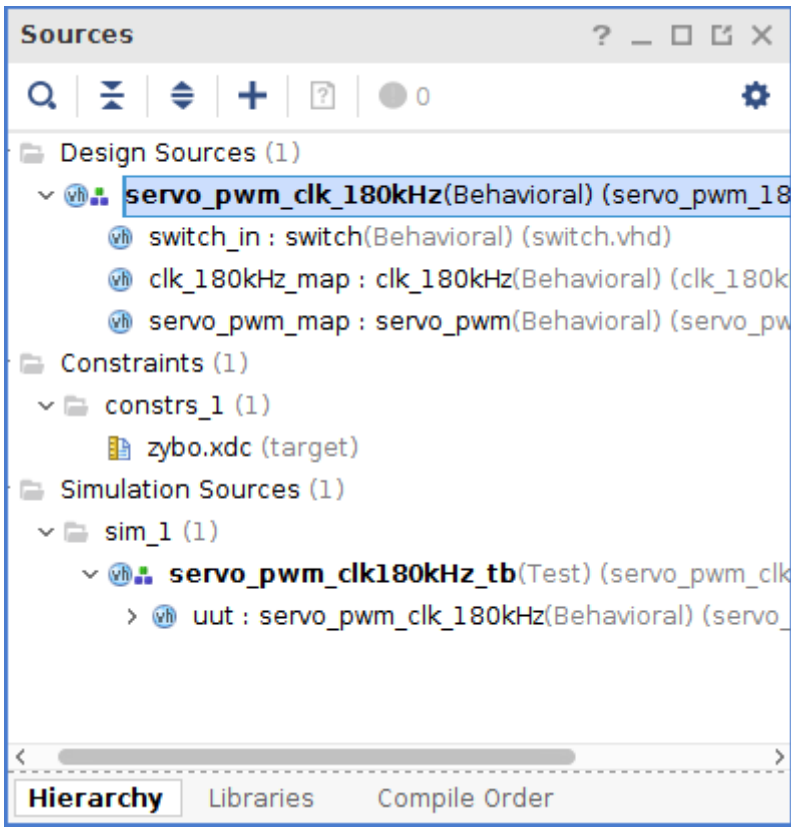
TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 문한나

mhn97@naver.com



Add source로 파일을 추가한다

servo_pwm_clk_180kHz.vhd

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity servo_pwm_clk_180kHz is
    port(
        clk : in std_logic;
        reset : in std_logic;
        sw : in std_logic_vector(3 downto 0);
        servo : out std_logic
    );
end servo_pwm_clk_180kHz;

architecture Behavioral of servo_pwm_clk_180kHz is
    component switch
        port(
            sw : in std_logic_vector(3 downto 0);
            pos : out std_logic_vector(7 downto 0)
        );
    end component;

    component clk_180kHz
        port(
            clk : in std_logic;
            reset : in std_logic;
```

```
            clk_out : out std_logic
        );
    end component;

    component servo_pwm
        port(
            clk : in std_logic;
            reset : in std_logic;
            pos : in std_logic_vector(7 downto 0);
            servo : out std_logic
        );
    end component;

    signal clk_out : std_logic := '0';
    signal pos : std_logic_vector(7 downto 0);
begin
    switch_in : switch port map(
        sw => sw, pos => pos
    );

    clk_180kHz_map : clk_180kHz port map(
        clk => clk, reset => reset, clk_out => clk_out
    );

    servo_pwm_map : servo_pwm port map(
        clk => clk_out, reset => reset,
        pos => pos, servo => servo
    );

end Behavioral;
```

clk_180kHz.vhd

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity clk_180kHz is
    port(
        clk : in std_logic;
        reset : in std_logic;
        clk_out : out std_logic
    );
end clk_180kHz;

architecture Behavioral of clk_180kHz is
    signal temp : std_logic;
    signal counter : integer range 0 to 346 := 0;
begin
    frequency_divider : process(reset, clk)
    begin
        if(reset = '1') then
            temp <= '0';
            counter <= 0;
```

```
        elsif rising_edge(clk) then
            if(counter = 346) then
                temp <= NOT(temp);
                counter <= 0;
            else
                counter <= counter + 1;
            end if;
        end if;
    end process;

    clk_out <= temp;

end Behavioral;
```

servo_pwm.vhd

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity servo_pwm is
    port(
        clk : in std_logic;
        reset : in std_logic;
        pos : in std_logic_vector(7 downto 0);
        servo : out std_logic
    );
end servo_pwm;

architecture Behavioral of servo_pwm is
    signal cnt : unsigned(11 downto 0);
    signal pwmi : unsigned(9 downto 0);
begin
    pwmi <= "00" & unsigned(pos) + 180;
    counter : process(reset, clk) begin
        if(reset = '1') then
            cnt <= (others => '0');
        elsif rising_edge(clk) then
            if(cnt = 3599) then
                cnt <= (others => '0');
            else
                cnt <= cnt + 1;
            end if;
        end if;
    end process;

    servo <= '1' when (cnt < pwmi) else '0';

end Behavioral;
```

switch.vhd

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/04/2018 01:16:17 PM
-- Design Name:
-- Module Name: swt - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity switch is
    port(
        sw : in std_logic_vector(3 downto 0);
        pos : out std_logic_vector(7 downto 0)
    );
end switch;

architecture Behavioral of switch is

begin
    P1 : process begin
        switch : case sw is
            when "0000" => pos <= X"00";
            when "0001" => pos <= X"10";
            when "0010" => pos <= X"20";
```

```
            when "0011" => pos <= X"30";
            when "0100" => pos <= X"40";
            when "0101" => pos <= X"50";
            when "0110" => pos <= X"60";
            when "0111" => pos <= X"70";
            when "1000" => pos <= X"80";

            when "1001" => pos <= X"90";
            when "1010" => pos <= X"A0";
            when "1011" => pos <= X"B0";
            when "1100" => pos <= X"C0";
            when "1101" => pos <= X"D0";
            when "1110" => pos <= X"E0";
            when others => pos <= X"00";
        end case switch;
    end process;

end Behavioral;
```

servo_pwm_clk180kHz_tb.vhd

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity servo_pwm_clk180kHz_tb is
end servo_pwm_clk180kHz_tb;

architecture Test of servo_pwm_clk180kHz_tb is
    component servo_pwm_clk_180kHz
        port(
            clk : in std_logic;
            reset : in std_logic;
            pos : in std_logic_vector(9 downto 0);
            servo : out std_logic
        );
    end component;

    signal clk : std_logic := '0';
    signal reset : std_logic := '0';
    signal pos : std_logic_vector(9 downto 0) := (others => '0');

    signal servo : std_logic;

    constant clk_period : time := 8 ns;
```

```
begin
    -- Unit Under Test
    uut: servo_pwm_clk_180kHz port map(
        clk => clk,
        reset => reset,
        pos => pos,
        servo => servo
    );

    clk_process : process begin
        clk <= '0';
        wait for clk_period / 2;
        clk <= '1';
        wait for clk_period / 2;
    end process;

    stimuli : process begin
        reset <= '1';
        wait for 50 ns;
        reset <= '0';
        wait for 50 ns;
        pos <= b"00_0000_0000";
        wait for 20 ms;
        pos <= b"00_0011_1100";
        wait for 20 ms;
        pos <= b"00_0111_1000";
        wait for 20 ms;
        pos <= b"00_1011_0100";
        wait;
    end process;

end Test;
```

zybo master xdc로 구글에 검색

zybo.xdc

```
## This file is a general .xdc for the ZYBO Rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used signals according to the project


##Clock signal
set_property -dict {PACKAGE_PIN L16 IOSTANDARD LVCMOS33}
[get_ports clk]
create_clock -period 8.000 -name sys_clk_pin -waveform {0.000
4.000} -add [get_ports clk]


##Switches
```

```
set_property -dict {PACKAGE_PIN G15 IOSTANDARD LVCMOS33}
[get_ports {sw[0]}]
set_property -dict {PACKAGE_PIN P15 IOSTANDARD LVCMOS33}
[get_ports {sw[1]}]
set_property -dict {PACKAGE_PIN W13 IOSTANDARD LVCMOS33}
[get_ports {sw[2]}]
set_property -dict {PACKAGE_PIN T16 IOSTANDARD LVCMOS33}
[get_ports {sw[3]}]


##Buttons
#set_property -dict { PACKAGE_PIN R18   IOSTANDARD LVCMOS33 }
[get_ports { btn[0] }]; #IO_L20N_T3_34 Sch=BTN0
#set_property -dict { PACKAGE_PIN P16   IOSTANDARD LVCMOS33 }
[get_ports { btn[1] }]; #IO_L24N_T3_34 Sch=BTN1
#set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 }
[get_ports { btn[2] }]; #IO_L18P_T2_34 Sch=BTN2
#set_property -dict { PACKAGE_PIN Y16   IOSTANDARD LVCMOS33 }
[get_ports { btn[3] }]; #IO_L7P_T1_34 Sch=BTN3


##LEDs
#set_property -dict { PACKAGE_PIN M14   IOSTANDARD LVCMOS33 }
[get_ports { led[0] }]; #IO_L23P_T3_35 Sch=LED0
#set_property -dict { PACKAGE_PIN M15   IOSTANDARD LVCMOS33 }
[get_ports { led[1] }]; #IO_L23N_T3_35 Sch=LED1
#set_property -dict { PACKAGE_PIN G14   IOSTANDARD LVCMOS33 }
[get_ports { led[2] }]; #IO_0_35=Sch=LED2
#set_property -dict { PACKAGE_PIN D18   IOSTANDARD LVCMOS33 }
[get_ports { led[3] }]; #IO_L3N_T0_DQS_AD1N_35 Sch=LED3


##I2S Audio Codec
#set_property -dict { PACKAGE_PIN K18   IOSTANDARD LVCMOS33 }
[get_ports ac_bclk]; #IO_L12N_T1_MRCC_35 Sch=AC_BCLK
#set_property -dict { PACKAGE_PIN T19   IOSTANDARD LVCMOS33 }
[get_ports ac_mclk]; #IO_25_34 Sch=AC_MCLK
#set_property -dict { PACKAGE_PIN P18   IOSTANDARD LVCMOS33 }
[get_ports ac_muten]; #IO_L23N_T3_34 Sch=AC_MUTEN
#set_property -dict { PACKAGE_PIN M17   IOSTANDARD LVCMOS33 }
[get_ports ac_pbdat]; #IO_L8P_T1_AD10P_35 Sch=AC_PBDAT
#set_property -dict { PACKAGE_PIN L17   IOSTANDARD LVCMOS33 }
[get_ports ac_pblrc]; #IO_L11N_T1_SRCC_35 Sch=AC_PBLRC
#set_property -dict { PACKAGE_PIN K17   IOSTANDARD LVCMOS33 }
[get_ports ac_recdat]; #IO_L12P_T1_MRCC_35 Sch=AC_RECDAT
#set_property -dict { PACKAGE_PIN M18   IOSTANDARD LVCMOS33 }
[get_ports ac_reclrc]; #IO_L8N_T1_AD10N_35 Sch=AC_RECLRC


##Audio Codec/external EEPROM IIC bus
#set_property -dict { PACKAGE_PIN N18   IOSTANDARD LVCMOS33 }
[get_ports ac_scl]; #IO_L13P_T2_MRCC_34 Sch=AC_SCL
```

```
#set_property -dict { PACKAGE_PIN N17   IOSTANDARD LVCMOS33 }
[get_ports ac_sda]; #IO_L23P_T3_34 Sch=AC_SDA


##Additional Ethernet signals
#set_property -dict { PACKAGE_PIN F16   IOSTANDARD LVCMOS33 }
[get_ports eth_int_b]; #IO_L6P_T0_35 Sch=ETH_INT_B
#set_property -dict { PACKAGE_PIN E17   IOSTANDARD LVCMOS33 }
[get_ports eth_rst_b]; #IO_L3P_T0_DQS_AD1P_35 Sch=ETH_RST_B


##HDMI Signals
#set_property -dict { PACKAGE_PIN H17   IOSTANDARD TMDS_33 }
[get_ports hdmi_clk_n]; #IO_L13N_T2_MRCC_35 Sch=HDMI_CLK_N
#set_property -dict { PACKAGE_PIN H16   IOSTANDARD TMDS_33 }
[get_ports hdmi_clk_p]; #IO_L13P_T2_MRCC_35 Sch=HDMI_CLK_P
#set_property -dict { PACKAGE_PIN D20   IOSTANDARD TMDS_33 }
[get_ports { hdmi_d_n[0] }]; #IO_L4N_T0_35 Sch=HDMI_D0_N
#set_property -dict { PACKAGE_PIN D19   IOSTANDARD TMDS_33 }
[get_ports { hdmi_d_p[0] }]; #IO_L4P_T0_35 Sch=HDMI_D0_P
#set_property -dict { PACKAGE_PIN B20   IOSTANDARD TMDS_33 }
[get_ports { hdmi_d_n[1] }]; #IO_L1N_T0_AD0N_35 Sch=HDMI_D1_N
#set_property -dict { PACKAGE_PIN C20   IOSTANDARD TMDS_33 }
[get_ports { hdmi_d_p[1] }]; #IO_L1P_T0_AD0P_35 Sch=HDMI_D1_P
#set_property -dict { PACKAGE_PIN A20   IOSTANDARD TMDS_33 }
[get_ports { hdmi_d_n[2] }]; #IO_L2N_T0_AD8N_35 Sch=HDMI_D2_N
#set_property -dict { PACKAGE_PIN B19   IOSTANDARD TMDS_33 }
[get_ports { hdmi_d_p[2] }]; #IO_L2P_T0_AD8P_35 Sch=HDMI_D2_P
#set_property -dict { PACKAGE_PIN E19   IOSTANDARD LVCMOS33 }
[get_ports hdmi_cec]; #IO_L5N_T0_AD9N_35 Sch=HDMI_CEC
#set_property -dict { PACKAGE_PIN E18   IOSTANDARD LVCMOS33 }
[get_ports hdmi_hpd]; #IO_L5P_T0_AD9P_35 Sch=HDMI_HPD
#set_property -dict { PACKAGE_PIN F17   IOSTANDARD LVCMOS33 }
[get_ports hdmi_out_en]; #IO_L6N_T0_VREF_35 Sch=HDMI_OUT_EN
#set_property -dict { PACKAGE_PIN G17   IOSTANDARD LVCMOS33 }
[get_ports hdmi_scl]; #IO_L16P_T2_35 Sch=HDMI_SCL
#set_property -dict { PACKAGE_PIN G18   IOSTANDARD LVCMOS33 }
[get_ports hdmi_sda]; #IO_L16N_T2_35 Sch=HDMI_SDA


##Pmod Header JA (XADC)
#set_property -dict { PACKAGE_PIN N15   IOSTANDARD LVCMOS33 }
[get_ports { ja_p[0] }]; #IO_L21P_T3_DQS_AD14P_35 Sch=JA1_R_p
#set_property -dict { PACKAGE_PIN L14   IOSTANDARD LVCMOS33 }
[get_ports { ja_p[1] }]; #IO_L22P_T3_AD7P_35 Sch=JA2_R_P
#set_property -dict { PACKAGE_PIN K16   IOSTANDARD LVCMOS33 }
[get_ports { ja_p[2] }]; #IO_L24P_T3_AD15P_35 Sch=JA3_R_P
#set_property -dict { PACKAGE_PIN K14   IOSTANDARD LVCMOS33 }
[get_ports { ja_p[3] }]; #IO_L20P_T3_AD6P_35 Sch=JA4_R_P
#set_property -dict { PACKAGE_PIN N16   IOSTANDARD LVCMOS33 }
[get_ports { ja_n[0] }]; #IO_L21N_T3_DQS_AD14N_35 Sch=JA1_R_N
#set_property -dict { PACKAGE_PIN L15   IOSTANDARD LVCMOS33 }
[get_ports { ja_n[1] }]; #IO_L22N_T3_AD7N_35 Sch=JA2_R_N
```

```
#set_property -dict { PACKAGE_PIN J16   IOSTANDARD LVCMOS33 }
[get_ports { ja_n[2] }]; #IO_L24N_T3_AD15N_35 Sch=JA3_R_N
#set_property -dict { PACKAGE_PIN J14   IOSTANDARD LVCMOS33 }
[get_ports { ja_n[3] }]; #IO_L20N_T3_AD6N_35 Sch=JA4_R_N


##Pmod Header JB
#set_property -dict { PACKAGE_PIN T20   IOSTANDARD LVCMOS33 }
[get_ports { jb_p[0] }]; #IO_L15P_T2_DQS_34 Sch=JB1_p
#set_property -dict { PACKAGE_PIN U20   IOSTANDARD LVCMOS33 }
[get_ports { jb_n[0] }]; #IO_L15N_T2_DQS_34 Sch=JB1_N
#set_property -dict { PACKAGE_PIN V20   IOSTANDARD LVCMOS33 }
[get_ports { jb_p[1] }]; #IO_L16P_T2_34 Sch=JB2_P
#set_property -dict { PACKAGE_PIN W20   IOSTANDARD LVCMOS33 }
[get_ports { jb_n[1] }]; #IO_L16N_T2_34 Sch=JB2_N
#set_property -dict { PACKAGE_PIN Y18   IOSTANDARD LVCMOS33 }
[get_ports { jb_p[2] }]; #IO_L17P_T2_34 Sch=JB3_P
#set_property -dict { PACKAGE_PIN Y19   IOSTANDARD LVCMOS33 }
[get_ports { jb_n[2] }]; #IO_L17N_T2_34 Sch=JB3_N
#set_property -dict { PACKAGE_PIN W18   IOSTANDARD LVCMOS33 }
[get_ports { jb_p[3] }]; #IO_L22P_T3_34 Sch=JB4_P
#set_property -dict { PACKAGE_PIN W19   IOSTANDARD LVCMOS33 }
[get_ports { jb_n[3] }]; #IO_L22N_T3_34 Sch=JB4_N


##Pmod Header JC
#set_property -dict { PACKAGE_PIN W15   IOSTANDARD LVCMOS33 }
[get_ports { jc_n[0] }]; #IO_L10N_T1_34 Sch=JC1_N
#set_property -dict { PACKAGE_PIN T11   IOSTANDARD LVCMOS33 }
[get_ports { jc_p[1] }]; #IO_L1P_T0_34 Sch=JC2_P
#set_property -dict { PACKAGE_PIN T10   IOSTANDARD LVCMOS33 }
[get_ports { jc_n[1] }]; #IO_L1N_T0_34 Sch=JC2_N
#set_property -dict { PACKAGE_PIN W14   IOSTANDARD LVCMOS33 }
[get_ports { jc_p[2] }]; #IO_L8P_T1_34 Sch=JC3_P
#set_property -dict { PACKAGE_PIN Y14   IOSTANDARD LVCMOS33 }
[get_ports { jc_n[2] }]; #IO_L8N_T1_34 Sch=JC3_N
#set_property -dict { PACKAGE_PIN T12   IOSTANDARD LVCMOS33 }
[get_ports { jc_p[3] }]; #IO_L2P_T0_34 Sch=JC4_P
#set_property -dict { PACKAGE_PIN U12   IOSTANDARD LVCMOS33 }
[get_ports { jc_n[3] }]; #IO_L2N_T0_34 Sch=JC4_N


##Pmod Header JD
#set_property -dict { PACKAGE_PIN T14   IOSTANDARD LVCMOS33 }
[get_ports { jd_p[0] }]; #IO_L5P_T0_34 Sch=JD1_P
#set_property -dict { PACKAGE_PIN T15   IOSTANDARD LVCMOS33 }
[get_ports { jd_n[0] }]; #IO_L5N_T0_34 Sch=JD1_N
#set_property -dict { PACKAGE_PIN P14   IOSTANDARD LVCMOS33 }
[get_ports { jd_p[1] }]; #IO_L6P_T0_34 Sch=JD2_P
#set_property -dict { PACKAGE_PIN R14   IOSTANDARD LVCMOS33 }
[get_ports { jd_n[1] }]; #IO_L6N_T0_VREF_34 Sch=JD2_N
#set_property -dict { PACKAGE_PIN U14   IOSTANDARD LVCMOS33 }
[get_ports { jd_p[2] }]; #IO_L11P_T1_SRCC_34 Sch=JD3_P
```

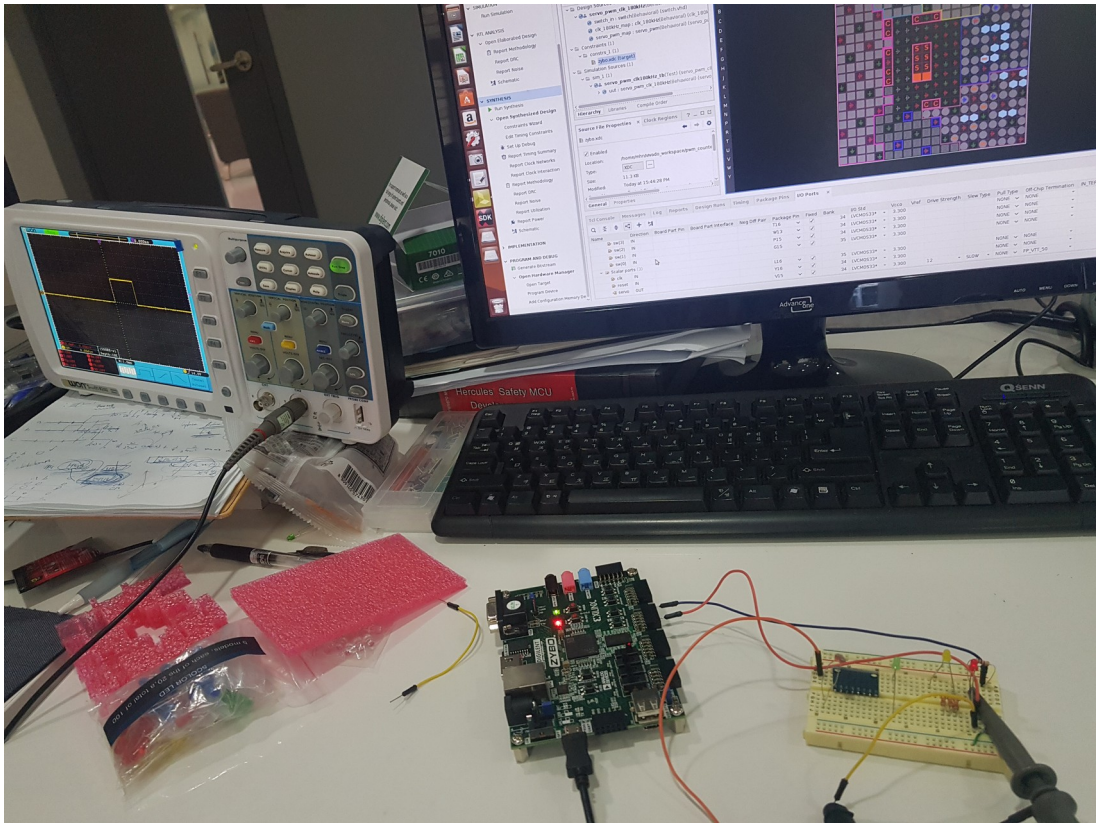
```
#set_property -dict { PACKAGE_PIN U15   IOSTANDARD LVCMOS33 }
[get_ports { jd_n[2] }]; #IO_L11N_T1_SRCC_34 Sch=JD3_N
#set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 }
[get_ports { jd_p[3] }]; #IO_L21P_T3_DQS_34 Sch=JD4_P
#set_property -dict { PACKAGE_PIN V18   IOSTANDARD LVCMOS33 }
[get_ports { jd_n[3] }]; #IO_L21N_T3_DQS_34 Sch=JD4_N


##Pmod Header JE
#set_property -dict { PACKAGE_PIN V12   IOSTANDARD LVCMOS33 }
[get_ports { je[0] }]; #IO_L4P_T0_34 Sch=JE1
#set_property -dict { PACKAGE_PIN W16   IOSTANDARD LVCMOS33 }
[get_ports { je[1] }]; #IO_L18N_T2_34 Sch=JE2
#set_property -dict { PACKAGE_PIN J15   IOSTANDARD LVCMOS33 }
[get_ports { je[2] }]; #IO_25_35 Sch=JE3
#set_property -dict { PACKAGE_PIN H15   IOSTANDARD LVCMOS33 }
[get_ports { je[3] }]; #IO_L19P_T3_35 Sch=JE4
#set_property -dict { PACKAGE_PIN V13   IOSTANDARD LVCMOS33 }
[get_ports { je[4] }]; #IO_L3N_T0_DQS_34 Sch=JE7
#set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 }
[get_ports { je[5] }]; #IO_L9N_T1_DQS_34 Sch=JE8
#set_property -dict { PACKAGE_PIN T17   IOSTANDARD LVCMOS33 }
[get_ports { je[6] }]; #IO_L20P_T3_34 Sch=JE9
#set_property -dict { PACKAGE_PIN Y17   IOSTANDARD LVCMOS33 }
[get_ports { je[7] }]; #IO_L7N_T1_34 Sch=JE10


##USB-OTG overcurrent detect pin
#set_property -dict { PACKAGE_PIN U13   IOSTANDARD LVCMOS33 }
[get_ports otg_oc]; #IO_L3P_T0_DQS_PUDC_B_34 Sch=OTG_OC


##VGA Connector
#set_property -dict { PACKAGE_PIN M19   IOSTANDARD LVCMOS33 }
[get_ports { vga_r[0] }]; #IO_L7P_T1_AD2P_35 Sch=VGA_R1
#set_property -dict { PACKAGE_PIN L20   IOSTANDARD LVCMOS33 }
[get_ports { vga_r[1] }]; #IO_L9N_T1_DQS_AD3N_35 Sch=VGA_R2
#set_property -dict { PACKAGE_PIN J20   IOSTANDARD LVCMOS33 }
[get_ports { vga_r[2] }]; #IO_L17P_T2_AD5P_35 Sch=VGA_R3
#set_property -dict { PACKAGE_PIN G20   IOSTANDARD LVCMOS33 }
[get_ports { vga_r[3] }]; #IO_L18N_T2_AD13N_35 Sch=VGA_R4
#set_property -dict { PACKAGE_PIN F19   IOSTANDARD LVCMOS33 }
[get_ports { vga_r[4] }]; #IO_L15P_T2_DQS_AD12P_35 Sch=VGA_R5
#set_property -dict { PACKAGE_PIN H18   IOSTANDARD LVCMOS33 }
[get_ports { vga_g[0] }]; #IO_L14N_T2_AD4N_SRCC_35 Sch=VGA_G0
#set_property -dict { PACKAGE_PIN N20   IOSTANDARD LVCMOS33 }
[get_ports { vga_g[1] }]; #IO_L14P_T2_SRCC_34 Sch=VGA_G1
#set_property -dict { PACKAGE_PIN L19   IOSTANDARD LVCMOS33 }
[get_ports { vga_g[2] }]; #IO_L9P_T1_DQS_AD3P_35 Sch=VGA_G2
#set_property -dict { PACKAGE_PIN J19   IOSTANDARD LVCMOS33 }
[get_ports { vga_g[3] }]; #IO_L10N_T1_AD11N_35 Sch=VGA_G3
#set_property -dict { PACKAGE_PIN H20   IOSTANDARD LVCMOS33 }
[get_ports { vga_g[4] }]; #IO_L17N_T2_AD5N_35 Sch=VGA_G4
```

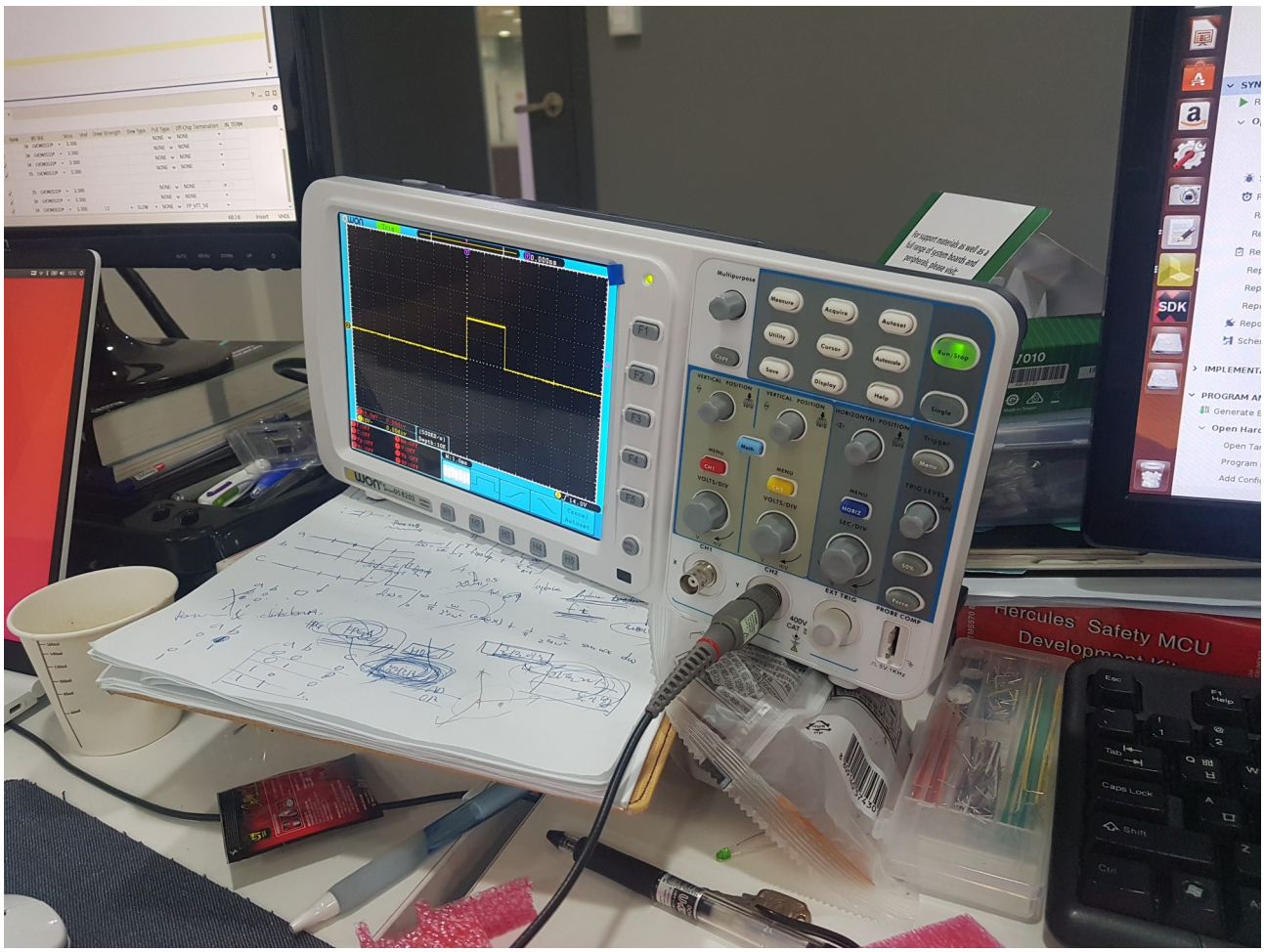
```
#set_property -dict { PACKAGE_PIN F20   IOSTANDARD LVCMOS33 }
[get_ports { vga_g[5] }]; #IO_L15N_T2_DQS_AD12N_35 Sch=VGA=G5
#set_property -dict { PACKAGE_PIN P20   IOSTANDARD LVCMOS33 }
[get_ports { vga_b[0] }]; #IO_L14N_T2_SRCC_34 Sch=VGA_B1
#set_property -dict { PACKAGE_PIN M20   IOSTANDARD LVCMOS33 }
[get_ports { vga_b[1] }]; #IO_L7N_T1_AD2N_35 Sch=VGA_B2
#set_property -dict { PACKAGE_PIN K19   IOSTANDARD LVCMOS33 }
[get_ports { vga_b[2] }]; #IO_L10P_T1_AD11P_35 Sch=VGA_B3
#set_property -dict { PACKAGE_PIN J18   IOSTANDARD LVCMOS33 }
[get_ports { vga_b[3] }]; #IO_L14P_T2_AD4P_SRCC_35 Sch=VGA_B4
#set_property -dict { PACKAGE_PIN G19   IOSTANDARD LVCMOS33 }
[get_ports { vga_b[4] }]; #IO_L18P_T2_AD13P_35 Sch=VGA_B5
#set_property -dict { PACKAGE_PIN P19   IOSTANDARD LVCMOS33 }
[get_ports vga_hs]; #IO_L13N_T2_MRCC_34 Sch=VGA_HS
#set_property -dict { PACKAGE_PIN R19 IOSTANDARD LVCMOS33 }
[get_ports vga_vs]; #IO_0_34 Sch=VGA_VS

set_property IOSTANDARD LVCMOS33 [get_ports reset]
set_property IOSTANDARD LVCMOS33 [get_ports servo]

set_property PACKAGE_PIN Y16 [get_ports reset]
set_property PACKAGE_PIN V15 [get_ports servo]
```

Run Synthesis

Run implementation

Generate Bitstream




Hercules Safety MCU
Development
SDK

How_to_boot_Zybo_with_SD_Card.pdf