以下编译 petalinux 实现 Ubuntu桌面显示

##
0. 准备工作
##

0.1，先按 ug1144（2020.2） 安装petalinux2020.2 以及所需依赖库

https://china.xilinx.com/support/answers/73296.html


```
$ sudo ./plnx-env-setup.sh
```

0.2，下载 Xilinx 本地编译库 sstate-rel-v2020.2.tar.gz， 并将该压缩包先考到Ubuntu Linux 里解压（如果在Windows里直接解压会导致链接文件解压失败 ）后保存在本地Ubuntu目录下，如：

home/zyf/petalinux/sstate/v2020.2/

对 sstate里的文件夹说明如下：

/downloads	-- petalinux 编译所需的文件和工具
/aarch64	-- MPSOC 器件编译库
/arm	-- Zynq 器件编译库（只编译MPSOC器件时可不用）
/mb-xxxx	-- MicroBlaze 器件编译库（只编译MPSOC器件时可不用）
/versal	-- Versal 器件编译库（只编译MPSOC器件时可不用）

##
1. 建立工程
##

1.1，vivado2020.1工程编译（包括bitstream）后，生成 .xsa文件

1.1，在自己的工程目录下 /prj/ 建立 vivado目录，并把

```
# ------------------------------------------------------------------
# 1. 建立工程  prj1
# ------------------------------------------------------------------
```

进入Ubuntu虚拟机，

在当前目录下 建立 vivado目录，并把.xsa文件 考入该目录中

在终端输入命令：

petalinux-create --type project --template zynqMP --name my_bsp

或

petalinux-create --type project -s xilinx-zcu104-v2020.2-final.bsp --name my_bsp

----
2. 配置工程
----
cd my_bsp
petalinux-config --get-hw-description=../vivado

-- 设置 boot 和 kernel 启动位置：

在选项 **Subsystem AUTO Hardware Settings ---> Advanced bootable images storage Settings ---> boot image settings ---> image storage media** 选择 **(primary sd)**

kernel image settings ---> image storage media 选择 **(primary sd)**

-- 设置开发板型号：

在选项 **DTG Settings ---> MACHINE_NAME** 填入 zcu104-revc 或 zcu102-rev1.0

-- 设置 Kernel Bootargs 参数值

在选项 **DTG Settings ---> Kernel Bootargs ---> [ ] generate boot args automatically** 取消使用，选择手动设置 **user set kernel bootargs (NEW)**

按回车键进入后，填入以下内容：

earlycon console=ttyPS0,115200 clk_ignore_unused cma=1024M root=/dev/mmcblk0p2 rw earlyprintk uio_pdrv_genirq.of_id=xlnx,generic-uio cpuidle.off=1 cpufreq.off=1 zynqmp_dpsub.power_on_delay_ms=40 rootwait

-- 在 **UBOOT_CONFIG_TARGET** 设置项里，填入 xilinx_zynqmp_virt_defconfig

-- 设置 从SD卡上启动Linux根文件系统

在选项 **Image Packaging Configuration ---> Root filesystem type (INITRAMFS) --->** 用空格键确定选择 **EX4 (SD/EMMC/USB/SATA)** 配置项，按回车退出

在下一项中确认 SD卡的设备节点 mmcblk0p2 和Bootargs参数设置保持一致：

(/dev/mmcblk0p2) Device node of SD device (NEW)

-- 在 **Yocto Setting ---> YOCTO_MACHINE_NAME** 设置项里，填入 zcu104-zynqmp 或 zcu102-zynqmp

==========

3，设置PetaLinux 从本地库 Sstate 进行编译

1）设置镜像文件的本地下载路径（即PetaLinux编译时需要的常用工具）

Yocto Settings ---> Add pre-mirror url ---> 回车进入

删除原有路径，添加自己 sstate 路径，示例如下：

file:///home/zyf/petalinux/v2020.2/sstate/downloads

默认配置：(http://petalinux.xilinx.com/sswreleases/rel-v${PETALINUX_VER%%.*}/downloads) pre-mirror url path

按回车确认并退出，再返回到 Yocto Settings

2）设置编译库的本地下载路径（即PetaLinux编译时所需要的库文件）

Yocto Settings ---> Local sstate feeds settings ---> () local sstate feeds url ---> 回车进入

添加自己 sstate 路径，示例如下：

file:///home/zyf/petalinux/v2020.2/sstate/aarch64

按回车确认并退出，再返回到 Yocto Settings

3）禁止从网络下载编译库

在 **Yocto Settings** 按上下键移动到 **[*] Enable Network sstate feeds**

按空格键禁用该配置项，显示为： **[ ] Enable Network sstate feeds**

默认配置：(http://petalinux.xilinx.com/sswreleases/rel-v${PETALINUX_VER%%.*}/aarch64/sstate-cache) network sstate feeds url

4）使用非网络的BB (BitBake)来编译

在 **Yocto Settings** 按上下键移动到 **[ ] Enable BB NO NETWORK**

按空格键使能该配置项，显示为： **[*] Enable BB NO NETWORK**

5） 到此配置结束，保存后直接退出 menuconfig，等待配置运行结束，显示如下信息：

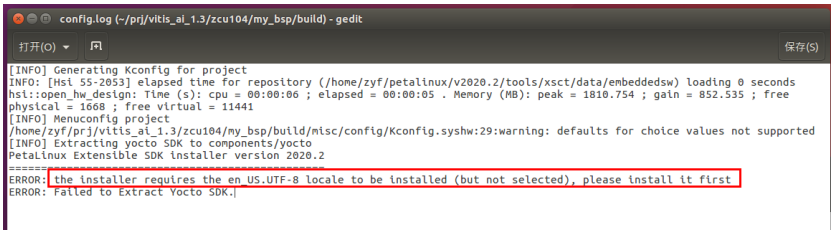
```
|-------------------------------------------------------------------------------------------------------------------------------
-----------------------------
|     *** End of the configuration.
|     *** Execute 'make' to start the build or try 'make help'.
|
|     [INFO] sourcing bitbake
|     [INFO] generating plnxtool conf
|     [INFO] generating meta-plnx-generated layer
|     [INFO] generating user layers
|     [INFO] generating machine configuration
|     [INFO] generating bbappends for project . This may take time !
|     [INFO] generating u-boot configuration files
|     [INFO] generating kernel configuration files
|     [INFO] generating kconfig for Rootfs
|     [INFO] silentconfig rootfs
|     [INFO] generating petalinux-user-image.bb
|-------------------------------------------------------------------------------------------------------------------------------
-----------------------------
```

```
zyf@zyf-virtual-machine:~/prj0$
zyf@zyf-virtual-machine:~/prj0$ petalinux-create --type project -s xilinx-zcu104-v2020.2-final.bsp  --name bsp
INFO: Create project: bsp
INFO: New project successfully created in /home/zyf/prj0/bsp
zyf@zyf-virtual-machine:~/prj0$
zyf@zyf-virtual-machine:~/prj0$
zyf@zyf-virtual-machine:~/prj0$
zyf@zyf-virtual-machine:~/prj0$ cd bsp
zyf@zyf-virtual-machine:~/prj0/bsp$
zyf@zyf-virtual-machine:~/prj0/bsp$
zyf@zyf-virtual-machine:~/prj0/bsp$ petalinux-config --get-hw-description=../vivado
INFO: Sourcing build tools
INFO: Getting hardware description...
INFO: Rename zcu104_dpu.xsa to system.xsa
[INFO] Generating Kconfig for project
[INFO] Menuconfig project
configuration written to /home/zyf/prj0/bsp/project-spec/configs/config

*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.

[INFO] Extracting yocto SDK to components/yocto
[INFO] Sourcing build environment
[INFO] Generating kconfig for Rootfs
[INFO] Silentconfig rootfs
[INFO] Generating plnxtool conf
[INFO] Adding user layers
[INFO] Generating workspace directory
zyf@zyf-virtual-machine:~/prj0/bsp$
zyf@zyf-virtual-machine:~/prj0/bsp$
zyf@zyf-virtual-machine:~/prj0/bsp$
zyf@zyf-virtual-machine:~/prj0/bsp$
zyf@zyf-virtual-machine:~/prj0/bsp$
```

注： 如果显示 ERROR: Failed to Extract Yocto SDK.

则查看 bulid 目录下的 config.log 文件

得知原因是



saw

6）打开工程中的文件：**./build/conf/plnxtool.conf** 验证是否成功修改了sstate-cache的目标地址。

该文件相关sstate内容如下所示：

//

#Add Pre-mirrors of tool

SOURCE_MIRROR_URL = "file:///opt/pkg/petalinux/2019.1/components/yocto/downloads/"

#Add Pre-mirrors from petalinux-config

PREMIRRORS = " git://.*/.* file:///home/zyf/petalinux-sstate-cache/rel-v2019.1/downloads \n \

ftp://.*/.* file:///home/zyf/petalinux-sstate-cache/rel-v2019.1/downloads \n \

http://.*/.* file:///home/zyf/petalinux-sstate-cache/rel-v2019.1/downloads \n \

https://.*/.* file:///home/zyf/petalinux-sstate-cache/rel-v2019.1/downloads \n"

#Sttate mirror settings

SSTATE_MIRRORS = "file://universal/(.*) file:///opt/pkg/petalinux/2019.1/components/yocto/source/aarch64/sstate-cache/universal-4.8/\1\n"

SSTATE_MIRRORS .= "\nfile://.* file:///opt/pkg/petalinux/2019.1/components/yocto/source/aarch64/sstate-cache/PATH\n"

SSTATE_MIRRORS_append = " \

file://.* file:///home/zyf/petalinux-sstate-cache/rel-v2019.1/aarch64/PATH \n \

\

"

SIGGEN_UNLOCKED_RECIPES += "arm-trusted-firmware busybox init-ifupdown python3"

UNINATIVE_DLDIR = "/opt/pkg/petalinux/2019.1/components/yocto/source/aarch64/downloads/uninative"

BB_NO_NETWORK = "1"

//

6） 在完成后，重用SState离线下载文件

在 **dpu_bsp / project-spec / meta-user / conf /** 路径下编辑 **petalinuxbsp.conf** 文件，添加如下内容：

//

PREMIRRORS_prepend = " \

git://.*/.* file:///home/zyf/petalinux/v2020.2/sstate/downloads \n \

gitsm://.*/.* file:///home/zyf/petalinux/v2020.2/sstate/downloads \n \

ftp://.*/.* file:///home/zyf/petalinux/v2020.2/sstate/downloads \n \
http://.*/.* file:///home/zyf/petalinux/v2020.2/sstate/downloads \n \
https://.*/.* file:///home/zyf/petalinux/v2020.2/sstate/downloads \n "

或者
PREMIRRORS_prepend = " git://.*/.* file:///home/zyf/petalinux/v2020.2/sstate/downloads \n \
gitsm://.*/.* file:///home/zyf/petalinux/v2020.2/sstate/downloads \n \
ftp://.*/.* file:///home/zyf/petalinux/v2020.2/sstate/downloads \n \
http://.*/.* file:///home/zyf/petalinux/v2020.2/sstate/downloads \n \
https://.*/.* file:///home/zyf/petalinux/v2020.2/sstate/downloads \n"

//

petalinux-build -c bootloader -x distclean

petalinux-build -x mrproper -f

petalinux-config -c kernel

在GUI界面中配置Kernel操作

#1，Disable -- Initramfs（关键配置项），PMIC，PCI：

@ ---> General setup ---> Initial RAM file system and RAM disk (initramfs/initrd) support 手动 Disable 设置（关键配置项）

@ ---> Device Drivers ---> PCI support' This needs to be disabled for this version 默认 Enable 设置

#2，Enable -- Tracer，Input device，Multimedia，Sound card，USB，PHY：

Device Drivers ---> Input device support ---> Mouse interface' 手动 Enable 设置 <*> Mouse interface

Kernel hacking ---> Tracers ---> Kernel Function Tracer 手动 Enable 设置

Device Drivers ---> Input device support ---> Event interface' 默认 Enable 设置

Device Drivers ---> Input device support ---> Keyboards' 默认 Enable 设置

Device Drivers ---> Multimedia support ---> Media USB Adapters ---> USB Video Class (UVC) 默认 Enable 设置

Device Drivers ---> Multimedia support ---> Cameras/video grabbers support' 默认 Enable 设置

Device Drivers ---> Multimedia support ---> V4L platform devices
默认 Enable 设置

Device Drivers ---> HID support ---> Generic HID driver
默认 Enable 设置

Device Drivers ---> HID support ---> USB HID support ---> USB HID transport layer 默认 Enable 设置

Device Drivers ---> Sound card support ---> Advanced Linux Sound Architecture' enabling ALSA support 默认 Enable 设置

Device Drivers ---> PHY Subsystem ---> Xilinx ZynqMP PHY driver'
默认 Enable 设置

Device Drivers ---> USB support and enable all required classes
默认 Enable 设置

Device Drivers ---> Hardware Monitoring support ---> PMBus support ---> Maxim MAX20751'
默认 Disable 设置

-- It's would cause CPU IDLE when JTAG is connected. So it is recommended to disable the selection.

b) Ensure the following are ***TURNED OFF*** by entering '**n**' in the [ ] menu selection for:

CPU Power Management --> CPU Idle --> CPU idle PM support
默认 Enable 设置

CPU Power Management --> CPU Frequency scaling --> CPU Frequency scaling
默认 Enable 设置

petalinux-config -c rootfs

Filesystem Packages ---> libs ---> libmali-xlnx => [*] libmali-xlnx | [*] libmali-xlnx-dbg | [*] libmali-xlnx-dev

Filesystem Packages ---> misc ---> hicolor-icon-theme => [*] hicolor-icon-theme | [*] hicolor-icon-theme-dev

Filesystem Packages ---> misc ---> hdmi-module => [*] kernel-module-hdmi

Petalinux Package Groups ---> packagegroup-petalinux-matchbox => [*] packagegroup-petalinux-matchbox 默认配置

packagegroup-petalinux-v4lutils => [*] packagegroup-petalinux-v4lutils 默认配置

packagegroup-petalinux-x11 => [*] packagegroup-petalinux-x11
默认配置

packagegroup-petalinux-opencv => [*] packagegroup-petalinux-opencv **手动配置**

在根菜单配置 DPU 编译

apps ---> [*] autostart 默认配置

modules ---> [*] dpu 默认配置

user packages ---> [*] dnndk 默认配置

```
&amba {
...
dpu {
compatible = "xilinx,dpu";
base-addr = <0x8f000000>;//CHANGE THIS ACCORDING TO YOUR
DESIGN
dpucore {
compatible = "xilinx,dpucore";
interrupt-parent = <&intc>;
interrupts = <0x0 106 0x1 0x0 107 0x1>;
core-num = <0x2>;
};
};
softmax {
compatible = "xilinx, smfc";
interrupt-parent = <&intc>;
interrupts = <0x0 110 0x1>;
core-num = <0x1>;
....
}
```

petalinux-config --silentconfig

Add below lines in ***build/conf/local.conf***

EXTRA_IMAGE_FEATURES += "package-management"

---

4. 编译工程

---

petalinux-build -x distclean

petalinux-build

WARNING: /home/zyf/prj0/bsp/components/yocto/layers/meta-xilinx/meta-xilinx-bsp/recipes-kernel/linux/linux-xlnx_2020.2.bb:do_compile is tainted from a forced run | ETA: 0:00:03

5.制作BOOT.bin

---

cd image/linux/

petalinux-package --boot --fsbl zynqmp_fsbl.elf --u-boot u-boot.elf --pmufw pmufw.elf --fpga system.bit --atf bl31.elf --force

---

最后的SD卡启动文件

将 boot.scr，BOOT.bin，image.ub （包含了 system.dtb, urootfs.cpio.gz, devicetree）考到SD卡 Boot 第一个分区 FAT32

将 petalinux 编译所生成的根文件系统解压到 第二个分区 EXT4
sudo tar xvf rootfs.tar.gz -C /media/zyf/Rootfs

如果tar打包里有根目录，则去除根目录的命令为：
$ sudo tar xvf rootfs.tar.gz --strip-components=1 -C /media/zyf/rootfs

---
如果要备份SD卡原有根文件系统，可通过 tar 命令 打包
进入SD卡rootfs分区，在终端敲入命令：
sudo tar -zvcf /home/zyf/bckup/rootfs.tar.gz ./
[打包后的文件存放的位置]