zybo 디바이스드라이버 제작

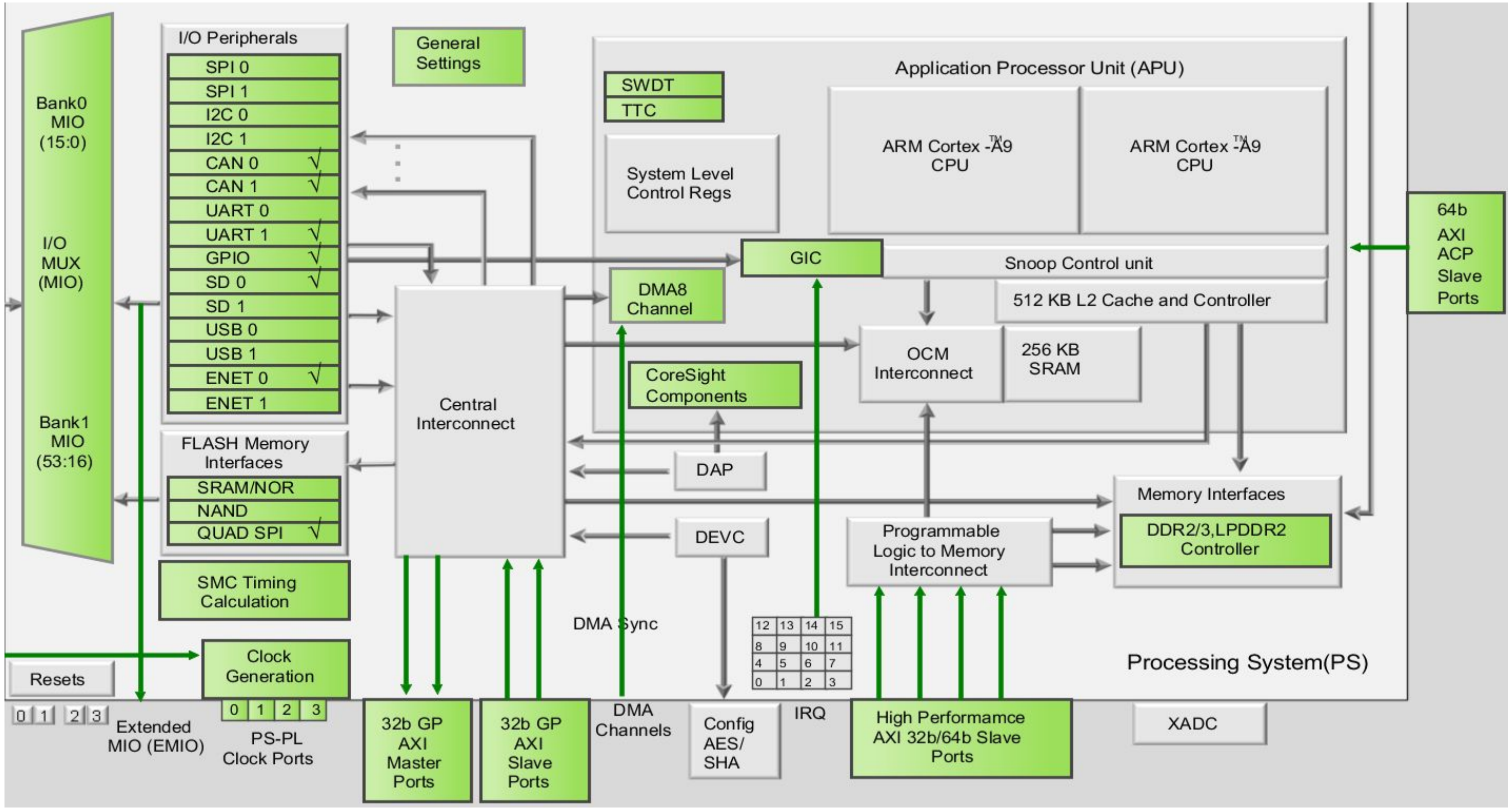
임세진
sjj1507@gmail.com

목차

FPGA

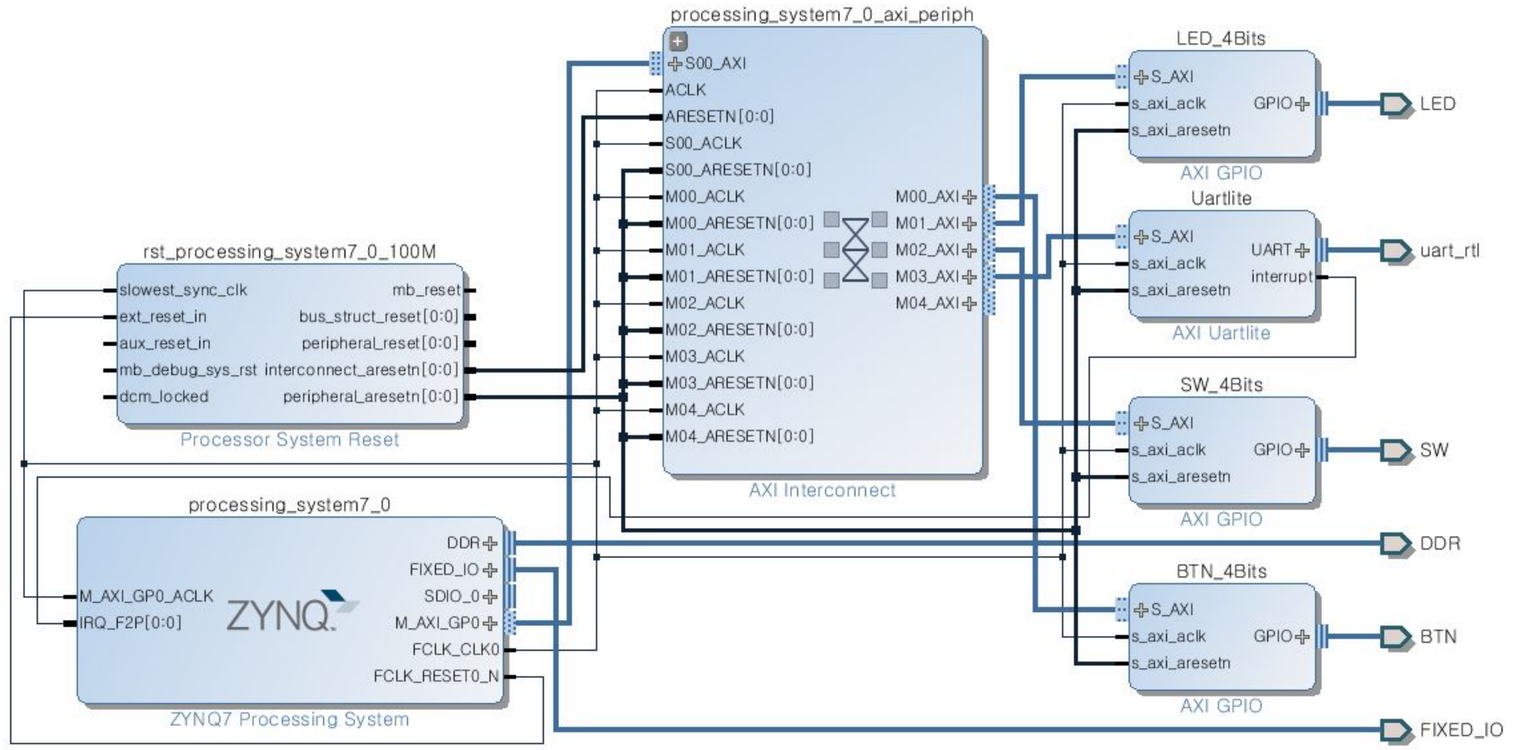
1. FPGA(field programmable gate array)는 설계 가능 논리 소자와 프로그래밍가능 내부선이 포함된 반도체 소자
2. zynq는 ARM-v7 와
3. PS와 PL로 구분
    a. PS(Processing System)은 Cortex-A9의 SOC를 사용
    b. PL(Programmable Logic)은 프로그램이 가능한 논리 제어기
4. vivado downloads
    a. http://www.xilinx.com/ -> support -> downloads & licensing -> vivado
5. peta linux
    a. 4-a에서 Embedded Development에서 2015.4 메뉴

2. FPGA PS와 PL



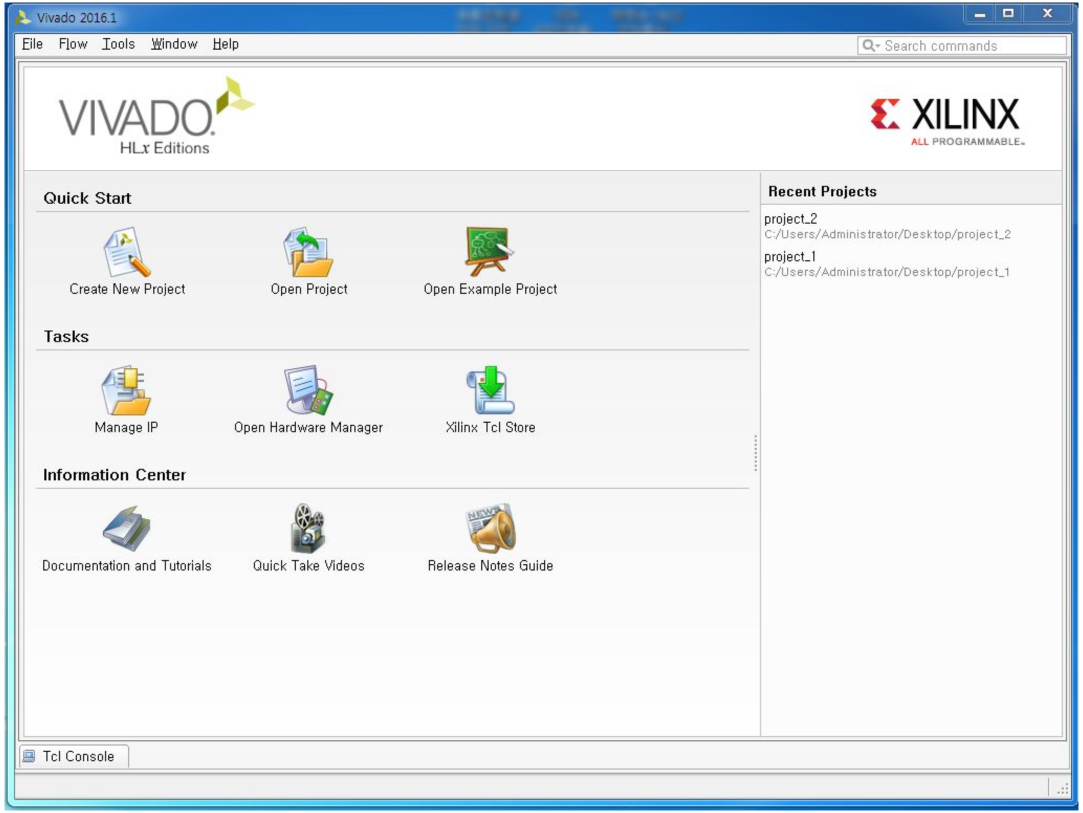
I/O Peripherals
SPI 0
SPI 1
I2C 0
I2C 1
CAN 0 √
CAN 1 √
UART 0
UART 1 √
GPIO √
SD 0 √
SD 1
USB 0
USB 1
ENET 0 √
ENET 1
General Settings
Bank0 MIO (15:0)
I/O MUX (MIO)
Bank1 MIO (53:16)
FLASH Memory Interfaces
SRAM/NOR
NAND
QUAD SPI √
SMC Timing Calculation
Clock Generation
Resets
0 1 2 3
Extended MIO (EMIO)
0 1 2 3
PS-PL Clock Ports
Central Interconnect
32b GP AXI Master Ports
32b GP AXI Slave Ports
DMA Sync
DMA Channels
Config AES/ SHA
IRQ
12 13 14 15
8 9 10 11
4 5 6 7
0 1 2 3
Application Processor Unit (APU)
SWDT
TTC
System Level Control Regs
ARM Cortex™-A9 CPU
ARM Cortex™-A9 CPU
GIC
Snoop Control unit
512 KB L2 Cache and Controller
DMA8 Channel
OCM Interconnect
256 KB SRAM
CoreSight Components
DAP
DEVC
Memory Interfaces
DDR2/3,LPDDR2 Controller
Programmable Logic to Memory Interconnect
High Performamce AXI 32b/64b Slave Ports
64b AXI ACP Slave Ports
Processing System(PS)
XADC

2. FPGA PS와 PL



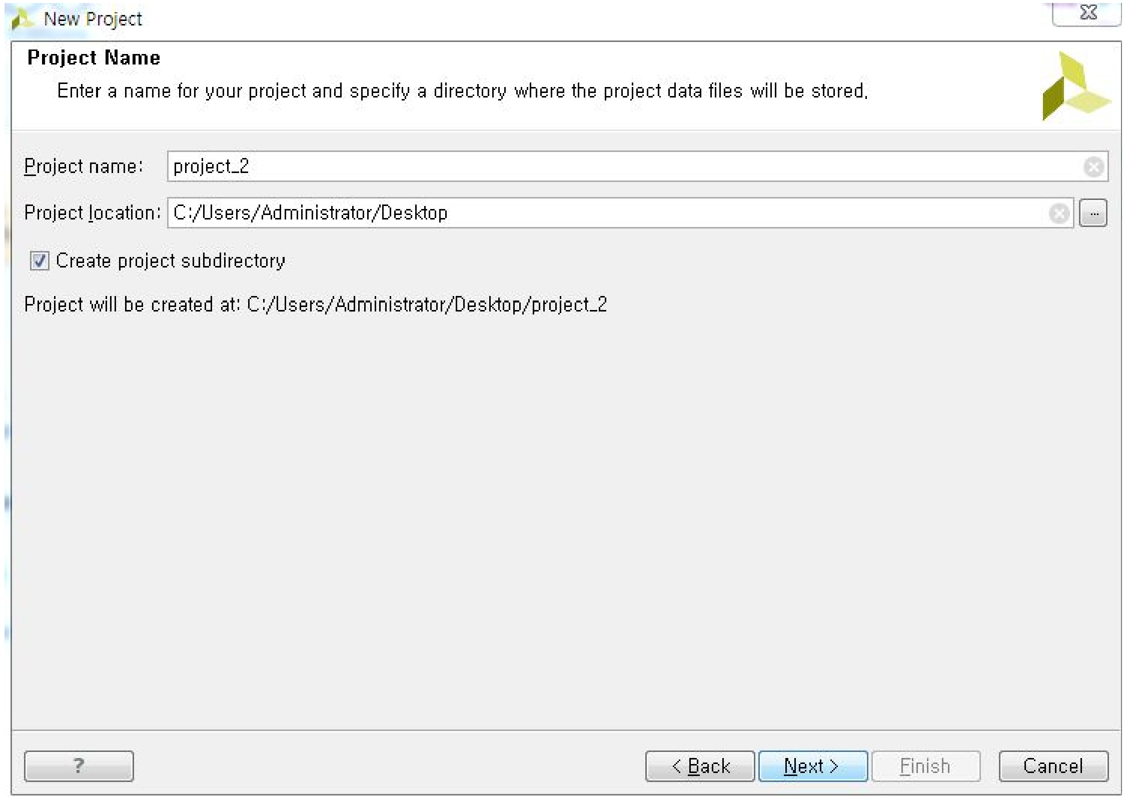
processing_system7_0_axi_periph
S00_AXI
ACLK
ARESETN[0:0]
S00_ACLK
S00_ARESETN[0:0]
M00_ACLK
M00_ARESETN[0:0]
M01_ACLK
M01_ARESETN[0:0]
M02_ACLK
M02_ARESETN[0:0]
M03_ACLK
M03_ARESETN[0:0]
M04_ACLK
M04_ARESETN[0:0]
M00_AXI
M01_AXI
M02_AXI
M03_AXI
M04_AXI
AXI Interconnect
rst_processing_system7_0_100M
slowest_sync_clk
ext_reset_in
aux_reset_in
mb_debug_sys_rst
dcm_locked
mb_reset
bus_struct_reset[0:0]
peripheral_reset[0:0]
interconnect_aresetn[0:0]
peripheral_aresetn[0:0]
Processor System Reset
processing_system7_0
M_AXI_GP0_ACLK
IRQ_F2P[0:0]
ZYNQ
DDR
FIXED_IO
SDIO_0
M_AXI_GP0
FCLK_CLK0
FCLK_RESET0_N
ZYNQ7 Processing System
LED_4Bits
S_AXI
s_axi_aclk
s_axi_aresetn
GPIO
AXI GPIO
LED
Uartlite
S_AXI
s_axi_aclk
s_axi_aresetn
UART
interrupt
AXI Uartlite
uart_rtl
SW_4Bits
S_AXI
s_axi_aclk
s_axi_aresetn
GPIO
AXI GPIO
SW
DDR
BTN_4Bits
S_AXI
s_axi_aclk
s_axi_aresetn
GPIO
AXI GPIO
BTN
FIXED_IO

3. PL을 이용한 LED점등



1) Quick Start -> Create New Project

3. PL을 이용한 LED점등


New Project

Project Name

Enter a name for your project and specify a directory where the project data files will be stored.

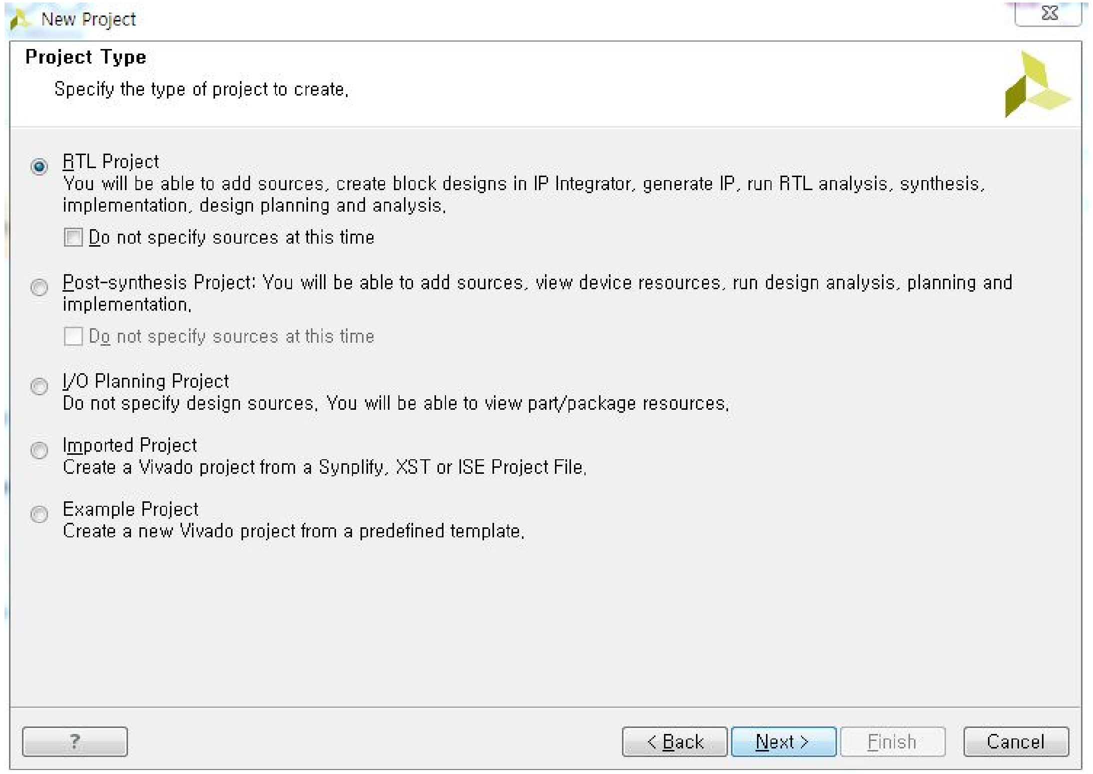
Project name: project_2

Project location: C:/Users/Administrator/Desktop

☑ Create project subdirectory

Project will be created at: C:/Users/Administrator/Desktop/project_2

3. PL을 이용한 LED점등



New Project

Project Type

Specify the type of project to create.

[✓ RTL Project]
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.

☐ Do not specify sources at this time

Post-synthesis Project: You will be able to add sources, view device resources, run design analysis, planning and implementation.

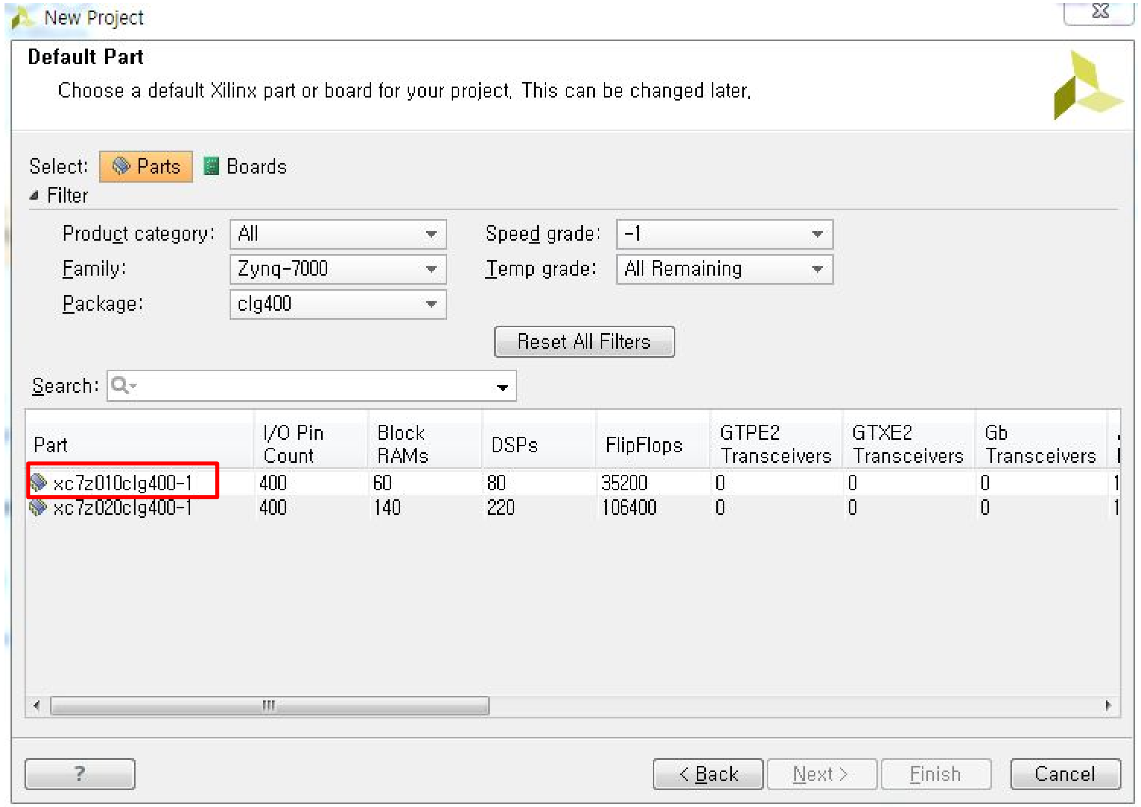
☐ Do not specify sources at this time

I/O Planning Project
Do not specify design sources. You will be able to view part/package resources.

Imported Project
Create a Vivado project from a Synplify, XST or ISE Project File.

Example Project
Create a new Vivado project from a predefined template.

3. PL을 이용한 LED점등



Default Part

Choose a default Xilinx part or board for your project. This can be changed later.

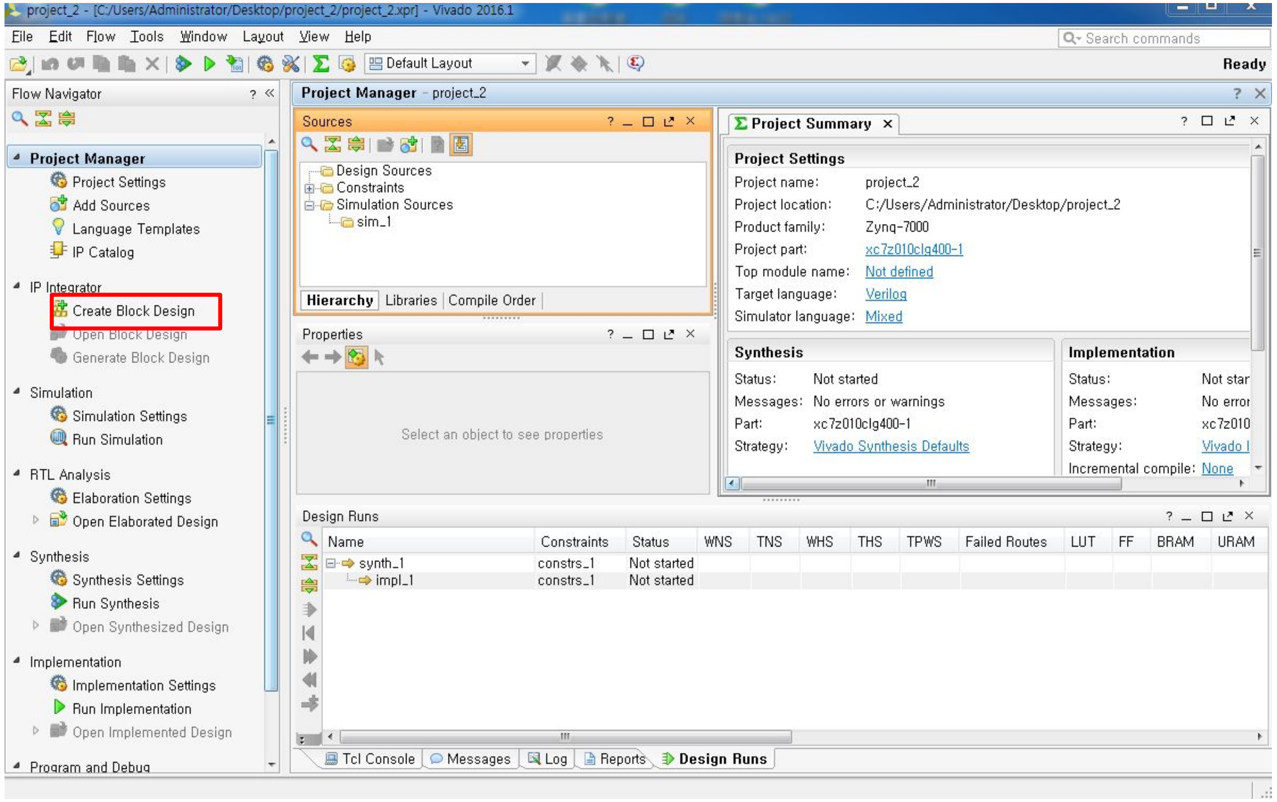
Product category: All
Family: Zynq-7000
Package: clg400
Speed grade: -1
Temp grade: All Remaining

Part	I/O Pin Count	Block RAMs	DSPs	FlipFlops	GTPE2 Transceivers	GTXE2 Transceivers	Gb Transceivers
xc7z010clg400-1	400	60	80	35200	0	0	0
xc7z020clg400-1	400	140	220	106400	0	0	0

zybo 보드타입
- xc7z010clg400-1

zedboard 보드타입
- xc7z020clg400-1

3. PL을 이용한 LED점등



Project Settings

Project name:	project_2
Project location:	C:/Users/Administrator/Desktop/project_2
Product family:	Zynq-7000
Project part:	xc7z010clg400-1
Top module name:	Not defined
Target language:	Verilog
Simulator language:	Mixed

Synthesis

Status:	Not started
Messages:	No errors or warnings
Part:	xc7z010clg400-1
Strategy:	Vivado Synthesis Defaults

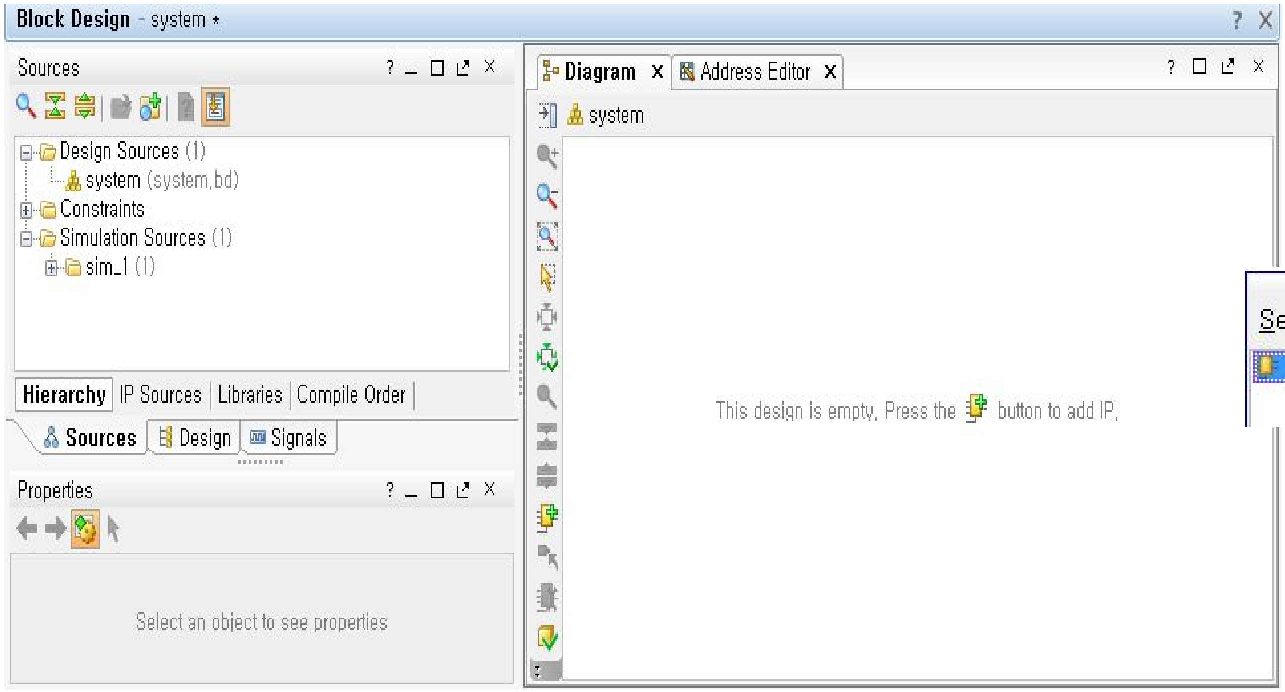
Implementation

Status:	Not star
Messages:	No error
Part:	xc7z010
Strategy:	Vivado I
Incremental compile:	None

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Failed Routes	LUT	FF	BRAM	URAM
synth_1	constrs_1	Not started										
impl_1	constrs_1	Not started										

1) Create Block Design
2) popup window
3) Design name
4) OK! click

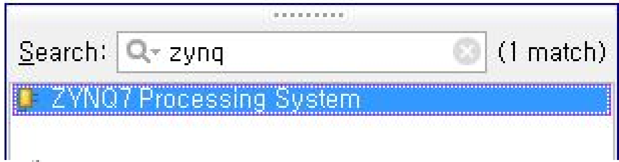
3. PL을 이용한 LED점등



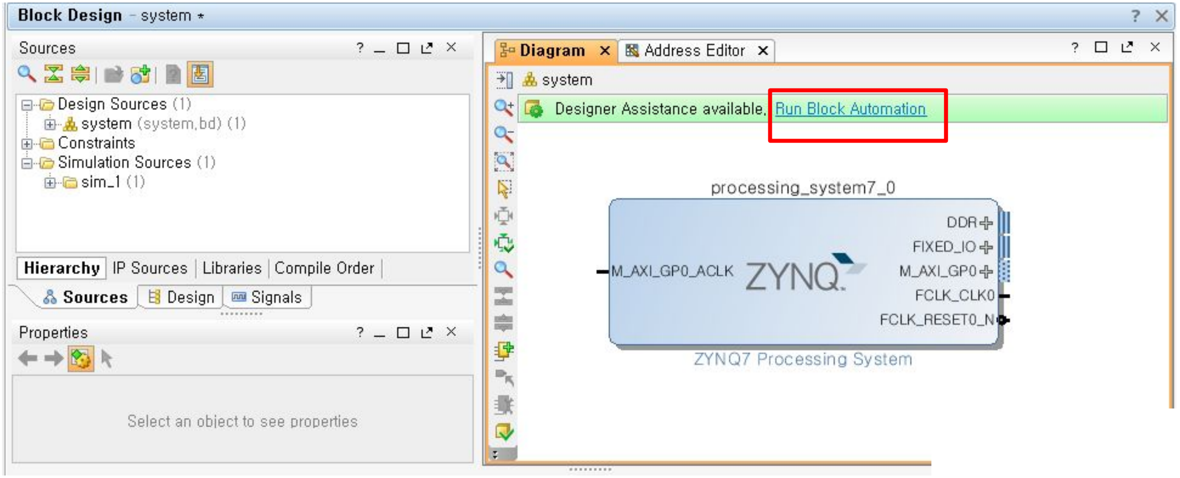
IP추가 방법

1) ctrl+i
2) Diagram -> right click

search에 zynq 검색



3. PL을 이용한 LED점등


Block Design - system *

Sources

- Design Sources (1)
  - system (system.bd) (1)
- Constraints
- Simulation Sources (1)
  - sim_1 (1)

Hierarchy | IP Sources | Libraries | Compile Order

Sources | Design | Signals

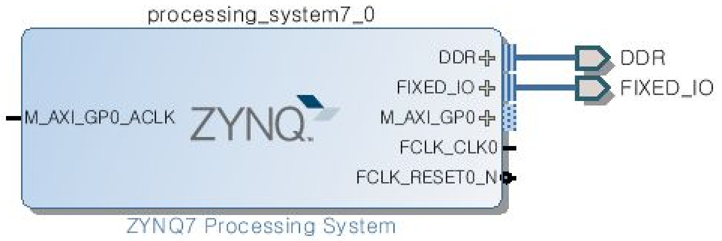
Properties

Select an object to see properties

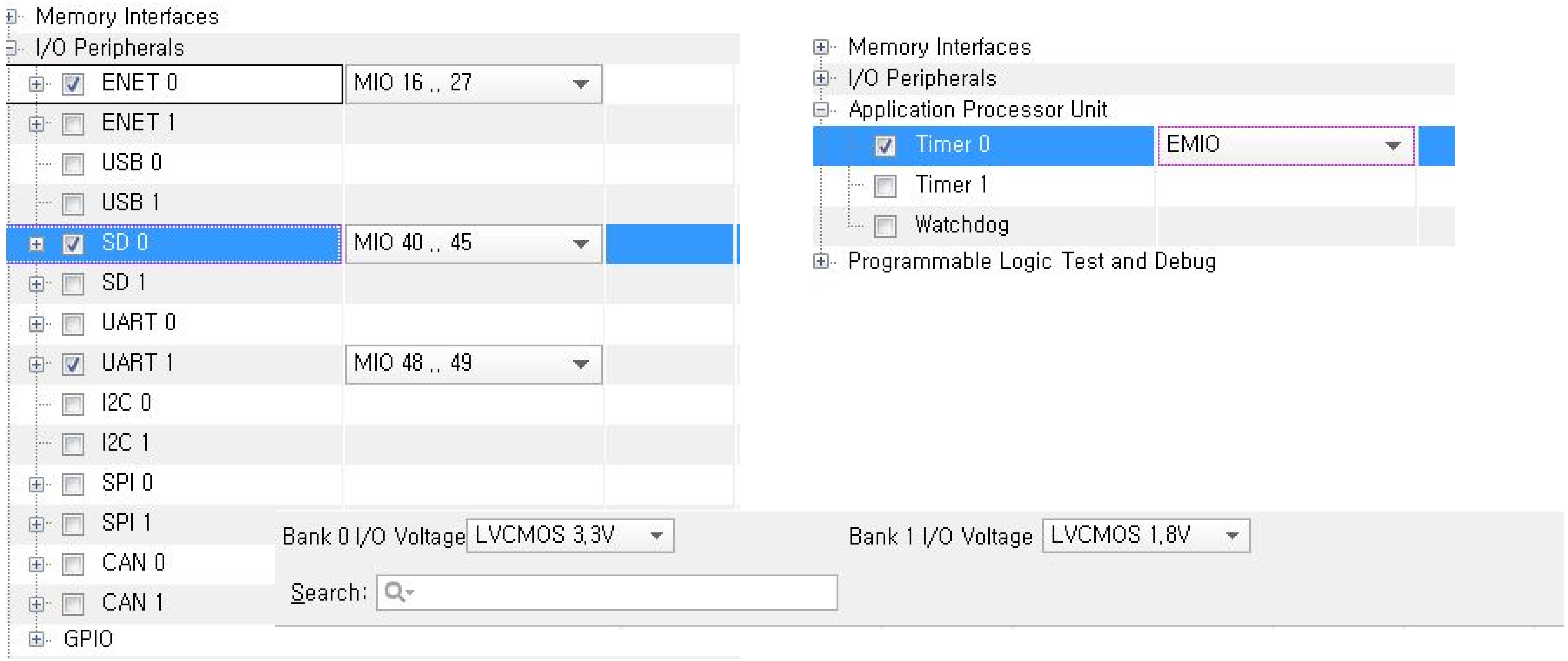
Diagram | Address Editor

system

Designer Assistance available. Run Block Automation



2. FPGA PS와 PL



Memory Interfaces
I/O Peripherals

✓	ENET 0	MIO 16 .. 27
☐	ENET 1	
☐	USB 0	
☐	USB 1	
✓	SD 0	MIO 40 .. 45
☐	SD 1	
☐	UART 0	
✓	UART 1	MIO 48 .. 49
☐	I2C 0	
☐	I2C 1	
☐	SPI 0	
☐	SPI 1	
☐	CAN 0	
☐	CAN 1	

GPIO

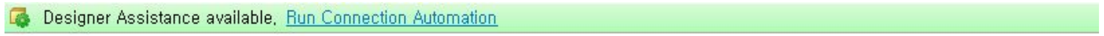
Memory Interfaces
I/O Peripherals
Application Processor Unit

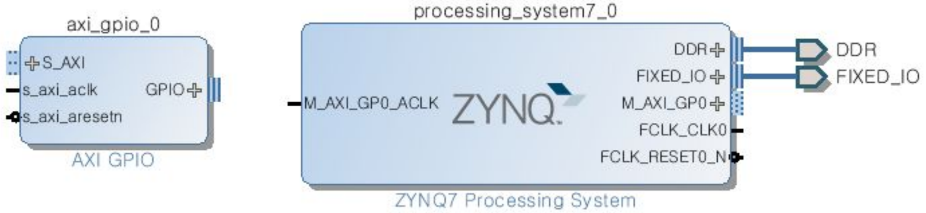
✓	Timer 0	EMIO
☐	Timer 1	
☐	Watchdog	

Programmable Logic Test and Debug

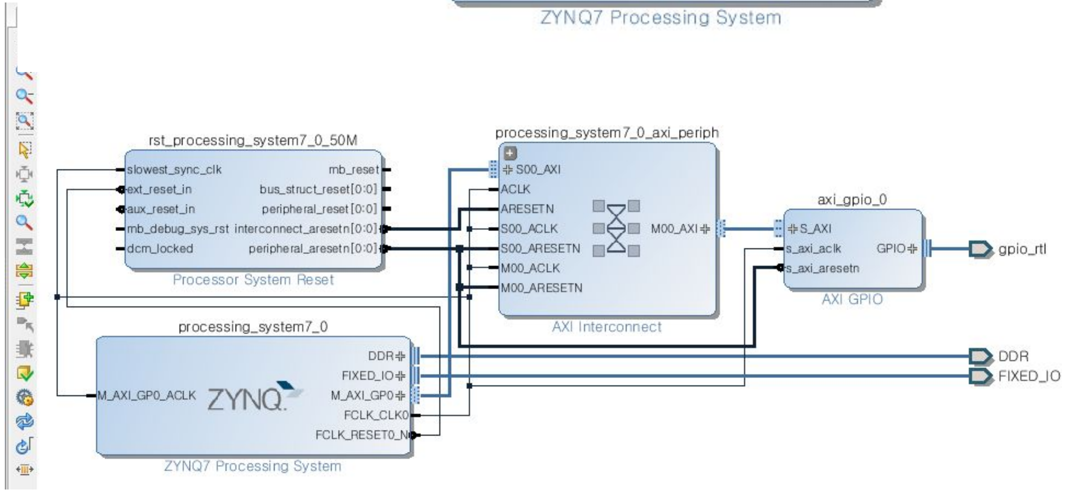
Bank 0 I/O Voltage LVCMOS 3.3V
Bank 1 I/O Voltage LVCMOS 1.8V
Search:

3. PL을 이용한 LED점등

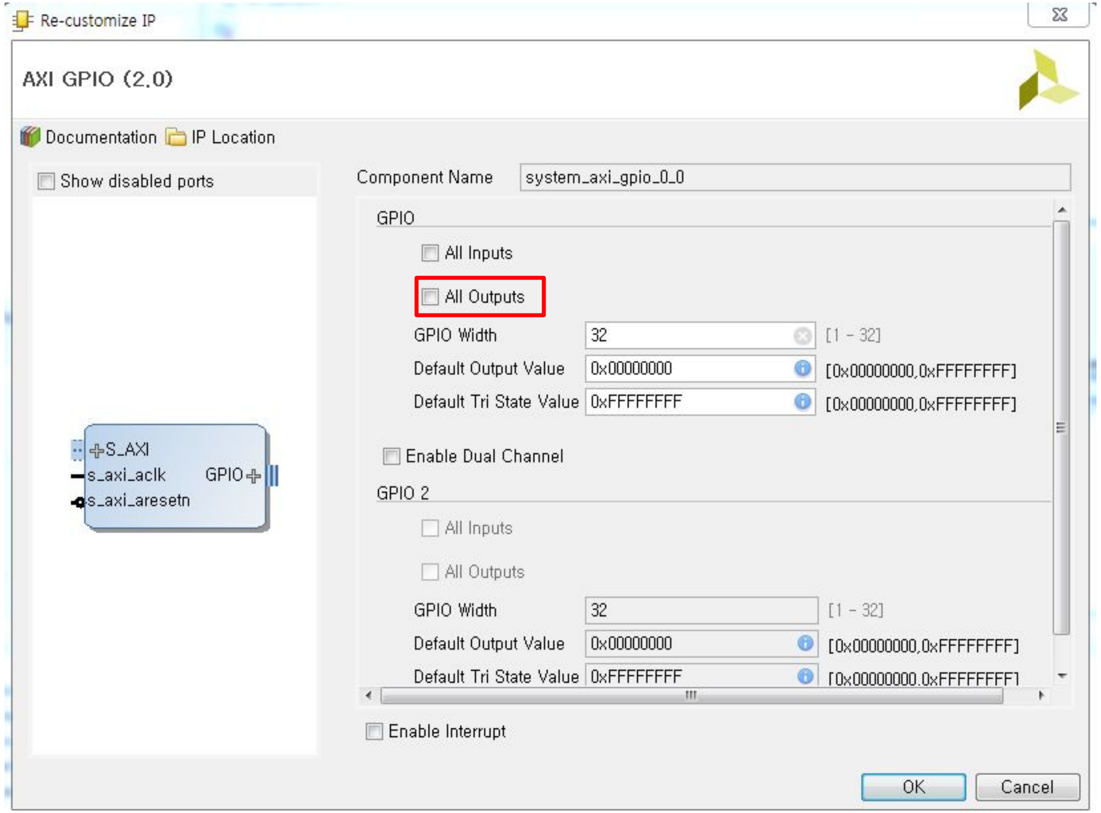

Designer Assistance available. Run Connection Automation



1) gpio IP를 추가후
2) run connection automation



3. PL을 이용한 LED점등



Re-customize IP

AXI GPIO (2.0)

Documentation | IP Location

- [ ] Show disabled ports

Component Name: system_axi_gpio_0_0

GPIO

- [ ] All Inputs
- [ ] All Outputs

GPIO Width	32	[1 - 32]
Default Output Value	0x00000000	[0x00000000,0xFFFFFFFF]
Default Tri State Value	0xFFFFFFFF	[0x00000000,0xFFFFFFFF]

- [ ] Enable Dual Channel

GPIO 2

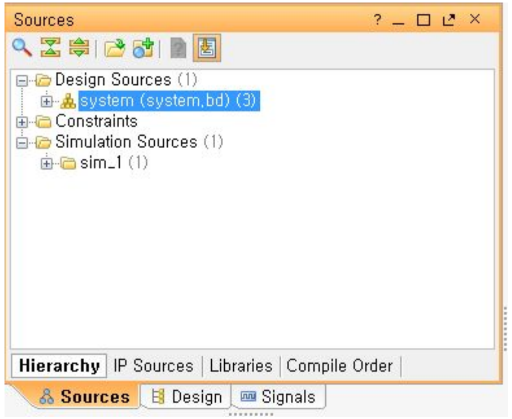
- [ ] All Inputs
- [ ] All Outputs

GPIO Width	32	[1 - 32]
Default Output Value	0x00000000	[0x00000000,0xFFFFFFFF]
Default Tri State Value	0xFFFFFFFF	[0x00000000,0xFFFFFFFF]

- [ ] Enable Interrupt

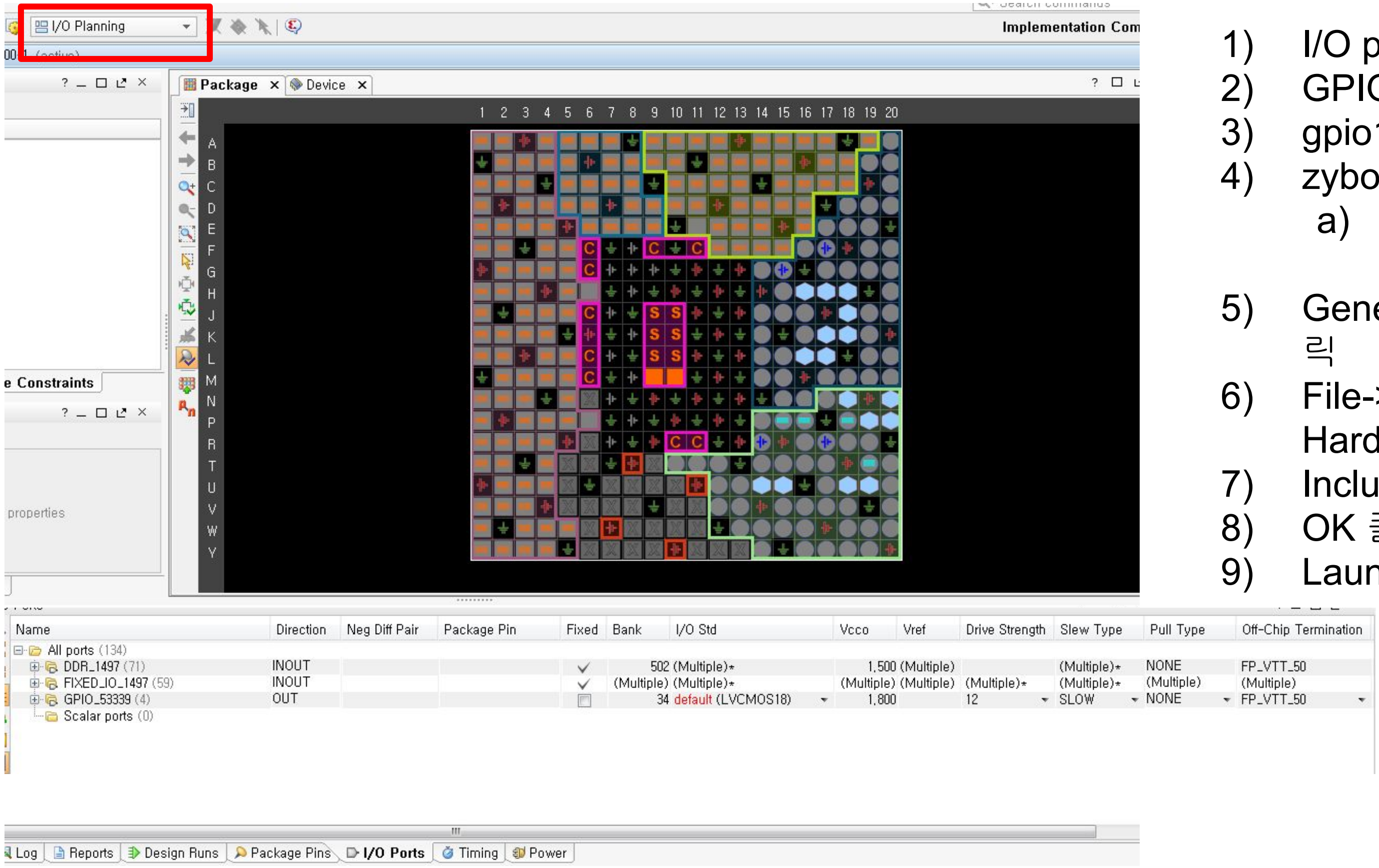
1) All Outputs 체크
2) GPIO Width 4비트 설정
3) Default Output Value
   a) 0x0000000F

3. PL을 이용한 LED점등



1) Validate Design 클릭
2) Design Sources아래
3) system우클릭
4) Generate Output Products 클릭
   a) Generate 클릭
5) 2번 다시
6) Create HDL Wrapper
7) Run Synthesis
8) Run Implementation
9) Open Implementation

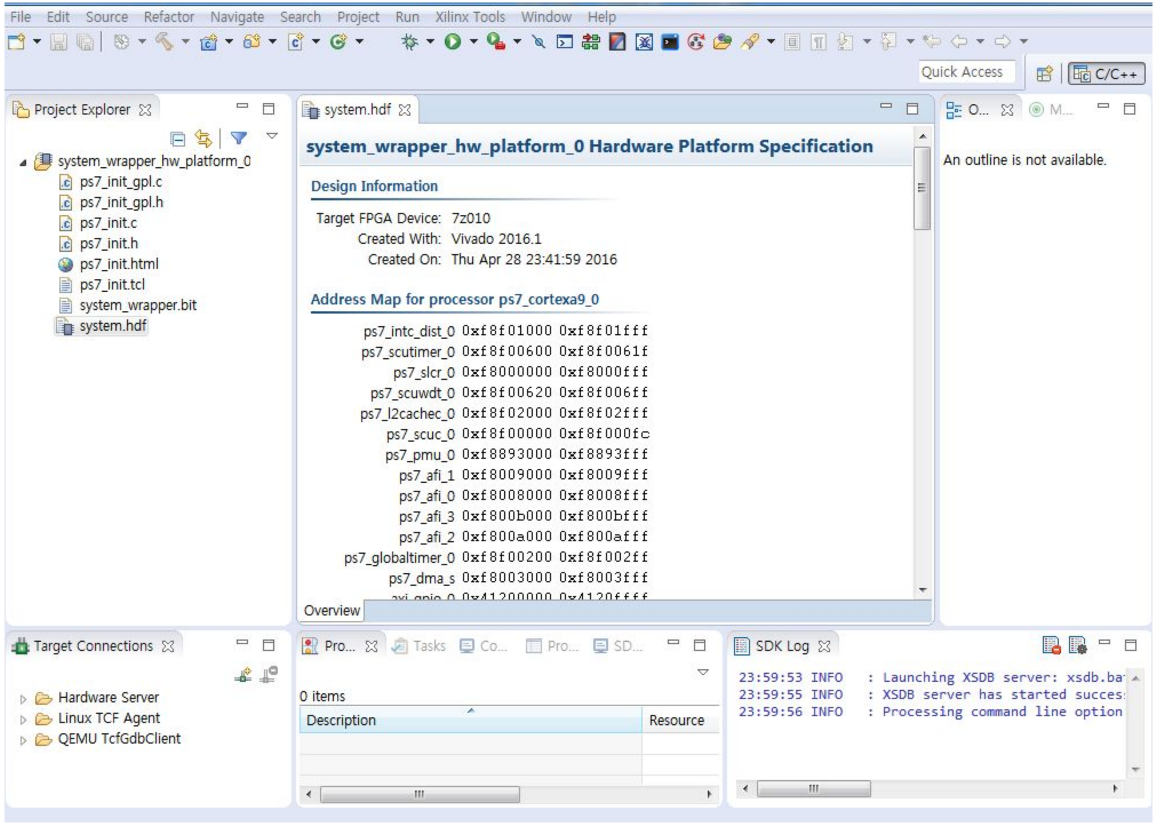
3. PL을 이용한 LED점등



Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination
All ports (134)												
DDR_1497 (71)	INOUT			✓	502	(Multiple)*	1,500	(Multiple)		(Multiple)*	NONE	FP_VTT_50
FIXED_IO_1497 (59)	INOUT			✓	(Multiple)	(Multiple)*	(Multiple)	(Multiple)	(Multiple)*	(Multiple)*	(Multiple)	(Multiple)
GPIO_53339 (4)	OUT			☐	34	default (LVCMOS18)	1,800		12	SLOW	NONE	FP_VTT_50
Scalar ports (0)												

1) I/O planning 설정
2) GPIO 의 PIN을 설정
3) gpio1~4번까지 핀설정
4) zybo LED 핀번호
   a) D18, G14, M15, M14
5) Generate Bitstream 클릭
6) File->Export->Export Hardware
7) Include Bitstream 체크
8) OK 클릭
9) Launch SDK 클릭

4. SDK 펌웨어코딩



system_wrapper_hw_platform_0
- ps7_init_gpl.c
- ps7_init_gpl.h
- ps7_init.c
- ps7_init.h
- ps7_init.html
- ps7_init.tcl
- system_wrapper.bit
- system.hdf

system_wrapper_hw_platform_0 Hardware Platform Specification

Design Information

Target FPGA Device: 7z010
Created With: Vivado 2016.1
Created On: Thu Apr 28 23:41:59 2016

Address Map for processor ps7_cortexa9_0

ps7_intc_dist_0	0xf8f01000	0xf8f01fff
ps7_scutimer_0	0xf8f00600	0xf8f0061f
ps7_slcr_0	0xf8000000	0xf8000fff
ps7_scuwdt_0	0xf8f00620	0xf8f006ff
ps7_l2cachec_0	0xf8f02000	0xf8f02fff
ps7_scuc_0	0xf8f00000	0xf8f000fc
ps7_pmu_0	0xf8893000	0xf8893fff
ps7_afi_1	0xf8009000	0xf8009fff
ps7_afi_0	0xf8008000	0xf8008fff
ps7_afi_3	0xf800b000	0xf800bfff
ps7_afi_2	0xf800a000	0xf800afff
ps7_globaltimer_0	0xf8f00200	0xf8f002ff
ps7_dma_s	0xf8003000	0xf8003fff
axi_gpio_0	0x41200000	0x4120ffff

An outline is not available.

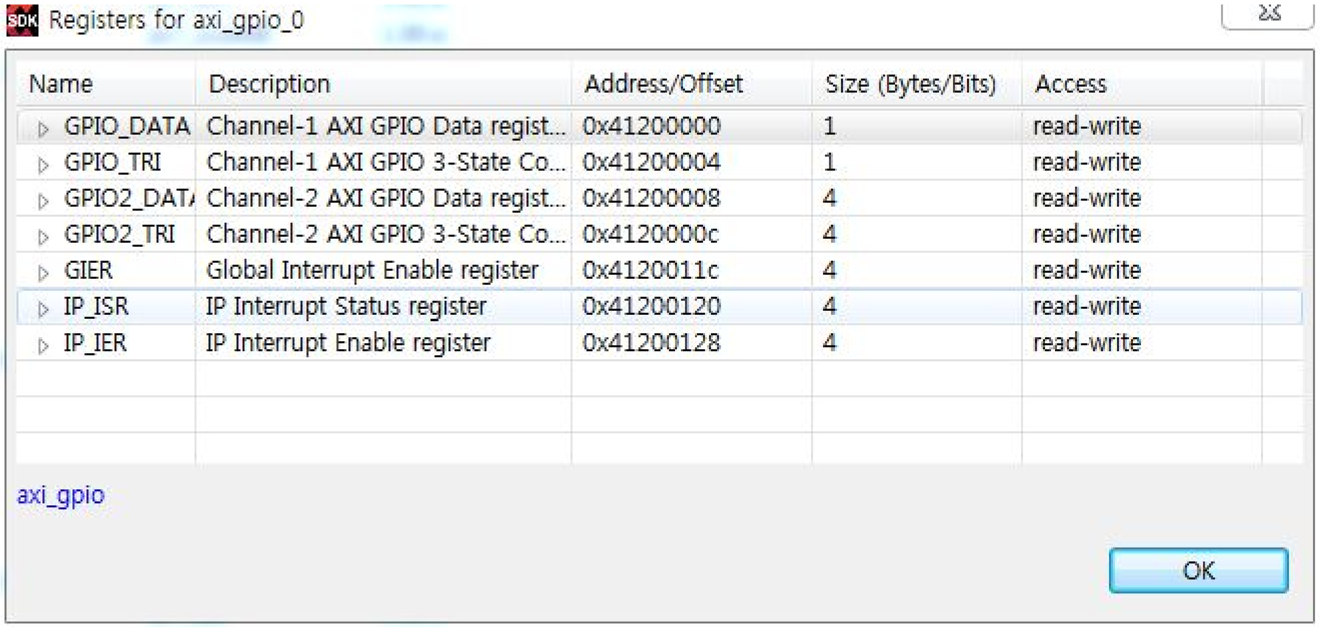
- Hardware Server
- Linux TCF Agent
- QEMU TcfGdbClient

0 items

Description	Resource

```
23:59:53 INFO  : Launching XSDB server: xsdb.ba
23:59:55 INFO  : XSDB server has started succes
23:59:56 INFO  : Processing command line option
```

4. SDK 펌웨어코딩

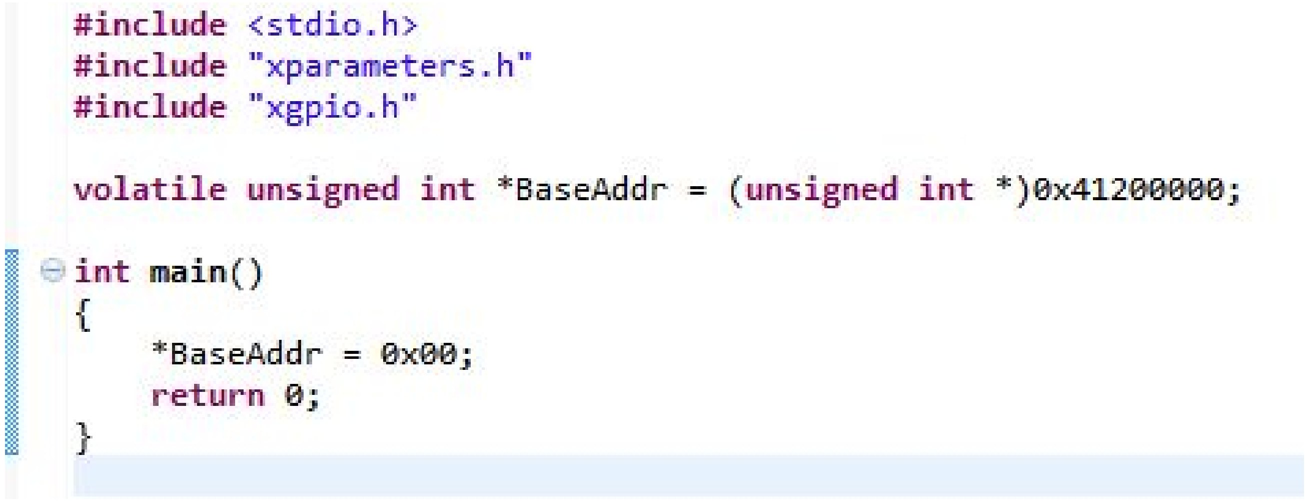

Registers for axi_gpio_0

Name	Description	Address/Offset	Size (Bytes/Bits)	Access
GPIO_DATA	Channel-1 AXI GPIO Data regist...	0x41200000	1	read-write
GPIO_TRI	Channel-1 AXI GPIO 3-State Co...	0x41200004	1	read-write
GPIO2_DAT/	Channel-2 AXI GPIO Data regist...	0x41200008	4	read-write
GPIO2_TRI	Channel-2 AXI GPIO 3-State Co...	0x4120000c	4	read-write
GIER	Global Interrupt Enable register	0x4120011c	4	read-write
IP_ISR	IP Interrupt Status register	0x41200120	4	read-write
IP_IER	IP Interrupt Enable register	0x41200128	4	read-write

axi_gpio

GPIO레지스터

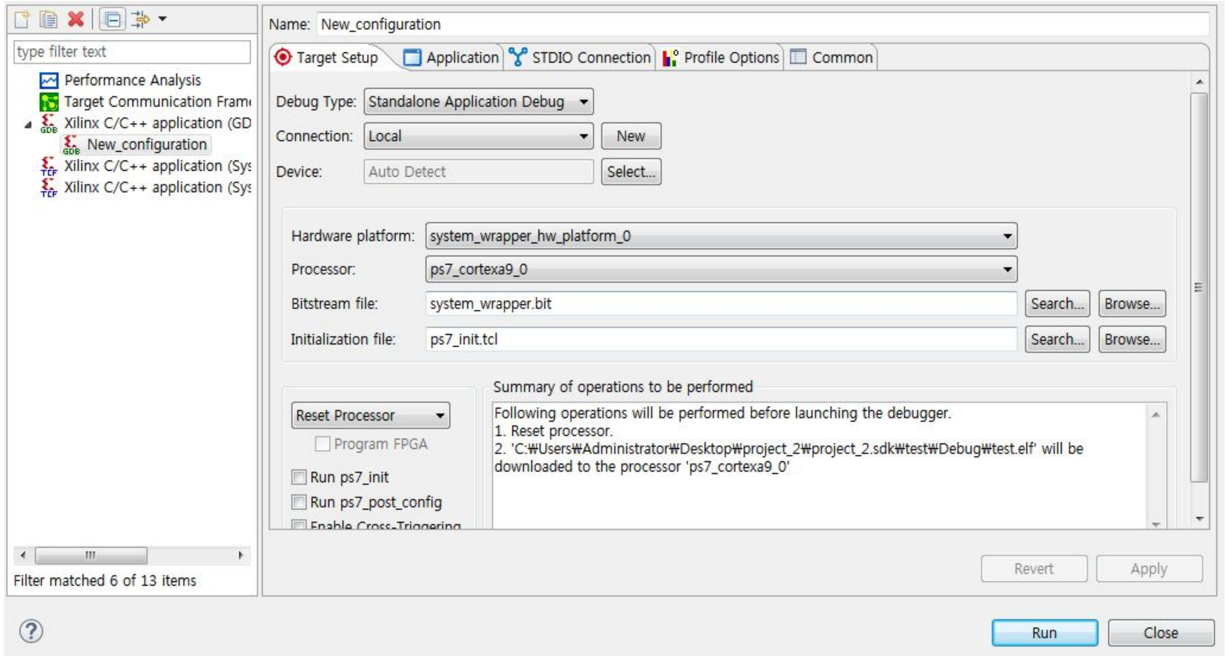
4. SDK 펌웨어코딩



```
#include <stdio.h>
#include "xparameters.h"
#include "xgpio.h"

volatile unsigned int *BaseAddr = (unsigned int *)0x41200000;

int main()
{
    *BaseAddr = 0x00;
    return 0;
}
```

LED 소등 코드

4. SDK 펌웨어코딩



Name: New_configuration

Target Setup | Application | STDIO Connection | Profile Options | Common

type filter text

- Performance Analysis
- Target Communication Frame
- Xilinx C/C++ application (GD
  - New_configuration
- Xilinx C/C++ application (Sys
- Xilinx C/C++ application (Sys

Filter matched 6 of 13 items

Debug Type: Standalone Application Debug

Connection: Local [New]

Device: Auto Detect [Select...]

Hardware platform: system_wrapper_hw_platform_0

Processor: ps7_cortexa9_0

Bitstream file: system_wrapper.bit [Search...] [Browse...]

Initialization file: ps7_init.tcl [Search...] [Browse...]

Reset Processor

- [ ] Program FPGA
- [ ] Run ps7_init
- [ ] Run ps7_post_config
- [ ] Enable Cross-Triggering

Summary of operations to be performed

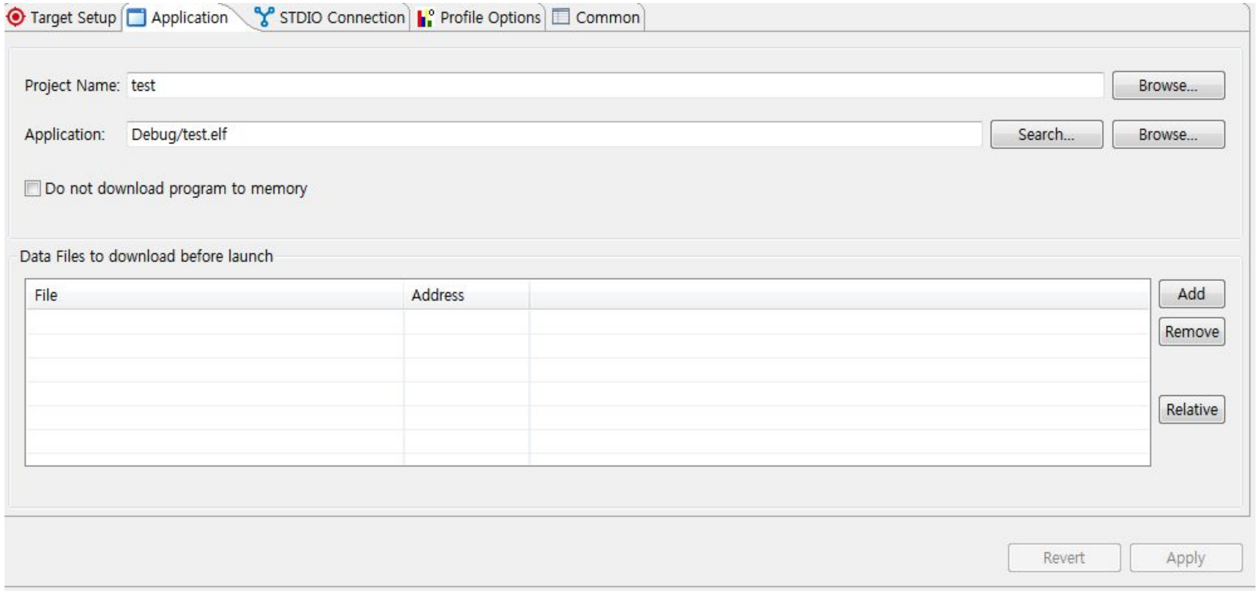
Following operations will be performed before launching the debugger.
1. Reset processor.
2. 'C:₩Users₩Administrator₩Desktop₩project_2₩project_2.sdk₩test₩Debug₩test.elf' will be downloaded to the processor 'ps7_cortexa9_0'

Revert | Apply

Run | Close

1) xilinx c/c++ application (GDB) 우클릭
2) NEW
3) 왼쪽 화면처럼 설정

4. SDK 펌웨어코딩



Project Name: test

Application: Debug/test.elf

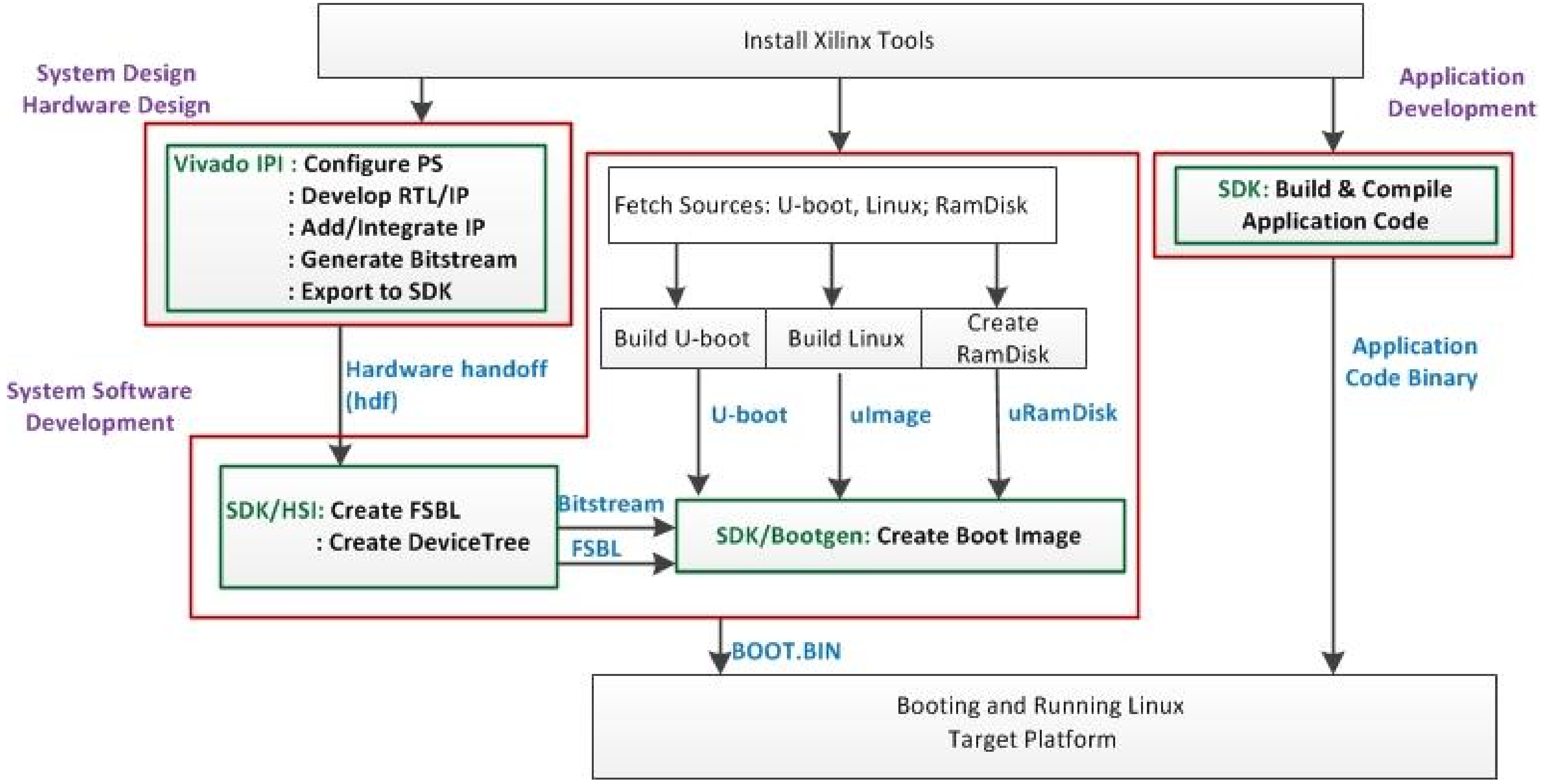
☐ Do not download program to memory

Data Files to download before launch

File	Address	

1) xilinx c/c++ application (GDB) 우클릭
2) NEW
3) 왼쪽 환면처럼 설정

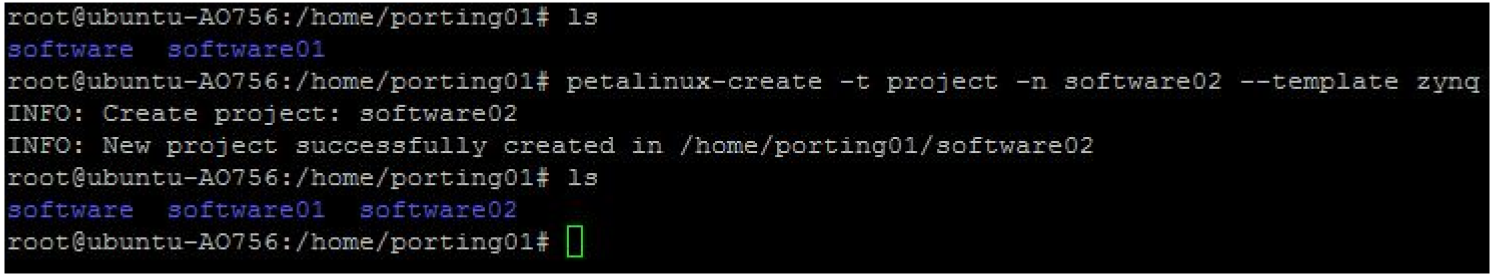
5. 리눅스 디바이스 드라이버



5. 리눅스 디바이스 드라이버

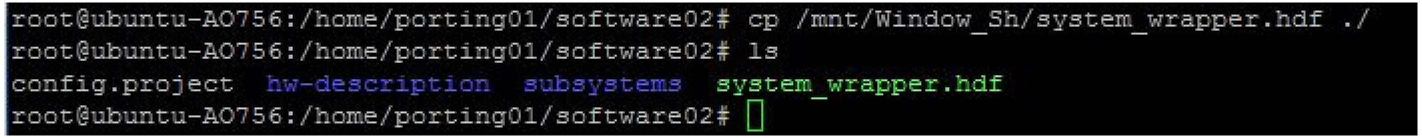
1) vivado 프로젝트 -> sdk폴더
2) hdf를 리눅스에 전송
3) petalinux설치후 해당폴더에 setting.sh를 source를 이용 실행
4) 리눅스커널 컴파일할 임의의디렉토리 생성(해당 디렉토리에서 작업 진행)

5. 리눅스 디바이스 드라이버



```
root@ubuntu-AO756:/home/porting01# ls
software  software01
root@ubuntu-AO756:/home/porting01# petalinux-create -t project -n software02 --template zynq
INFO: Create project: software02
INFO: New project successfully created in /home/porting01/software02
root@ubuntu-AO756:/home/porting01# ls
software  software01  software02
root@ubuntu-AO756:/home/porting01# 
```

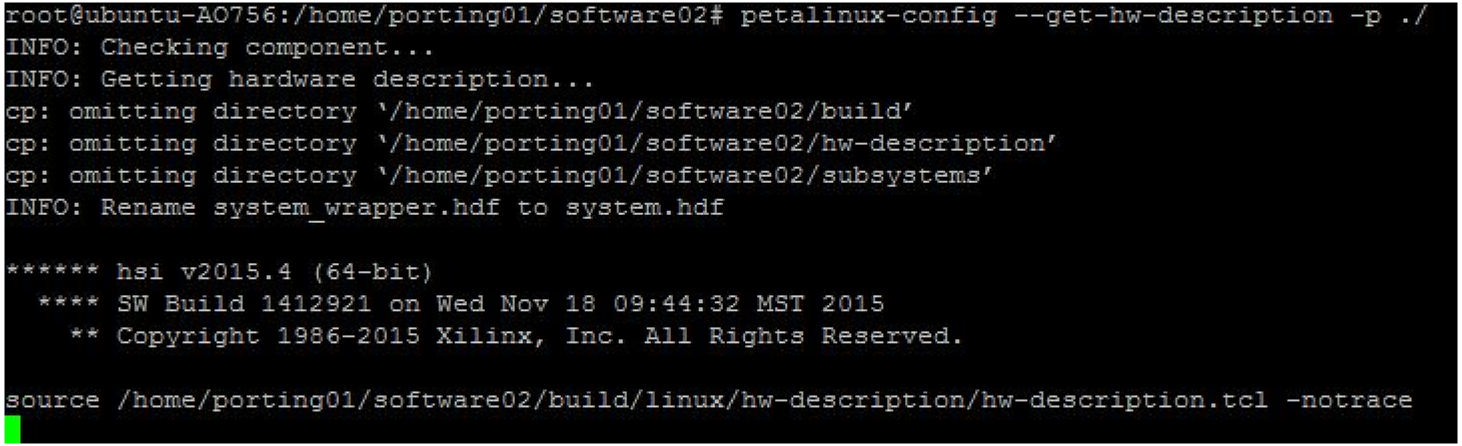
petalinux-create -t project -n software02 -–template zynq



```
root@ubuntu-AO756:/home/porting01/software02# cp /mnt/Window_Sh/system_wrapper.hdf ./
root@ubuntu-AO756:/home/porting01/software02# ls
config.project  hw-description  subsystems  system_wrapper.hdf
root@ubuntu-AO756:/home/porting01/software02# 
```

hdf 파일을 리눅스로 복사함

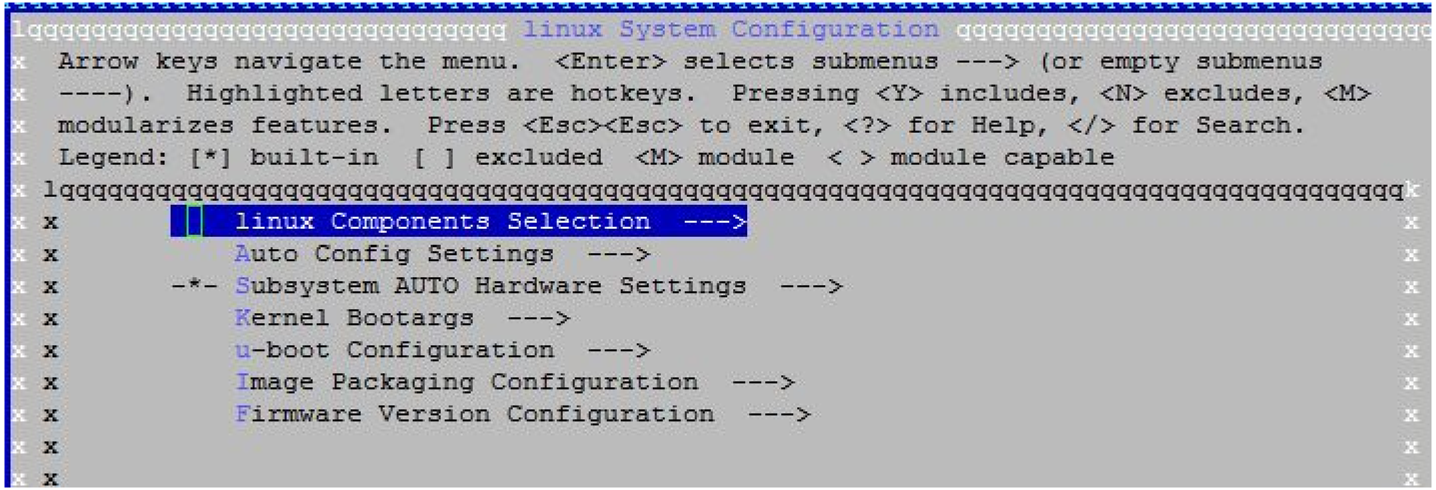
5. 리눅스 디바이스 드라이버



```
root@ubuntu-AO756:/home/porting01/software02# petalinux-config --get-hw-description -p ./
INFO: Checking component...
INFO: Getting hardware description...
cp: omitting directory '/home/porting01/software02/build'
cp: omitting directory '/home/porting01/software02/hw-description'
cp: omitting directory '/home/porting01/software02/subsystems'
INFO: Rename system_wrapper.hdf to system.hdf

****** hsi v2015.4 (64-bit)
  **** SW Build 1412921 on Wed Nov 18 09:44:32 MST 2015
    ** Copyright 1986-2015 Xilinx, Inc. All Rights Reserved.

source /home/porting01/software02/build/linux/hw-description/hw-description.tcl -notrace
```

petalinux-config --get-hw-description -p ./

5. 리눅스 디바이스 드라이버

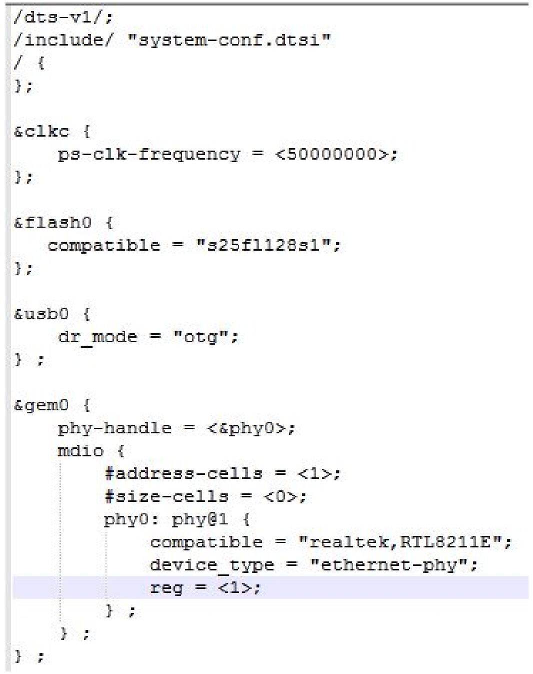


```
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq linux System Configuration qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
x  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus
x  ----).  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M>
x  modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.
x  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x x        [ ]  linux Components Selection  --->                                         x
x x             Auto Config Settings  --->                                               x
x x        -*-  Subsystem AUTO Hardware Settings  --->                                   x
x x             Kernel Bootargs  --->                                                    x
x x             u-boot Configuration  --->                                               x
x x             Image Packaging Configuration  --->                                      x
x x             Firmware Version Configuration  --->                                     x
x x                                                                                      x
x x                                                                                      x
```

5. 리눅스 디바이스 드라이버

프로젝트 최상위 디렉토리에서
subsystems->linux->configs->device_tree->system-top.dts 오픈

5. 리눅스 디바이스 드라이버

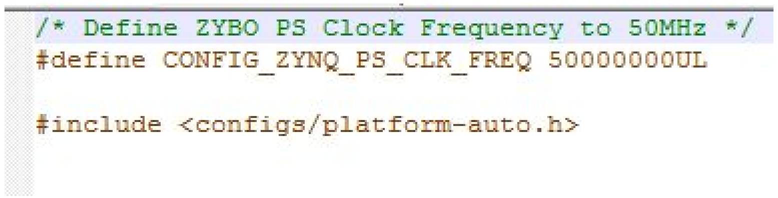


```
/dts-v1/;
/include/ "system-conf.dtsi"
/ {
};

&clkc {
    ps-clk-frequency = <50000000>;
};

&flash0 {
   compatible = "s25fl128s1";
};

&usb0 {
    dr_mode = "otg";
} ;

&gem0 {
    phy-handle = <&phy0>;
    mdio {
        #address-cells = <1>;
        #size-cells = <0>;
        phy0: phy@1 {
            compatible = "realtek,RTL8211E";
            device_type = "ethernet-phy";
            reg = <1>;
        } ;
    } ;
} ;
```

5. 리눅스 디바이스 드라이버

프로젝트 최상위 디렉토리에서
subsystems->linux->configs->u-boot->platform-top.h

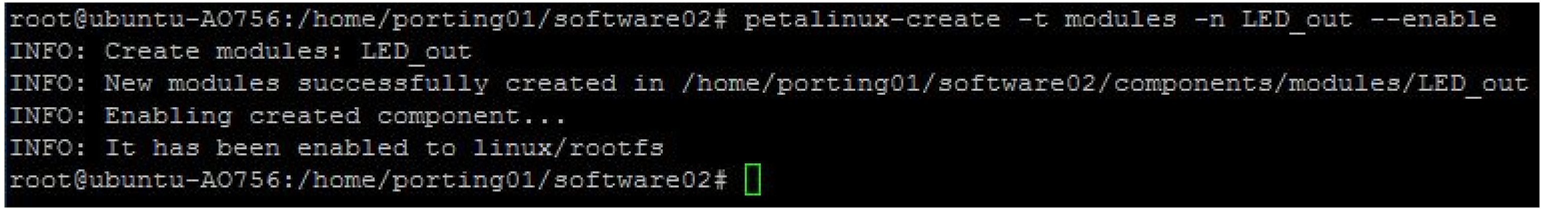
5. 리눅스 디바이스 드라이버



```
/* Define ZYBO PS Clock Frequency to 50MHz */
#define CONFIG_ZYNQ_PS_CLK_FREQ 50000000UL

#include <configs/platform-auto.h>
```

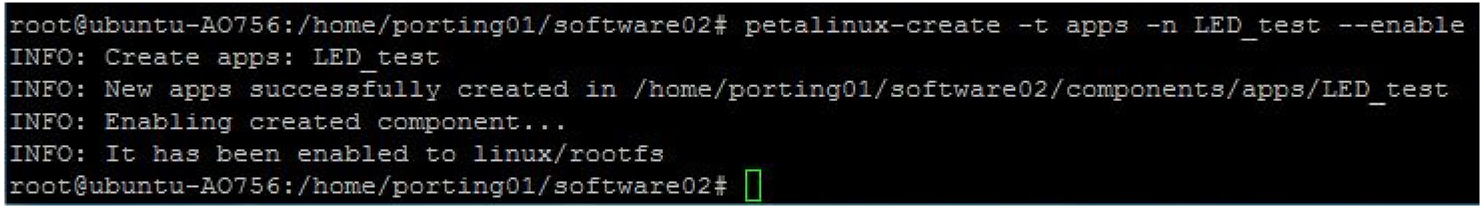
5. 리눅스 디바이스 드라이버

작업 디렉토리에서
petalinux-create -t modules -n 모듈이름 --enable



```
root@ubuntu-AO756:/home/porting01/software02# petalinux-create -t modules -n LED_out --enable
INFO: Create modules: LED_out
INFO: New modules successfully created in /home/porting01/software02/components/modules/LED_out
INFO: Enabling created component...
INFO: It has been enabled to linux/rootfs
root@ubuntu-AO756:/home/porting01/software02# 
```

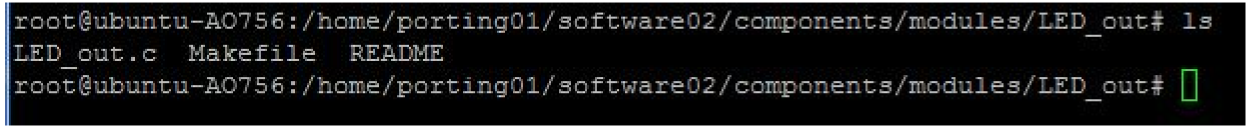
작업 디렉토리에서
petalinux-create -t apps -n 앱이름 --enable



```
root@ubuntu-AO756:/home/porting01/software02# petalinux-create -t apps -n LED_test --enable
INFO: Create apps: LED_test
INFO: New apps successfully created in /home/porting01/software02/components/apps/LED_test
INFO: Enabling created component...
INFO: It has been enabled to linux/rootfs
root@ubuntu-AO756:/home/porting01/software02# 
```

5. 리눅스 디바이스 드라이버

작업 디렉토리에서
components->modules->모듈이름



```
root@ubuntu-A0756:/home/porting01/software02/components/modules/LED_out# ls
LED_out.c   Makefile   README
root@ubuntu-A0756:/home/porting01/software02/components/modules/LED_out# 
```

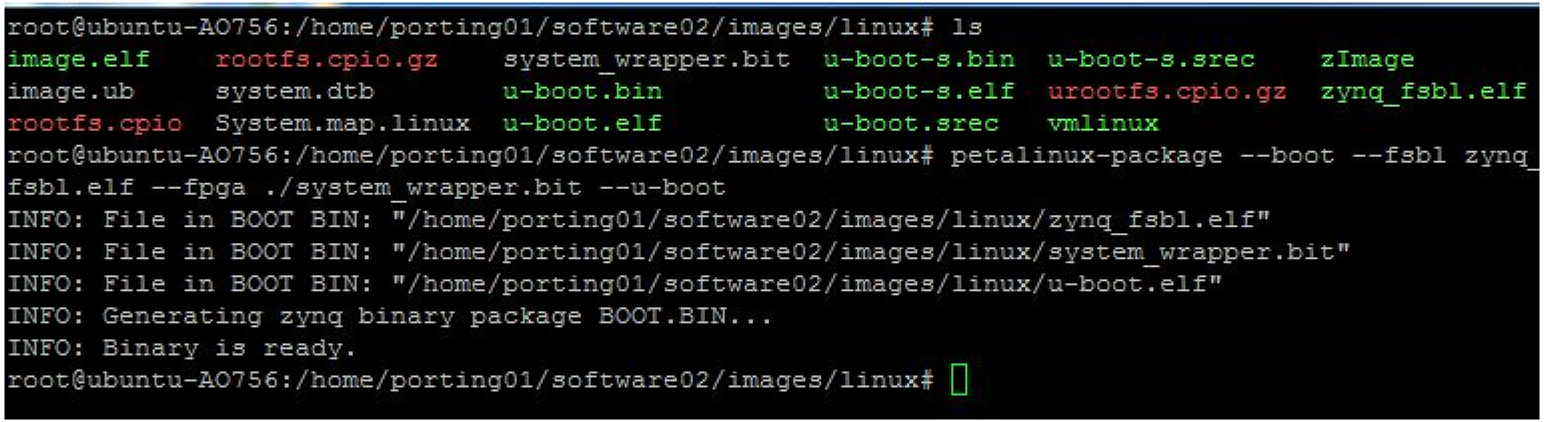
5. 리눅스 디바이스 드라이버

modules, app을 작성후
petalinux-config -c rootfs 에서 추가한 파일들을 컴파일 가능하게 등록 한다.

petalinux-build로 커널을 빌드한다.

5. 리눅스 디바이스 드라이버

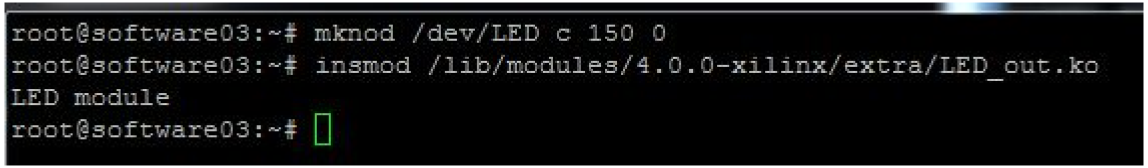
빌드후
images->linux 디렉토리에서
petalinux-package --boot --fsbl zynq_fsbl.elf --fpga ./system_wrapper.bit --u-boot



```
root@ubuntu-AO756:/home/porting01/software02/images/linux# ls
image.elf    rootfs.cpio.gz    system_wrapper.bit  u-boot-s.bin   u-boot-s.srec     zImage
image.ub     system.dtb        u-boot.bin          u-boot-s.elf   urootfs.cpio.gz   zynq_fsbl.elf
rootfs.cpio  System.map.linux  u-boot.elf          u-boot.srec    vmlinux
root@ubuntu-AO756:/home/porting01/software02/images/linux# petalinux-package --boot --fsbl zynq_
fsbl.elf --fpga ./system_wrapper.bit --u-boot
INFO: File in BOOT BIN: "/home/porting01/software02/images/linux/zynq_fsbl.elf"
INFO: File in BOOT BIN: "/home/porting01/software02/images/linux/system_wrapper.bit"
INFO: File in BOOT BIN: "/home/porting01/software02/images/linux/u-boot.elf"
INFO: Generating zynq binary package BOOT.BIN...
INFO: Binary is ready.
root@ubuntu-AO756:/home/porting01/software02/images/linux# 
```

5. 리눅스 디바이스 드라이버

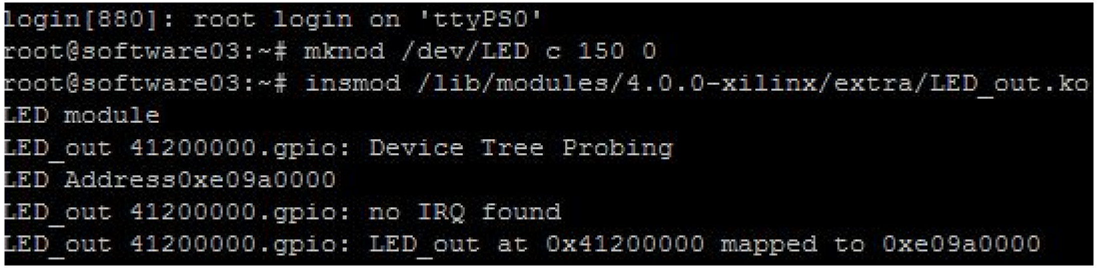
생성된 BOOT.BIN을 SD카드에 복사
커널부팅



```
root@software03:~# mknod /dev/LED c 150 0
root@software03:~# insmod /lib/modules/4.0.0-xilinx/extra/LED_out.ko
LED module
root@software03:~# 
```

디바이스 노드 등록
mknod /dev/LED c 150 0
insmod /lib/modules/4.0.0-xilinx/extra/LED_out.ko

5. 리눅스 디바이스 드라이버



```
login[880]: root login on 'ttyPS0'
root@software03:~# mknod /dev/LED c 150 0
root@software03:~# insmod /lib/modules/4.0.0-xilinx/extra/LED_out.ko
LED module
LED_out 41200000.gpio: Device Tree Probing
LED Address0xe09a0000
LED_out 41200000.gpio: no IRQ found
LED_out 41200000.gpio: LED_out at 0x41200000 mapped to 0xe09a0000
```