一． 前期工作

1. include bitstream生成.xsa文件。
2. 进入虚拟机，预先安装petalinux2019.2。
3. source /petalinux安装目录/settings.sh
4. petalinux-create -t project -n axidma\_axiuartlite\_plnx -- template zynq 生成petalinux工程。
5. cd axidma\_axiuartlite\_plnx
6. 将1中生成的.xsa放入axidma\_axiuartlite\_plnx文件夹。

二． 基础配置

1. petalinux-config --get-hw-description=.，开启基础配置界面。
2. 进入Subsystem AUTO Hardware Settings。
3. 选择Serial Settings，选择ps7\_uart\_1，退出。
4. 选择Ethernet Settings，选择ps7\_ethernet\_1；去勾选Obtain IP address automatically，并配置网口地址192.168.1.16，网关192.168.1.1，退出。
5. 选择SD/SDIO Settings，选择ps7\_sd\_0，保存，2次退出。
6. 选择Image Packaging Configuration，在Root filesystem type选择EXT（SD/eMMC/QSPI/SATA/USB），保证linux文件掉电不丢失（RAM会丢失），2次退出

三． DMA驱动添加

有两种方案添加axidma驱动，传统方案，petalinux方案；目前使用传统方案：

传统方案：

1. 在<https://github.com/Xilinx>处下载linux-xlnx内核，选择Tags=xilinx-v2019.2，并单独分文件夹保存。
2. 参考<https://blog.csdn.net/long_fly/article/details/80482248>，第7节，在<https://github.com/bperez77/xilinx_axidma>处下载master分支的dma驱动源码，单独分文件夹保存。
3. 将dma源码文件夹xilinx-axidma-master拷贝进linux-xlnx-xilinx-v2019.2 文件夹，更名为extra。
4. cd extra
5. make CROSS\_COMPILE=arm-linux-gnueabihf- ARCH=arm KBUILD\_DIR=/内核源码路径/linux-xlnx-xilinx-v2019.2/ driver，编译驱动。
6. make CROSS\_COMPILE=arm-linux-gnueabihf- ARCH=arm examples，编译驱动和示例应用。
7. 拷贝outputs文件夹出来备用。

\*注：驱动所需的编译器在编译后的内核中，应用程序所需的编译器是SDK/gnu/aarch32中

Petalinux方案：

1. petalinux-create -t modules -n xilinx-axidma --enable, 生成axidma模块。
2. 参考<https://blog.csdn.net/qq_39337844/article/details/90238338>，在<https://download.csdn.net/download/qq_39337844/11181780>链接处下载axidma驱动源码，单独分文件夹保存，重命名文件夹为xilinx-axidma。
3. 从axidma\_axiuartlite\_plnx项目文件夹出发，进入./project-spec/meta-user/recipes-modules，将刚刚保存的xilinx-axidma替换该文件夹。
4. 进入xilinx-axidma，修改.bb文件如下并保存

SRC\_URI = "file://Makefile \

file://axi\_dma.c \

file://axidma\_chrdev.c \

file://axidma\_dma.c \

file://axidma\_of.c \

file://axidma.h \

file://axidma\_ioctl.h \

file://COPYING \

"

1. 进入files文件夹，删除xilinx\_axidma.c文件
2. 修改Makefile文件，修改第一行如下

DRIVER\_NAME = xilinx-axidma

$(DRIVER\_NAME)-objs = axi\_dma.o axidma\_chrdev.o axidma\_dma.o axidma\_of.o

obj-m := $(DRIVER\_NAME).o

\*注：若全套使用2019.2的vivado和petalinux工具链，则以上提到的xilinx-axidma驱动源文件都需要进行以下修改：

1. 添加 "#include <linux/of\_address.h>" into driver/axi\_dma.c

2. 在axidma\_chrdev.c, 修改 "of\_dma\_configure(dev->device, NULL,) ;"参数为 "of\_dma\_configure(dev->device, NULL, true) ;"

四． 设备树配置

1. petalinux-config -c device-tree，配置设备树
2. 进入./project-spec/meta-user/recipes-bsp/device-tree/files，修改system-user.dtsi如下并保存。

/include/ "system-conf.dtsi"

/ {

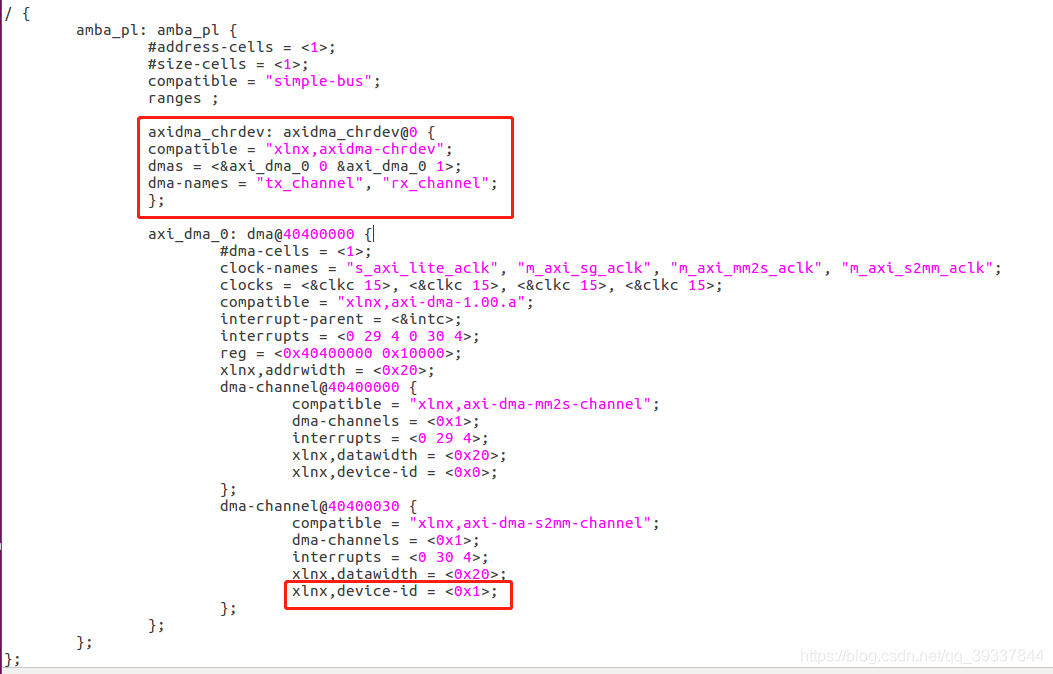
chosen {

bootargs = "console=ttyPS0,115200 root=/dev/mmcblk0p2 rw earlyprintk rootfstype=ext4 rootwait";

};

};

1. 在axidma\_axiuartlite\_plnx文件夹，进入components/plnx\_workspace/device-tree/device-tree-generation，修改pl.dtsi文件如下图，保存。



五． 内核配置

1. petalinux-config -c kernel，编译内核
2. 选择Memory Management Options---->Maximum count of the CMA areas，将默认的7改为25。（DMA原因）
3. 选择Device Drivers---->Generic Driver Options---->Size in Mega Bytes，将默认的16改为25。（DMA原因）
4. 勾选Device Drivers---->Generic Driver Options---->Character devices---->Serial drivers---->Xilinx uartlite serial port support。（axi\_uartlite原因）保存后连续退出。

六． 文件系统配置

1. 选择Filesystem Packages/misc，勾选与qt有关的库

七． 自启动设置

1. 参考<https://blog.csdn.net/WANGYONGZIXUE/article/details/121237131>，使用petalinux-create -t apps --template install -n autorun-init --enable，添加autotun-init应用程序。
2. 从axidma\_axiuartlite\_plnx出发，修改./project-spec/meta-user/recipes-apps/ autorun-init / autorun-init.bb如下，使该应用程序可以开机启动：

#

# this file is the autorun-init recipe.

#

SUMMARY = "Simple autorun-init application"

SECTION = "PETALINUX/apps"

LICENSE = "MIT"

LIC\_FILES\_CHKSUM = "file://${COMMON\_LICENSE\_DIR}/MIT;md5=0835ade698e0bcf8506ecda2f7b4f302"

SRC\_URI = "file://autorun-init \

"

S = "${WORKDIR}"

FILESEXTRAPATHS\_prepend := "${THISDIR}/files:"

inherit update-rc.d

INITSCRIPT\_NAME = "autorun-init"

INITSCRIPT\_PARAMS = "start 99 S ."

do\_install() {

install -d ${D}${sysconfdir}/init.d

install -m 0755 ${S}/autorun-init ${D}${sysconfdir}/init.d/autorun-init

}

FILES\_${PN} += "${sysconfdir}/\*"

1. 修改files中的autorun\_init文件如下，该文件开机启动（开机运行autorun.sh脚本）：

cp /run/media/mmcblk0p1/autorun.sh ~

~/autorun.sh

八． 其他设置

1. 运行petalinux-config，选择Auto Config Settings，去勾选Device tree autoconfig，使第四步的自定义设备树可以被编译。

九． 编译

1. petalinux-build编译
2. petalinux-package --boot --fsbl ./images/linux/zynq\_fsbl.elf --fpga ./images/linux/system\_top.bit --u-boot –force打包编译生成BOOT.bin和image.ub，以及文件系统rootfs.tar.gz，保存在./image/linux中。

十． 收尾工作

1. 一张全新的sd卡，对其进行分盘，BOOT盘格式为FAT32共512M，剩余为文件系统盘格式为ext4。
2. 插入sd卡，显示sd卡对应/dev/sdb，两个盘分别为sdb1，sdb2（视情况会有不同）。
3. 拷贝BOOT.bin，image.ub进入BOOT盘。
4. 拷贝第四节outputs文件夹进入BOOT盘。
5. 自定义auturun.sh文件并放入BOOT盘（可以写加载dma驱动，开机自启动软件等等）。
6. 拷贝需要运行的云高仪程序进入BOOT盘。
7. 运行sudo mkdir –p /media/zxy，生成文件系统盘挂载文件夹。
8. 运行sudo umount /dev/sdb2，解除挂载文件系统盘。
9. 运行sudo mkfs.ext4 /dev/sdb2，格式化文件系统盘为ext4。
10. 运行sudo mount /dev/sdb2 /media/zxy/，挂载文件系统盘。
11. 运行sudo tar -xzvf ./images/linux/rootfs.tar.gz -C /media/zxy/，拷贝文件系统进文件系统盘。

十一． 上板测试

1. 插入sd卡，putty串口监测。
2. 上板后打印信息，用户名密码都是root。
3. 主机ping 192.168.1.16，能ping通说明网口正常，可使用ssh连接板卡。
4. 拷贝outputs文件夹进入~目录。
5. 若第四节采用传统方法生成dma驱动，则insmod ./outputs/axidma.ko；若采用petalinux方法，则insmod /lib/modules/xilinx-v2019.2-xxx/extra/axidma.ko，来加载dma驱动。
6. 若采用环通的PL端FPGA程序，则可以使用outputs中的axidma\_benckmark和axidma\_transfer软件进行dma环通测试。具体测试方法为：

传输16384个字节做环通测试。

./axidma\_benchmarker –b 16384 –s 16384

传输1.txt文件中内容给2.txt

./axidma\_transfer 1.txt 2.txt

1. 运行ls /dev，若能查看到ttyUL1，说明axi\_uartlite外设被正常添加。