QSPI & EMMC boot mode with Ubuntu Rootfs

Please change the swither to 1,0 which is QSPI mode. However, the board should use the SD switcher mode (but plug out the SD card) as JTAG mode to flash the QSPI bootable files.

Note:

1.Suggested use Petalinux **2022.1** to build project, but it is has some bugs for generating bootable files, so the **Bootgen tool** suggest used Petalinux 2020.2.

2. If you directly use Petalinux 2020.2 to build project, it will be fine to use. However, the **Serial port** will has some issue when you used **baud rate**. Petalinux 2020.2 is default use 9600, and it can't change to 115200 (It is bug, if you change it by manual on the config file, the boot file can't right output and running). So, you should care for your **baud rate** setting.

3. For flash the bootable files into board'QSPI, which is based on the **Vitis**'s program flash tool. Please use the **Vitis 2022.1**, which old version has some bugs and broke the flashed files.

Pure QSPI boot hardware to load petalinux, and then format EMMC

Create Petalinux project: 1-4 steps same as the previous guidance.

1. Loading the Petalinux environment by command: `source <path-to-installed-PetaLinux>/settings.sh`

2. Copy Vivado project hardware files (*.xsa) into the workplace folder. Sometimes, Vivado project files in HDF format, which is old-style output by Vivado. We are currently all of XSA format for the ODK hardware project.

3. Create an **Empty** project based a petalinux template, ODK hardware belongs to "zynqMP" parameter for the Zynq UltraScale+ MPSoC series. You can directly use the example command as follows: `petalinux-create --type project --template zynqMP --name <PROJECT_NAME>`Note: you can't create a project with the BSP file, due to ODK's chip is not a development kit board. So, there are suggested to create an **Empty Template** project first.

4. Importing hardware configuration by ODK's Vivado project. Please care for the command, which automatically searches the folder path to look for the hardware file. So, it is just only one hardware file can be copied into the workplace folder. Command as follows: `petalinux-config --get-hw-description= <PATH-TO-XSA Directory>`Such as you can run it as `petalinux-config --get-hw-description=../`(the *.xsa file on the Upper level Directory's folder). Note: Inhere, there is no space with the "=" on the command.

Config the project to be suitable with QSPI boot mode

ODK2 has **128MB** for the QSPI partition (there is 64MB dual paralle), it is should be separated into 5 partitions: BOOT, bootenv, Image, boot.scr, and rootfs.

Run: `petalinux-config` into menuconfiguration GUI

5. Go to `Subsystem AUTO Hardware settings` into `Flash Settings` for partition offset and size settings:

Flash Partition Name	Partition Offset	Size
BOOT	0x0	0x1E00000 (30MB)
bootenv	0x1E00000	0x40000 (256KB)
Image	0x1E40000	0x2040000 (33MB)
boot.scr	0x3E80000	0x180000 (1.5MB)
rootfs	0x4000000	0x4000000 (64MB)

6. Go to `u-boot Configuration`, and then `u-boot script configuration`. Inhere, please setting all parameter as follows into `QSPI /OSPI image offsets`:

- `QSPI/OSPI Kernel offset` according to the Image kernel offset setting, which corresponds: `0x1E40000`
- `QSPI/OSPI Kernel size` according to the Image kernel size setting, which corresponds: `0x2040000`
- `QSPI/OSPI Ramdisk offset` according to the offset of rootfs partition, which corresponds: `0x4000000`
- `QSPI/OSPI Ramdisk size` according to the size of rootfs partition, which corresponds: `0x4000000`
- `QSPI/OSPI fit image offset` same as the Image kernel offset setting, which corresponds: `0x1E40000`
- `QSPI/OSPI fit image size` same as the Image kernel size setting, which corresponds: `0x2040000`

Note: Petalinux 2022.1 is totally diffence with Petalinux 2020.2 in the Offset and Size setting on the u-boot configuration. If you used Petalinux 2020.2 for QSPI project building, please use the below two steps:

Optional step for Petalinux **2020.2**:

a. Adjust the values for the kernel offset and the FIT image size in `<PROJECT_NAME>/project-spec/meta-user/recipes-bsp/u-boot/u-boot-zynq-scr.bbappend`:

Find Line: ## For zynqMP

- `QSPI_KERNEL_OFFSET` according to the above configuration, which corresponds: `0x1E40000`
- `QSPI_RAMDISK_OFFSET` according to the above configuration, which corresponds: `0x4000000`
- `QSPI_KERNEL_SIZE` according to the size of Image kernel partition size, which corresponds: `0x2040000`
- `QSPI_RAMDISK_SIZE` according to the size of rootfs partition size, which corresponds: `0x4000000`

Find line **QSPI_FIT_IMAGE_SIZE** in below, and then the setting corresponds: `0x4000000`

b. Go to `Advanced bootable images storage Settings`change boot image storage settings from SD to **primary flash** for "`boot image settings`" and "`kernel image settings`".

7. Please go to `<PROJECT_NAME>/project-spec/meta-user/recipes-bsp/device-tree/files`, and replace the **system-user.dtsi**. The QSPI devicetree files can be download as follows:

Note: If you used ODK3 board, please used below devicetree file. The ODK3 has some special setting, such as USB. (Please change use it to replace **system-user.dtsi** as well, the path is same as above.)

8. Run: `petalinux-config -c u-boot` to config BOOT_SCRIPT_OFFSET, go to `ARM architecture` to find `Boot script offset`, and then setting it as **0x3E80000**, which corresponds with the above table.

Optional step for Petalinux 2020.2: Go to `Environment`, setting `Environment is not stored` for empty selected, and then selected `Environment is in SPI flash` .

Note: This step is for Petalinux 2020.2 only, if you use Petalinux 2022.1, please ignore this step.

9. Go to `Boot options`, and then `Boot media`, please select `Support for booting from QSPI flash`.

10. If you have some special request on the kernel function, which is can be confined by command: `petalinux-config -c kernel`

Note: I have create new guidance for USB WIFI setting as follow section, which is some driver source should choose in the kernel steps.

11. Due to we should use the QSPI's Rootfs to format EMMC, which is request some applications to install on the Rootfs. The config command: p`etalinux-config -c rootfs`

Go to: `Filesystem Packages -> Base -> e2fsprogs`, and then select `e2fsprogs` and `e2fsprogs-mke2fs`

Go to `Filesystem Packages -> Base -> util-linux`, and then select `util-linux-mkfs`, `util-linux-fdisk`

Go to `Filesystem Packages -> utils -> dosfstools`, and then select `dosfstools`

12. Now, you can build project by command: `petalinux-build`

13. Go to `<PROJECT_NAME>/images/linux/`, used the command: `bootgen -arch zynqmp -image bootgen.bif -o BOOT.bin -w` run the **bootgen.bif** file to generate Bootable files (BOOT.bin and zynq_fsbl.elf both in this path).

Note: For Petalinux 2022.1, it is should use **u-boot-dtb.elf**. Conversely, for old version (Petalinux 2020.2), it is should use u-boot.elf. There are diffence for **u-boot-dtb.elf** that is Self-extractable U-Boot elf that has the U-Boot binary with DTB.

14. Please open the **Vitis** software, and go to **Xilinx**, and then open **Progarm Flash**. Inhere:

- Image Files: BOOT.bin files
- Offset: 0x0
- Flash Type: qspi-x8-dual
- Init Files: zynq_fsbl.elf

Finally, click **Program** to flash QSPI bootable files into board

Note: This flash should connect board JTAG cable with PC, and board switcher mode is 1,1 on the ODK2 and . Meanwhile, the ODK3 is diffence switcher setting on the JTAG mode, which is 0,0

15. After flash successful show on the console, you can change board switcher mode to 1,0 for ODK2, and 0,1 forODK3. Then, the QSPI boot will work on the board.

16. In the Series port, you can use petalinux to format EMMC now. If you are missed fdisk and mkfs tools, which is problem for **PATH**. Please try this command to update PATH environment: `export PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin`

17. Format EMMC:

Use `fdisk` to check the EMMC partition, it is should be in the `/dev/mmcblk0`. (Please check your EMMC disk first)

- `sudo fdisk /dev/mmcblk0`
- n to create a new partition, then use p for the primary partition, First cylinder use default, and Last cylinder for requested disk size.
- `w` to write the changes.

18. Now, you can check the partition by `/dev/mmcblk0p1`. (Note, due to we plan for QSPI+EMMC boot, so that EMMC is only used for Rootfs. If you plan only to use EMMC in the future, you can create two partitions on the EMMC, and the second EMMC partition will be `/dev/mmcblk0p2`

19. Format EMMC two partitions to **FAT32** and **Ext4**, which are similar with SD card setting. Boot partition for FAT32 and Rootfs for Ext4. It is controlled by following command: `mkfs.vfat -F32 /dev/mmcblk0p1`, and `mkfs.ext4 /dev/mmcblk0p2`

20. Mount the EMMC into petalinux, command: `mount /dev/mmcblk0p2 /mnt`

Note: For this project, we don't use `mmcblk0p1` partition, due to the Boot files loaded by the QSPI partition.

21. Based on the board's USB port, the board can load the USB storage to copy files. Please using the USB 2.0 device at this stage (Petalinux is limited support for USB drivers). The USB device can be find in the `/dev`path, normally it is show as `sda1`. Mount the USB device into the system by command: `sudo mount /dev/sda1 /tmp`. After mounted USB, you can go to `/tmp` to find your files, and just copy files to EMMC partition. (Such as Ubuntu rootfs files)

Optional Steps for ODK2 (This method not work on the ODK3, due to ODK3 don't designed a Ethernet port): It is can based on the tftp service to transfer files into EMMC, command: `tftp -g -r filename IP`

Note: Direct use ethernet cable to connect the board with Host PC, and then `ifconfig` to confirm the IP address with each eth port, and then try to ping each other. If it is work, please tftp transfer the Rootfs tar.gz file to EMMC. (Inhere, you can use the previous Ubuntu rootfs).

22. Unzip the Rootfs file in the `/mnt`, it is a command for: `tar -zxvf rootfs.tar.gz -C /mnt`

Pure QSPI boot is done, and then if you plan use the EMMC to load rootfs for Ubuntu embedded system, there are new boot files should be built.

23. Please back into petalinux project, and then run: `petalinux-config`

- Go to **Image Packaging configuration** ---> **Root filesystem type** select **SD card**, and then setting the `/dev/mmcblk0p2` in the Device node of SD device.

Note: the partition parameter is should correspond with EMMC rootfs partition.

25. Build again for the petalinux project by command: `petalinux-build`

26. Please repeat the above step 13, 14, 15, which is flash new boot files into board, and the BOOT.bin files will auto find the Rootfs partition with `/dev/mmcblk0p2`.

Note: please careful for the switcher mode, which is only 1,1 state for ODK2 and 0,0 for ODK3 can be flashed with JTAG mode, and then should setting back to 1,0 for ODK2 and 0,1 for ODK3 for QSPI boot mode.

Now, please enjoy the QSPI + EMMC Ubuntu system.

USB 3.0 + USB WIFI and Ethernet for ODK3

There are some steps should be setting and config on the Devicetree and Kernel, and then can use for USB 3.0 function in the ODK3 with Ubuntu (Petalinux is limited support for USB, which is just can used USB 2.0 work mode in the rootfs)

1. Please follow the below devicetree setting for load USB 3.0 and WIFI device, which is can direct replace the devicetree file in the `<PROJECT_NAME>/project-spec/meta-user/recipes-bsp/device-tree/files/system-user.dtsi`

Explanation: ZYNQMP is based on the dwc3.0 as USB 3.0 controller. If missed this configuration, the board USB will only work on USB2.0. And `w lan0 = &usb0;` is USB WIFI support.

2. Kernel driver support setting by `petalinux-config -c kernel`.

Go to `Deivce Drivers`, and find `Network device support`, select `USB Network Adapters`. Due to our USB Hub is used Realtek chiplets, so please open all of Realtek RTL8XXX setting inhere (Such as RTL8150, and RTL8152/8153).

Go to `Deivce Drivers`, and find `Network device support`, select `Wireless LAN`. Same as above, please open all Realtek USB support inhere. Such as 8187, 8188, 8192…, Please go to `Realtek rtlwifi family of devices`, open all Realtek wireless option.

Note: If you are use other chiplets USB device, please check the chip details and find correspond driver in the kernel. Due to petlainux kernel is limited support for the USB device, maybe some of your USB can't direct work on it. But you can install Linux driver into Ubuntu system, after the . ko files install, Ubuntu will support for the USB device without petalinux kernel support. (.ko files will integred into kernel).

Now, please enjoy the USB 3.0 + USB WIFI & Ethernet