TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

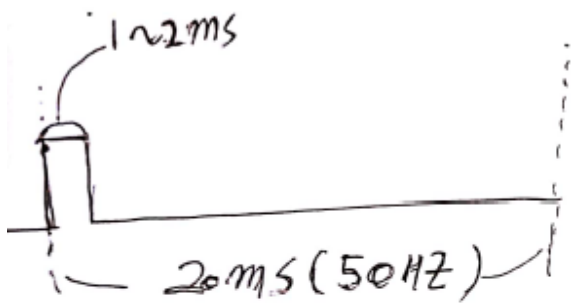
강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 – GJ (박현우)
uc820@naver.com

목차

Zynq FPGA on Petalinux for PWM

1) Pwm clock 분주 sequence

PWM period

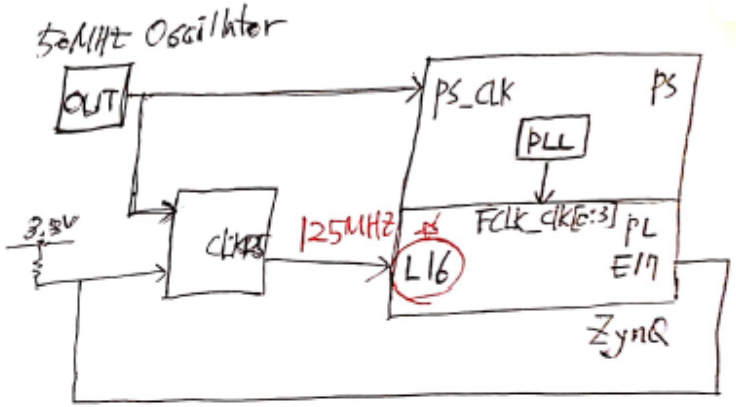


$f = \frac{1}{T} \Rightarrow 1°$ 단위제어 $\Rightarrow f = \left(\frac{1ms}{180}\right)^{-1}$

$\therefore f = 180kHz$

즉, 180kHz 신호를 만들어 주어야 함.

Zybo



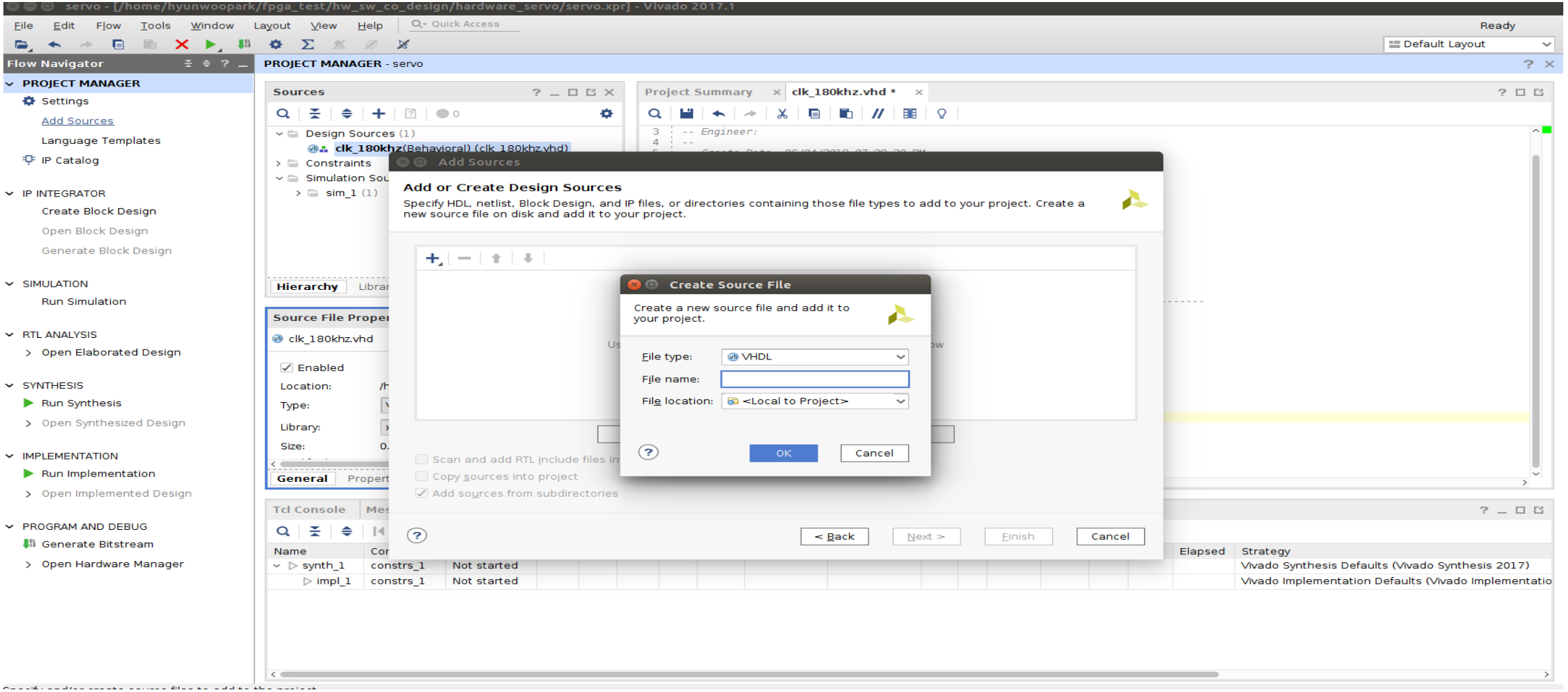
125MHz로 180kHz 만들기

$125 \times 10^6 = 180 \times 10^3 \times scale$

$\therefore scale = 694$

즉, 125MHz를 694번 count하면 180kHz가 만들어짐.

2) servo_pwm with test_bench & VHDL Code 1



2) servo_pwm with test_bench & VHDL Code 2

servo - [/home/hyunwoopark/fpga_test/hw_sw_co_design/hardware_servo/servo.xpr] - Vivado 2017.1

File Edit Flow Tools Window Layout View Help Quick Access

Synthesis Complete

Default Layout

Flow Navigator

- PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION
  - Run Simulation
- RTL ANALYSIS
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

PROJECT MANAGER - servo

Sources

- Design Sources (1)
  - clk_180khz(Behavioral) (clk_180khz.vhd)
- Constraints
- Simulation Sources (1)
  - sim_1 (1)

Hierarchy | Libraries | Compile Order

Source File Properties

clk_180khz.vhd

Enabled

Location: /home/hyunwoopark/fpga_test/hw_sw_co

Type: VHDL

Library: xil_defaultlib

Size: 1.2 KB

General | Properties

Project Summary | clk_180khz.vhd

```
entity clk_180khz is
    port(
        clk_in : 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_in)
    begin
        if(reset = '1') then
            temp <= '0';
            counter <= 0;
        elsif rising_edge(clk_in) 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;
```

Tcl Console | Messages | Log | Reports | Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	Strategy
synth_1	constrs_1	synth_design Complete!								13	10	0.00	0	0	6/4/18 7:36 PM	00:00:20	Vivado Synthesis Defaults (Viva
impl_1	constrs_1	Not started															Vivado Implementation Defaults

2) servo_pwm with test_bench & VHDL Code 3

servo - [/home/hyunwoopark/fpga_test/hw_sw_co_design/hardware_servo/servo.xpr] - Vivado 2017.1

File Edit Flow Tools Window Layout View Help Quick Access

Synthesis Out-of-date details

Default Layout

Flow Navigator

- PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION
  - Run Simulation
- RTL ANALYSIS
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

PROJECT MANAGER - servo

Sources

- Design Sources (2)
  - clk_180khz(Behavioral) (clk_180khz.vhd)
  - server_pwm(Behavioral) (server_pwm.vhd)
- Constraints
- Simulation Sources (2)
  - sim_1 (2)

Hierarchy | Libraries | Compile Order

Source File Properties

server_pwm.vhd

Enabled

Location: /home/hyunwoopark/fpga_test/hw_sw_co

Type: VHDL

Library: xil_defaultlib

Size: 0.9 KB

General | Properties

Project Summary | clk_180khz.vhd | server_pwm.vhd *

```
12  --
13  -- Dependencies:
14  --
15  -- Revision:
16  -- Revision 0.01 - File Created
17  -- Additional Comments:
18  --
19  ----------------------------------------------------------------------------------
20
21
22  library IEEE;
23  use IEEE.STD_LOGIC_1164.ALL;
24  use IEEE.NUMERIC_STD.ALL;
25
26  entity server_pwm is
27
28  end server_pwm;
29
30  architecture Behavioral of server_pwm is
31
32  begin
33
34
35  end Behavioral;
36
```

Tcl Console | Messages | Log | Reports | Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	Strategy
synth_1	constrs_1	Synthesis Out-of-date								13	10	0.00	0	0	6/4/18 7:36 PM	00:00:20	Vivado Synthesis Defaults (Vivac
impl_1	constrs_1	Not started															Vivado Implementation Defaults

2) servo_pwm with test_bench & VHDL Code 4

File Edit Flow Tools Window Layout View Help Quick Access

Synthesis Out-of-date details

Default Layout

Flow Navigator

PROJECT MANAGER - servo

- PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION
  - Run Simulation
- RTL ANALYSIS
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

Sources — Updating

- Design Sources (2)
  - clk_180khz(Behavioral) (clk_180khz.vhd)
  - server_pwm(Behavioral) (server_pwm.vhd)
- Constraints
- Simulation Sources (2)
  - sim_1 (2)

Hierarchy | Libraries | Compile Or

Source File Properties

server_pwm.vhd

Enabled

Location: /home/hyunwoopark/

Type: VHDL

Library: xil_defaultlib

Size: 1.4 KB

General | Properties

Project Summary | clk_180khz.vhd | server_pwm.vhd

```
24  use IEEE.NUMERIC_STD.ALL;
25
26  entity server_pwm is
27          port(
28          clk : in std_logic;
29          reset : in std_logic;
30          pos : in std_logic_vector(9 downto 0);
```

Define Module

Define a module and specify I/O Ports to add to your source file.
For each port specified:
MSB and LSB values will be ignored unless its Bus column is checked.
Ports with blank names will not be written.

Module Definition

Entity name: servo_pwm_180khz

Architecture name: Behavioral

I/O Port Definitions

Port Name	Direction	Bus	MSB	LSB
	in	☐	0	0

OK | Cancel

Tcl Console | Messages | Log

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	Strategy
synth_1	constrs_1	Synthesis Out-of-date								13	10	0.00	0	0	6/4/18 7:36 PM	00:00:20	Vivado Synthesis Defaults (Vivac
impl_1	constrs_1	Not started															Vivado Implementation Defaults

2) servo_pwm with test_bench & VHDL Code 5

File Edit Flow Tools Window Layout View Help Quick Access

Synthesis Out-of-date details

Default Layout

Flow Navigator

- PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION
  - Run Simulation
- RTL ANALYSIS
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

PROJECT MANAGER - servo

Sources

- Design Sources (3)
  - clk_180khz(Behavioral) (clk_180khz.vhd)
  - server_pwm(Behavioral) (server_pwm.vhd)
  - servo_pwm_180khz(Behavioral) (servo_pwm_180kh
- Constraints
- Simulation Sources (3)
  - sim_1 (3)
    - servo_pwm_180khz(Behavioral) (servo_pwr
    - server_pwm(Behavioral) (server_pwm.vhd)

Hierarchy Libraries Compile Order

Source File Properties

servo_pwm_clk180khz_tb.vhd

- Enabled
- Location: /home/hyunwoopark/fpga_test/hw_sw_co
- Type: VHDL
- Library: xil_defaultlib
- Size: 1.1 KB

General Properties

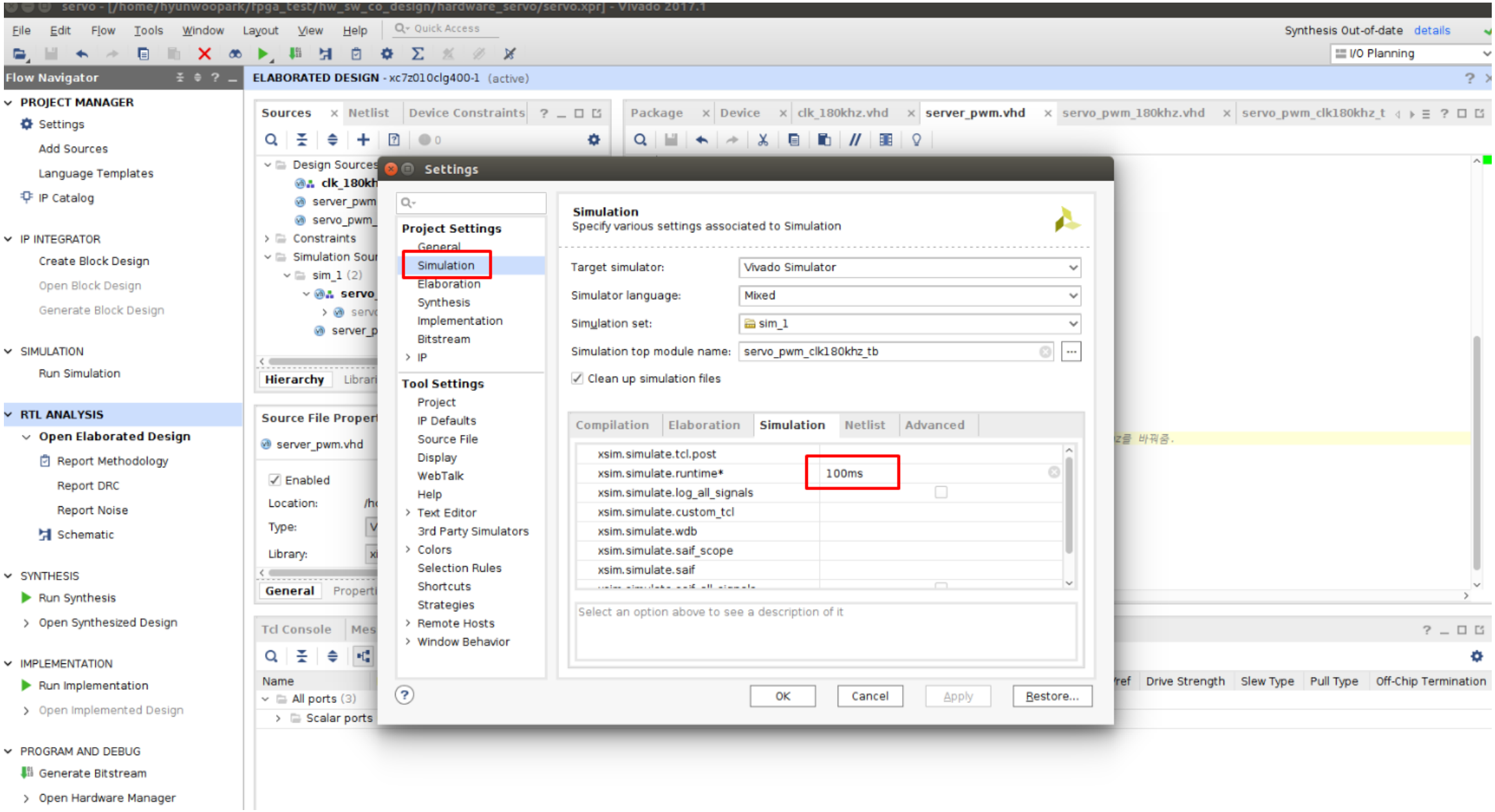
Project Summary | clk_180khz.vhd | server_pwm.vhd | servo_pwm_180khz.vhd | servo_pwm_clk180khz_tb.vhd *

```
20
21
22  library IEEE;
23  use IEEE.STD_LOGIC_1164.ALL;
24
25  entity servo_pwm_clk180khz_tb is
26  end servo_pwm_clk180khz_tb;
27
28
29  architecture Behavioral of servo_pwm_clk180khz_tb is
30
31          component servo_pwm_180khz
32              port(
33                  clk : in std_logic;
34                  reset : in std_logic;
35                  pos : in std_logic_vector(9 downto 0);
36                  servo : out std_logic
37              );
38          end component;
39
40      signal clk : std_logic := '0';
41      signal reset : std_logic := '0';
42      signal pos : std_logic_vector(9 downto 0) := (others => '0');
43
44      signal servo : std_logic;
45
46      constant clk_period : time := 8 ns;    --  f = 125 Mhz => 8 ns;
47
48  begin
49      -- Unit under Test
50      uut: servo_pwm_180khz port map(
51          clk => clk,
52          reset => reset,
```

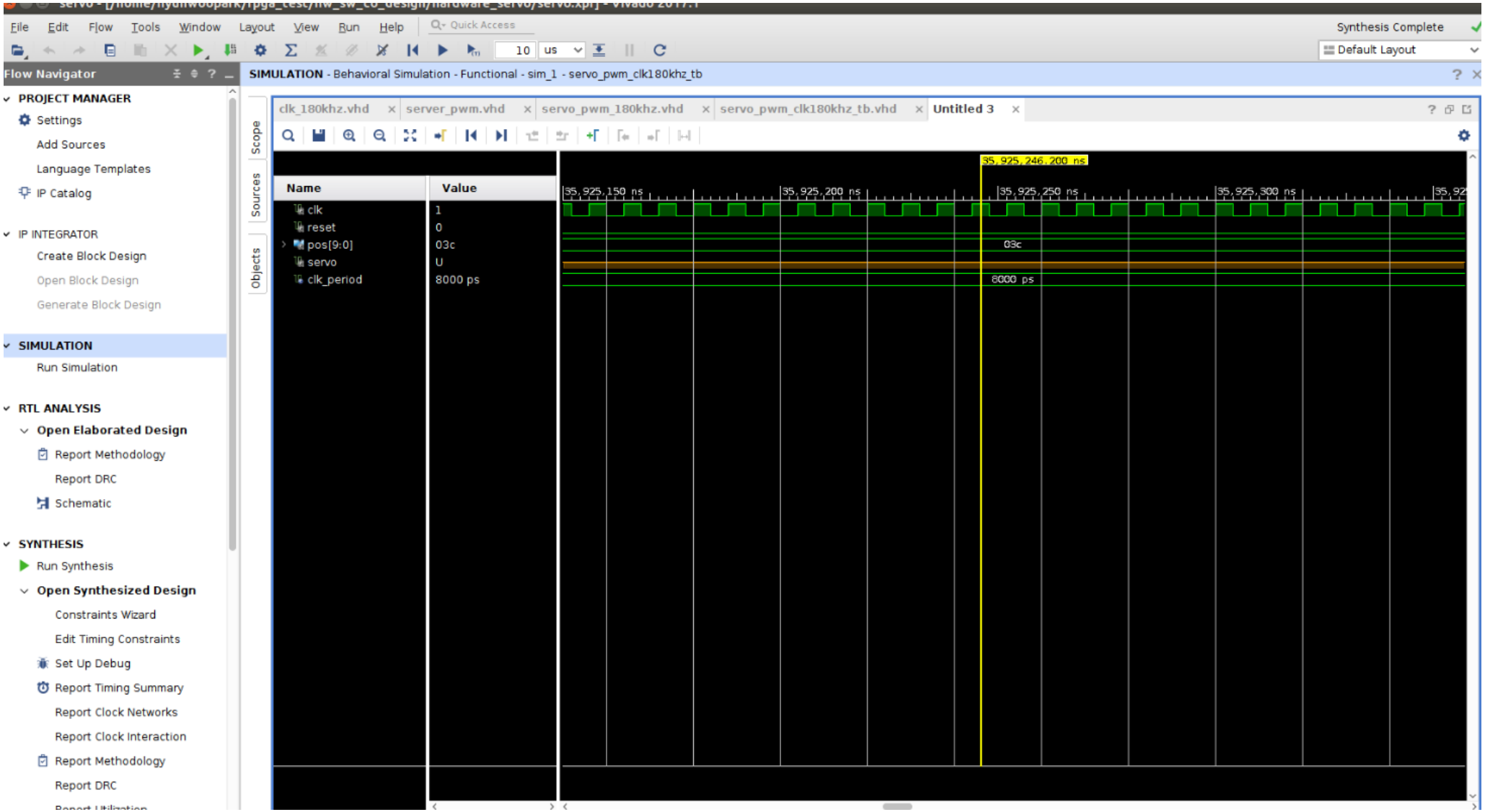
Tcl Console | Messages | Log | Reports | Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	Strategy
synth_1	constrs_1	Synthesis Out-of-date								13	10	0.00	0	0	6/4/18 7:36 PM	00:00:20	Vivado Synthesis Defaults (Viva
impl_1	constrs_1	Not started															Vivado Implementation Default

2) servo_pwm with test_bench & VHDL Code 6



2) servo_pwm with test_bench & VHDL Code 7



2) servo_pwm with test_bench & VHDL Code 8

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;
```

Pwm_servo.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;
```

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;
```

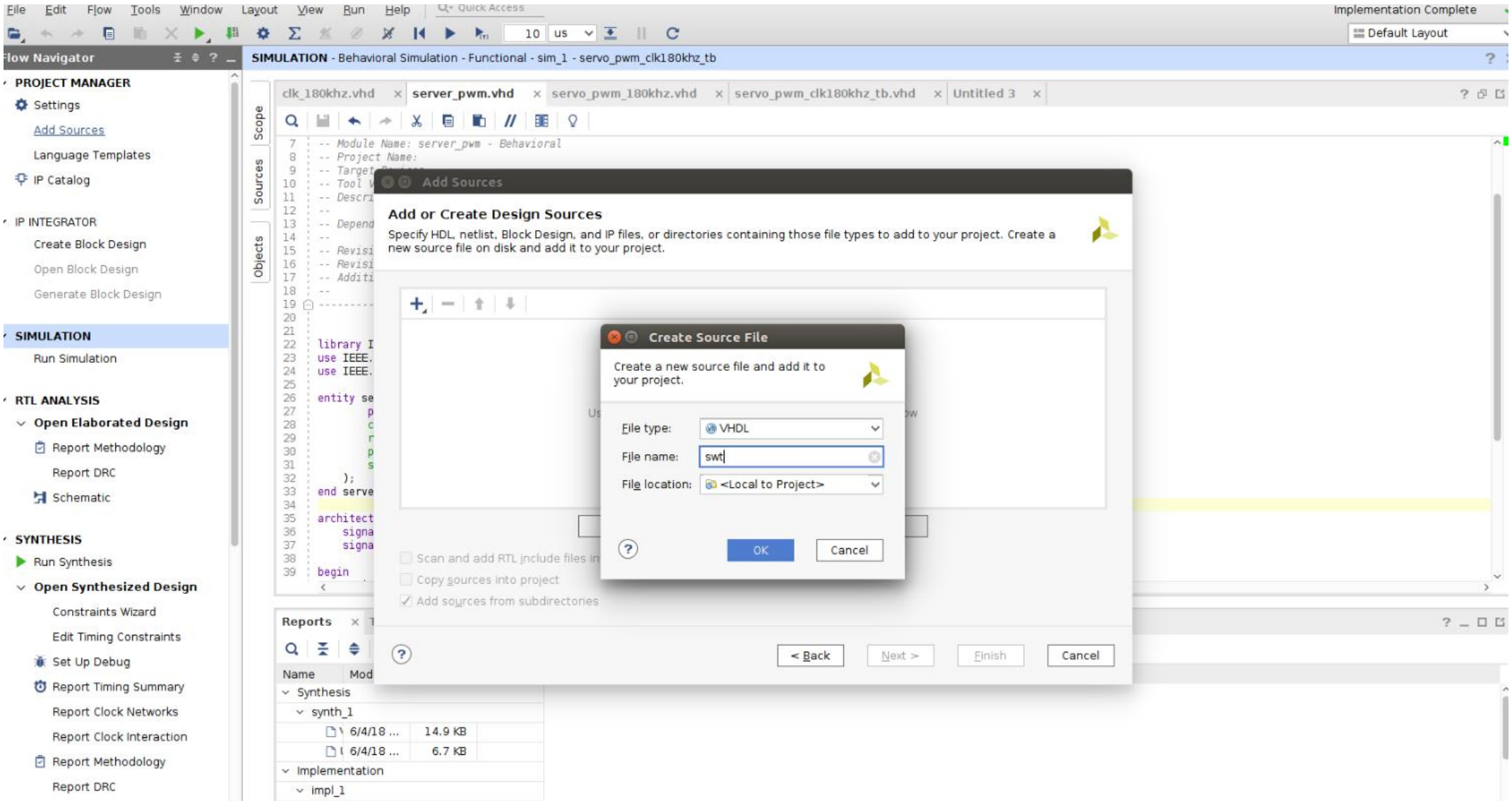
2) servo_pwm with test_bench & VHDL Code 9

Servo_pwm_clk_180kHz_tb.vhd

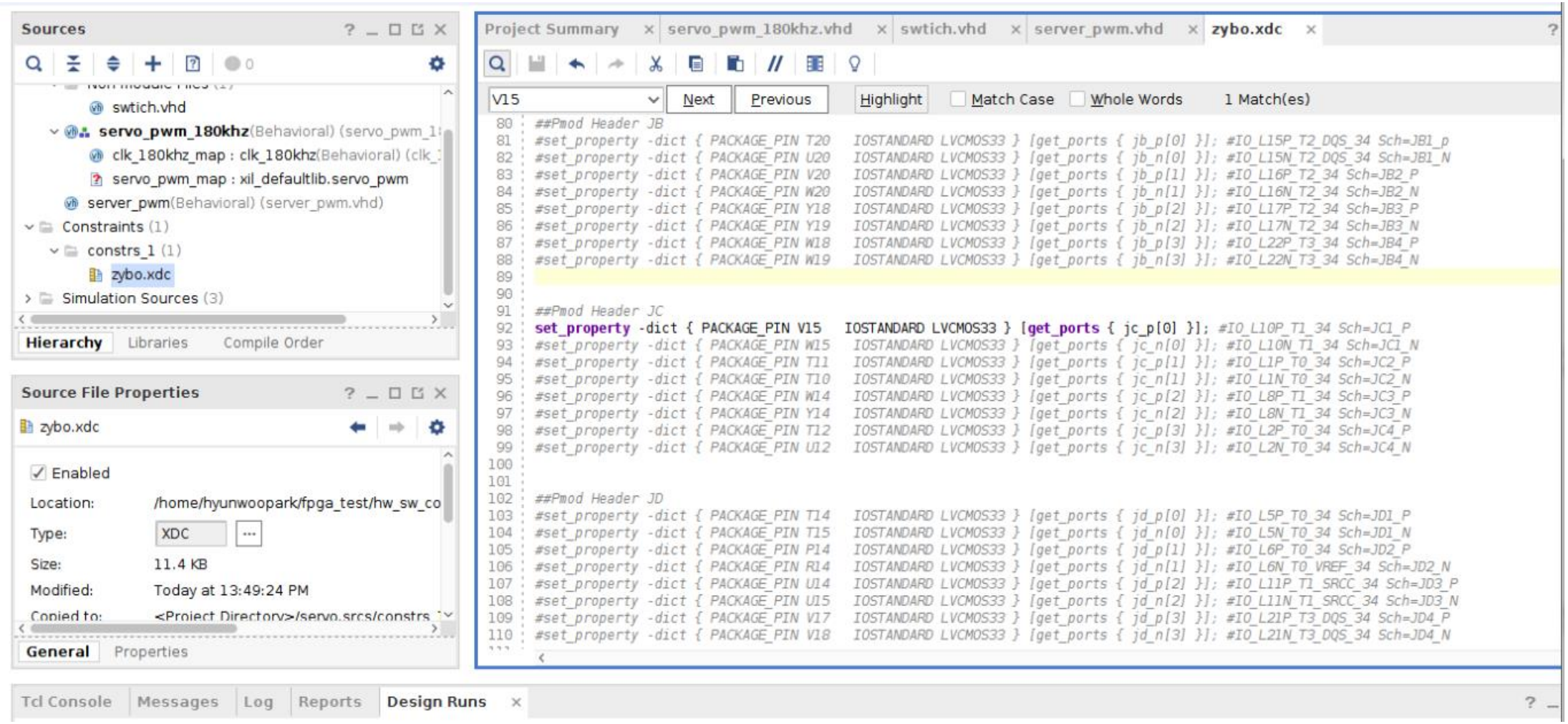
```
library IEEE;

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_clk180kHz
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_clk180kHz 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;
```

3) pwm switch 만들기 & VHDL Code 1



3) pwm switch 만들기 & VHDL Code 2



3) pwm switch 만들기 & VHDL Code 3

PROJECT MANAGER - servo

Sources

- swtich.vhd
- servo_pwm_180khz(Behavioral) (servo_pwm_1
  - clk_180khz_map : clk_180khz(Behavioral) (clk_
  - servo_pwm_map : xil_defaultlib.servo_pwm
- server_pwm(Behavioral) (server_pwm.vhd)
- Constraints (1)
  - constrs_1 (1)
    - zybo.xdc
- Simulation Sources (3)

Hierarchy | Libraries | Compile Order

Source File Properties

zybo.xdc

- Enabled
- Location: /home/hyunwoopark/fpga_test/hw_sw_co
- Type: XDC
- Size: 11.4 KB
- Modified: Today at 13:49:24 PM
- Copied to: <Project Directory>/servo.srcs/constrs_

General | Properties

Project Summary | servo_pwm_180khz.vhd | swtich.vhd | server_pwm.vhd | zybo.xdc *

V15 | Next | Previous | Highlight | Match Case | Whole Words | 1 Match(es)

```
## 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 }]; #IO_L11P_T1_SRCC_35 Sch=sysclk
create_clock -add -name sys_clk_pin -period 8.00 -waveform {0 4} [get_ports { clk }];


##Switches
set_property -dict { PACKAGE_PIN G15   IOSTANDARD LVCMOS33 } [get_ports { sw[0] }]; #IO_L19N_T3_VREF_35 Sch=SW0
set_property -dict { PACKAGE_PIN P15   IOSTANDARD LVCMOS33 } [get_ports { sw[1] }];  #IO_L24P_T3_34 Sch=SW1
set_property -dict { PACKAGE_PIN W13   IOSTANDARD LVCMOS33 } [get_ports { sw[2] }]; #IO_L4N_T0_34 Sch=SW2
set_property -dict { PACKAGE_PIN T16   IOSTANDARD LVCMOS33 } [get_ports { sw[3] }]; #IO_L9P_T1_DQS_34 Sch=SW3


##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
```

Tcl Console | Messages | Log | Reports | Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	Strategy
synth_1	constrs_1	Synthesis Out-of-date								13	10	0.00	0	0	6/4/18 1:30 PM	00:00:17	Vivado Synthesis Defaults (Vivad
impl_1	constrs_1	Not started															Vivado Implementation Defaults

3) pwm switch 만들기 & VHDL Code 4

clk_180kHz.vhd

pwm_servo.vhd

servo_pwm_clk_180kHz.vhd

switch.vhd

zybo.xdc

```vhdl
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;
```

```vhdl
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;
```

```vhdl
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;
```

3) pwm switch 만들기 & VHDL Code 5

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.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;
```

```xdc
##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]}]


set_property IOSTANDARD LVCMOS33 [get_ports reset]


set_property PACKAGE_PIN T19 [get_ports reset]
```

5) PWM ip로 내부 LED & 외부 LED 결과