sudo nautilus

秘钥认证失败时使用

ssh-keygen -R 192.168.2.8

===运行minicom=========================================================

sudo chmod 777 /dev/ttyUSB0

sudo minicom -s 主界面 ctrl+A

===板子挂载主机NFS目录==================================================

主机

修改nfs服务器配置文件

sudo nano /etc/exports

在该文件最后添加一行： /home/xt500/aaa/nfs7010 \*(rw,sync,no\_root\_squash)

exportfs -r

service rpcbind start

service nfs start

或者/etc/init.d/nfs-kernel-server start

输入"showmount -e"命令可以查看 nfs server 路径

-------------------------------------------------

板子：

手动挂载，需要加参数 -o nolock。例如

mount -t nfs -o nolock 192.168.2.10:/home/xt500/aaa/nfs7010 /mnt

自动挂载：修改启动参数kernel bootargs ，将启动参数改为手动配置，然后增加参数

nfsroot=192.168.2.8:/home/xt500/aaa/nfs7010,proto=tcp,nfsvers=3,nolock ip=192.168.2.8

其中192.168.2.8为HOST端IP，192.168.2.8为TARGET端IP地址

===编译cgminer===========================================================

source /home/xt500/aaa/petalinux/mime7010\_prj/images/linux/sdk/environment-setup-cortexa9t2hf-neon-xilinx-linux-gnueabi

先编译curl

./configure --prefix=/home/xt500/ccc/out --host=arm-xilinx-linux-gnueabi CC=arm-xilinx-linux-gnueabi-gcc

然后编译cgminer

export LIBCURL\_CFLAGS=-I/home/xt500/ccc/out/include/curl

export LIBCURL\_LIBS=-L/home/xt500/ccc/out/lib

./configure --prefix=/home/xt500/aaa/cgminer/out --host=arm-xilinx-linux-gnueabi LIBCURL\_LIBS="-L/home/xt500/ccc/out/lib -lcurl" LIBS=-lz --enable-ants2

编译PetaLinux

source /tools/Xilinx/PetaLinux/2020.2/tool/settings.sh

---------创建工程----------------------------

cd ~/aaa/petalinux

petalinux-create --type project --template zynq --name mime7010\_prj

---------引用硬件信息----------------------------

复制 bit，xsa 到mime7010\_prj

cd ~/aaa/petalinux/mime7010\_prj

petalinux-config --get-hw-description=.

---------第1遍 配置----------------------------

petalinux-config

Subsystem AUTO Hardware Settings --->

Ethernet Settings ---> 改静态ip，成品要改随机MAC

Advanced bootable images storage Settings ---> 不用改，成品根据实际修改

Image Packaging Configuration --->

Root filesystem type (INITRD) --->改为sd

修改后vitis不能联机调试

(/tftpboot) tftpboot directory 改为 /home/xt500/aaa/tftpboot

petalinux-config -c u-boot

Boot media --->

[\*] Support for booting from SD/EMMC

SPL / TPL --->

[\*] Support SHA1 [\*] Support SHA256

petalinux-config -c kernel

Device Drivers ->

I2C support -> i2C Hardware Bus support ->

Xilinx I2C Controller Pressing 启用CONFIG\_I2C\_XILINX=y

+-SPI support--->

<\*> User mode SPI device driver support

Userspace I/O drivers

<\*> Userspace I/O platform driver with generic IRQ handing

<\*> Userspace platform driver with generic irq and dynamic memory

Power management options --->

Suspend to RAM and standby

将 system-user.dtsi 替换为：

/include/ "system-conf.dtsi"

/

&axi\_gpio\_0 {

compatible = "generic-uio";

};

petalinux-config -c rootfs

Filesystem Packages -> base -> i2c-tools -> i2c-tools

Image Features --->

[ ] ssh-server-dropbear │ │

[\*] ssh-server-openssh │ │

[\*] hwcodecs │ │

[ ] package-management │ │

-\*- debug-tweaks │ │

[\*] auto-login

petalinux-build

----安装到SD卡---------------------------

cd ~/aaa/petalinux/mime7010\_prj/images/linux

rm BOOT.BIN

petalinux-package --boot --format BIN --fsbl zynq\_fsbl.elf --fpga system.bit --u-boot

rm -rfd /media/xt500/Fat32\_boot

cp boot.scr /media/xt500/Fat32\_boot

cp BOOT.BIN /media/xt500/Fat32\_boot

cp image.ub /media/xt500/Fat32\_boot

sudo rm -rfd /media/xt500/Ext4\_roofs

cp rootfs.cpio /media/xt500/Ext4\_roofs

cd /media/xt500/Ext4\_roofs

sudo pax -rvf rootfs.cpio

cd ~/aaa/petalinux/mime7010\_prj

----创建APP---------------------------------------------------------------------

petalinux-create -t apps --template c --name myapp-gpio --enable

----编译-SDK-----------------------------------------------------------------

cd ~/aaa/petalinux/mime7010\_prj

petalinux-build --sdk

petalinux-package --sysroot

===============================================================================

创建sdk输出目录，建立boot、image两个子目录，建立linux.bif文件，

拷贝boot.scr,system.bit,zynq\_fsbl.elf到boot，

拷贝image.ub,rootfs.cpio.gz到image，

了 sysroot，在 Vitis 中需要使用它来进行交叉编译。

使用petalinux-boot --jtag --fpga --bitstream petalinux\_wrapper.bit即可通过JTAG下载bit流到板子上

使用petalinux-boot --jtag --prebuilt 3即可把linux镜像下载到板子上。  
使用了：system.dtb u-boot.elf zImage zynq\_fsbl.elf

注意：使用QEMU和JTAG时，要先封装预建镜像，使用如下命令：  
petalinux-package --prebuilt --fpga  
之后，会自动创建pre-built文件夹，并自动添加一些文件。

执行petalinux-boot --qemu --kernel或者petalinux-boot --qemu --prebuilt 3 命令，启动kernel的效果如下图：

三、利用SD卡在硬件上启动petalinux镜像

将文件夹linux/images下的BOOT.BIN 和 image.ub 拷贝到SD卡的FAT32分区中。

设置启动方式为SD卡。

如果启动失败，提示找不到内核镜像。可能是bootcmd环境变量不正确。

四、使用TFTP启动硬件上的PETALINUX镜像

普通文件传输协议（TFTP）

TFTP 启动可节省大量时间，因为比 JTAG 启动快很多，而且不必为内核源中的每次更改都刷新镜像。

①　petalinux-boot --jtag --prebuilt 2

②　输入任意字符，使其停在UBOOT中。如下：

③　print serverip 检查TFTP服务器IP地址

Zynq> print serverip

serverip=192.168.157.129

④　set serverip ; saveenv 设置主机的IP地址

Zynq> set serverip 10.113.214.140 ;saveenv

Saving Environment to FAT...

writing uboot.env

done

1

2

3

4

⑤　run netboot 启动内核

Zynq> run netboot

\*\*\* ERROR: `ipaddr' not set

1

2

针对这个error，使用如下命令：

Zynq> set ipaddr 10.113.214.205

Zynq> run netboot

Using ethernet@e000b000 device

TFTP from server 10.113.214.252; our IP address is 10.113.214.205

Filename 'image.ub'.

Load address: 0x10000000

1

2

3

4

5

6

又会出现下面这个错误，是UBUNTU上的TFTP服务器还没有搭建。

Loading: T T T T T T T T T

TFTP server died; starting again

1

2

————————————————

版权声明：本文为CSDN博主「Ta o」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/weixin\_41922484/article/details/103148396

setenv bootargs 'console=ttyPS0,115200 root=/dev/nfs rw nfsroot=192.168.2.125:/home/mklinux/nfs/zynq\_rootfs ip=192.168.2.116: 192.168.2.125:192.168.2.100:255.255.255.0::eth0:off'　　//通过网络挂载文件系统