DPU on PYNQ (2)

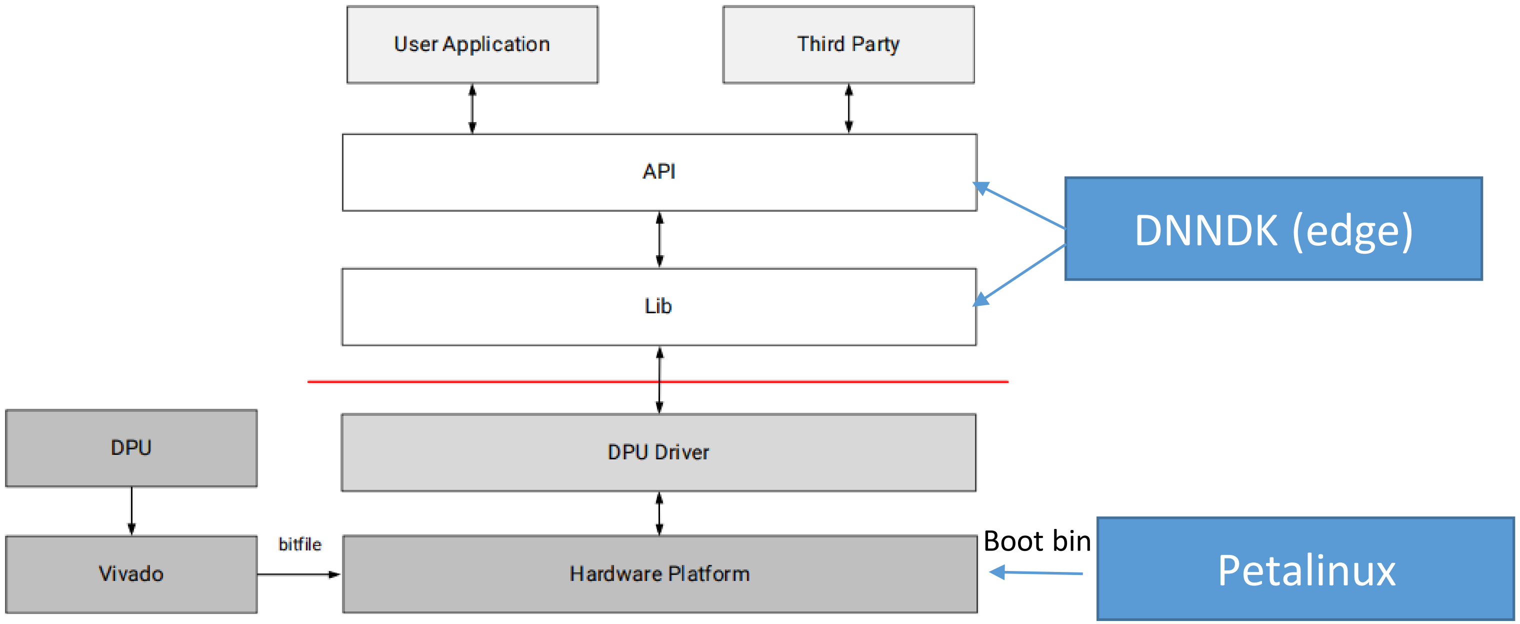
Petalinux

Outline

- DPU project Architecture
- What is petalinux?
- Build pynqz2_dpu boot bin.

DPU Project Architecture

- Vivado: Generate HW design and export bitstream
- Petalinux: Build boot bin base on HW design and create embedded Linux system for FPGA board.
- DNNDK: Interact with DPU



Yocto Project

- Yocto is a open source development tools to create Linux distributions for embedded and IoT software for underlying architecture of the embedded hardware.
- Yocto project is hosted by Linux Foundation and many well-known companies are members of it.

What is Petalinux

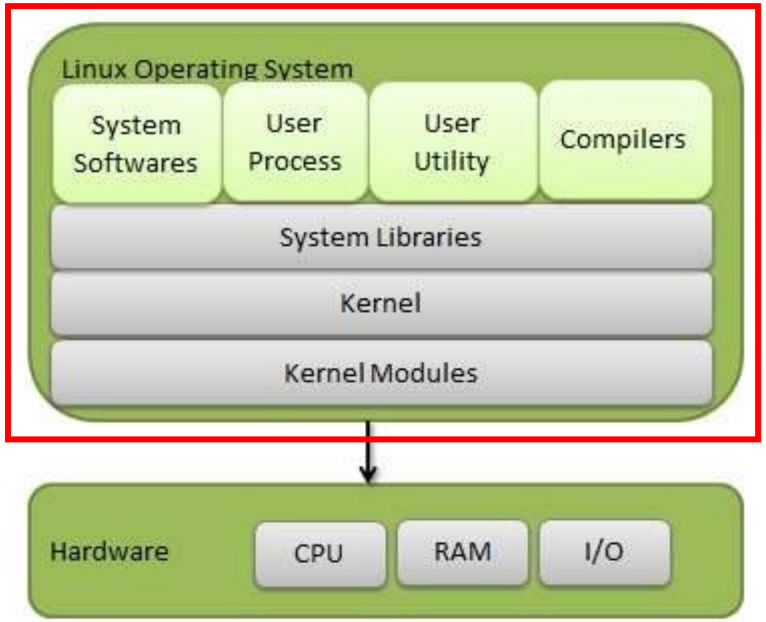
- Petalinux is a embedded Linux development solution for Xilinx Zynq SoCs.

- Petalinux contain a set of high level commands that are built on top of the Yocto Linux distribution. It allows user to customize, build, and deploy Embedded Linux Solutions/Linux images for Xilinx processing system.

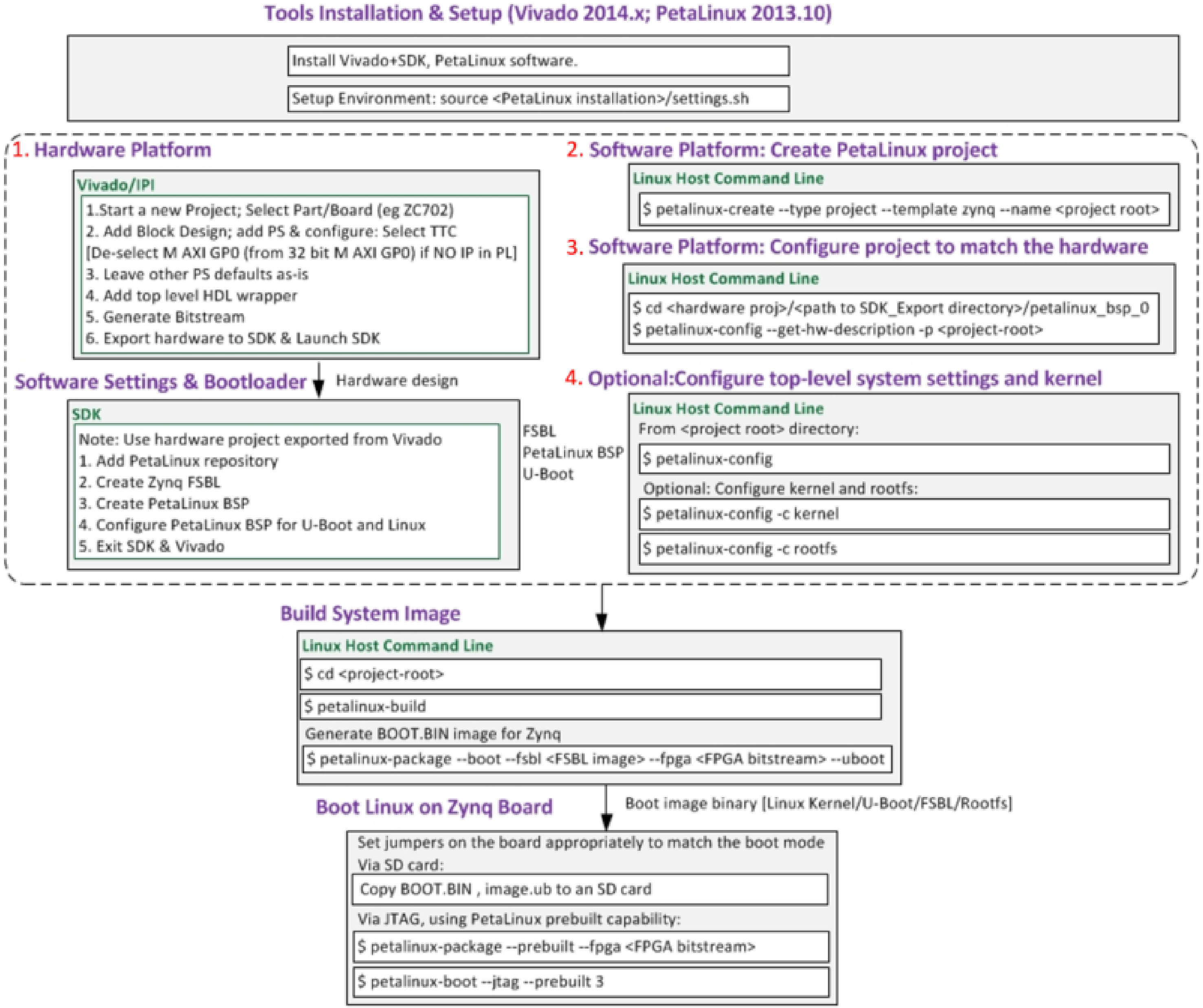
- Petalinux is used for accelerate design productivity, and works with the Xilinx hardware design tools (Vivado).

Design Flow

- A Complete petalinux design flow for a Xilinx Zynq device.



Components of Linux System



Tools Installation & Setup (Vivado 2014.x; PetaLinux 2013.10)

Install Vivado+SDK, PetaLinux software.

Setup Environment: source <PetaLinux installation>/settings.sh

1. Hardware Platform

Vivado/IPI

1.Start a new Project; Select Part/Board (eg ZC702)
2. Add Block Design; add PS & configure: Select TTC
[De-select M AXI GP0 (from 32 bit M AXI GP0) if NO IP in PL]
3. Leave other PS defaults as-is
4. Add top level HDL wrapper
5. Generate Bitstream
6. Export hardware to SDK & Launch SDK

Software Settings & Bootloader

Hardware design

SDK

Note: Use hardware project exported from Vivado
1. Add PetaLinux repository
2. Create Zynq FSBL
3. Create PetaLinux BSP
4. Configure PetaLinux BSP for U-Boot and Linux
5. Exit SDK & Vivado

FSBL
PetaLinux BSP
U-Boot

2. Software Platform: Create PetaLinux project

Linux Host Command Line

```
$ petalinux-create --type project --template zynq --name <project root>
```

3. Software Platform: Configure project to match the hardware

Linux Host Command Line

```
$ cd <hardware proj>/<path to SDK_Export directory>/petalinux_bsp_0
$ petalinux-config --get-hw-description -p <project-root>
```

4. Optional:Configure top-level system settings and kernel

Linux Host Command Line

From <project root> directory:

```
$ petalinux-config
```

Optional: Configure kernel and rootfs:

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

```
$ petalinux-config -c rootfs
```

Build System Image

Linux Host Command Line

```
$ cd <project-root>
```

```
$ petalinux-build
```

Generate BOOT.BIN image for Zynq

```
$ petalinux-package --boot --fsbl <FSBL image> --fpga <FPGA bitstream> --uboot
```

Boot Linux on Zynq Board

Boot image binary [Linux Kernel/U-Boot/FSBL/Rootfs]

Set jumpers on the board appropriately to match the boot mode

Via SD card:

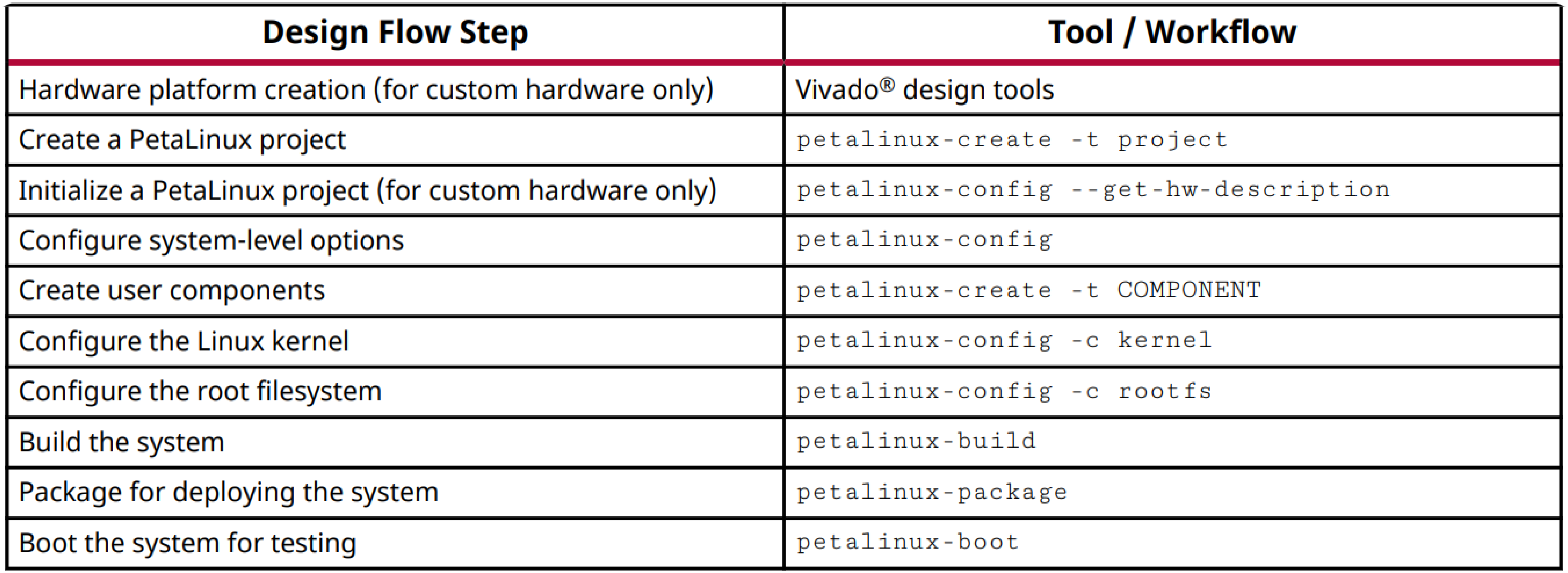
Copy BOOT.BIN , image.ub to an SD card

Via JTAG, using PetaLinux prebuilt capability:

```
$ petalinux-package --prebuilt --fpga <FPGA bitstream>
```

```
$ petalinux-boot --jtag --prebuilt 3
```

Design Flow



	Design Flow Step	Tool / Workflow
V	Hardware platform creation (for custom hardware only)	Vivado® design tools
V	Create a PetaLinux project	`petalinux-create -t project`
V	Initialize a PetaLinux project (for custom hardware only)	`petalinux-config --get-hw-description`
V	Configure system-level options	`petalinux-config`
	Create user components	`petalinux-create -t COMPONENT`
	Configure the Linux kernel	`petalinux-config -c kernel`
V	Configure the root filesystem	`petalinux-config -c rootfs`
V	Build the system	`petalinux-build`
V	Package for deploying the system	`petalinux-package`
	Boot the system for testing	`petalinux-boot`

Install Petalinux

Requirement: Ubuntu 18.04

Change ubuntu mirror source(optional):

- sudo sed -i ‘s/us.archive.ubuntu.com/free.nchc.org.tw/g’ /etc/apt/sources.list

Install dependency:

- sudo apt update
- sudo apt upgrade
- sudo apt install htop vim openssh-server iproute2 gcc g++ net-tools libncurses5-dev zlib1g:i386 libssl-dev flex bison libselinux1 xterm autoconf libtool texinfo zlib1g-dev gcc-multilib build-essential screen pax gawk python3 python3-pexpect python3-pip python3-git python3-jinja2 xz-utils debianutils iputils-ping libegl1-mesa libsdl1.2-dev pylint3 cpio chrpath socat python –y
- pip3 install --upgrade pip

Install petalinux(Installer Download link):

- ./petalinux-v2019.2-final-installer.run *~/Petalinux_2019.2/*

Install Petalinux

- Change /bin/sh to bash

PetaLinux tools require that your host system /bin/sh be bash, which in Ubuntu it is dash.

1. In the terminal, run this command: chsh -s /bin/bash.
2. Reboot.
3. Open a terminal and run these commands to make /bin/sh link to /bin/bash:

```
sudo dpkg-reconfigure dash
```

→ no

Petalinux Reference

- how-to-install-petalinux-2020.1 (reference)

https://www.fpgadeveloper.com/how-to-install-petalinux-2020.1/

- Petalinux-2019.2 user guide

https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_2/ug1144-petalinux-tools-reference-guide.pdf

- Getting Started with the Yocto Project - New Developer Screencast Tutorial- https://vimeo.com/36450321
- Xilinx Embedded Linux Build flows: PetaLinux Tools- https://www.youtube.com/watch?v=FhKLwR8w5CI

Build Pynqz2-dpu boot bin

Create Petalinux project

- source ~/Petalinux_2019.2/settings.sh
- petalinux-create --type project --template zynq --name pynqz2_dpu
- cd pynqz2_dpu/
- petalinux-config --get-hw-description=<path to pynqz2_dpu.xsa>
- Change Boot settings
  - DTG Settings->Kernel Bootargs->disable generate boot args automatically
  - and enter the following bootargs in user set kernel bootargs
    - *console=ttyPS0,115200 root=/dev/mmcblk0p2 rw earlyprintk quiet rootfstype=ext4 rootwait cma=256M*
  - Image Packaging Configuration->Root filesystem type (EXT4 (SD card))

Modify device tree

- Change the content of *project-spec/meta-user/recipes-bsp/device-tree/files/system-user.dtsi* to the text in right block.

```
/include/ "system-conf.dtsi"
&amba {
              xlnk {
                            compatible = "xlnx,xlnk-1.0";
              };
};
&amba{
              dpu{
                #address-cells = <1>;
                #size-cells = <1>;
                compatible = "xilinx,dpu";
                base-addr = <0x4f000000>;    //CHANGE THIS ACCORDING TO YOUR DESIGN

                dpucore {
                  compatible = "xilinx,dpucore";
                  interrupt-parent = <&intc>;
                  interrupts = <0 29 4>; //CHANGE THIS ACCORDING TO YOUR DESIGN
                  core-num = <0x1>; //CHANGE THIS ACCORDING TO YOUR DESIGN
                };
              };
};
//usb device tree
/{
 usb_phy0: usb_phy@0 {
  compatible = "ulpi-phy";
  #phy-cells = <0>;
  reg = <0xe0002000 0x1000>;
  view-port = <0x0170>;
  drv-vbus;
 };
};
&usb0 {
 dr_mode = "host";
 usb-phy = <&usb_phy0>;
};
```

Device-Tree

Add DPU driver

zcu102-dpu-trd-2019-1-timer Download link

- Use 7-zip to open *zcu102-dpu-trd-2019-1-timer/apu/dpu_petalinux_bsp/xilinx-dpu-trd-zcu102-v2019.1.bsp*
- Copy *dpu_petalinux_bsp/zcu102-dpu-trd-2019-1/project-spec/meta-user/recipes-modules this folder to ~/petalinux_project/pynqz2_dpu/project-spec/meta-user/*
- Add *CONFIG_dpu in ~/petalinux_project/pynqz2_dpu/project-spec/meta-user/conf/user-rootfsconfig*
- Add *IMAGE_INSTALL_append = "dpu"* in *~/petalinux_project/pynqz2_dpu/project-spec/meta-user/conf/petalinuxbsp.conf*

Configure rootfs

- **petalinux-config -c rootfs**
  - modules -> dpu
  - Filesystem Packages -> console -> utils -> pkgconfig -> pkgconfig, pkgconfig-dev
  - Filesystem Packages -> libs -> gtk+3 -> gtk+3, gtk+3-demo, gtk+3-dev, gtk+3-dbg

Add USB driver

- petalinux-config -c kernel
  - Device Drivers -> USB support -> [*]USB announce new devices
- petalinux-build -c kernel -x finish
- Modify *project-spec/meta-user/recipes-kernel/linux/linux-xlnx/*.cfg* file to the texts at right.

```
CONFIG_USB_OTG=y
# CONFIG_USB_OTG_FSM is not set
# CONFIG_USB_ZERO_HNPTEST is not set
CONFIG_MEDIA_USB_SUPPORT=y
CONFIG_USB_VIDEO_CLASS=y
CONFIG_USB_VIDEO_CLASS_INPUT_EVDEV=y
CONFIG_USB_GSPCA=m
CONFIG_V4L_PLATFORM_DRIVERS=y
CONFIG_VIDEO_ADV7604=y
CONFIG_USB_HID=y
CONFIG_USB_OHCI_LITTLE_ENDIAN=y
CONFIG_USB_SUPPORT=y
CONFIG_USB_COMMON=y
CONFIG_USB_ARCH_HAS_HCD=y
CONFIG_USB=y
CONFIG_USB_ANNOUNCE_NEW_DEVICES=y
CONFIG_USB_DEFAULT_PERSIST=y
CONFIG_USB_EHCI_HCD=y
CONFIG_USB_EHCI_ROOT_HUB_TT=y
CONFIG_USB_EHCI_PCI=y
CONFIG_USB_EHCI_HCD_PLATFORM=y
CONFIG_USB_ACM=m
CONFIG_USB_PRINTER=m
CONFIG_USB_WDM=m
CONFIG_USB_TMC=m
CONFIG_USB_STORAGE=y
CONFIG_USB_CHIPIDEA=y
CONFIG_USB_CHIPIDEA_OF=y
CONFIG_USB_CHIPIDEA_PCI=y
CONFIG_USB_CHIPIDEA_HOST=y
CONFIG_USB_PHY=y
CONFIG_NOP_USB_XCEIV=y
CONFIG_AM335X_CONTROL_USB=y
CONFIG_AM335X_PHY_USB=y
CONFIG_USB_GPIO_VBUS=y
CONFIG_USB_ULPI=y
CONFIG_USB_ULPI_VIEWPORT=y
```

project-spec/meta-user/recipes-kernel/linux/linux-xlnx/*.cfg file content

Build boot bin

- petalinux-build
- petalinux-package --boot --force --fsbl images/linux/zynq_fsbl.elf --fpga images/linux/*.bit --u-boot
- Copy */images/linux/BOOT.BIN*, */images/linux/image.ub* these two files.