Hierarchical Design Prototype

Synopsis

This note covers an example on a development of hierarchical design prototype on Field Programmable Gate Array (FPGA) and Vivado Integrated Design Environment (IDE) [1].

Introduction

Hardware prototype on FPGA requires an IDE project that instantiates the device to be verified. This requires adding source files from previous different designs. Prototype testing design has as the top level an instance of the device-under-test. The device comprises components and the design needs inclusions of Hardware Description Language (HDL) source files.

IDE Procedure

The procedure begins form crating a project for the prototype hardware with a VHDL code below (Box 1). The prototype is an instance of the adder previously developed whose the inputs are n-bit vector operands, where n is a generic parameter. The prototype is for 3-bit operands.

```
----------------------------------------------
-- Company: Drexel ECE
-- Engineer: prawat
-- adder test
----------------------------------------------
Library ieee;
Use ieee.std_logic_1164.all, work.p1_pack.all;

entity adder3 is
Port ( a, b : in std_logic_vector(2 downto 0);
       c : out std_logic_vector(2 downto 0);
       c_out : out std_logic;
       go,reset,ck, btn0, btn1 : in std_logic);
end adder3;

architecture Behavioral of adder3 is
component  adder
Generic (K: natural := 4);
Port ( a, b : in std_logic_vector(k-1 downto 0);
       C : out std_logic_vector(k-1 downto 0);
       C_out : out std_logic;
       Go,reset,ck : in std_logic);
End component;
signal en: std_logic; -- single step enable

begin
u2: adder generic map(3)
          port map (a,b,c,c_out,go,reset,en);

begin
u2: adder generic map(3)
          port map (a,b,c,c_out,go,reset,en);

-- single step, debounce (db)
-- btn0 to enter and btn1 to reset
process(ck)
type db_state is (not_rdy, rdy, pulse);
variable db_ns: db_state;
begin
if ck='1' and ck'event then
  case db_ns is
   when not_rdy => en <= '0';
   if btn1 = '1' then db_ns := rdy; end if;
   when rdy => en <= '0';
   if btn0 = '1' then db_ns := pulse; end if;
   when pulse => en <= '1';
   db_ns := not_rdy;
   when others => null;
  end case;
end if;
end process;

end Behavioral;
```

Box 1. Prototype Hardware HDL

The entity adder3 contains the adder component and the single-step process triggered by ck. The single-step process is a state machine that generates the signal en according to the btn0 and btn1 button presses. The signal en is a single pulse signal that drives the adder component, described in the HDL code by

U2: adder generic map (3)
    port map(a, b, c, c_out, go, reset, en);

The top-level adder3 entity contains the adder component. In the design sources of the Vivado IDE project would have a question mark on the component U2: adder since it does not yet bind to any source.

After adding the adder.vhd source file from the adder project directory the IDE now displays the components contained in the adder with question marks. The questions marks disapear after adding the component source files; serial_adder, sh_reg and control. In addition, the design requires adding the package p1_pack.vhd to the project. Figure 1 shows the hierarchy of the adder3.

Generally for design containing further subcomponents, proceed to add the source files on the order of the hierarchy.

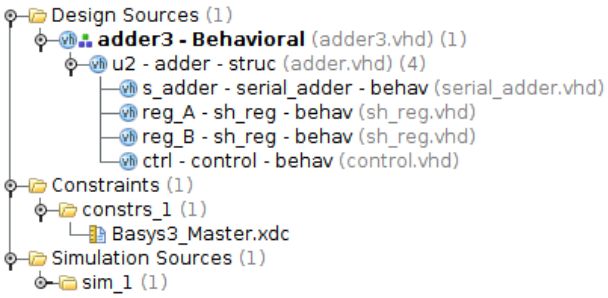

Fig. 1 Design Hierarchy Vivado IDE

Hardware Verification

Hardware testing on an FPGA board uses a single-step mechanism that allows user to position the a, b, reset and go switches. The hardware testing proceeds as

1) switch a and b vectors, and reset to high, then apply a single step
2) switch reset to low, go to high then step
3) switch go to low and continue stepping until the result becomes correct and no longer changes
4) repeat Step 1 with new test operands

Debugging Strategy

The design should progress in stages to isolate design errors. Plausible stages are

1. Serial Adder simulation and implementation on FPGA board
2. Serial Adder with the controller simulation and implementation on FPGA board
3. Adder simulation and
4. Adder3 prototype.

Conclusion

A development of an FPGA prototype of hierarchical design requires adding (importing) source files of the components. The note covered an example on an IDE that requires adding subcomponents source files in a particular order of the hierarchic.

References

[1] Vivado IDE, http://www.xilinx.com/support/documentation/sw_manuals/xilinx2016_1/ug893-vivado-ide.pdf

[2] Modelsim, http://www.microsemi.com/document-portal/doc_view/131619-modelsim-user