Lauro Cabral
February 1, 2018
CECS 461

Lab 1

1. After which step we finished building the hardware platform for the embedded system we build in the lab?

In step 3, we generate output products and create the HDL wrapper to be exported to the SDK.

2. Briefly describe the major components in the hardware platform.

When I added IP into the hardware platform I add the ZYNQ7 Processing System we only added *UART 1* and *DDR memory.*

3. What is the top-level design file for hardware platform? Copy the contents in this file that support the answer you provide in the previous question.

Vivado IP integrator block diagram files (.bd)

```
1.          --Copyright 1986-2015 Xilinx, Inc. All Rights Reserved.
2.  ----------------------------------------------------------------------------------
3.  --Tool Version: Vivado v.2015.2 (win64) Build 1266856 Fri Jun 26 16:35:25 MDT 2015
4.  --Date        : Tue Jan 30 22:44:43 2018
5.  --Host        : Lauro-Laptop running 64-bit major release  (build 9200)
6.  --Command     : generate_target system_wrapper.bd
7.  --Design      : system_wrapper
8.  --Purpose     : IP block netlist
9.  ----------------------------------------------------------------------------------
10. library IEEE;
11. use IEEE.STD_LOGIC_1164.ALL;
12. library UNISIM;
13. use UNISIM.VCOMPONENTS.ALL;
14. entity system_wrapper is
15.   port (
16.     DDR_addr : inout STD_LOGIC_VECTOR ( 14 downto 0 );
17.     DDR_ba : inout STD_LOGIC_VECTOR ( 2 downto 0 );
18.     DDR_cas_n : inout STD_LOGIC;
19.     DDR_ck_n : inout STD_LOGIC;
20.     DDR_ck_p : inout STD_LOGIC;
21.     DDR_cke : inout STD_LOGIC;
22.     DDR_cs_n : inout STD_LOGIC;
23.     DDR_dm : inout STD_LOGIC_VECTOR ( 3 downto 0 );
24.     DDR_dq : inout STD_LOGIC_VECTOR ( 31 downto 0 );
25.     DDR_dqs_n : inout STD_LOGIC_VECTOR ( 3 downto 0 );
26.     DDR_dqs_p : inout STD_LOGIC_VECTOR ( 3 downto 0 );
27.     DDR_odt : inout STD_LOGIC;
28.     DDR_ras_n : inout STD_LOGIC;
29.     DDR_reset_n : inout STD_LOGIC;
30.     DDR_we_n : inout STD_LOGIC;
31.     FIXED_IO_ddr_vrn : inout STD_LOGIC;
32.     FIXED_IO_ddr_vrp : inout STD_LOGIC;
33.     FIXED_IO_mio : inout STD_LOGIC_VECTOR ( 53 downto 0 );
34.     FIXED_IO_ps_clk : inout STD_LOGIC;
35.     FIXED_IO_ps_porb : inout STD_LOGIC;
36.     FIXED_IO_ps_srstb : inout STD_LOGIC
37.   );
38. end system_wrapper;
39.
40. architecture STRUCTURE of system_wrapper is
41.   component system is
42.   port (
43.     DDR_cas_n : inout STD_LOGIC;
44.     DDR_cke : inout STD_LOGIC;
45.     DDR_ck_n : inout STD_LOGIC;
```

```
    DDR_ck_p : inout STD_LOGIC;
    DDR_cs_n : inout STD_LOGIC;
    DDR_reset_n : inout STD_LOGIC;
    DDR_odt : inout STD_LOGIC;
    DDR_ras_n : inout STD_LOGIC;
    DDR_we_n : inout STD_LOGIC;
    DDR_ba : inout STD_LOGIC_VECTOR ( 2 downto 0 );
    DDR_addr : inout STD_LOGIC_VECTOR ( 14 downto 0 );
    DDR_dm : inout STD_LOGIC_VECTOR ( 3 downto 0 );
    DDR_dq : inout STD_LOGIC_VECTOR ( 31 downto 0 );
    DDR_dqs_n : inout STD_LOGIC_VECTOR ( 3 downto 0 );
    DDR_dqs_p : inout STD_LOGIC_VECTOR ( 3 downto 0 );
    FIXED_IO_mio : inout STD_LOGIC_VECTOR ( 53 downto 0 );
    FIXED_IO_ddr_vrn : inout STD_LOGIC;
    FIXED_IO_ddr_vrp : inout STD_LOGIC;
    FIXED_IO_ps_srstb : inout STD_LOGIC;
    FIXED_IO_ps_clk : inout STD_LOGIC;
    FIXED_IO_ps_porb : inout STD_LOGIC
  );
  end component system;
begin
system_i: component system
     port map (
      DDR_addr(14 downto 0) => DDR_addr(14 downto 0),
      DDR_ba(2 downto 0) => DDR_ba(2 downto 0),
      DDR_cas_n => DDR_cas_n,
      DDR_ck_n => DDR_ck_n,
      DDR_ck_p => DDR_ck_p,
      DDR_cke => DDR_cke,
      DDR_cs_n => DDR_cs_n,
      DDR_dm(3 downto 0) => DDR_dm(3 downto 0),
      DDR_dq(31 downto 0) => DDR_dq(31 downto 0),
      DDR_dqs_n(3 downto 0) => DDR_dqs_n(3 downto 0),
      DDR_dqs_p(3 downto 0) => DDR_dqs_p(3 downto 0),
      DDR_odt => DDR_odt,
      DDR_ras_n => DDR_ras_n,
      DDR_reset_n => DDR_reset_n,
      DDR_we_n => DDR_we_n,
      FIXED_IO_ddr_vrn => FIXED_IO_ddr_vrn,
      FIXED_IO_ddr_vrp => FIXED_IO_ddr_vrp,
      FIXED_IO_mio(53 downto 0) => FIXED_IO_mio(53 downto 0),
      FIXED_IO_ps_clk => FIXED_IO_ps_clk,
      FIXED_IO_ps_porb => FIXED_IO_ps_porb,
      FIXED_IO_ps_srstb => FIXED_IO_ps_srstb
    );
end STRUCTURE;
```

4. What does a bit stream file do? Is there a bit stream file generated for the hardware platform we built in this lab?

The bit stream file one of the files generated in a sequence of files used for designing and configuring an FPGA.

No bit steam file was generated for this lab because we only used the processing system.

5. What is the step that transitions our design from hardware platform to software platform? What are the tools used for hardware design and software design respectively? What kind of information are passed from hardware design tool to software design tool?

The step that transitions from hardware platform to software platform is when you export the hardware to the SDK. The tools used for hardware design is Vivado/IP integrator and software design is the Xilinix SDK.

6. How many projects are created for building the software platform? Briefly describe the key design information provide in each one of them and the role each one of them play in the software platform.

3 projects were created for building the software platform.

i. *system_wrapper_hw_platform_0*

includes the ps7_init function which initializes the PS as part of the first stage boot-loader

ii. *mem_test_bsp (board support project)*

is a collection of libraries and drivers that will form the lowest layer of your application software stack. Your software applications must link against or run on top of a given software platform using the APIs that it provides. Therefore, before you can create and use software applications in SDK, you must create a board support package.

iii. mem_test

is the application that we will use to verify the functionality of the design.

7. What is a BSP? What information does it provide and what role does it play in building the software platform?

is a collection of libraries and drivers that will form the lowest layer of your application software stack. Your software applications must link against or run on top of a given software platform using the APIs that it provides. Therefore, before you can create and use software applications in SDK, you must create a board support package.

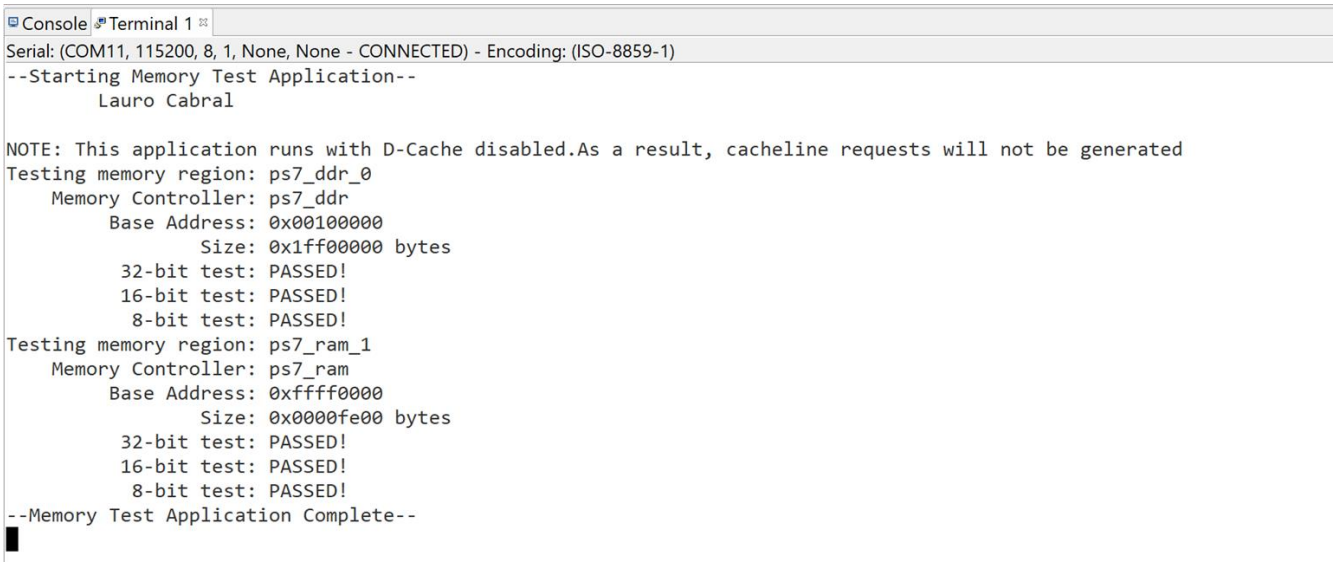
8. What is the top-level design file for software platform? Briefly describe the key information in this file and support your answer by copying the related contents in this file.

Software application projects are your final application containers. The project directory that is created contains (or links to) your C/C++ source files, executable output file, and associated utility files, such as the make files used to build the project.

9. What is an .elf file? Is there an .elf file generated in this lab?

Contains an executable CPU code image. Yes a .elf file was generated for this lab.

10. Please provide a screen shot of your embedded system output.


Console
Terminal 1
Serial: (COM11, 115200, 8, 1, None, None - CONNECTED) - Encoding: (ISO-8859-1)
--Starting Memory Test Application--
Lauro Cabral
NOTE: This application runs with D-Cache disabled.As a result, cacheline requests will not be generated
Testing memory region: ps7_ddr_0
Memory Controller: ps7_ddr
Base Address: 0x00100000
Size: 0x1ff00000 bytes
32-bit test: PASSED!
16-bit test: PASSED!
8-bit test: PASSED!
Testing memory region: ps7_ram_1
Memory Controller: ps7_ram
Base Address: 0xffff0000
Size: 0x0000fe00 bytes
32-bit test: PASSED!
16-bit test: PASSED!
8-bit test: PASSED!
--Memory Test Application Complete--