Petalinux Work Flow Lab

Introduction

ZynqMP PS 로만 구성된 Vivado Project HW 를 구성하고, Petalinux 명령어를 사용하여 sdcard 로 부팅가능한 Linux 이미지를 생성하고 Ultra96 보드를 가지고 테스트한다.

Objectives

- Export Vivado Project HW
- Create/Configure Petalinux Project
- Configure Device Tree
- Configure U-boot
- Configure Kernel
- Configure/Build Petalinux-user-image
- Create BOOT.BIN
- Prepare sdcard
- Test

Source Step 0

```
$ cd ~/work
$ git clone https://github.com/inipro/zynqmp_linux.git
$ cd zynqmp_linux
```

Export Vivado Project HW Step 1

1-1. Ultra96v1(hw1_v1.tcl) 또는 Ultra96v2(hw1_v2.tcl) Vivado Project 생성/Bitstream 생성/HW Export

1-1-1. 터미널 창에서 Vivado: Project 생성

```
$ vivado -nolog -nojournal -mode -batch -source hw1_v2.tcl
$ cd hw1
$ vivado hw1.xpr
```

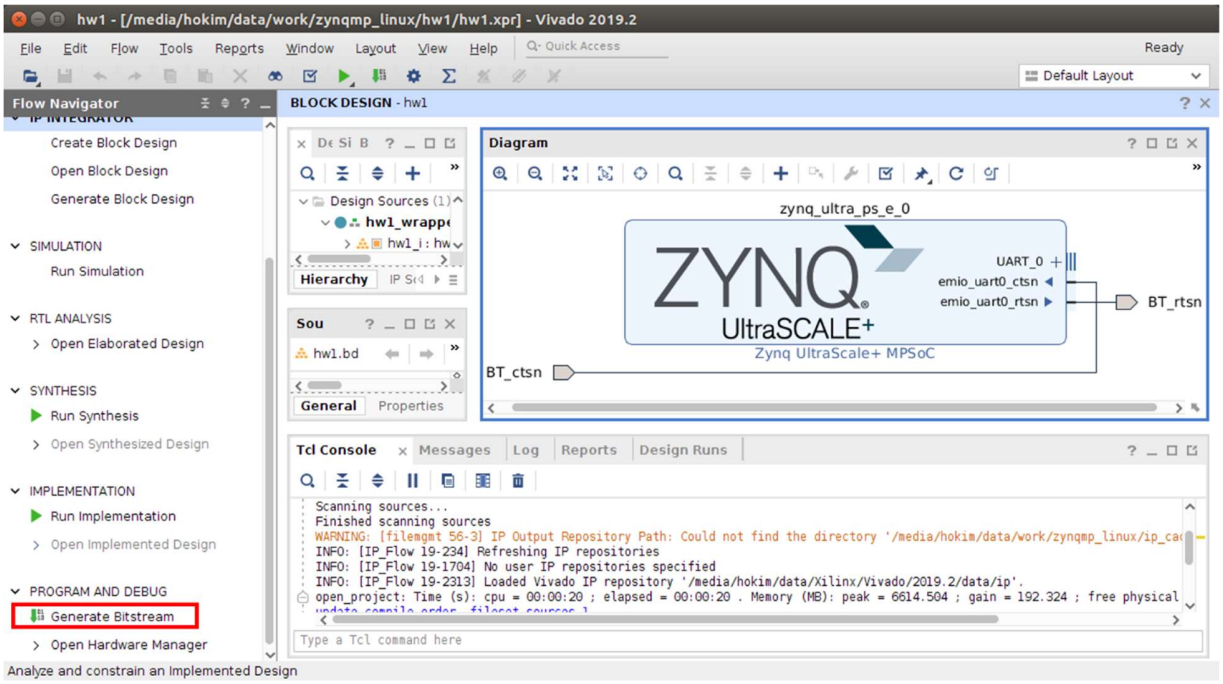


Figure 1. Vivado Project

1-1-2. **Flow Navigator** 의 **PROGRAM AND DEBUG** 에 있는 **Generate Bitstream** 을 클릭

1-1-3. **File** 메뉴에 있는 **Export/Export Hardware** 클릭. **Include bitstream** 을 선택하고 **OK** 클릭

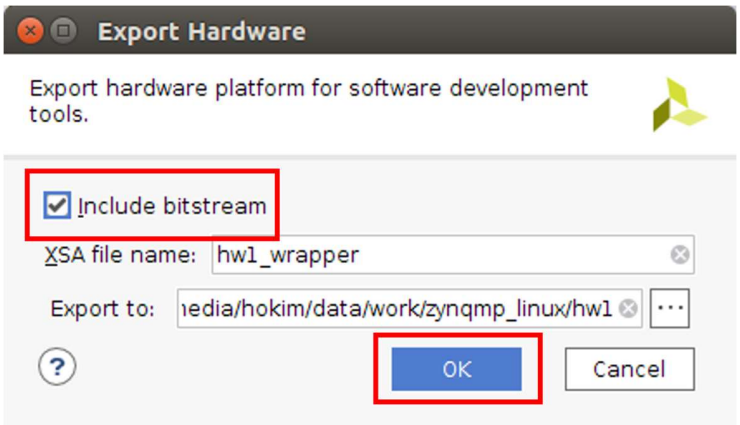


Figure 2. Export Hardware

www.inipro.net
hokim@inipro.net

Create/Configure Petalinux Project Step 2

2-1. Petalinux (ZynqMP template) Project 를 생성하고, **Step1** 에서 생성된 **xsa** 파일에 기초하여 **Project** 를 **Configuration**

```
$ cd ~/work/zynqmp_linux/petalinux
$ petalinux-create -t project -n ultra96 --template zynqMP
$ cd ultra96
$ petalinux-config --get-hw-description=../../hw1/
```

2-1-1. Petalinux project(ultra96)을 생성. **../../hw1** 에 있는 **hw1_wrapper.xsa** 을 **import** 하면 **Configuration** 화면이 나타남

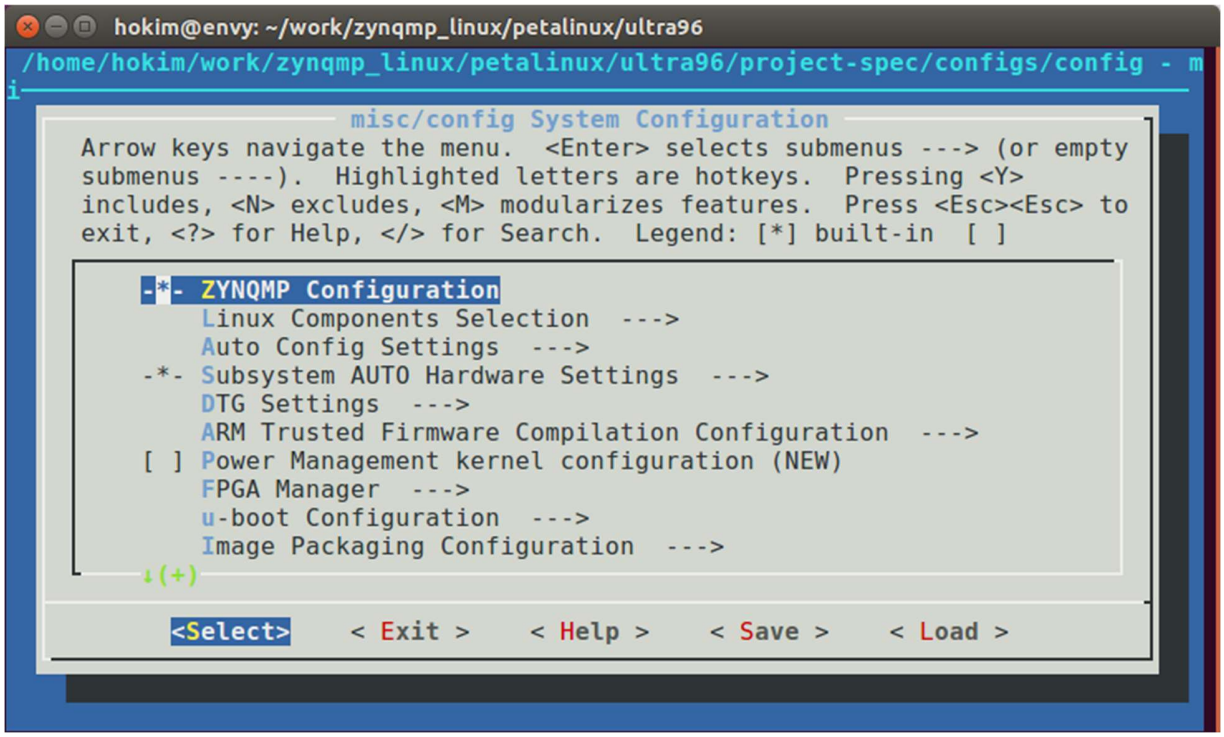


Figure 3. Petalinux Configuration

2-1-2. Subsystem AUTO Hardware Setting →Serial Settings →Primary stdin/stdout 을 **uart1** 으로 선택

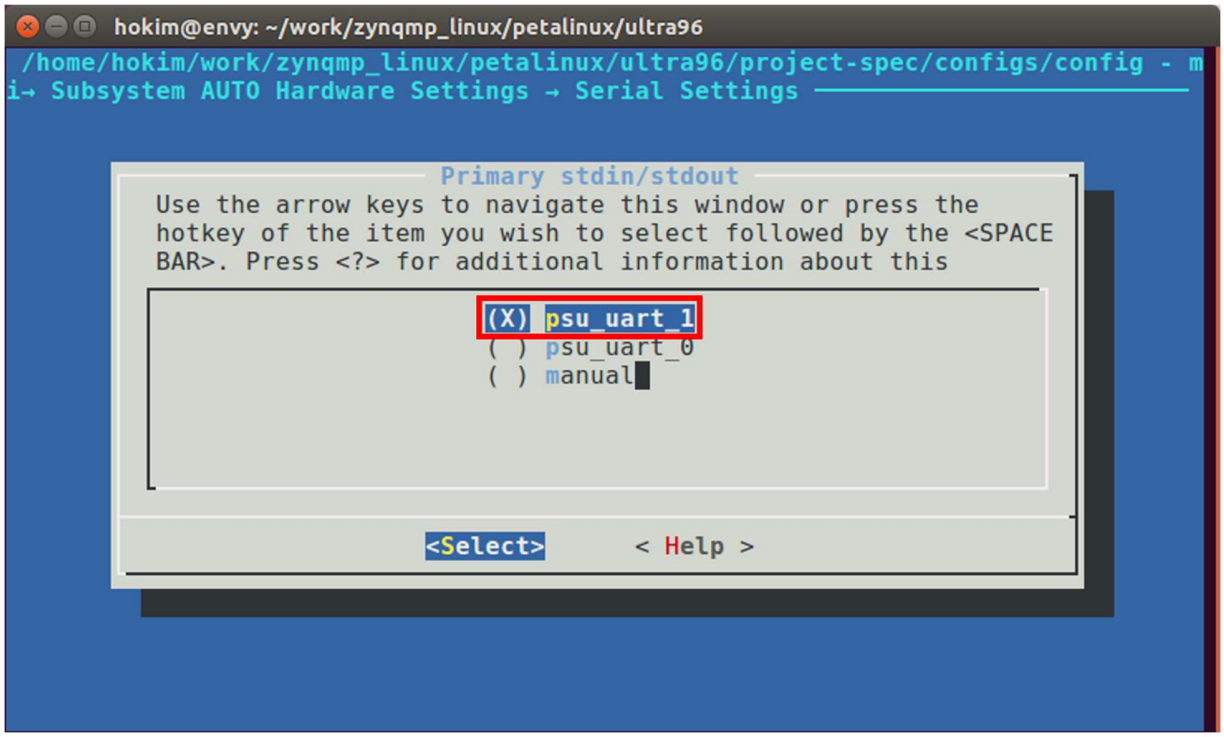


Figure 4. Petalinux Configuration(Primary stdin/stdout)

2-1-3. DTG_Settings →MACHINE_NAME 을 avnet-ultra96-rev1 으로 선택

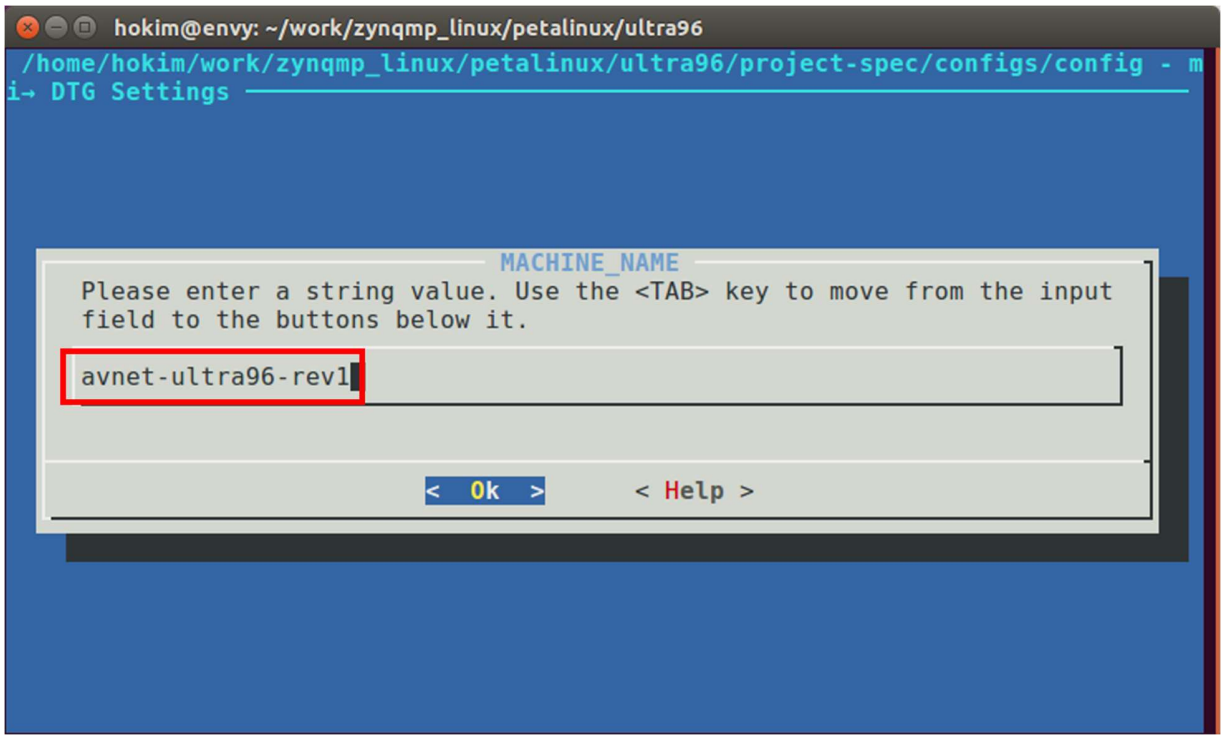


Figure 5. Petalinux Configuration(DTG MACHINE_NAME)

www.inipro.net
hokim@inipro.net

2-1-4. DTG_Settings →Kernel Bootargs → generate boot args automatically 를 disable 하고 DTG Setting → Kernel Bootargs → user set kernel bootargs 을 earlycon console=ttyPS0,115200 clk_ignore_unused root=/dev/mmcblk0p2 rw rootwait uio_pdrv_genirq.of_id=xlnx,generic-uio cma=512M 로 설정

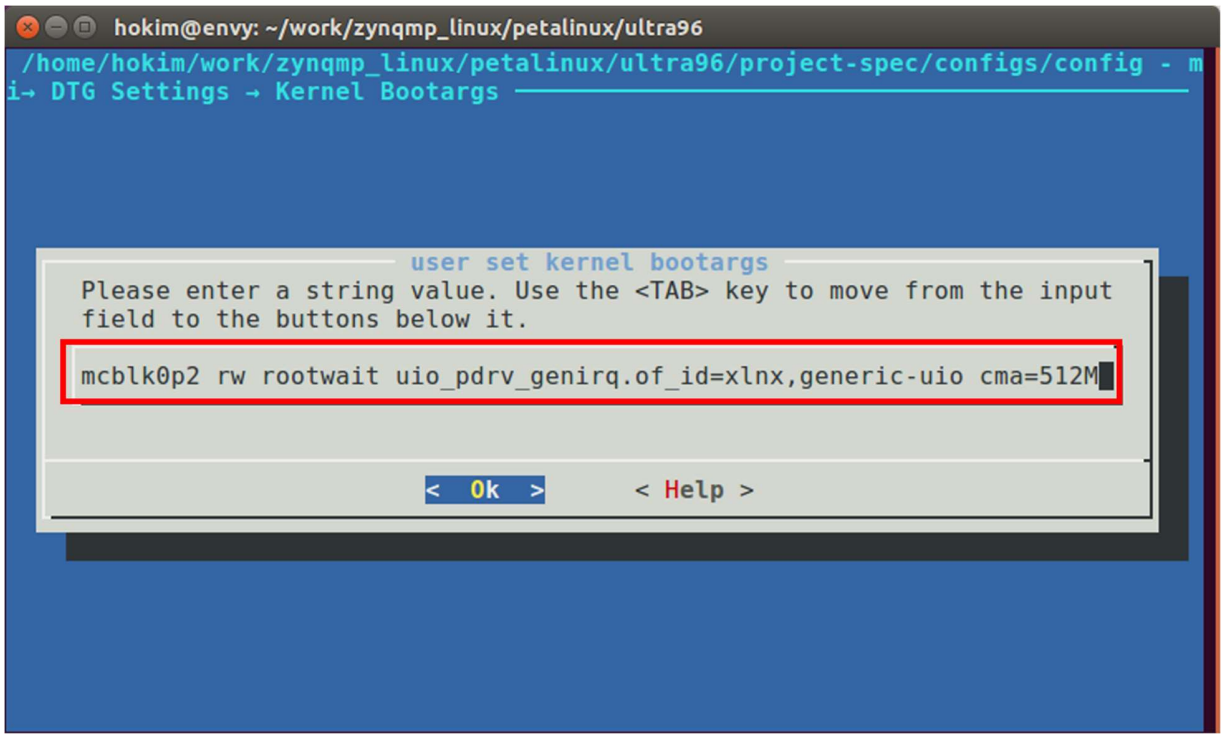


Figure 6. Petalinux Configuration(kernel bootargs)

2-1-5. u-boot Configuration → u-boot config target 을 avnet_ultra96_rev1_defconfig 로 설정

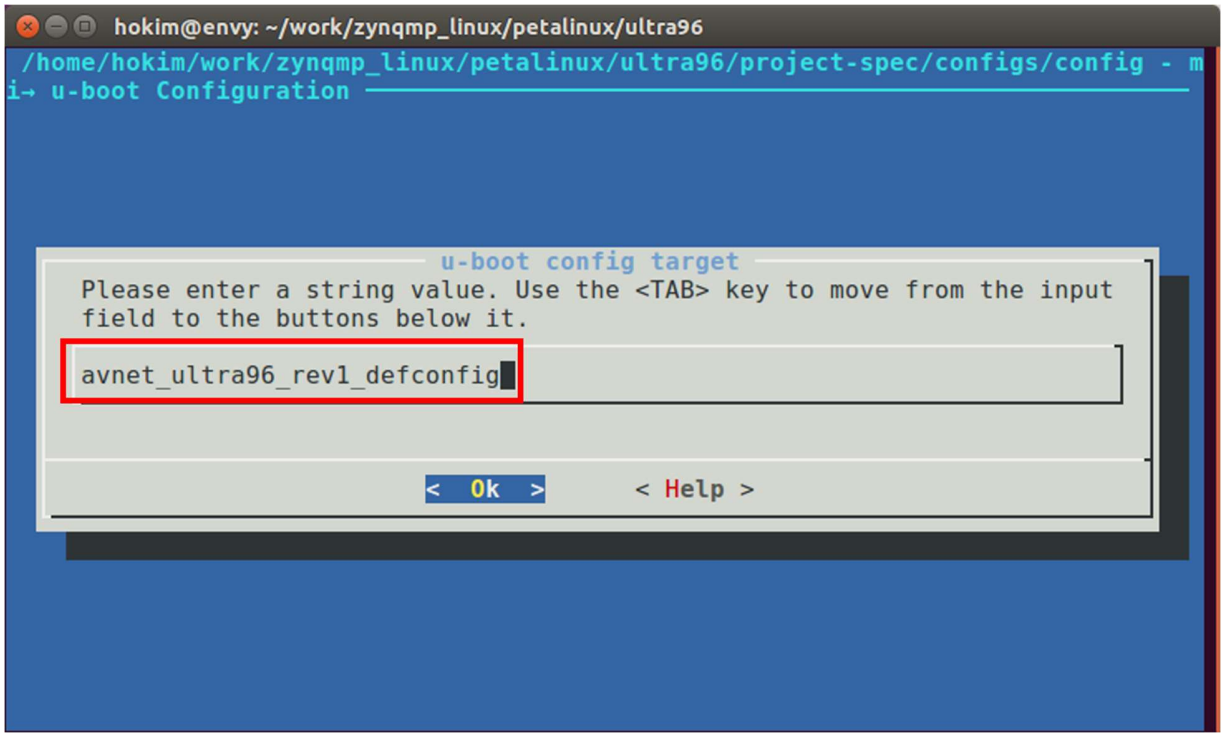


Figure 7. Petalinux Configuration(u-boot config)

2-1-6. Image Packaging Configuration → Root filesystem type 을 EXT 로 설정

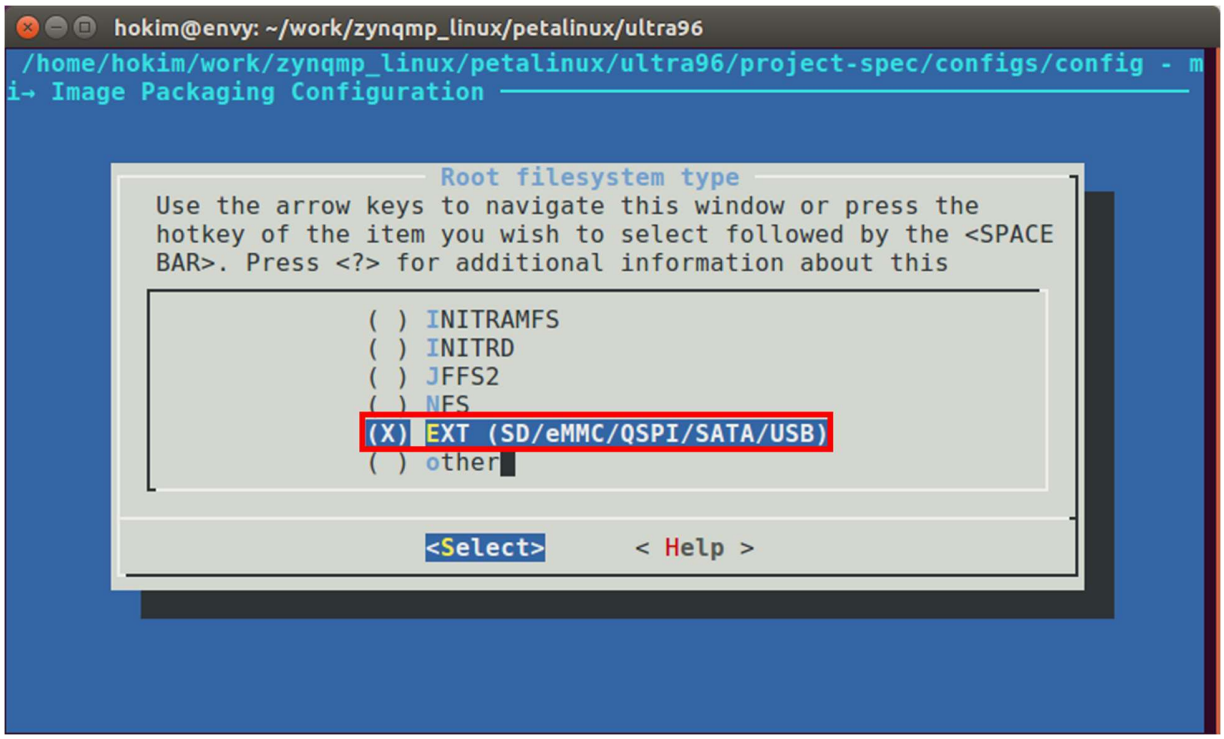


Figure 8. Petalinux Configuration(Root filesystem type)

www.inipro.net
hokim@inipro.net

2-1-7. Image Packaging Configuration →Copy final images to tftpboot 를 disable

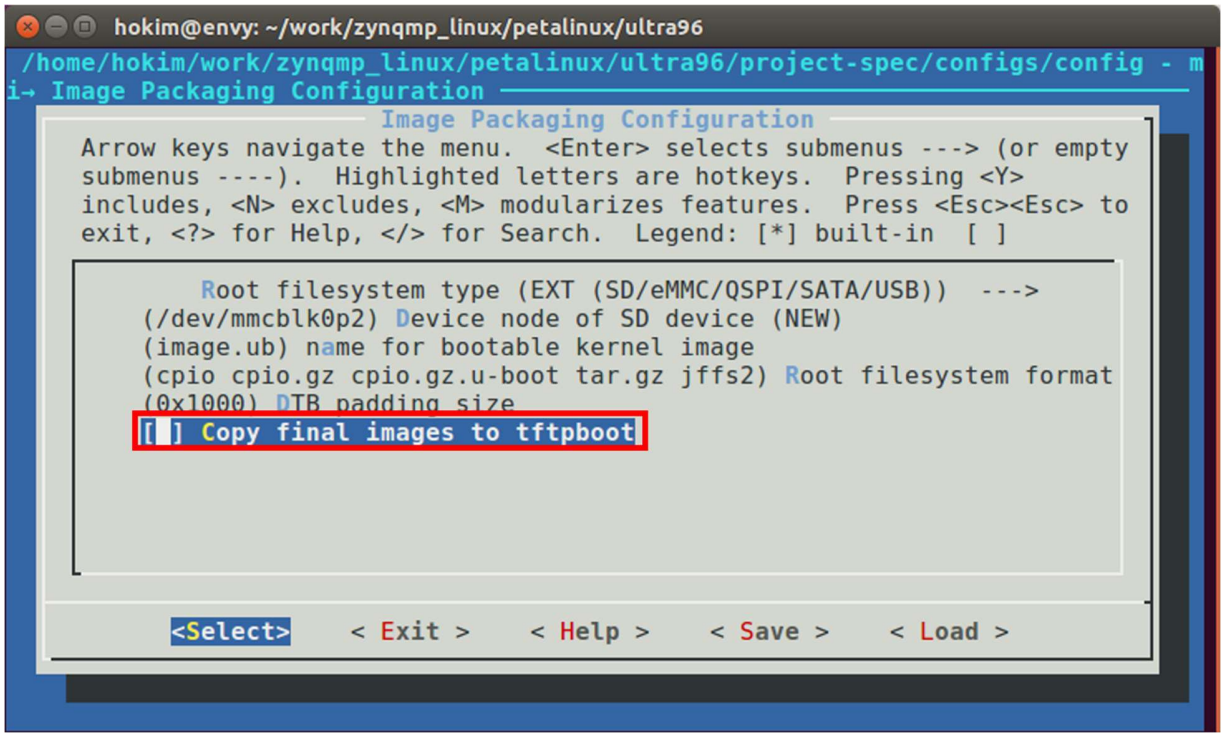


Figure 9. Petalinux Configuration(tftpboot disable)

2-1-8. Yocto Settings → YOCTO_MACHINE_NAME 을 ultra96-zynqmp 로 설정

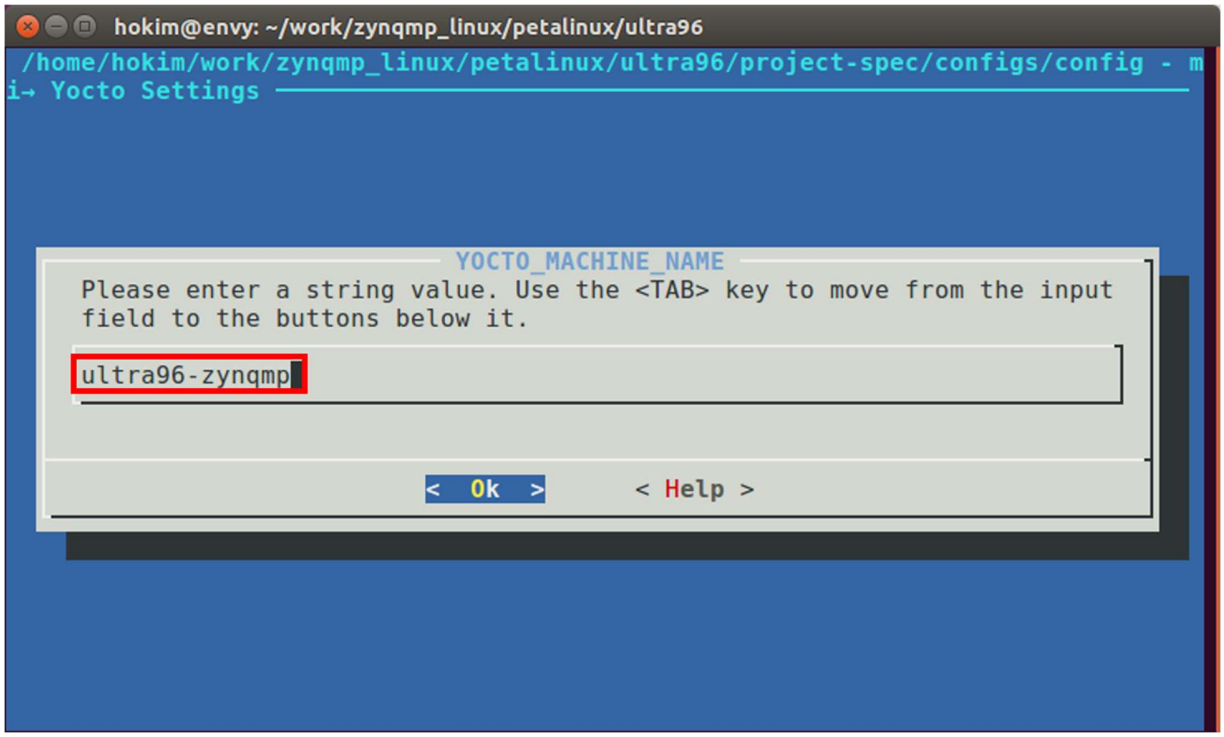


Figure 10. Petalinux Configuration(YOCTO_MACHINE_NAME)

2-1-9. Yocto Settings → Add pre-mirror url → pre-mirror url path 를 지우고 **Xilinx site** 에서 받은 **downloads**(디렉토리는 사용자의 것으로 지정)로 설정

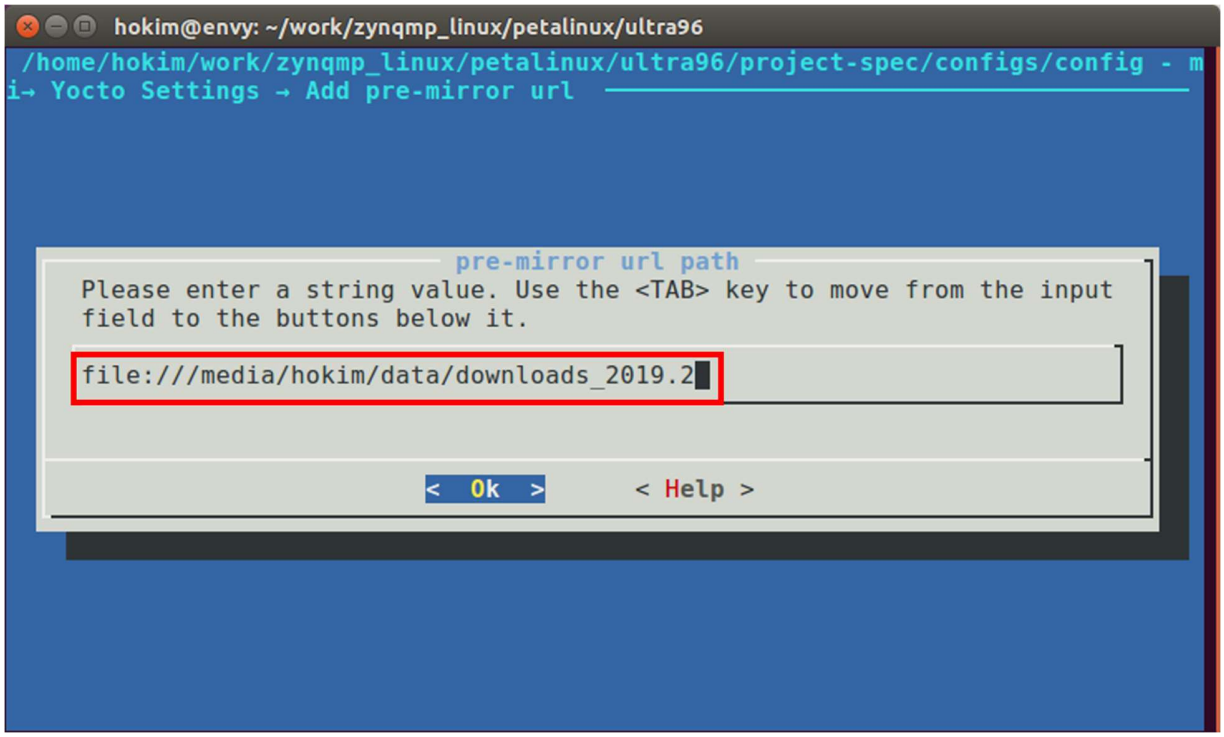


Figure 11. Petalinux Configuration(pre-mirror url)

www.inipro.net
hokim@inipro.net

2-1-10. Yocto Settings → local sstate feeds setting 을 **Xilinx site** 에서 받은 **sstate**(디렉토리는 사용자의 것으로 지정)로 설정

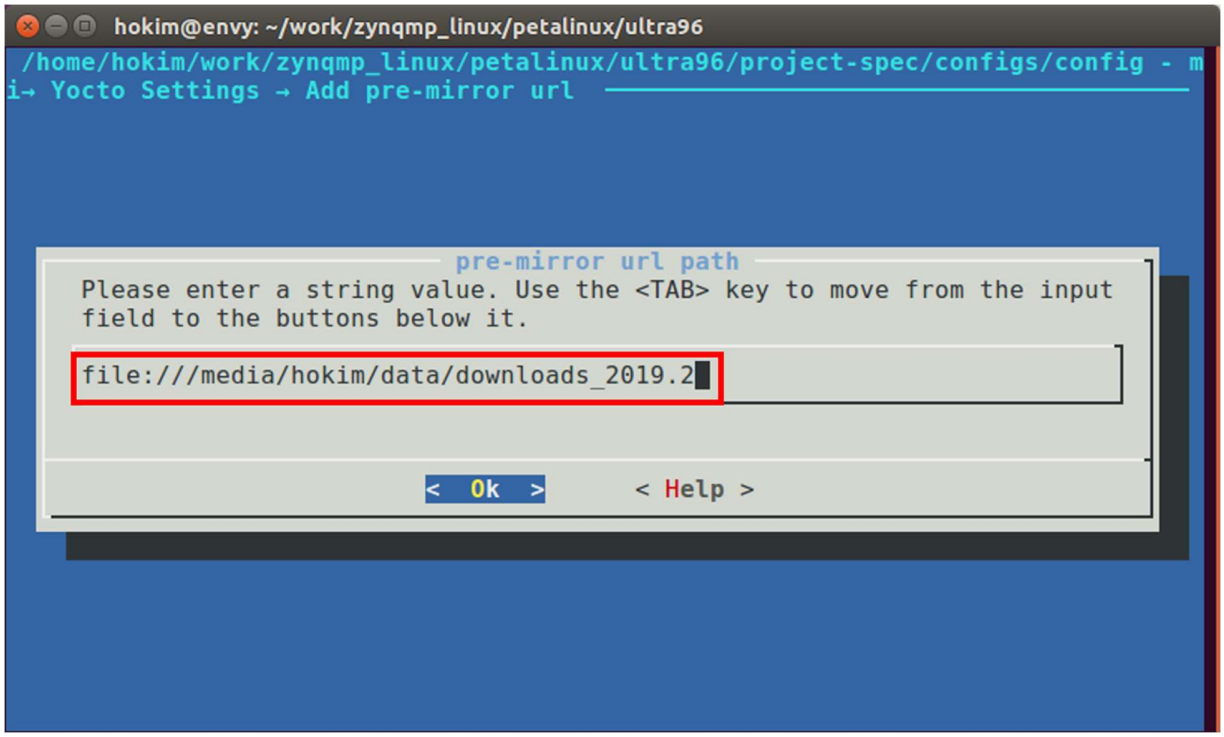


Figure 12. Petalinux Configuration(local sstate feeds)

2-1-11. Yocto Settings → User Layers → user layer 0 을 **${PROOT}/../meta-inipro** 로 설정

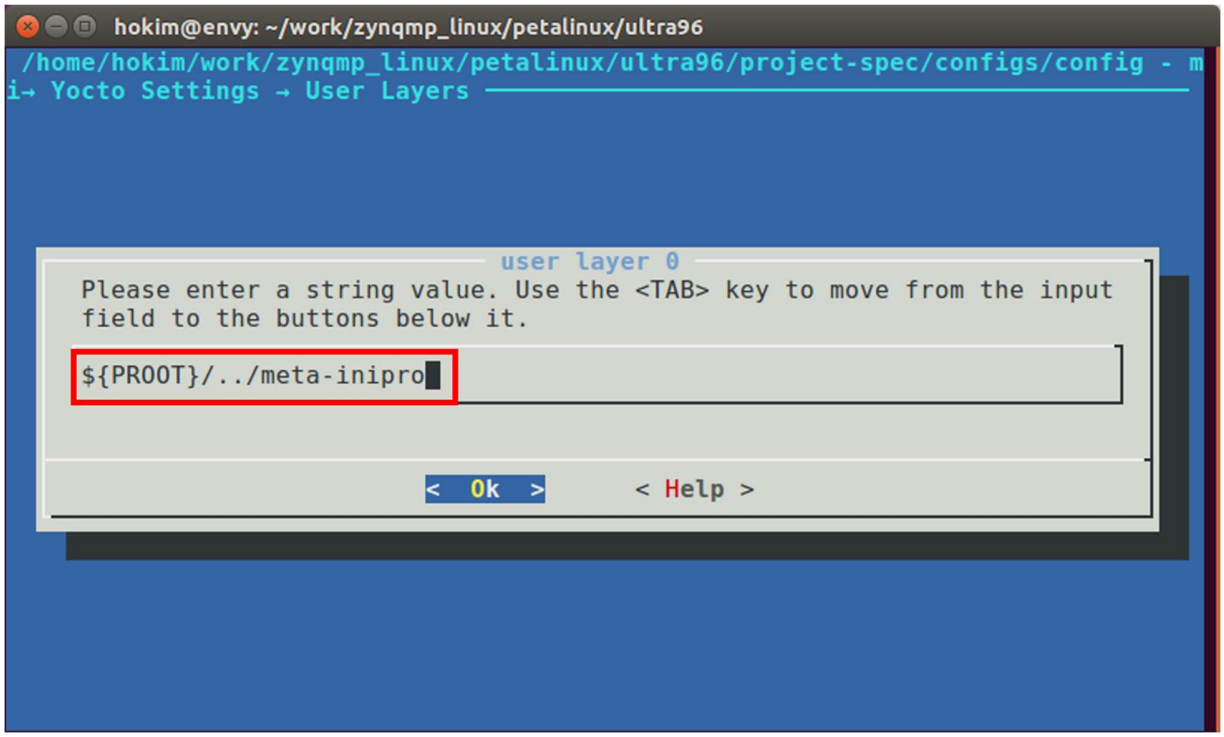


Figure 13. Petalinux Configuration(user layer)

2-1-12. Exit 를 반복하여 설정을 저장하고 **Configuration** 을 종료

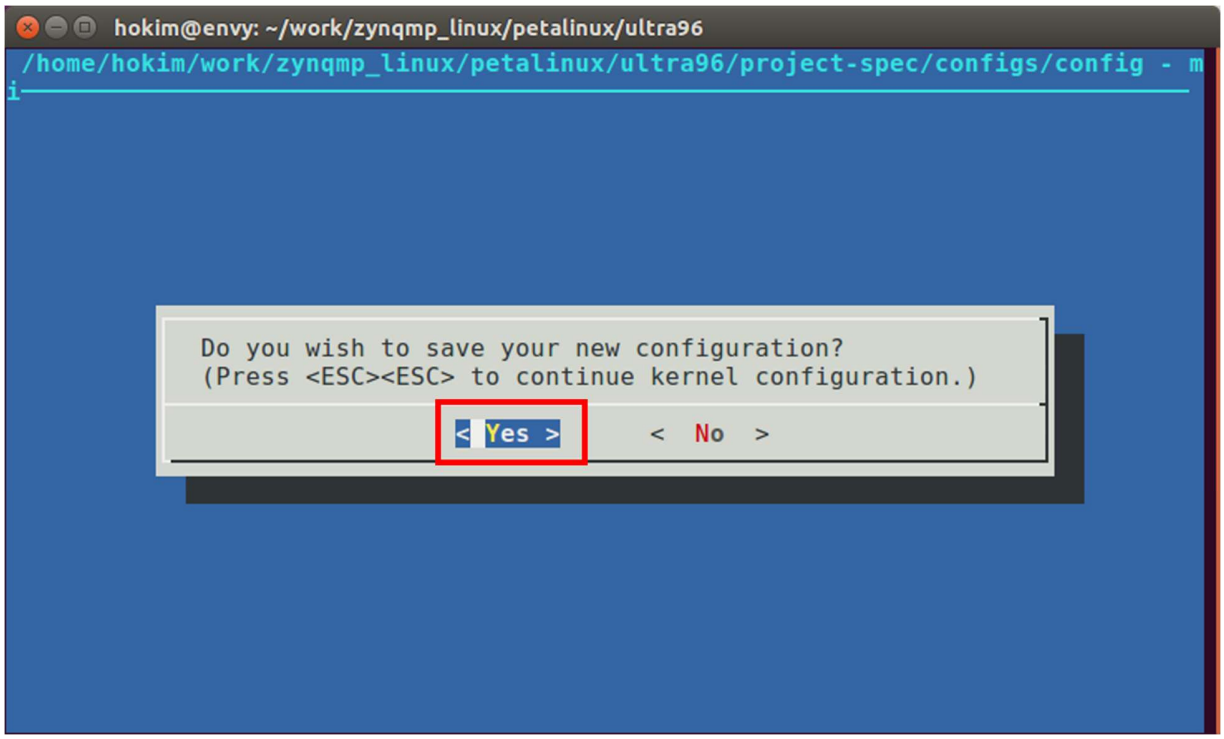


Figure 14. Petalinux Configuration(Exit)

www.inipro.net
hokim@inipro.net

2-1-13. Configuration 과정이 종료된 뒤 생성된 **project-spec/configs/config** 편집기(**vi** or **gedit**)로 열어 다음의 내용이 설정된 부분과 일치하는지 확인

```
53 #
54 # Serial Settings
55 #
56 CONFIG_SUBSYSTEM_SERIAL_PSU_UART_1_SELECT=y
57 # CONFIG_SUBSYSTEM_SERIAL_PSU_UART_0_SELECT is not set
```

```
140 #
141 # DTG Settings
142 #
143 CONFIG_SUBSYSTEM_MACHINE_NAME="avnet-ultra96-rev1"
144
145 #
146 # Kernel Bootargs
147 #
148 # CONFIG_SUBSYSTEM_BOOTARGS_AUTO is not set
149 CONFIG_SUBSYSTEM_USER_CMDLINE="earlycon console=ttyPS0,115200 clk_ignore_unu
    sed root=/dev/mmcblk0p2 rw rootwait uio_pdrv_genirq.of_id=xlnx,generic-uio c
    ma=512M"
```

```
172 # CONFIG_SUBSYSTEM_UBOOT_CONFIG_OTHER is not set
173 CONFIG_SUBSYSTEM_UBOOT_CONFIG_TARGET="avnet_ultra96_rev1_defconfig"
```

```
177 #
178 # Image Packaging Configuration
179 #
180 # CONFIG_SUBSYSTEM_ROOTFS_INITRAMFS is not set
181 # CONFIG_SUBSYSTEM_ROOTFS_INITRD is not set
182 # CONFIG_SUBSYSTEM_ROOTFS_JFFS2 is not set
183 # CONFIG_SUBSYSTEM_ROOTFS_NFS is not set
184 CONFIG_SUBSYSTEM_ROOTFS_EXT=y
185 # CONFIG_SUBSYSTEM_ROOTFS_OTHER is not set
186 CONFIG_SUBSYSTEM_SDROOT_DEV="/dev/mmcblk0p2"
187 CONFIG_SUBSYSTEM_UIMAGE_NAME="image.ub"
188 CONFIG_SUBSYSTEM_RFS_FORMATS="cpio cpio.gz cpio.gz.u-boot tar.gz jffs2"
189 CONFIG_SUBSYSTEM_DTB_PADDING_SIZE=0x1000
190 # CONFIG_SUBSYSTEM_COPY_TO_TFTPBOOT is not set
```

```
199 #
200 # Yocto Settings
201 #
202 CONFIG_YOCTO_MACHINE_NAME="ultra96-zynqmp"
```

```
217 #
218 # Add pre-mirror url
219 #
220 CONFIG_PRE_MIRROR_URL="file:///media/hokim/data/downloads_2019.2"
221
222 #
223 # Local sstate feeds settings
224 #
225
226 #
227 # Default sstate feeds ${PETALINUX}/components/yocto always added
228 #
229 CONFIG_YOCTO_LOCAL_SSTATE_FEEDS_URL="/media/hokim/data/sstate_aarch64_2019.2
```

Figure 15. Petalinux Configuration(confie file)

2-1-14. **config** 파일이 의도한 바와 일치하지 않았을 때 다음과 같은 명령으로 다시 **Configuration** 과정을 진행

```
$ petalinux-config
```

2-1-15. **project-spec/meta-user/conf/petalinuxbsp.conf** 파일을 편집기로 열어 다음을 추가(line 17-35). line 29-31 은 ultra96v2 보드만을 위해 사용되어야 함. line 34 의 디렉토리 경로는 사용자의 것을 사용할 것

```
17 MACHINE_FEATURES_remove = "mipi"
18
19 DISTRO_FEATURES_append = " bluez5 dbus"
20
21 EXTRA_IMAGE_FEATURES += "package-management"
22
23 PACKAGE_FEED_URIS = "http://192.168.2.50:5678"
24
25 IMAGE_ROOTFS_EXTRA_SPACE = "102400"
26
27 SIGGEN_UNLOCKED_RECIPES += "tzdata dnf-native dropbear dtc-native cmake-nati
   ve"
28
29 PREFERRED_VERSION_wilc-firmware = "15.2"
30
31 ULTRA96_VERSION_ultra96-zynqmp = "2"
32
33 SSTATE_MIRRORS_append = "  \
34 file://.* file:///media/hokim/data/sstate_aarch64_2019.2_2/PATH \n \
35 "
```

Figure 16. Petalinux Configuration(petalinuxbsp.conf)

www.inipro.net
hokim@inipro.net

Configure Device Tree Step 3

3-1. Device Tree 의 configure task 에 의해 xsa 파일로부터 ZynqMP PS 과 PL IP 들을 위한 Device Tree 생성하고 system-conf.dtsi 를 편집하여 Device Tree 를 수정/추가

3-1-1. 다음의 명령을 사용하여 **Device Tree Generation**. 생성된 Device Tree 의 결과**(*.dts, *.dtsi)**를 **components/plnx_workspace/device-tree/device-tree/** 에서 확인할 것. **zynqmp.dtsi** 는 zynqmp 의 PS 에 공통으로 적용되는 내용이 들어있고, **avnet-ultra96-rev1.dtsi** 는 ultra96v1 보드 PS 에 적용되는 내용이 들어있음. PL 에 ip 들이 있으면 pl.dtsi 가 그 정보를 가지면서 생성됨. 최상위에 있는 **system-top.dts** 이 차후에 다루어질 **system-user.dtsi** 를 포함하여 다른 dtsi 파일들을 include 함.

```
$ petalinux-build -c device-tree -x configure
```

```
/dts-v1/;
#include "zynqmp.dtsi"
#include "zynqmp-clk-ccf.dtsi"
#include "pcw.dtsi"
#include "avnet-ultra96-rev1.dtsi"
/ {
    chosen {
        bootargs = "earlycon clk_ignore_unused";
        stdout-path = "serial0:115200n8";
    };
    aliases {
        i2c0 = &i2c1;
        serial0 = &uart1;
        serial1 = &uart0;
        spi0 = &spi0;
        spi1 = &spi1;
    };
    memory {
        device_type = "memory";
        reg = <0x0 0x0 0x0 0x7ff00000>;
    };
};
#include "system-user.dtsi"
```

Figure 17. components/plnx_workspace/device-tree/device-tree/system-top.dts

3-1-2. **system-user.dtsi** 를 편집하여 **Device Tree** 수정. Ultra96v1 보드의 경우 **avnet-ultra96-rev1.dtsi** 가 이 보드에만 적용되는 정보를 가지고 있기 때문에 이 과정을 진행할 필요가 없음. 차후에 PL 이 추가 되면 이 파일에 그에 걸맞는

정보를 추가해야 함. Ultra96v2 를 다음과 같이 추가하여 Ultra96v1 정보를 Ultra96v2 의 정보로 수정

```
1  /include/ "system-conf.dtsi"
2  / {
3      /delete-node/ ltc2954;
4  };
5
6  &sdio_pwrseq {
7      chip_en-gpios = <&gpio 8 1>; // requires a patched pwrseq_simple.c for W
   ILC3000
8  };
9
10 &gpio {
11     /delete-property/gpio-line-names;
12 };
13
14 &i2csw_4 {
15     /delete-node/ pmic@5e;
16     irps5401_13: irps5401@13 {
17         compatible = "infineon,irps5401";
18         reg = <0x13>;
19     };
20     irps5401_14: irps5401@14 {
21         compatible = "infineon,irps5401";
22         reg = <0x14>;
23     };
24     ir38060_15: ir38060@15 {
25         compatible = "infineon,ir38060";
26         reg = <0x15>;
27     };
28 };
```

Figure 18. project-spec/meta-user/recipes-bsp/device-tree/system-user.dtsi.

Configure U-boot Step 4

4-1. 부팅단계에서 Linux kernel 을 call 하기 위한 부팅방법 및 그를 위해 필요한 device 정보를 platform-top.h 에 기술.

avnet_ultra96_rev1_defconfig configuration 에 ultra96 보드를 위한 대부분의 설정이 들어있으므로 여기서는 I2C buses(I2C Mux)에 대해서 다룬다.

4-1-1. **platform-top.h** 에 다음을 추가(line 26-39)

www.inipro.net
hokim@inipro.net

```
26 /* FIXME Will go away soon */
27 #define CONFIG_SYS_I2C_MAX_HOPS         1
28 #define CONFIG_SYS_NUM_I2C_BUSES        9
29 #define CONFIG_SYS_I2C_BUSES    { \
30                                 {0, {I2C_NULL_HOP} }, \
31                                 {0, {{I2C_MUX_PCA9548, 0x75, 0} } }, \
32                                 {0, {{I2C_MUX_PCA9548, 0x75, 1} } }, \
33                                 {0, {{I2C_MUX_PCA9548, 0x75, 2} } }, \
34                                 {0, {{I2C_MUX_PCA9548, 0x75, 3} } }, \
35                                 {0, {{I2C_MUX_PCA9548, 0x75, 4} } }, \
36                                 {0, {{I2C_MUX_PCA9548, 0x75, 5} } }, \
37                                 {0, {{I2C_MUX_PCA9548, 0x75, 6} } }, \
38                                 {0, {{I2C_MUX_PCA9548, 0x75, 7} } }, \
39                                 }
40
```

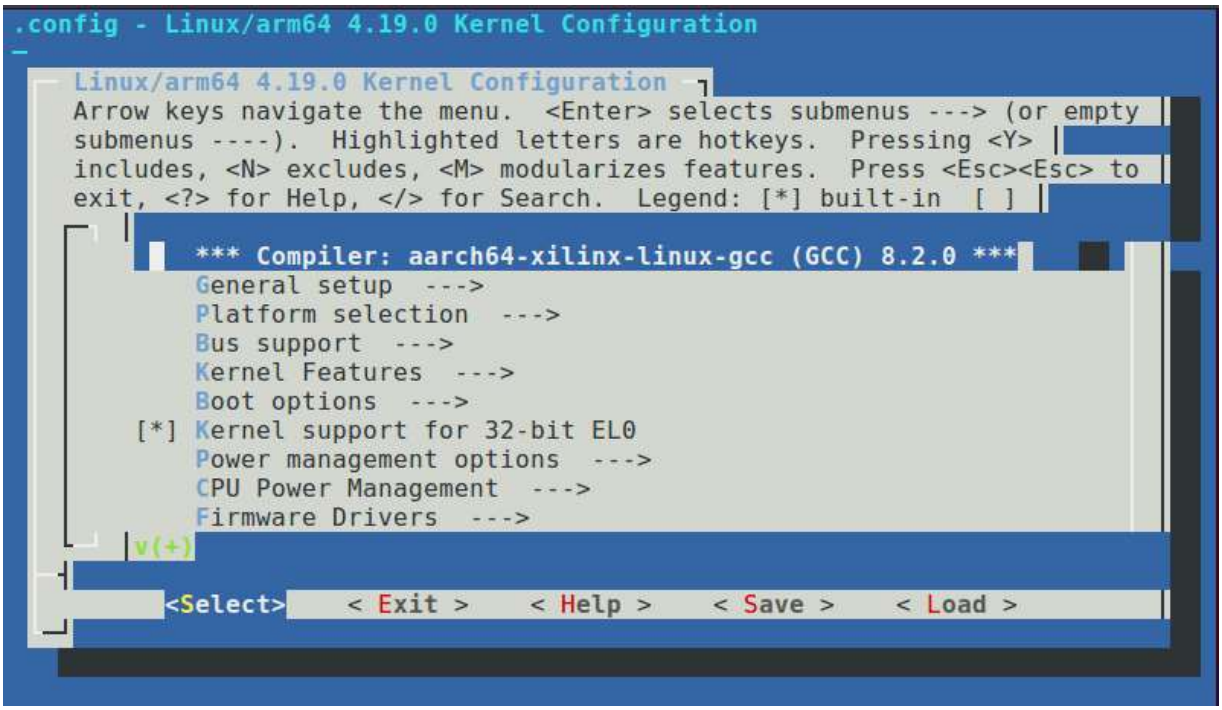
Figure 19. Project-spec/meta-user/recipes-bsp/u-boot/files/platform-top.h

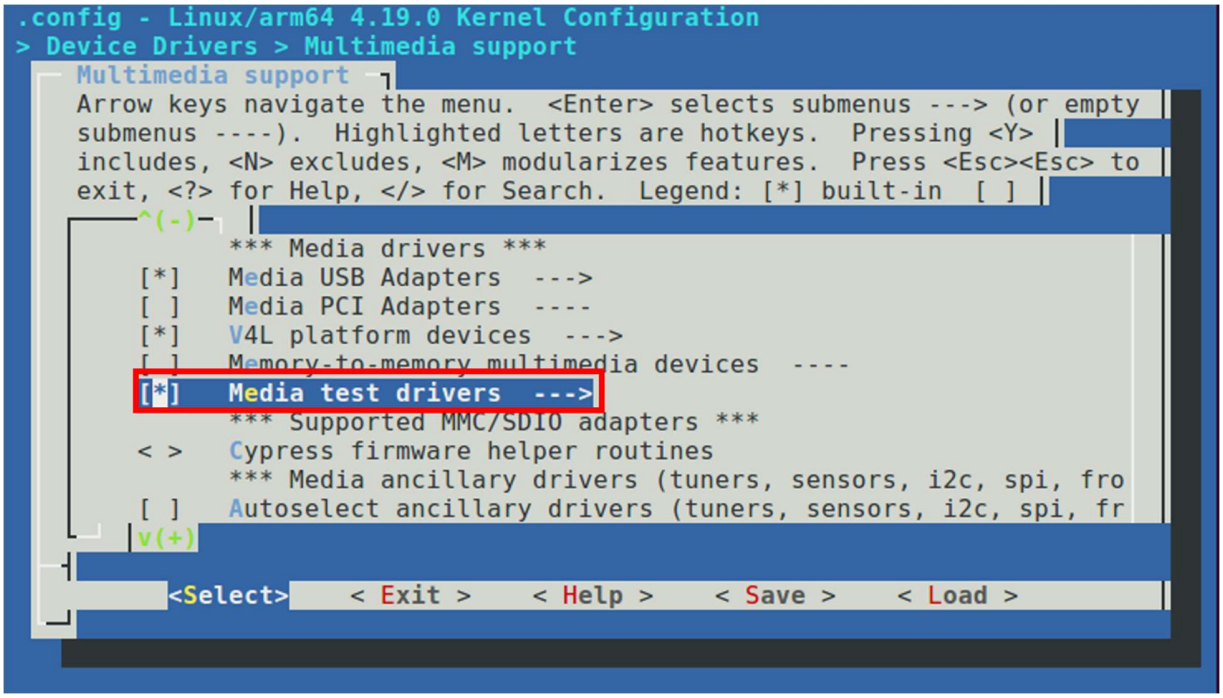
Configure Kernel Step 5

5-1. Kernel 은 default 로 xilinx kernel source(github.com/Xilinx/linux-xlnx)의 arch/arm64/configs/xilinx_zynqmp_defconfig 에 의해 설정되면 petalinux-config 명령어에 의해 그 설정을 변경할 수 있으면 kernel source code 를 수정(patch) 할 수 있음. 여기서 설정변경만 다룬다

5-1-1. Kernel configuration 을 위해 다음 명령을 실행하고 **configuration menu** 에서 **driver(Device Drivers→Multimedia support→Media test drivers→Virtual Video Test Driver)**를 **enable** 시키고 저장 종료

```
$ petalinux-config -c kernel
```





www.inipro.net
hokim@inipro.net

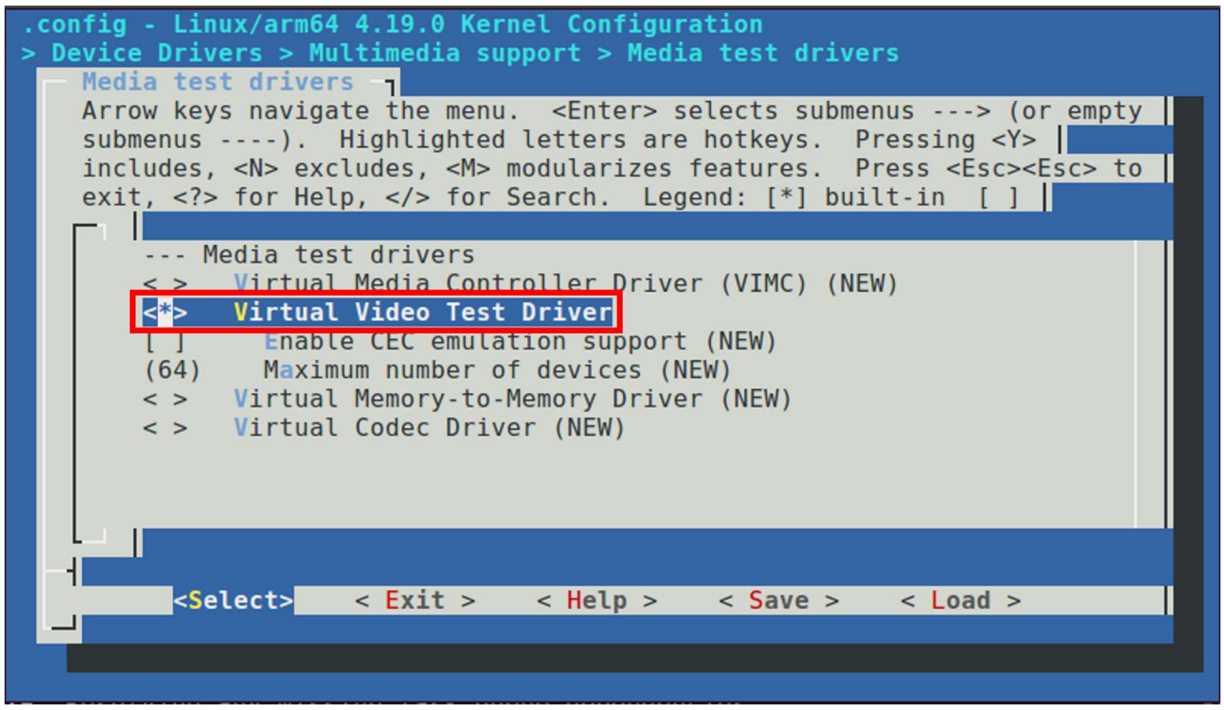


Figure 20. Kernel Configuration

5-1-2. 설정을 종료하면 **components/plnx_workspace/sources/linux-xlnx/** 아래에 **Kernel source** 가 있게 되고 **kernel** 설정은 그 아래에 **.config.new** 파일에 저장되지만 일시적으로 저장되어서 변경된 내용만 따로 저장하기위해 다음과 같이 한다. **linux-xlnx_2019.2.bbappend** 와 **devtool-fragment.cfg** 에서 저장내용 확인

```
$ petalinux-config
```

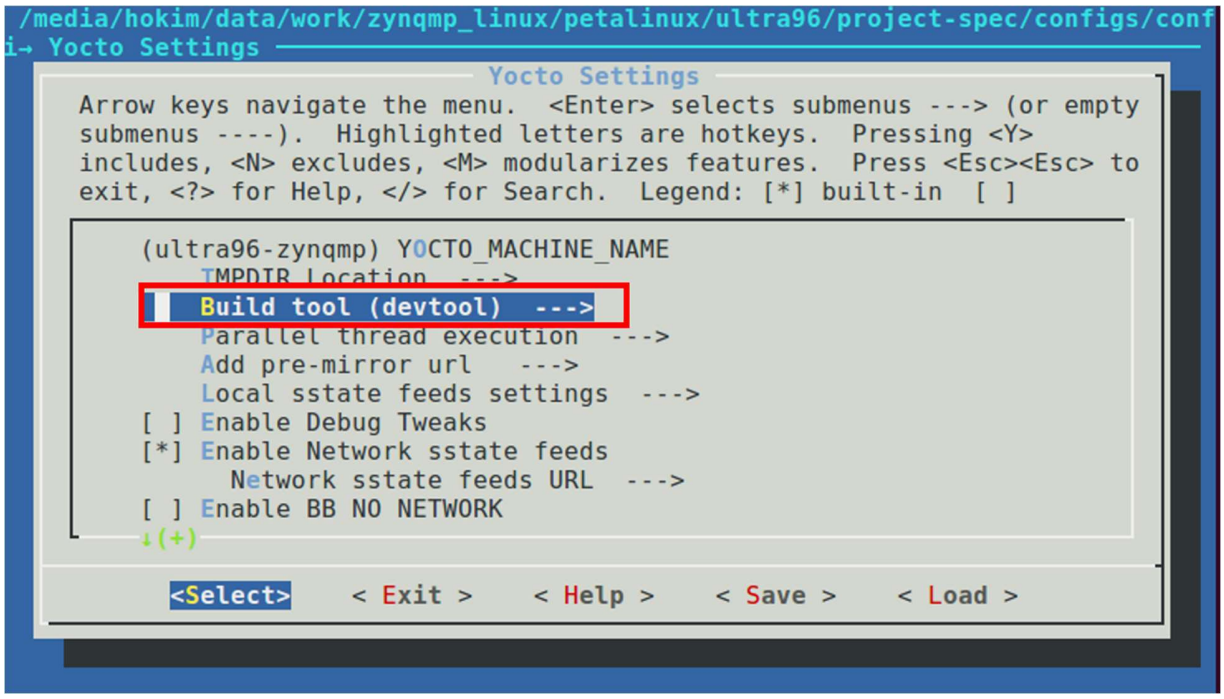


Figure 21. Build tool (bitbake → devtool)

```
$ petalinux-build -c kernel -x update-recipe
$ petalinux-config
```

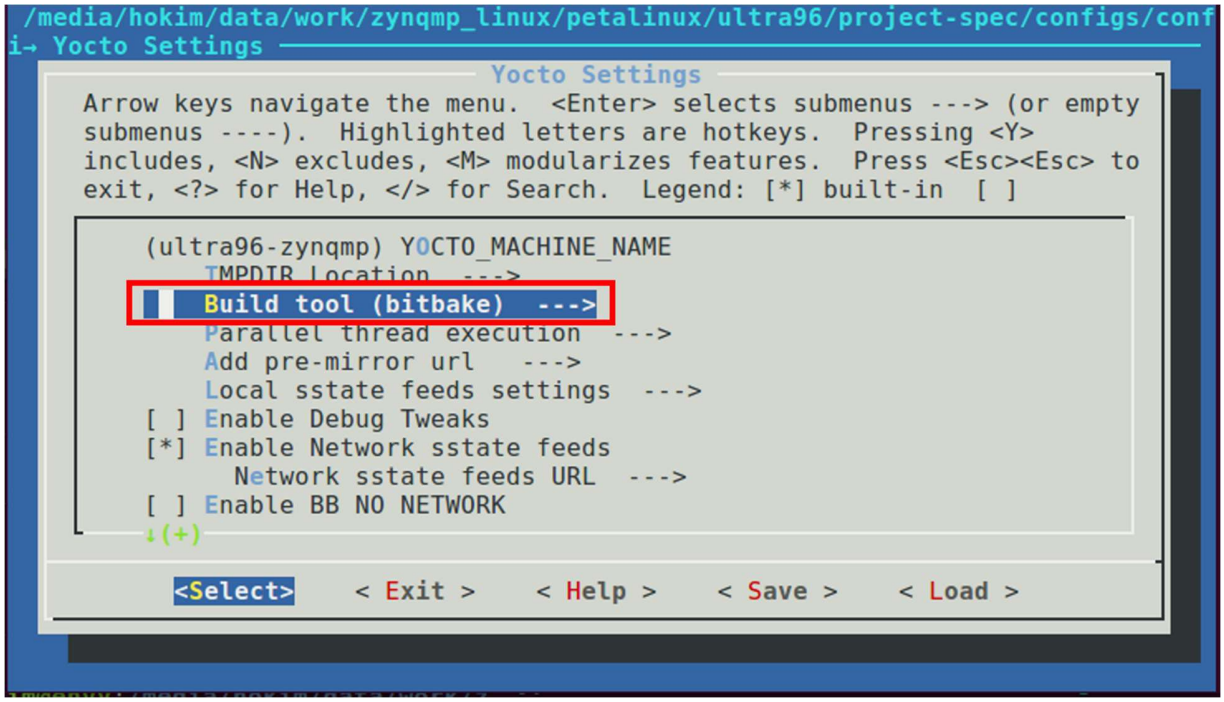


Figure 22. Build tool (devtool→bitbake)

www.inipro.net
hokim@inipro.net

```
1 FILESEXTRAPATHS_prepend := "${THISDIR}/${PN}:"
2
3 SRC_URI += "file://devtool-fragment.cfg"
4
```

Figure 23. project-spec/meta-user/recipes-kernel/linux/linux-xlnx_2019.2.bbappend

```
1 CONFIG_V4L_TEST_DRIVERS=y
2 # CONFIG_VIDEO_VIMC is not set
3 CONFIG_VIDEO_VIVID=y
4 # CONFIG_VIDEO_VIVID_CEC is not set
5 CONFIG_VIDEO_VIVID_MAX_DEVS=64
6 # CONFIG_VIDEO_VIM2M is not set
7 # CONFIG_VIDEO_VICODEC is not set
8 CONFIG_VIDEO_V4L2_TPG=y
```

Figure 24. project-spec/meta-user/recipes-kernel/linux/linux-xlnx/devtool-fragment.cfg

.

Configure/Build Petalinux-user-image Step 6

6-1. Petalinux user image 는 linux root filesystem(w/ apps, libs, dev tools...)을 사용자가 원하는 대로 구성할 수 있도록 한다. 또한 Petalinux user image 의 Build 는 step 1-5 까지 기술된 Device Tree, U-boot, Kernel 과 FSBL, PMU(Power Management Unit) FW, ATF(Arm Trusted Firmware) 등 linux 부팅과 실행을 위한 모든 것들의 Build 와 의존성을 갖는 최상위의 Linux Build recipe 이다. 다시 말하면 Petalinux user image 를 설치하면 앞에 기술한 모든 것들이 그에 따라서 설치되게 된다.

6-1-1. **Step2** 으로 생성된 **petalinux user image** 의 구성목록을 확인 line 13-28 까지가 기본적으로 image 에 설치될 **package** 목록. 각각의 **package** 들은 의존성을 갖는 다른 **package** 들을 가지고 있으므로 실제로 설치되는 package 들은 이보다 많음. line 29 는 linux image 가 부팅하고나서 로그인 할 수 있는 사용자와 암호(root:root)

```
1  DESCRIPTION = "PETALINUX image definition for Xilinx boards"
2  LICENSE = "MIT"
3
4  require recipes-core/images/petalinux-image-common.inc
5
6  inherit extrausers
7  COMMON_FEATURES = "\
8          ssh-server-dropbear \
9          hwcodecs \
10         "
11 IMAGE_LINGUAS = " "
12
13 IMAGE_INSTALL = "\
14         kernel-modules \
15         haveged \
16         mtd-utils \
17         canutils \
18         openssh-sftp-server \
19         pciutils \
20         run-postinsts \
21         udev-extraconf \
22         packagegroup-core-boot \
23         packagegroup-core-ssh-dropbear \
24         tcf-agent \
25         watchdog-init \
26         bridge-utils \
27         hellopm \
28         "
29 EXTRA_USERS_PARAMS ?= "usermod -P root root;"
```

Figure 25. project-spec/meta-plnx-generated/recipes-core/images/petalinux-user-image.bb

6-1-2. 위의 자동 생성된 기본적인 package 목록에 더 필요한 **목록을 petalinux-user-image.bbappend** 에 추가. 이 파일은 아래의 명령에 의해 생성되는 디렉토리 밑에 위치하여야 함. 또한 같은 파일에서 root 사용자 암호(xxxx)를 원하대로 수정(line 1)

```
$ mkdir -p project-spec/meta-user/recipes-core/images
```


www.inipro.net
hokim@inipro.net

```
EXTRA_USERS_PARAMS = "usermod -P xxxx root;"
IMAGE_INSTALL_append = " nano \
                       tzdata \
                       dtc \
                       kmod \
                       e2fsprogs-resize2fs \
                       i2c-tools \
                       iw \
                       wpa-supplicant \
                       ultra96-power-button \
                       bluez5 \
                       ${@bb.utils.contains('ULTRA96_VERSION', '2', 'wilc-firmware-wilc3000', '', d)} \
                       ${@bb.utils.contains('ULTRA96_VERSION', '2', 'wilc', '', d)} \
                       cmake \
                       packagegroup-petalinux-self-hosted \
                       packagegroup-petalinux-openamp \
                       packagegroup-petalinux-v4lutils \
                       packagegroup-petalinux-display-debug \
                       packagegroup-petalinux-x11 \
                       packagegroup-petalinux-opencv-dev \
                       packagegroup-petalinux-gstreamer-dev \
                       packagegroup-petalinux-qt-dev \
                       packagegroup-petalinux-qt-extended-dev \
                       packagegroup-core-tools-debug \
                       ffmpeg \
                       file \
                       ldd \
                       xrt \
                       xrt-dev \
                       zocl \
                       zocl-dev \
                       opencl-clhpp-dev \
                       opencl-headers-dev \
                       "
```

Figure 26. project-spec/meta-user/recipes-core/images/petalinux-user-image.bbappend

6-1-3. Petalinux user image 를 build 하기전에 부팅한 linux 에 wiifi 를 통해 접근할 필요가 있으므로 **wpa-supplicant.conf-sane** 파일에 사용자 환경의 **ssid** 와 **psk** 암호 등록을 확인할 것 예를 들어 KT 는 저자의 집, inipro 는 저자의 사무실

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
update_config=1

network={
    ssid="KT_GiGA_2G_Wave2_ECA3"
    key_mgmt=WPA-PSK
    psk="0gd99fh792"
}
network={
    ssid="inipro"
    key_mgmt=WPA-PSK
    psk="69569010"
}

```

Figure 27. ../meta-inipro/recipes-connectivity/wpa-supplicant/files/wpa_supplicant.conf.sane

6-1-4. Petalinux user image 를 build. build 가 완료된 후 **images/linux** 디렉토리 아래에 있는 결과물들 확인. 예를 들어 **rootfs.tar.gz** 는 root filesystem 이 압축된 파일. **u-boot.elf** 는 u-boot. **system.bit** 는 Bitstream. **bl3.elf** 는 ATF. **Image** 는 Kernel, **system.dtb** 는 Device Tree, **pmufw.elf** 는 PMU FW, **zynqmp_fsbl.elf** 는 fsbl, **image.ub** 는 Device Tree + Kernel

```
$ petalinux-build
```

Create BOOT.BIN Step 7

7-1. 부팅가능한 BOOT.BIN 은 zynqmp_fsbl.elf +pmufw.elf + system.bit + bl31.elf + u-boot.elf 로 구성된다. petalinux-package 명령어에 의해 생성된다

7-1-1. 다음의 명령어에 의해 **bootgen.bif** 로 구성된 **images/linux** 아래에 **BOOT.BIN** 생성

```
$ petalinux-package --force --boot --fsbl images/linux/zynqmp_fsbl.elf --u-boot images/linux/u-boot.elf --pmufw images/linux/pmufw.elf --fpga images/linux/system.bit
```

```
1 the_ROM_image:
2 {
3     [bootloader, destination_cpu=a53-0] /tmp/tmp.j7p2CNl3sV/zynqmp_fsbl.elf
4     [pmufw_image] /tmp/tmp.j7p2CNl3sV/pmufw.elf
5     [destination_device=pl] /tmp/tmp.j7p2CNl3sV/system.bit
6     [destination_cpu=a53-0, exception_level=el-3, trustzone] /tmp/tmp.j7p2CN
  l3sV/bl31.elf
7     [destination_cpu=a53-0, exception_level=el-2] /tmp/tmp.j7p2CNl3sV/u-boot
  .elf
8 }
```

Figure 28. buid/bootgen.bif

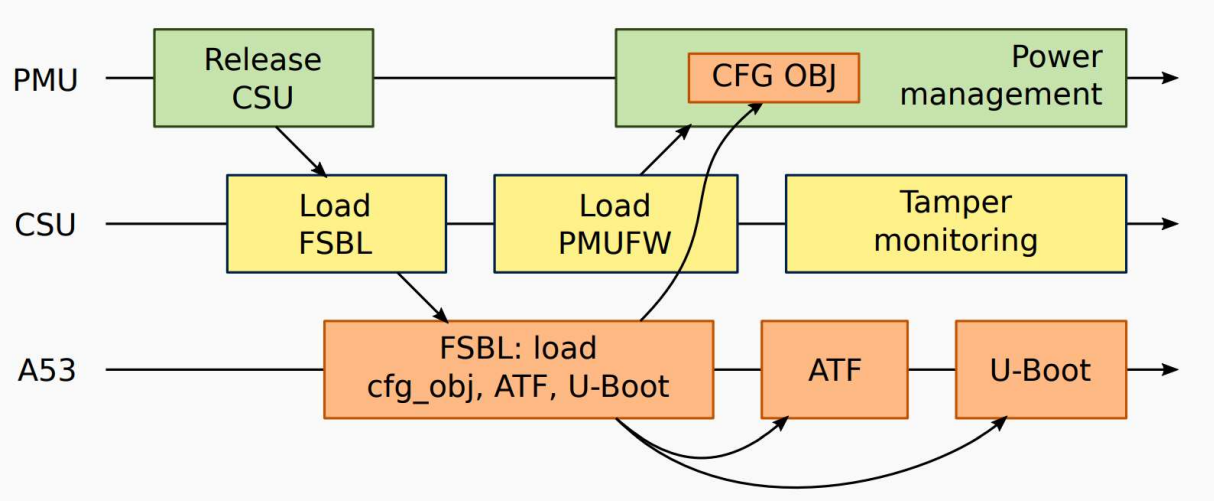


Figure 29. ZynqMP Booting Sequence

Prepare sdcard Step 8

8.1 sdcard 로 부팅가능한 Linux 를 만들때 booting 파티션과 root filesystem 파티션에 각각 BOOT.BIN, image.ub 와 rootfs.tar.gz 을 넣는다.

8-1-1. 다음의 명령어로 파티션 나누기, 파티션 포맷, 파일복사를 진행. 파티션을 나눈후 sdcard 를 뺏다가 다시 넣어야 함

```
$ sudo parted /dev/mmcblk0
(parted) print
Model: SD SL16G (sd/mmc)
Disk /dev/mmcblk0: 15.9GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start  End  Size  Type  File system  Flags

(parted) mkpart primary fat32 0 200MB
(parted) mkpart primary ext4 200MB 100%
(parted) print
Model: SD SL16G (sd/mmc)
Disk /dev/mmcblk0: 15.9GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start  End     Size    Type     File system  Flags
 1      512B   200MB   200MB   primary  fat32        lba
 2      200MB  15.9GB  15.7GB  primary  ext4         lba

(parted) quit
```

```
$ sudo mkfs.vfat -n card /dev/mmcblk0p1
$ sudo mkfs.ext4 -L root /dev/mmcblk0p2
$ cp images/linux/{BOOT.BIN,image.ub} /media/hokim/card/
$ sync
```


www.inipro.net
hokim@inipro.net

Test Step 9

9.1 Ultra96 보드에 sdcard 꽂고 부팅. usb uart 를 연결하여 부팅이 끝나면 로그인하고 다음 명령어를 사용하여 보드의 wlan0 ip 를 확인. Host 에서 그 ip 를 사용하여 ssh 로 연결

```
# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:352968 errors:0 dropped:0 overruns:0 frame:0
          TX packets:352968 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:48851674 (48.8 MB)  TX bytes:48851674 (48.8 MB)

wlan0     Link encap:Ethernet  HWaddr d0:57:7b:57:79:c0
          inet addr:172.30.1.25  Bcast:172.30.1.255  Mask:255.255.255.0
          inet6 addr: fe80::dc66:5ba4:57e4:a817/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6312201 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7832750 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4148458803 (4.1 GB)  TX bytes:9967167667 (9.9 GB)
```

```
$ ssh root@172.30.1.25
```