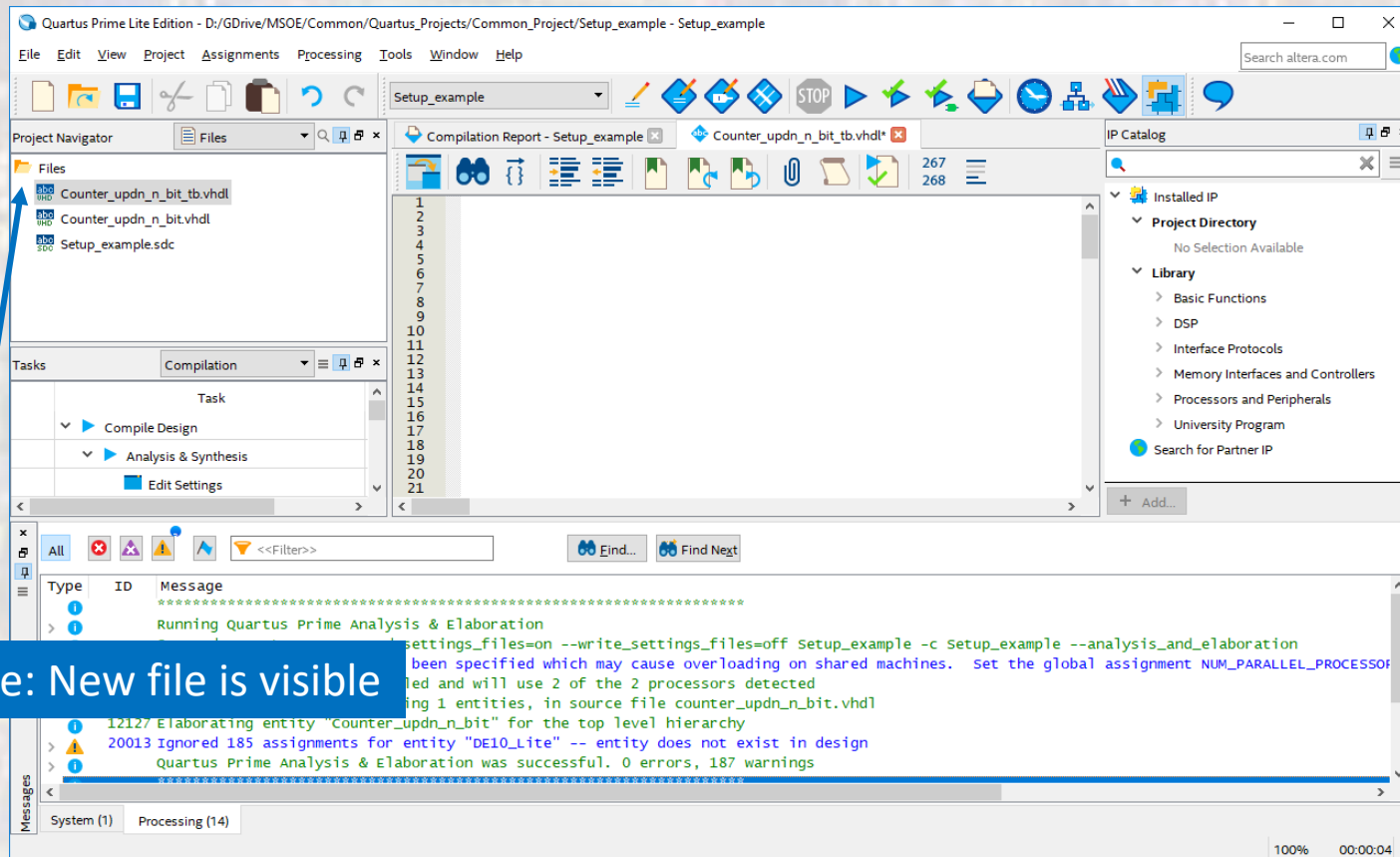
- Create Testbench

- File → New → VHDL File

- File → Save As and give it an informational file name

This should be the same name as your design file with _tb added

foo_nbit.vhdl
foo_nbit_tb.vhdl



Note: New file is visible

Quartus Counter Example

- Enter your TB

The screenshot displays the Quartus Prime Lite Edition interface. The main window shows the testbench file `counter_updn_n_bit_tb.vhdl` with the following content:

```
1
2
3  -- counter_updn_n_bit_tb.vhdl
4
5  -- created: 3/17/18
6  -- by: johnsontimoj
7  -- rev: 0
8
9  -- testbench for up down counter
10 -- of counter_updn_n_bit.vhdl
11
12
13 library ieee;
14 use ieee.std_logic_1164.all;
15
16 entity counter_updn_n_bit_tb is
17     generic(
18         N: natural := 8
19     );
20     -- no port entry - testbench
21 end entity;
```

The Messages window at the bottom shows the following output:

```
Running Quartus Prime Analysis & Elaboration
Command: quartus_map --read_settings_files=off Setup_example -c Setup_example --analysis_and_elaboration
18236 Number of processors has not been specified which may cause overloading on shared machines. set the global assignment NUM_PARALLEL_PROCESSORS
20030 Parallel compilation is enabled and will use 2 of the 2 processors detected
12021 Found 2 design units, including 1 entities, in source file counter_updn_n_bit.vhdl
12127 Elaborating entity "Counter_updn_n_bit" for the top level hierarchy
20013 Ignored 185 assignments for entity "DE10_Lite" -- entity does not exist in design
Quartus Prime Analysis & Elaboration was successful. 0 errors, 187 warnings
```

Quartus Counter Example

Continue to use generics in TB
TB signals in all caps
Define a constant for the clock PER
(no absolute times used)

- Enter your TB

```
-----  
--  
-- counter_updn_n_bit_tb.vhdl  
--  
-- created: 3/17/18  
-- by: johnsontim  
-- rev: 0  
--  
-- testbench for up down counter  
-- of counter_updn_n_bit.vhdl  
--  
-----  
library ieee;  
use ieee.std_logic_1164.all;  
  
entity counter_updn_n_bit_tb is  
  generic(  
    N: natural := 8  
  );  
  -- no port entry - testbench  
end entity;
```

```
architecture testbench of counter_updn_n_bit_tb is  
  signal CLK: std_logic;  
  signal RSTB: std_logic;  
  signal DIR: std_logic;  
  
  signal CNT: std_logic_vector((N - 1) downto 0);  
  constant PER: time := 20 ns;  
  
  -----  
  -- Component prototype  
  -----  
  component counter_updn_n_bit  
    generic ( n : NATURAL := 8 );  
    port  
    (  
      i_rstb : IN STD_LOGIC;  
      i_clk : IN STD_LOGIC;  
      i_dir : IN STD_LOGIC;  
      o_cnt : OUT STD_LOGIC_VECTOR((n - 1) downto 0)  
    );  
  end component;  
  
  -----  
  begin  
  
  -----  
  -- Device under test (DUT)  
  -----  
  DUT: counter_updn_n_bit  
    generic map(  
      n => N  
    )  
    port map(  
      i_clk => CLK,  
      i_rstb => RSTB,  
      i_dir => DIR,  
      o_cnt => CNT  
    );  
  
end architecture;
```

Quartus Counter Example

Separate clock, reset and run processes
Could be 0, 1, 2, Run processes

- Enter your TB

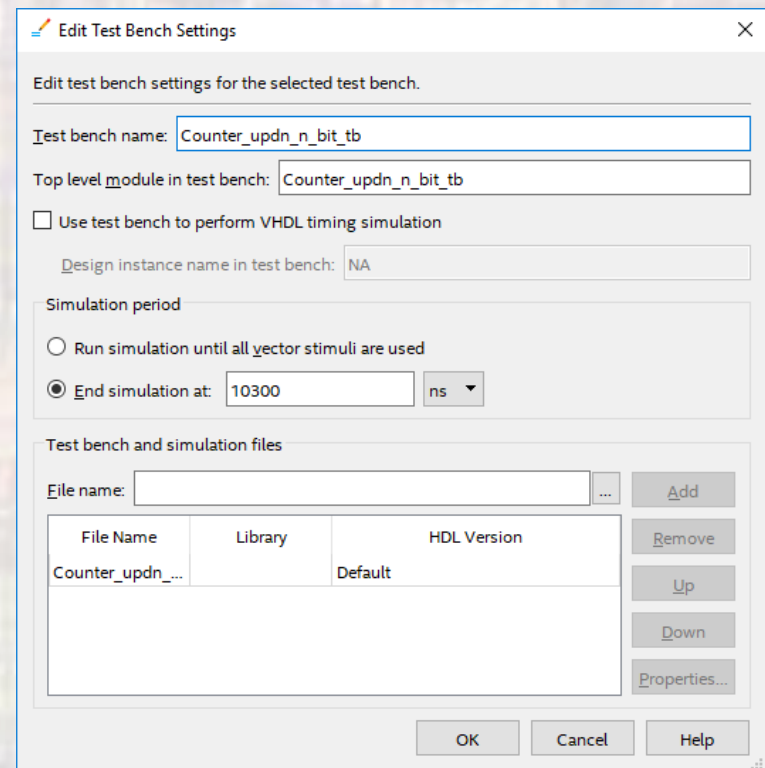
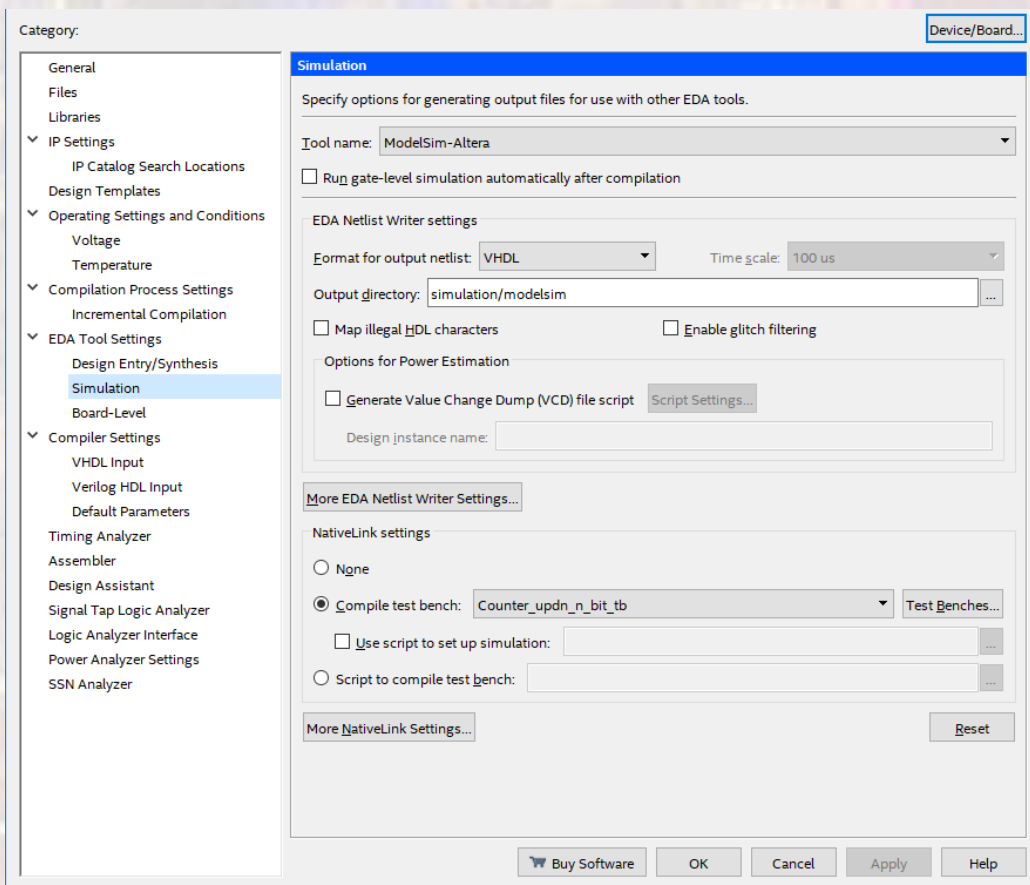
```
-----  
-- Test processes  
-----  
  
-- Clock process  
clock: process    -- note - no sensitivity list allowed  
begin  
    CLK <= '0';  
    wait for PER/2;  
    infinite: loop  
        CLK <= not CLK; wait for PER/2;  
    end loop;  
end process;  
  
-- Reset process  
reset: process    -- note - no sensitivity list allowed  
begin  
    RSTB <= '0'; wait for 2*PER;  
    RSTB <= '1'; wait;  
end process reset;  
  
-- Run Process  
run: process      -- note - no sensitivity list allowed  
begin  
    -- initialize inputs  
    DIR <= '0';  
  
    -- wait for reset  
    wait for 2*PER;  
  
    -- run code  
    wait for (2**N)*PER;  
    DIR <= '1';  
    wait for (2**N)*PER;  
  
end process run;  
  
-----  
-- End test processes  
-----  
  
end architecture;
```

Quartus Counter Example

- Enter your TB
 - Elaborate the design
 - With the original vhdl design set as the top level entity (**not the xxxx_tb.vhdl design**)
 - Select **Processing** → **Start** → **Start Analysis and Elaboration**
 - This causes Quartus to check the Test Bench code along with the original vhdl design

Quartus Counter Example

- Setup your simulation
 - Use the [ModelSim Testbench Setup](#) notes as a guide



Quartus Counter Example

- Run your simulation
 - Tools → Run Simulation Tools → RTL Simulation

ModelSim - INTEL FPGA STARTER EDITION 10.5b

File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help

Layout Simulate ColumnLayout Default

sim - Default

Instance

- counter_upd_n_b...
- DUT
- clock
- reset
- run
- standard
- textio
- std_logic_1164
- numeric_std

Design

- N
- CLK
- RSTB
- DIR
- CNT
- PER

Wave - Default

Signal	Value
/counter_upd_n_bit_tb/CLK	0
/counter_upd_n_bit_tb/RSTB	1
/counter_upd_n_bit_tb/DIR	1
/counter_upd_n_bit_tb/CNT	255

Transcript

```
# view signals  
# .main_pane.objects.interior.cs.body.tree  
# run 10300 ns  
VSIM 2>
```

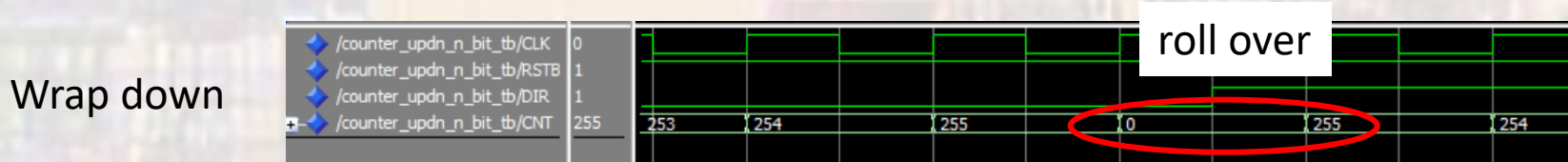
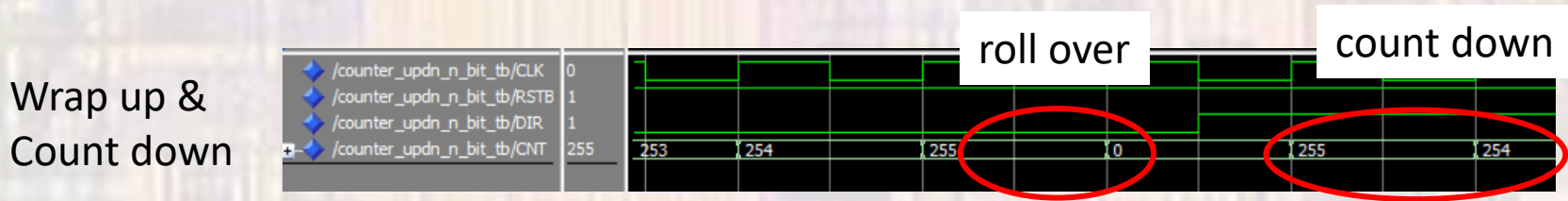
Now: 10.300 ns Delta: ?

sim:/counter_upd_n_bit_th

0 ns to 10815 ns

Quartus Counter Example

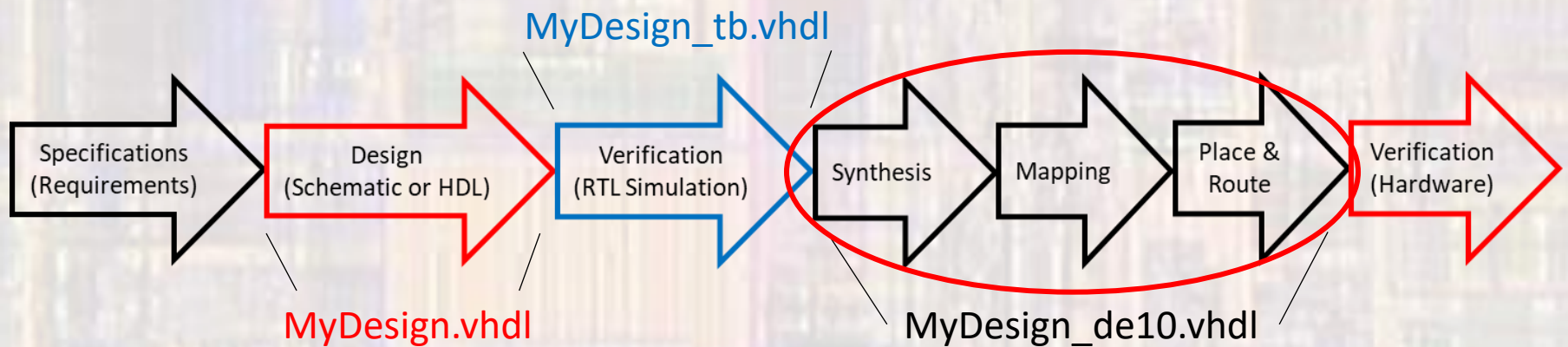
- Verify your design



signal names visible

Quartus Counter Example

- Flow

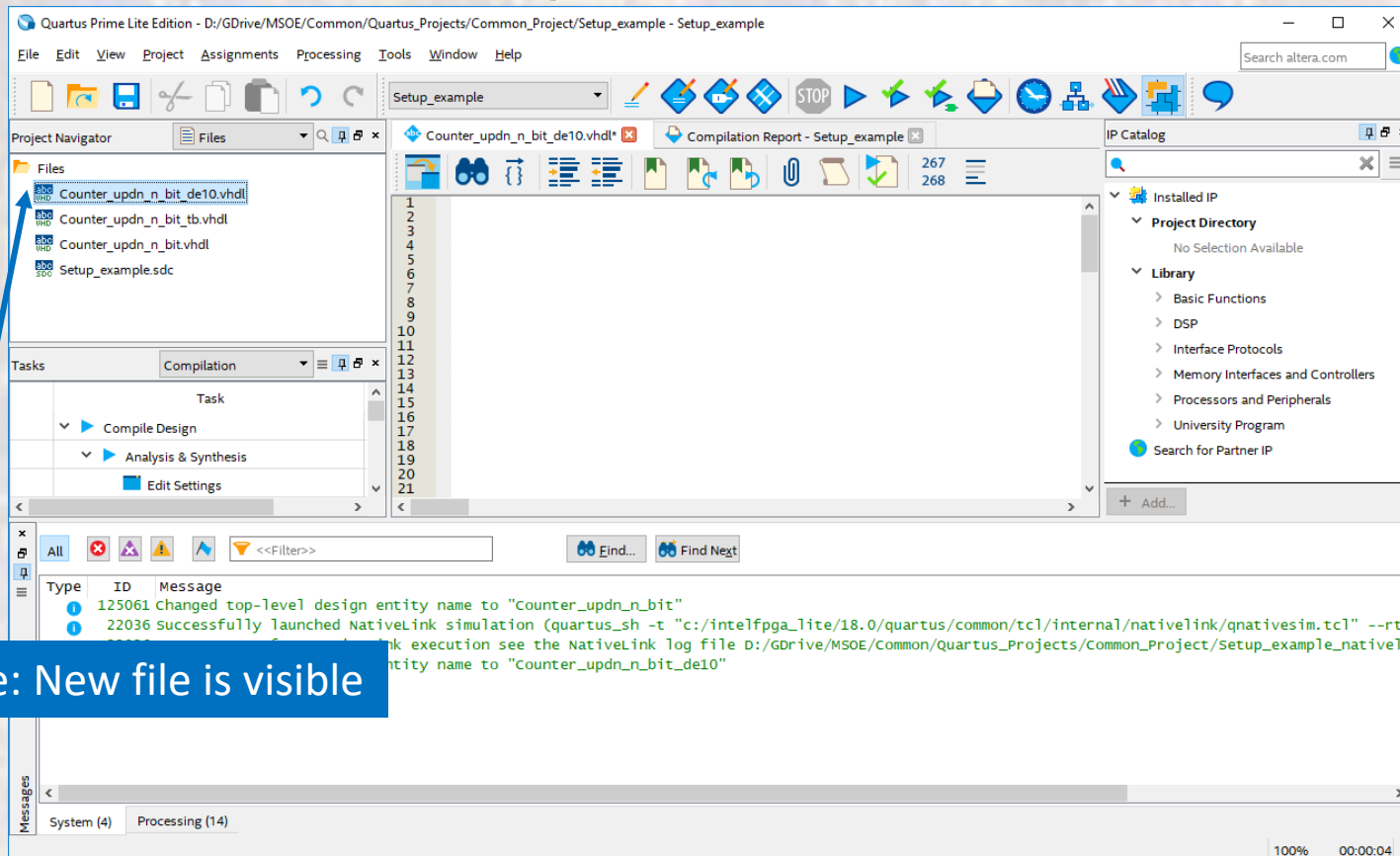


Quartus Counter Example

- Create DE10 implementation HDL
 - File → New → VHDL File
 - File → Save As and give it an informational file name

This should be the same name as your design file with _de10 added

foo_nbit.vhdl
foo_nbit_tb.vhdl
foo_nbit_de10.vhdl



Note: New file is visible

Quartus Counter Example

- Set the Top Level Entity
 - Right click on the de10 file in the Files window
 - Select **Set as Top Level Entity**

The screenshot shows the Quartus Prime Lite Edition interface. The 'Files' window is open, displaying a project structure with a file named 'Counter_updn_n_bit_de10.vhd'. A right-click context menu is open over this file, with the option 'Set as Top-Level Entity' highlighted. The menu also includes options like 'Open', 'Remove File from Project', and 'Properties...'. The 'Messages' window at the bottom shows a log of compilation messages, including '125061 Changed top-level design entity name to "counter_updn_n_bit"', '22036 Successfully launched NativeLink simulation', and '125061 Changed top-level design entity name to "counter_updn_n_bit_de10"'. A status bar at the bottom indicates 'Sets the current file entity as the top-level entity for the next compilation'.

Quartus Counter Example

- Enter your DE10 design

The screenshot displays the Quartus Prime Lite Edition interface. The main window shows the VHDL code for a counter, with the following content:

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21
```

Comments in the code include: "counter_updn_n_bit_de10.vhdl", "created: 3/17/18", "by: johnsontimoj", "rev: 0", "DE10 implementation for up down counter of counter_updn_n_bit.vhdl", "Uses a 3 Hz clock divider, sw0 for reset", "sw1 for the dir input", and "and LEDs for the count output".

The code defines an entity named "counter_updn_n_bit_de10" with a port "CLOCK_50" of type "std_logic".

The Messages window at the bottom shows the following messages:

Type	ID	Message
Info	125061	Changed top-level design entity name to "Counter_updn_n_bit"
Info	22036	Successfully launched NativeLink simulation (quartus_sh -t "c:/intelfpga_lite/18.0/quartus/common/tcl/internal/nativeLink/qnativesim.tcl" --rt)
Info	22036	For messages from NativeLink execution see the NativeLink log file D:/GDrive/MSOE/Common/Quartus_Projects/Common_Project/Setup_example_nativeLi
Info	125061	Changed top-level design entity name to "Counter_updn_n_bit_de10"

The status bar at the bottom indicates "Ln 1 Col 1 VHDL File" and "100% 00:00:04".

Quartus Counter Example

Use qsf pin names for the top level I/Os

- Enter your DE10 design

```
-----  
--  
-- counter_updn_n_bit_de10.vhdl  
--  
-- created: 3/17/18  
-- by: johnsontimoj  
-- rev: 0  
--  
-- DE10 implementation for up down counter  
-- of counter_updn_n_bit.vhdl  
--  
-- Uses a 3 Hz clock divider, sw0 for reset  
-- sw1 for the dir input  
-- and LEDs for the count output  
-----  
library ieee;  
use ieee.std_logic_1164.all;  
  
entity counter_updn_n_bit_de10 is  
    port(  
        CLOCK_50:    in    std_logic;  
        SW:          in    std_logic_vector(1 downto 0);  
  
        LEDR:        out   std_logic_vector(7 downto 0)  
    );  
end entity;
```

```
architecture hardware of counter_updn_n_bit_de10 is  
  
    signal CLK_SIG: std_logic; -- Intermediate clk signal  
  
    -----  
    -- Component prototypes  
    -----  
    component clk_3Hz  
        port(  
            i_clk_50MHz : IN STD_LOGIC;  
            i_rstb      : IN STD_LOGIC;  
            o_clk_3Hz   : OUT STD_LOGIC  
        );  
    end component;  
  
    component counter_updn_n_bit  
        generic ( n : NATURAL := 8 );  
        port  
        (  
            i_rstb : IN STD_LOGIC;  
            i_clk  : IN STD_LOGIC;  
            i_dir  : IN STD_LOGIC;  
            o_cnt  : OUT STD_LOGIC_VECTOR((n - 1) downto 0)  
        );  
    end component;  
    -----
```

Quartus Counter Example

- Enter your DE10 design

```
begin
-----
-- Device under test (DUT)
-----
CK: clk_3Hz
port map(
    i_clk_50MHz    => CLOCK_50,
    i_rstb         => SW(0),
    o_clk_3Hz      => CLK_SIG
);

DUT: counter_updn_n_bit
port map(
    i_clk    => CLK_SIG,
    i_rstb   => SW(0),
    i_dir    => SW(1),
    o_cnt    => LEDR(7 downto 0)
);
end architecture;
```


Quartus Counter Example

- Create a clock divider
 - File → New → VHDL File
 - File → Save As and give it an informational file name

The screenshot shows the Quartus Prime Lite Edition interface. The Project Navigator on the left lists files: `clk_3Hz.vhdl`, `Counter_updn_n_bit_de10.vhdl`, `Counter_updn_n_bit_tb.vhdl`, `Counter_updn_n_bit.vhdl`, and `Setup_example.sdc`. A blue arrow points to `clk_3Hz.vhdl`. The Messages window at the bottom shows the following text:

```
Type ID Message
125061 Changed top-level design entity name to "Counter_updn_n_bit"
22036 Successfully launched NativeLink simulation (quartus_sh -t "c:/intelfpga_lite/18.0/quartus/common/tcl/internal/nativeLink/qnativesim.tcl" --rt
iveLink execution see the NativeLink log file D:/GDrive/MSOE/Common/Quartus_Projects/Common_Project/Setup_example_native
ign entity name to "Counter_updn_n_bit_de10"
```

A blue callout box with white text says: **Note: New file is visible**

Quartus Counter Example

- Create 3Hz clock divider

```
-----  
--  
-- clk_3hz.vhd1  
--  
-- created 2/20/18  
-- tj  
--  
-- rev 0  
-----  
--  
-- 3Hz clock divider for use with DE10 Labs  
-- assumes a 50MHz external clock  
--  
-----  
-- Inputs: rstb, clk_50MHz  
-- Outputs: clk_out  
--  
-----  
library ieee;  
use ieee.std_logic_1164.all;  
use ieee.numeric_std.all;  
  
entity clk_3hz is  
  port (  
    i_clk_50MHz : in std_logic;  
    i_rstb      : in std_logic;  
  
    o_clk_3Hz : out std_logic  
  );  
end entity;
```

Quartus Counter Example

- Create 3Hz clock divider

```
architecture behavioral of clk_3hz is
    --
    -- constants and parameters
    --
    constant INPUT_FREQUENCY: natural := 50_000_000;
    constant OUTPUT_FREQUENCY: natural := 3;
    constant CLKS_PER_HALF_PERIOD: signed(24 downto 0) := to_signed(((INPUT_FREQUENCY / 2) / OUTPUT_FREQUENCY), 25);
    --
    -- internal signals
    --
    signal cnt: signed(24 downto 0);
    signal clk_sig: std_logic;
begin
    process(i_clk_50MHz, i_rstb)
    begin
        --
        -- reset
        --
        if (i_rstb = '0') then
            cnt <= (CLKS_PER_HALF_PERIOD - 1);
            clk_sig <= '0';
        elsif (rising_edge(i_clk_50MHz) ) then
            cnt <= cnt - 1;
            --
            -- check if half way
            --
            if (cnt < 0) then
                cnt <= ((CLKS_PER_HALF_PERIOD - 1) - 1);
                clk_sig <= not clk_sig;
            end if;
        end if;
    end process;

    --
    -- output logic
    --
    o_clk_3Hz <= clk_sig;
end behavioral;
```

Quartus Counter Example

- Compile
 - Check the design syntax, create RTL, Place and Route
 - Processing → Start Compilation

Fix all Errors

Check for timing issues

Type	ID	Message
Warning	332123	Deriving clock uncertainty. Please refer to report_sdc in the
Warning	332146	worst-case setup slack is 18.626
Warning	332146	worst-case hold slack is 0.166
Warning	332140	No Recovery paths to report
Warning	332140	No Removal paths to report
Warning	332146	worst-case minimum pulse width slack is 9.464
Warning	332102	Design is not fully constrained for setup requirements
Warning	332102	Design is not fully constrained for hold requirements

Review all Warnings

Quartus Counter Example

- Compile
 - Check the Flow Summary

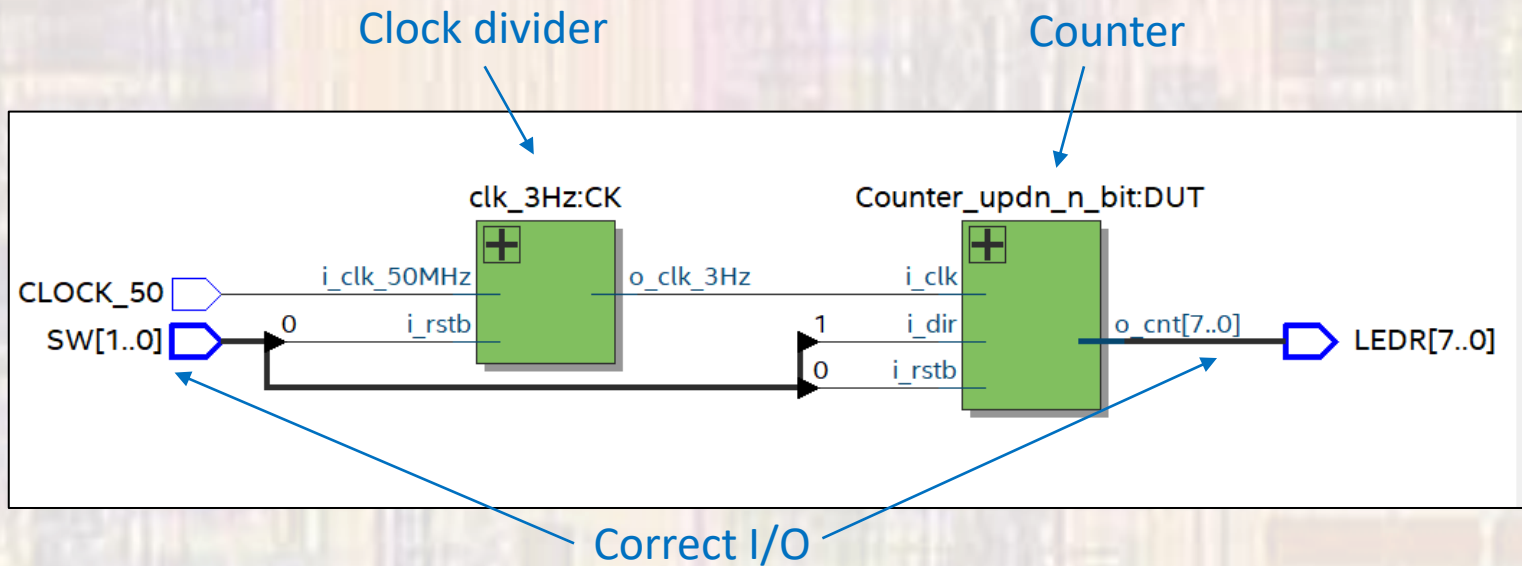
The screenshot shows the 'Flow Summary' window in Quartus. The 'Flow Status' is 'Successful' (circled in red). The 'Top-level Entity Name' is 'counter_updn_n_bit_de10' (circled in red). The 'Device' is '10M50DAF484C7G' (circled in red). The 'Total logic elements' is '35 / 49,760 (< 1 %)' (circled in red). The 'Total registers' is '34' (circled in red). The 'Total pins' is '11 / 360 (3 %)' (circled in red). The 'Total virtual pins' is '0' (circled in red). The 'Total memory bits' is '0 / 1,677,312 (0 %)' (circled in red). The 'Embedded Multiplier 9-bit elements' is '0 / 288 (0 %)' (circled in red). The 'Total PLLs' is '0 / 4 (0 %)' (circled in red). The 'UFM blocks' is '0 / 1 (0 %)' (circled in red). The 'ADC blocks' is '0 / 2 (0 %)' (circled in red). Annotations point to these values with yellow boxes: 'Check success' points to 'Successful', 'Correct Top Level Design' points to 'counter_updn_n_bit_de10', 'Correct Part Number' points to '10M50DAF484C7G', and 'Check to make sure utilization makes sense' points to the 'Total virtual pins' value.

Property	Value
Flow Status	Successful - Mon Jun 25 11:20:24 2018
Quartus Prime Version	18.0.0 Build 614 04/24/2018 SJ Lite Edition
Revision Name	Setup_example
Top-level Entity Name	counter_updn_n_bit_de10
Family	MAX 10
Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	35 / 49,760 (< 1 %)
Total registers	34
Total pins	11 / 360 (3 %)
Total virtual pins	0
Total memory bits	0 / 1,677,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 288 (0 %)
Total PLLs	0 / 4 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 2 (0 %)

Quartus Counter Example

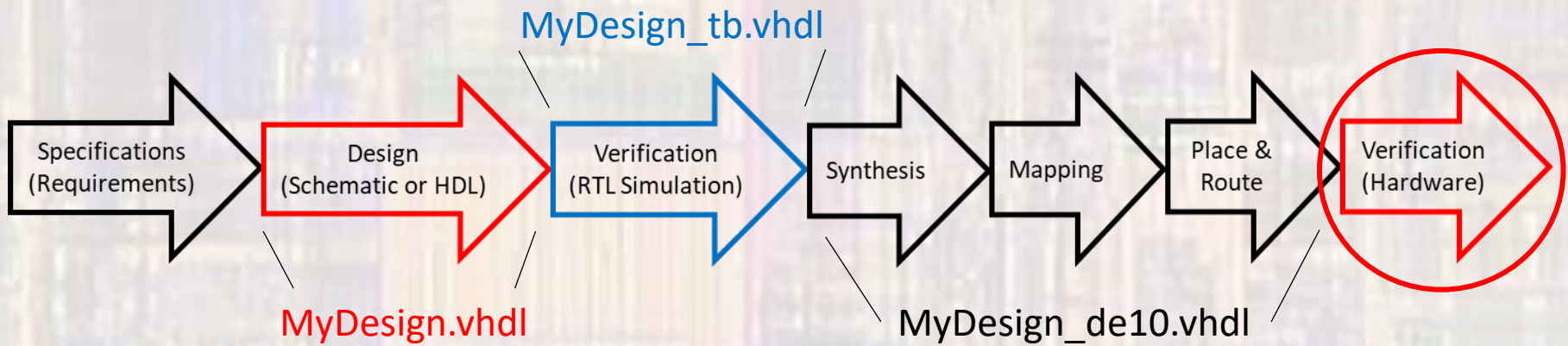
- Verify the RTL
 - Tools → Netlist Viewers → RTL Viewer

Note: this IS NOT sufficient for a design report – you need to show RTL for all blocks you created



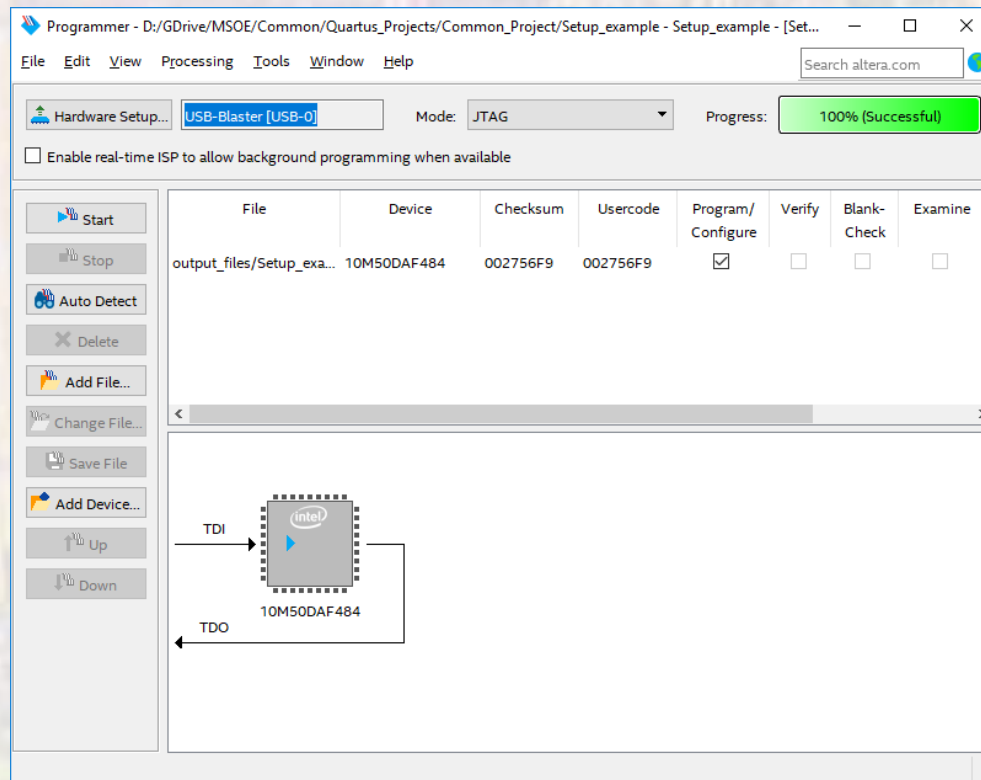
Quartus Counter Example

- Flow



Quartus Counter Example

- Program the board
 - Connect the DE10 board to computer
 - **Tools** → **Programmer**
 - Point to the compiled output file **xxxx.sof**
 - **Start**



Quartus Counter Example

- Verify Operation
 - video

