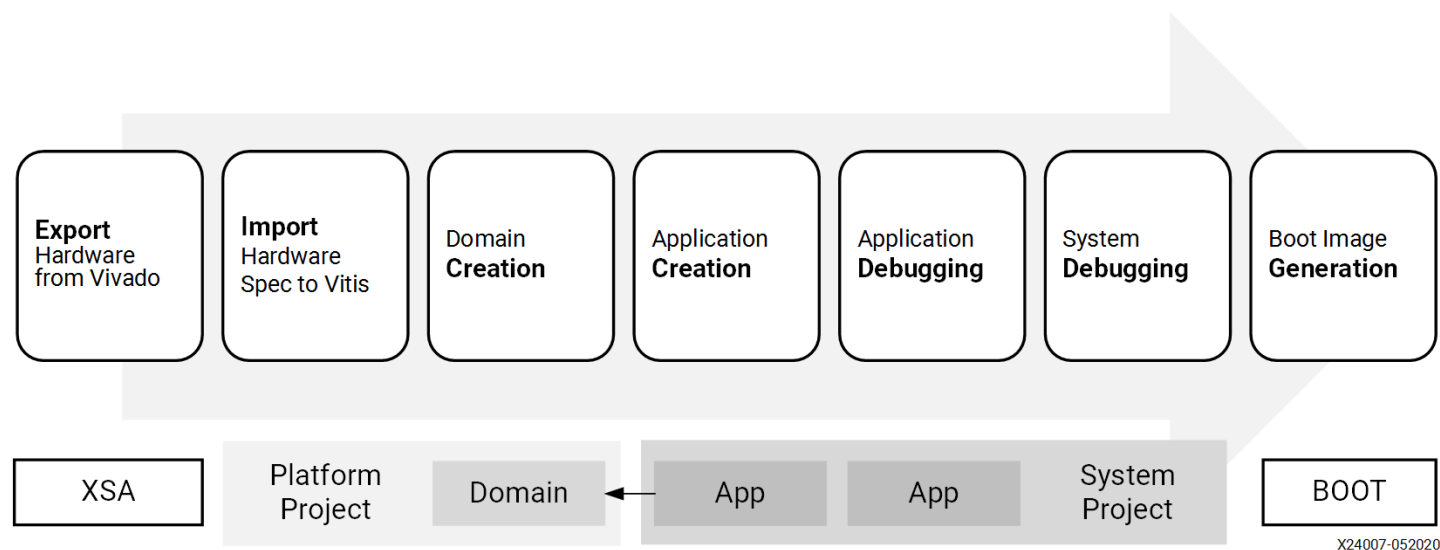


利用vitis快速部署应用到ZCU104

本教程主要介绍利用vitis工具流将加速应用部署到ZCU104上，如图1所示是vitis工具流的概览。



环境准备

安装所需依赖

```
sudo apt-get install ocl-icd-libopencl1 opencl-headers ocl-icd-opencl-dev
sudo add-apt-repository ppa:xorg-edgers/ppa
sudo apt-get update
sudo apt-get install libgl1-mesa-glx
sudo apt-get install libgl1-mesa-dri
sudo apt-get install libgl1-mesa-dev
sudo add-apt-repository --remove ppa:xorg-edgers/ppa
sudo apt install net-tools
sudo apt-get install -y unzip
sudo apt install gcc
sudo apt install g++
sudo apt install python
ln -s /usr/bin/python2 /usr/bin/python
sudo apt install putty
curl -1sLf \
  'https://dl.cloudsmith.io/public/balena/etcher/setup.deb.sh' \
  | sudo -E bash
sudo apt-get update
sudo apt-get install balena-etcher-electron
```

下载安装vitis相关工具

1. [安装vitis软件](#)，这里下载的版本号最好要与之后下载相关内容保持一致。
例如：这里下载了vitis2020.2，后续的软件和镜像最好也下载2020.2或者之前版本以保证兼容性。
2. [安装XRT软件](#)，XRT是Xilinx FPGA的运行时库。
3. 配置环境

```
source /tools/Xilinx/Vitis/2020.2/settings64.sh
source /tools/Xilinx/Vitis_HLS/2020.2/settings64.sh
source /opt/xilinx/xrt/setup.sh
```


4. [下载ZCU104平台描述文件](#)，解压ZCU104平台描述文件，并将其复制到 /opt/xilinx/platforms/ 下。
ps：对于这里使用vitis 2020.2版本的同学来说，请选择2020.1版本的ZCU104平台描述文件。
pss：2020.2版本的平台描述文件没有包含opencl domain，无法使用xrt方式运行并对PL编程。
5. [下载ZYNP平台通用镜像并解压](#)，选择ZYNQMP common image。

Common images for Embedded Vitis platforms - 2020.1

Important Information

The 'common image' packages below contain a prebuilt Linux kernel, root file system and sysroot that can be used with any Zynq or ZynqMP board for embedded Vitis platform developers.

Note: PetaLinux Tools installation is not necessary to use the common images. The licenses and sources used to create the common image content is provided below.

 [ZYNQMP common image](#) (TAR/GZIP - 977.56 MB)

MD5 SUM Value : 8bcf744c9c998f1ec3f9edb7cfd810a6

Download Type
Last Updated
Documentation

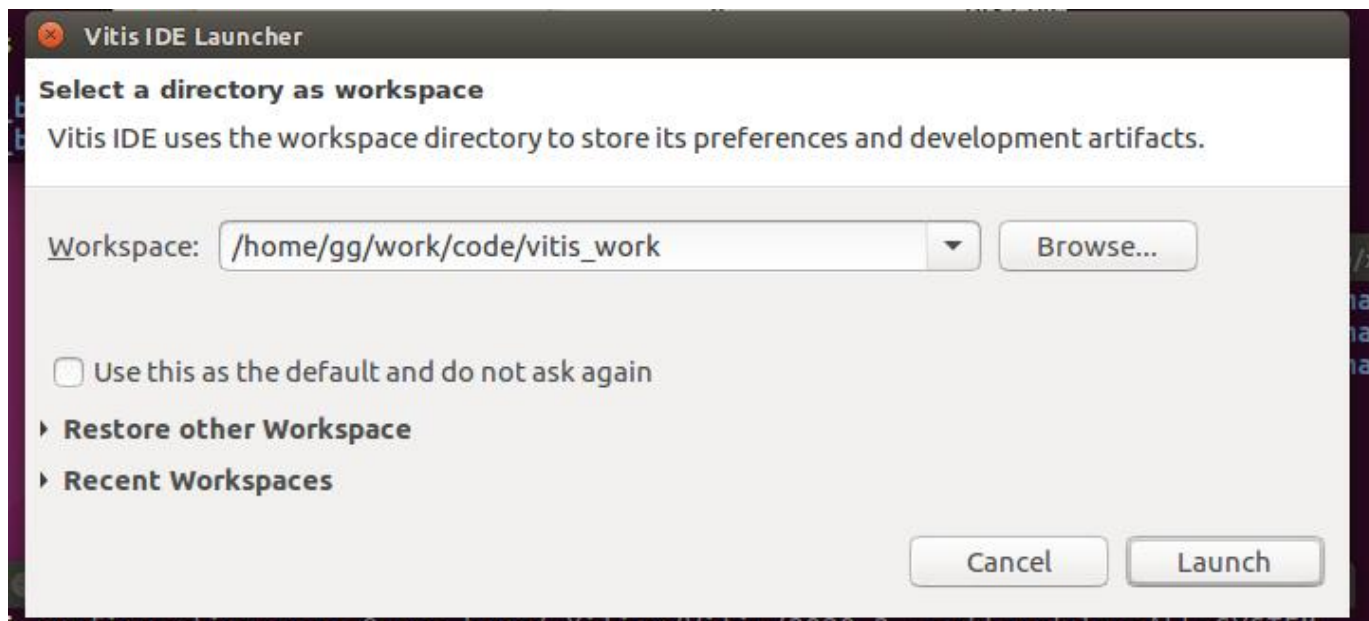
6. ZYNP平台通用镜像展开，经过这个步骤，获得后续进行工程的镜像和文件树等内容。

```
cd xilinx-zynqmp-common-v2020.2/
sudo gunzip ./rootfs.ext4.gz
./sdk.sh -y -dir ./ -p
```

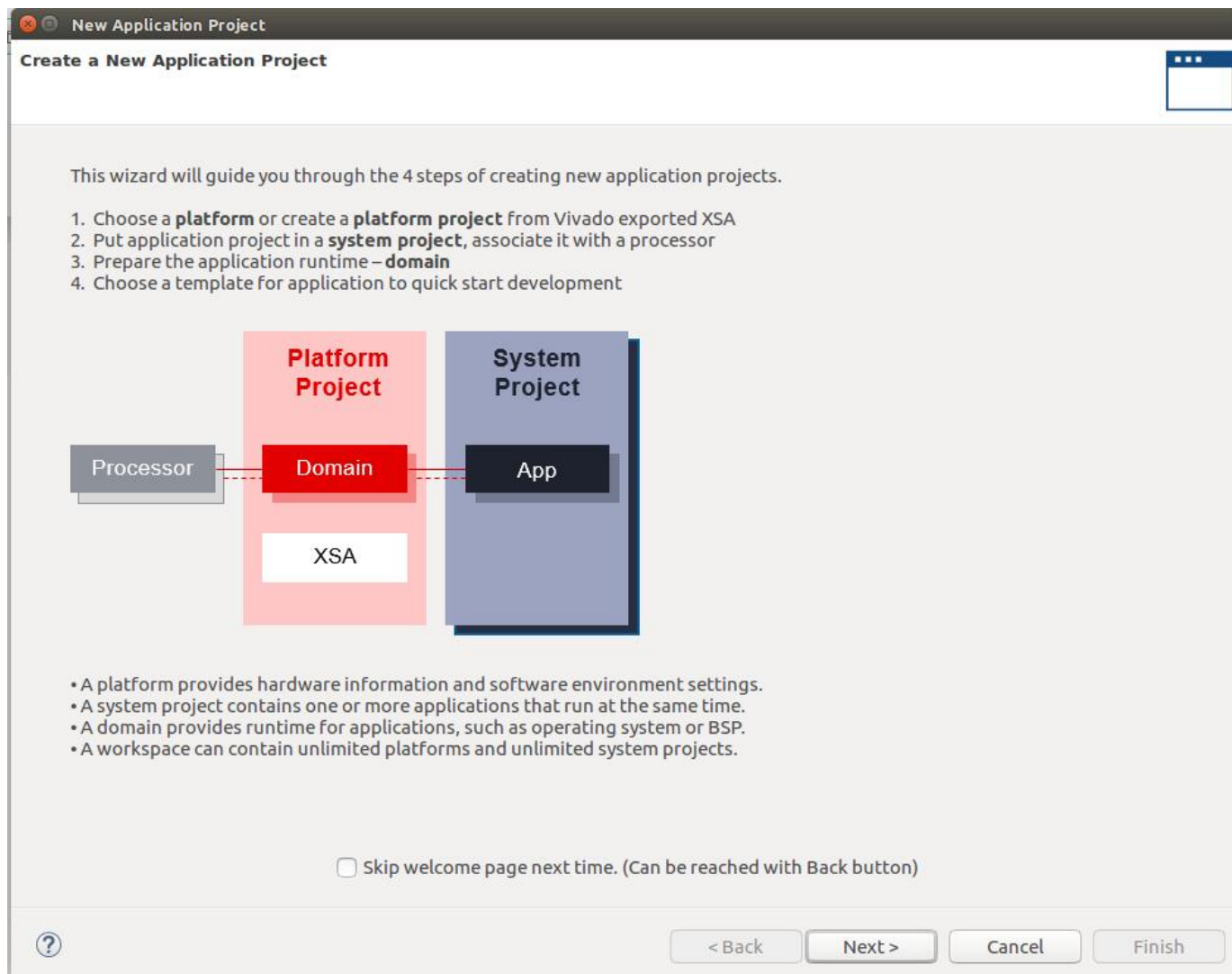
应用工程创建

建立工程

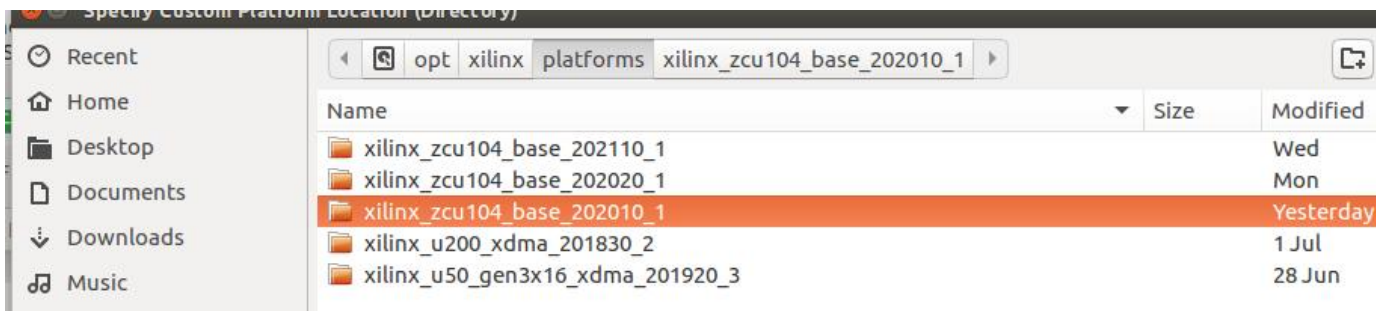
1. 在终端直接运行 `vitis` , 设置工作目录



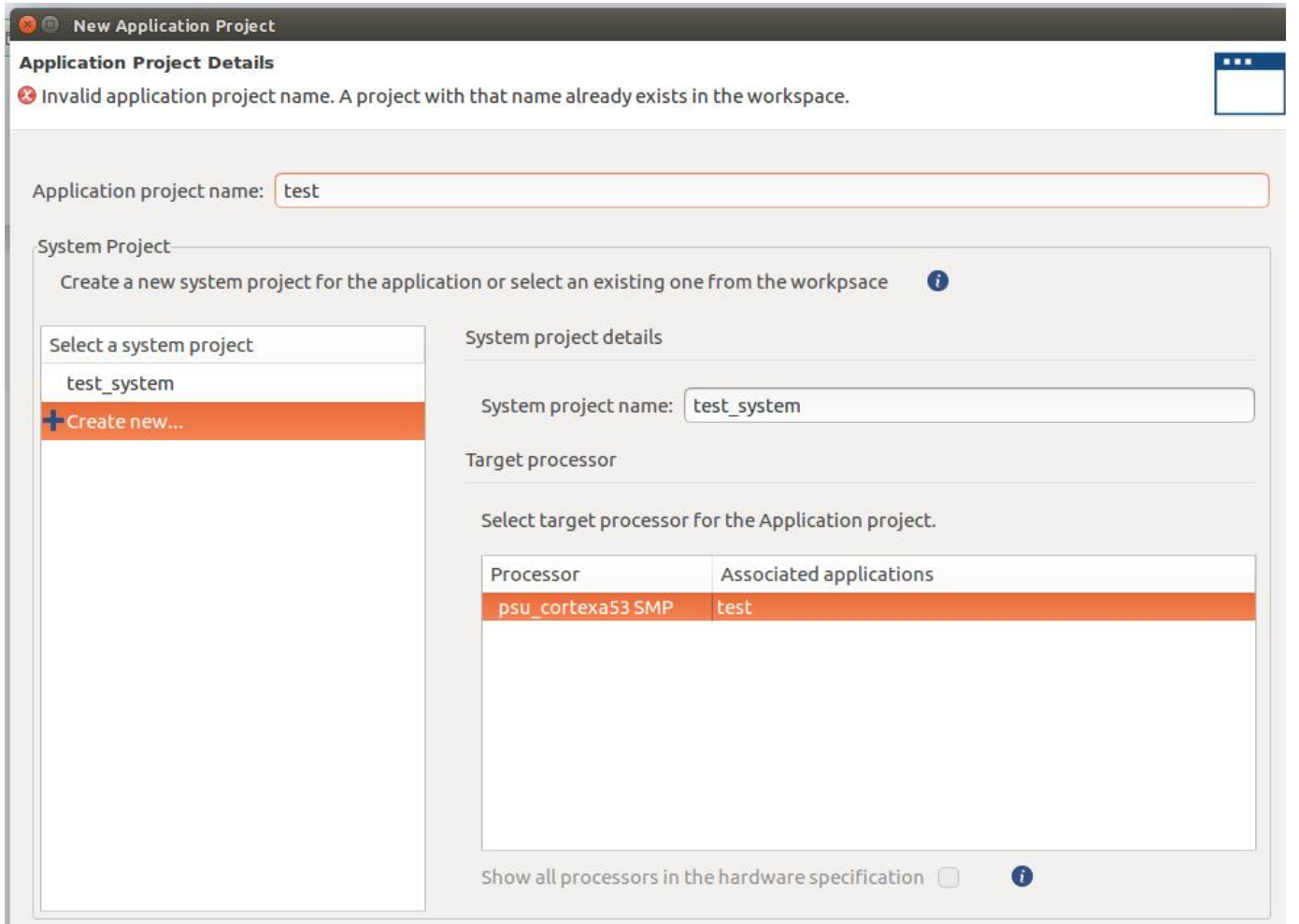
2. 新建应用工程, 并点击next



3. 点击Add添加之前已经下载好的ZCU104平台描述文件



4. 输入应用名称，例如 test



5. 选择镜像文件

- Sysroot -> /ZYNP平台通用镜像路径/xilinx-zynqmp-common-v2020.2/ir/sysroots/aarch64-xilinx-linux
- Root FS -> /ZYNP平台通用镜像路径/xilinx-zynqmp-common-v2020.2/rootfs.ext4
- Kernel Image -> /ZYNP平台通用镜像路径/xilinx-zynqmp-common-v2020.2/Image

Select the domain that the application would link to or create a new domain

Note: New domain created by this wizard will have all the requirements of the application template selected in the next step

Select a domain

xrt

Domain details

Name:

xrt

Display Name:

xrt

Operating System:

linux

Processor:

psu_cortexa53

Application settings

Sysroot path:

/home/gg/work/ZCU_image/xilinx-zynqmp-con

Browse...

Root FS:

/home/gg/work/ZCU_image/xilinx-zynqmp-con

Browse...

Kernel Image:

/home/gg/work/ZCU_image/xilinx-zynqmp-con

Browse...

6. 创建空白应用工程，选择Empty Application后点击Finish

New Application Project

Templates

Select a template to create your project.

Available Templates:

Find:

SW acceleration templates

Empty Application

Vector Addition

SW development templates

Empty Application (C++)

Linux Empty Application

Linux Hello World

Vitis IDE Examples...

Vitis IDE Libraries...

Empty Application

Creates a new Empty application

?

< Back

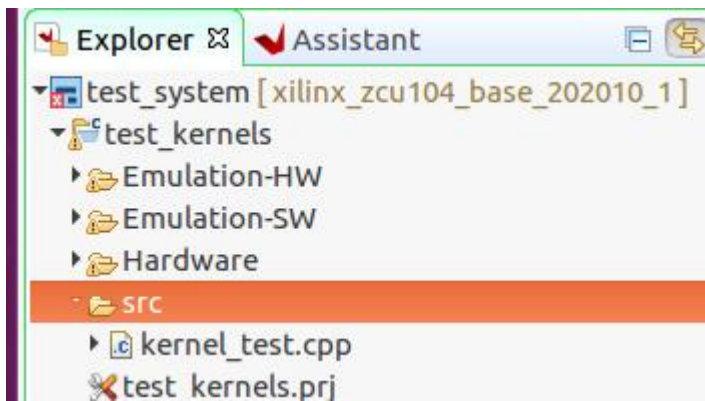
Next >

Cancel

Finish

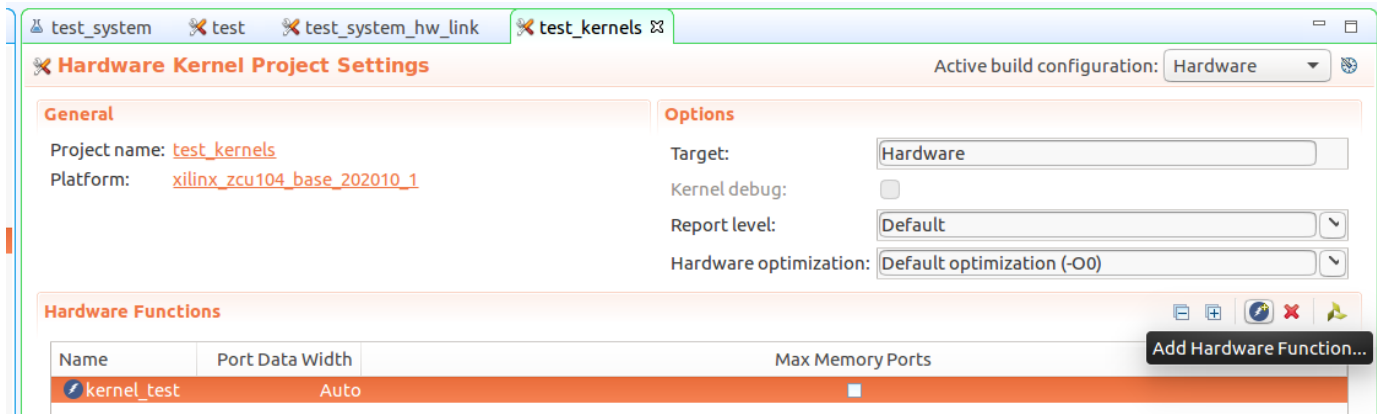
kernel端配置

1. 添加kernel代码，将编写好的kernel代码复制或导入到如图的src文件夹内



2. 打开上图中的test_kernels.prj配置kernel信息

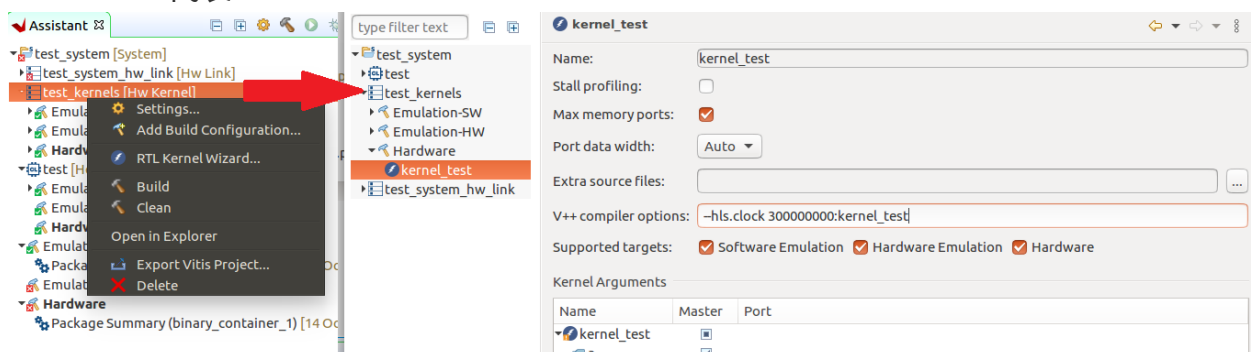
3. 注册kernel函数，点击Add Hardware Fuction，添加硬件单元的TOP函数



4. 可选配置

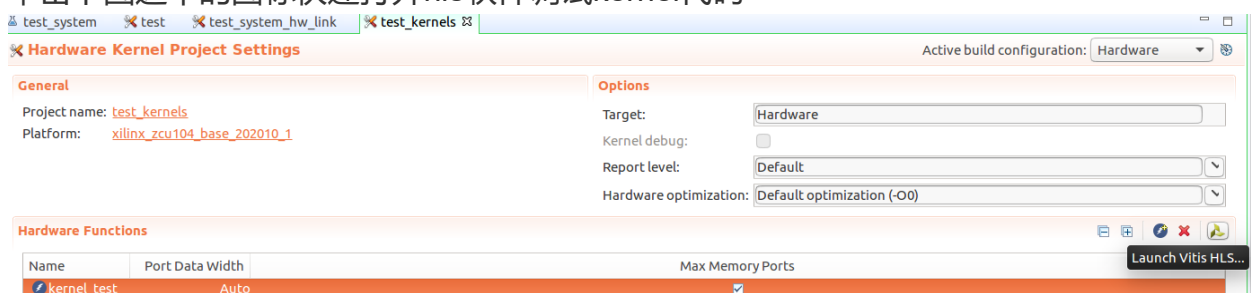
◦ 设置kernel编译时频率约束

- 在Assistant界面右键kernel项目部分
- 单击Settings进入编译设置界面
- 在kernel目录下的Hardware中选中\$YOUR_KERNEL_NAME选项
- 在v++ compiler options中添加 --hls.clock 300000000:\$YOUR_KERNEL_NAME，其中300000000代表300MHz



◦ 关联vitis_hls软件

- 打开之前的test_kernels.prj页面
- 单击下图选中的图标快速打开hls软件调试kernel代码



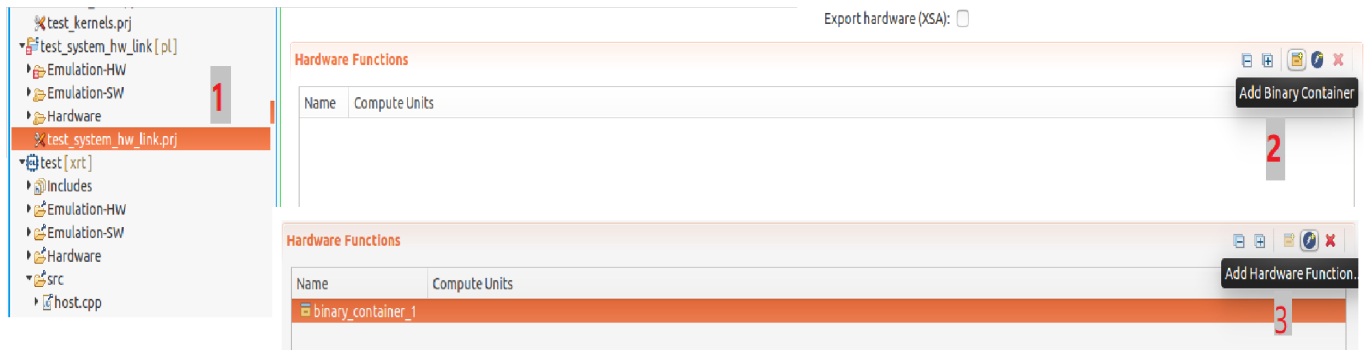
host端配置

1. 添加host代码，将编写好的host代码复制或导入到如图的src文件夹内



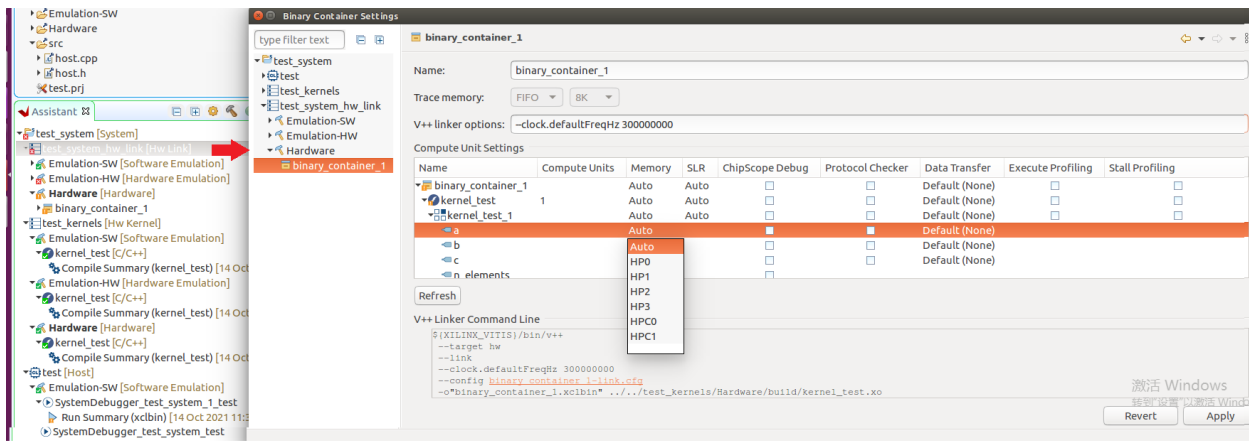
HW-link配置

1. 打开图中的test_system_hw_link.prj配置link信息
2. 点击Add Binary Container创建一个容器
3. 点击ADD Hardware Fuction添加硬件单元的Top Fuction



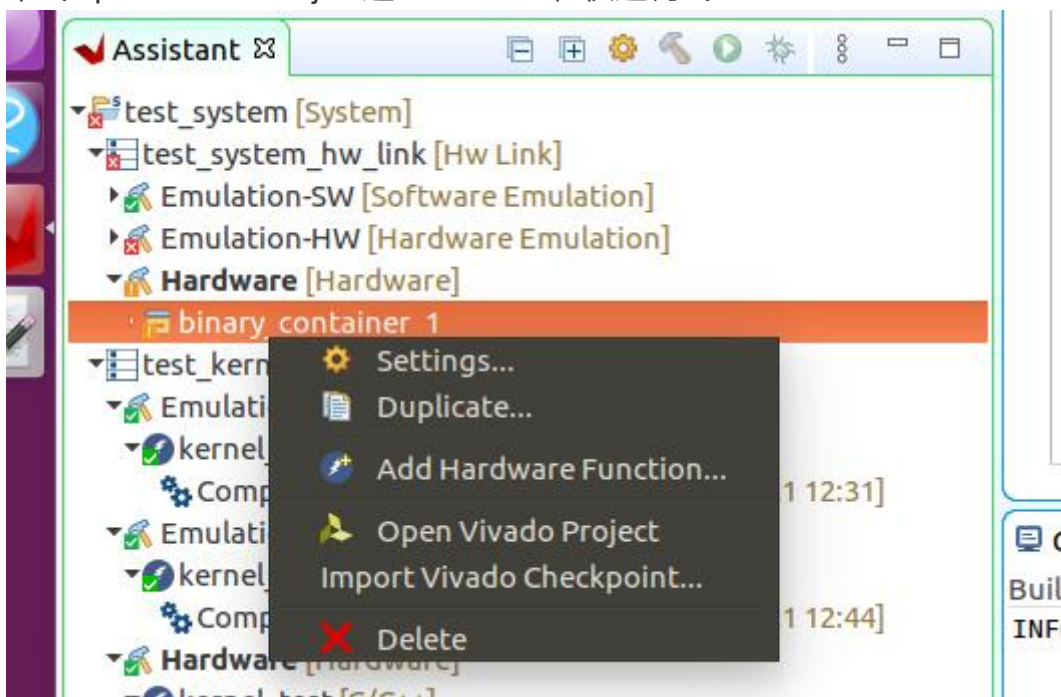
4. 可选配置

- 设置硬件实现频率约束
 - 在Assistant界面右键hw_link项目部分
 - 单击Settings进入编译设置界面
 - 在hw_link目录下的Hardware中选中\$YOUR_CONTAINER_NAME选项
 - 在v++ compiler options中添加 --clock.defaultFreqHz 300000000，其中300000000代表300MHz
- 设置kernel端口映射
 - 在下图中的Memory选项中可以配置kernel的端口映射信息



关联vivado软件

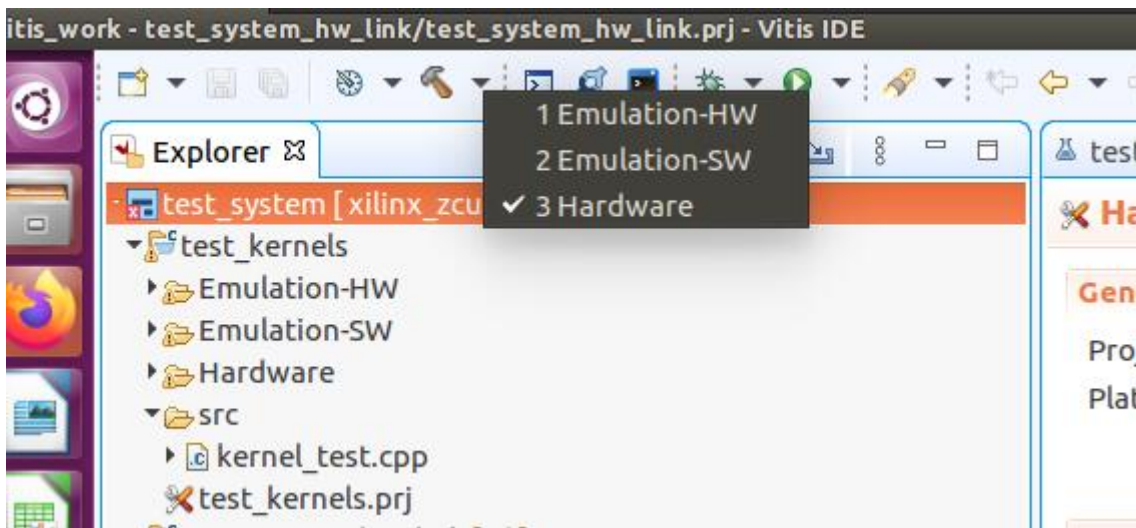
- 在Assistant界面右键hw_link项目下的container部分
- 单击Open Vivado Project进入vivado工程快速调试



应用工程编译

在Explorer界面选中System后，便可在菜单中点击build按钮，其中编译分为三种模式

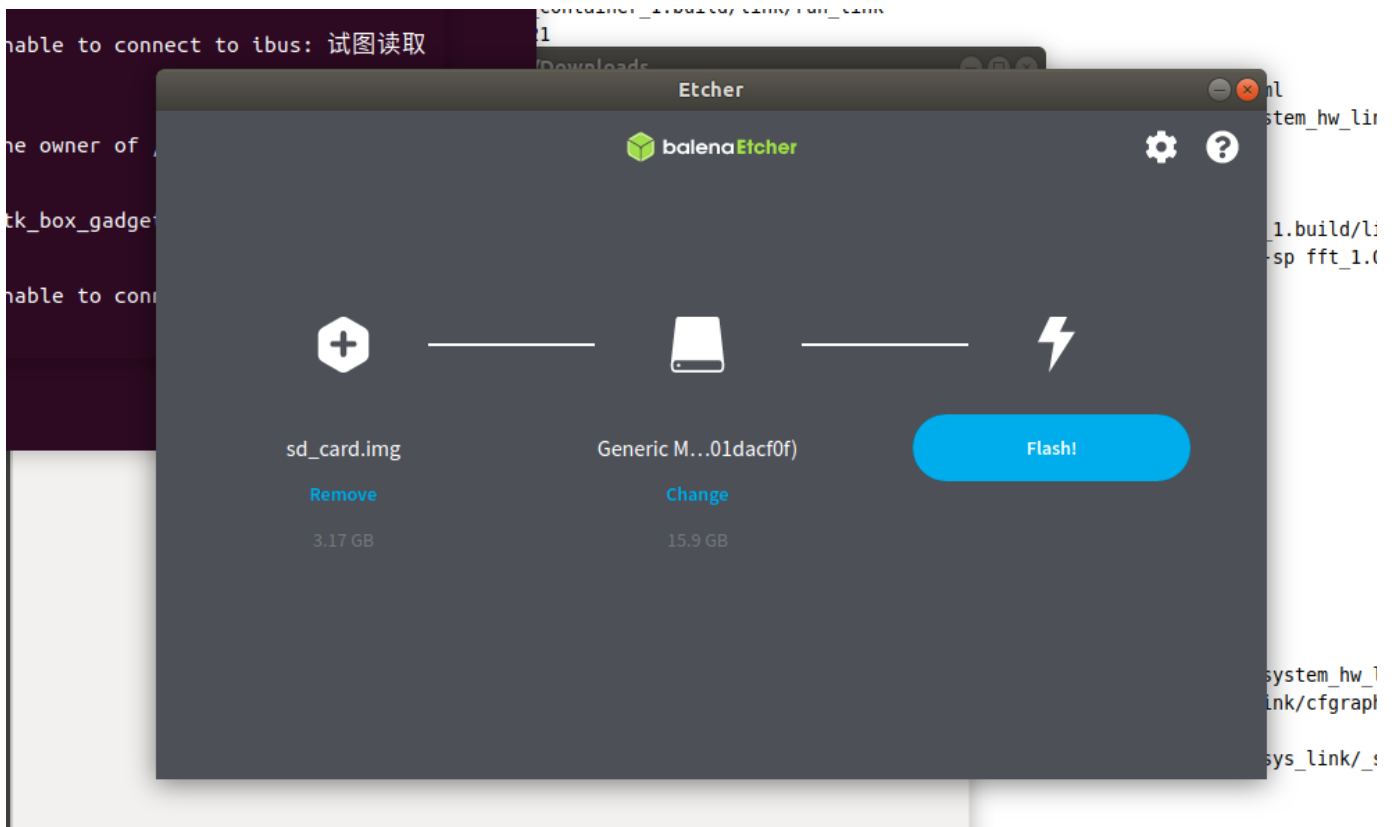
- Emulation-SW：软件仿真，类似于hls的纯软件仿真，主要是用于验证算法的正确性
- Emulation-HW：硬件仿真，仿真真实的硬件连接，用于检查硬件链接问题以及内存访问问题
- Hardware：硬件实现，编译可用于FPGA硬件的工程文件



硬件部署

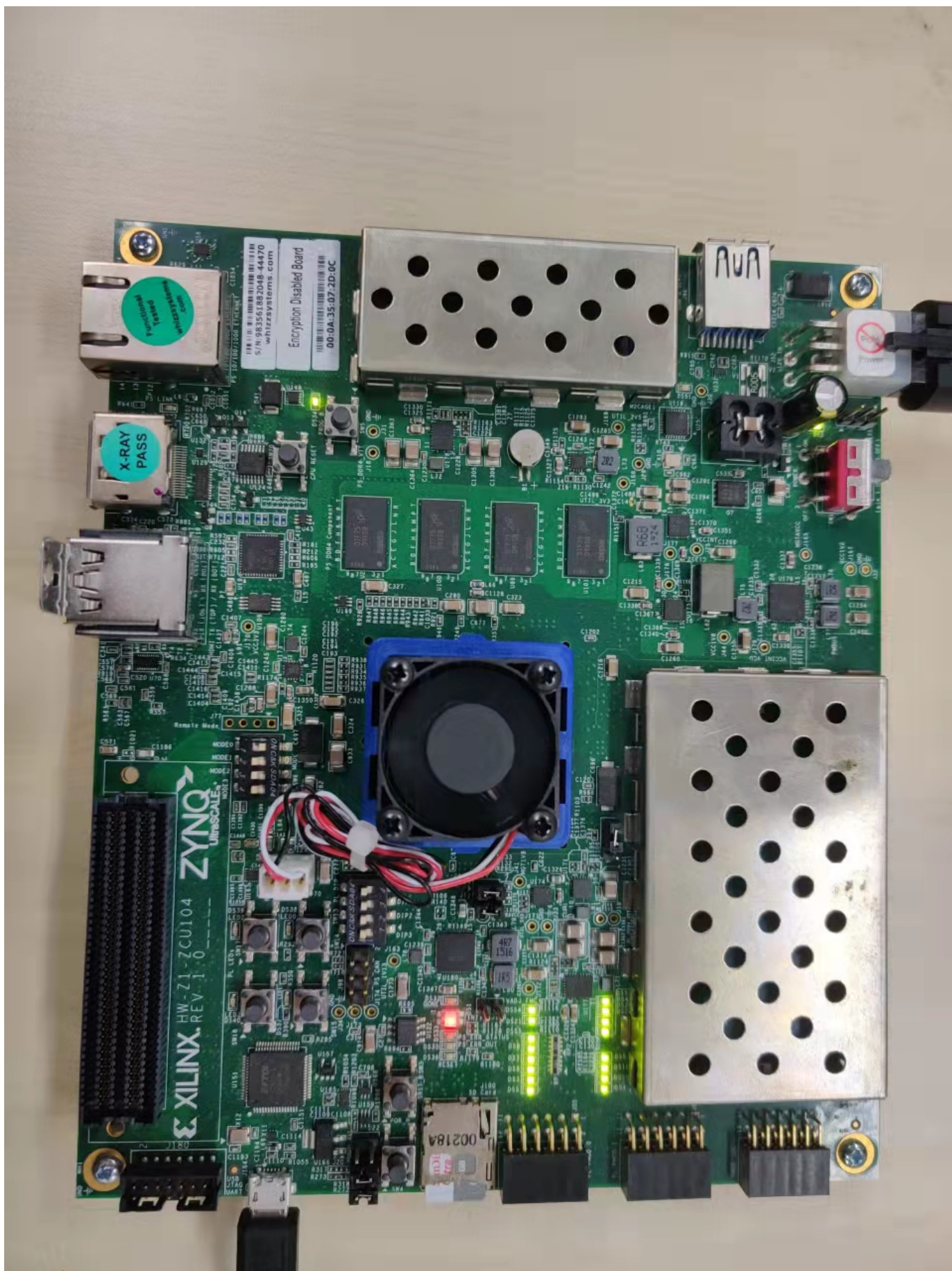
SD卡烧录

1. 将sd卡插到电脑
2. 打开etcher软件
3. 在软件中的image选项里选择， /PATH-to-YOUR-WORK/test_system/Hardware/package 下找到 sd_card.img文件
4. 在device选项里选择sd卡
5. 单击Flash进行烧录

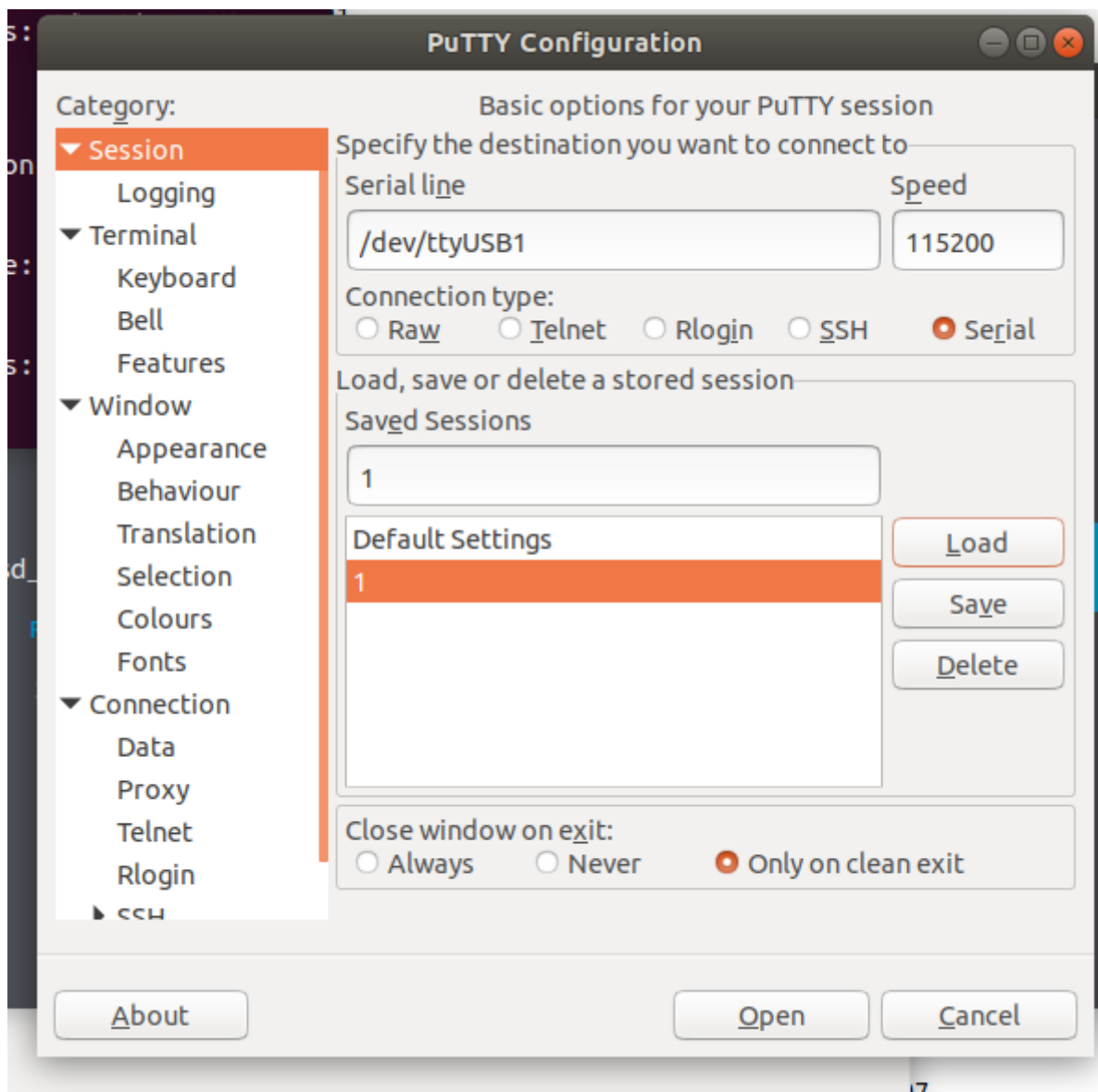


ZCU104板卡串口连接

1. 将ZCU104板卡与主机连接，并插上之前已经烧录好的sd卡



2. 命令行运行 `sudo putty`，打开putty后并如图配置，串口号随实际情况变化，本案例里是 `ttyUSB1`



运行结果

1. 板卡上电运行
2. 启动后运行如下命令

```
cd /mnt/sd-mmcb1k0p1/  
source ./init.sh
```

3. 执行host程序

```

Current version of pixman: 0.38.4
  Before reporting problems, check http://wiki.x.org
  to make sure that you have the latest version.
Markers: (--) probed, (**) from config file, (==) default setting,
  (++) from command line, (!!) notice, (II) informational,
  (WW) warning, (EE) error, (NI) not implemented, (??) unknown.
(==) Log file: "/var/log/Xorg.0.log", Time: Tue Feb  2 07:10:06 2021
(==) Using config file: "/etc/X11/xorg.conf"
(==) Using system config directory "/usr/share/X11/xorg.conf.d"
Starting internet superserver: inetd.
Starting syslogd/klogd: done
Starting tcf-agent: [ 17.288123] random: crng init done
[ 17.291526] random: 7 urandom warning(s) missed due to ratelimiting
OK

PetaLinux 2020.1 zynqmp-common-2020_1 ttyPS0

root@zynqmp-common-2020_1:~# The XKEYBOARD keymap compiler (xkbcomp) reports:
> Warning:      Unsupported high keycode 372 for name <I372> ignored
>              X11 cannot support keycodes above 255.
>              This warning only shows for the first high keycode.
Errors from xkbcomp are not fatal to the X server
D-BUS per-session daemon address is: unix:abstract=/tmp/dbus-k2V9ELhJBQ,guid=cf7
23f6b55e05ee01dee6dc526018fad1
matchbox: Cant find a keycode for keysym 269025056
matchbox: ignoring key shortcut XF86Calendar=!!$contacts

matchbox: Cant find a keycode for keysym 2809
matchbox: ignoring key shortcut telephone=!!$dates

matchbox: Cant find a keycode for keysym 269025050
matchbox: ignoring key shortcut XF86Start=!!matchbox-remote -desktop

dbus-daemon[637]: Activating service name='org.a11y.atspi.Registry' requested by
':1.0' (uid=0 pid=633 comm="matchbox-panel --start-applets showdesktop, windows"
)
dbus-daemon[637]: Successfully activated service 'org.a11y.atspi.Registry'
SpiRegistry daemon is running with well-known name - org.a11y.atspi.Registry
[settings daemon] Forking, run with -n to prevent fork

** (matchbox-desktop:632): WARNING **: 07:10:11.121: Error loading icon: Icon 'a
pplications-multimedia' not present in theme Sato

** (matchbox-desktop:632): WARNING **: 07:10:11.131: Error loading icon: Icon 'a
pplications-multimedia' not present in theme Sato

root@zynqmp-common-2020_1:~# cd /m
media/ mnt/
root@zynqmp-common-2020_1:~# cd /mnt/sd-mmcb1k0p1/
root@zynqmp-common-2020_1:/mnt/sd-mmcb1k0p1# ls
BOOT.BIN  binary_container_1.xclbin  fft      out.gold.dat      system.dtb
Image     boot.scr                  init.sh  platform_desc.txt
root@zynqmp-common-2020_1:/mnt/sd-mmcb1k0p1# source ./init.sh
root@zynqmp-common-2020_1:/mnt/sd-mmcb1k0p1# ./fft
cpu = 64.00.11.1. elapsed = 64.00.10.10 Memory (MB): peak = 1586.918. gain = 0.000. free physical = 1071.

```