Petalinux 编译镜像文件

1. Petalinux 集成式.

2. Petalinux 分步式

PS
- fsbl
- uboot
- 设备树.
- linux 内核.
- 根文件系统

常改

PL bit文件.

BOOT.BIN 中包含了 .bit
image.ub
- 内核 zImage
- 设备树 dtb.

根文件系统: → SD卡中 ext4分区.

.xsa文件: (2019.2之前为.hdf)
比特流文件.

```
sptl                                    //设置 petalinux 工作环境
petalinux-create -t project --template zynq -n ALIENTEK-ZYNQ-driver  //创建 petalinux 工程
cd ALIENTEK-ZYNQ-driver                 //进入 petalinux 工程目录下
petalinux-config --get-hw-description /mnt/hgfs/share18/xsa/Navigator_7010/  //导入 xsa 文件
```

注：后面更新 vivado 工程的时候，只需要从 vivado 工程中导出 xsa 文件，并在该 Petalinux 工程下使用“petalinux-config --get-hw-description <xsa 文件路径>”命令重新配置即可。

petalinux 工程配置窗口:

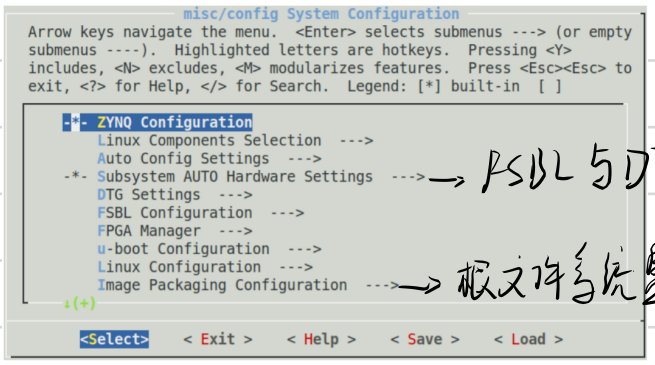

图 11.1.2 petalinux 工程配置窗口

fsbl, uboot 与 BOOT.BIN生成:

11.1.1 生成 BOOT.BIN

运行如下命令编译 fsbl 和 uboot：

```
petalinux-build -c bootloader
petalinux-build -c u-boot
```

然后执行下面命令生成 BOOT.bin：

```
petalinux-package --boot --fsbl --u-boot --dtb no --force
```

注意，petalinux 工具在打包的时候会自动把 dtb 文件加进入，所以这里加入“--dtb no”参数把 dtb 文件除去。

执行结果如下图所示：

BOOT.BIN 包含
- zynq_fsbl.elf
- u-boot.elf

生成 boot.scr:
- petalinux-build
- 修改.
- 加入 system.bit 的加载与启动命令

设备树:

编译 uboot 后生成:

在 Petalinux 工程中执行编译 uboot 后，会在工程的“components/plnx_workspace/device-tree/device-tree/”目录下生成设备树文件，如下图所示：

```
zy@zy-virtual-machine:~/petalinux/Navigator_7010_v3/ALIENTEK-ZYNQ-driver/components/plnx_workspace/device-tree/device-tree$ ls
device-tree.mss           ps7_init.c       skeleton.dtsi
drivers                   ps7_init_gpl.c   system-conf.dtsi
hardware_description.xsa  ps7_init_gpl.h   system-top.dts
include                   ps7_init.h       system_wrapper.bit
pcw.dtsi                  ps7_init.html    zynq-7000.dtsi
pl.dtsi                   ps7_init.tcl
```

图 11.2.1 设备树文件

← 由 .xsa 生成.

Linux内核:

内核源码: 5.4.0

本小节不使用前面移植好的 Linux 内核源码，我们用一份新的 Xilinx 官方 2020.2 版本内核源码（这个版本是 Xilinx 设定的版本，其 Linux 版本为 5.4.0），源码已经提供给大家了，路径为：开发板资料盘(A 盘)\4_SourceCode\3_Embedded_Linux\资源文件\Kernel\ linux-xlnx-xlnx_rebase_v5.4_2020.2.tar.gz 。大家也可以通过 https://github.com/Xilinx/linux-xlnx 网址进行下载，如下图所示：

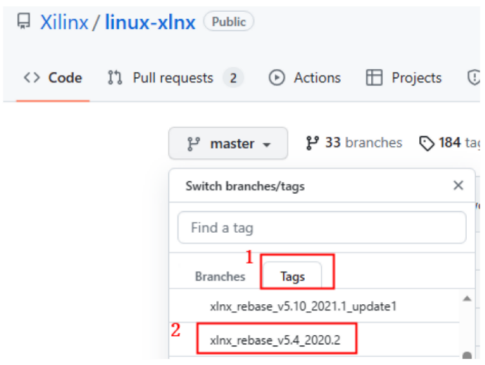

图 11.3.1 下载内核源码

将 11.2 小节中 pcw.dtsi，pl.dtsi，system-top.dts，zynq-7000.dtsi 和 system-conf.dtsi 五个设备树文件直接复制到内核源码“arch/arm/boot/dts”目录下，如下图所示：

```
zy@zy-virtual-machine:~/petalinux/Navigator_7010_v3/ALIENTEK-ZYNQ-driver/components/plnx_workspace/device-tree/device-tree$ cp pcw.dtsi pl.dtsi system-top.dts zynq-7000.dtsi system-conf.dtsi /home/zy/worksapce/kernel-driver/linux-xlnx-xlnx_rebase_v5.4_2020.2/arch/arm/boot/dts/
```

图 11.3.3 添加设备树文件

此外还要将工程“project-spec/meta-user/recipes-bsp/device-tree/files”目录下设备树文件“system-user.dtsi”复制到内核源码“arch/arm/boot/dts”目录下，如下图所示：

```
zy@zy-virtual-machine:~/petalinux/Navigator_7010_v3/ALIENTEK-ZYNQ-driver/project-spec/meta-user/recipes-bsp/device-tree/files$ cp system-user.dtsi /home/zy/worksapce/kernel-driver/linux-xlnx-xlnx_rebase_v5.4_2020.2/arch/arm/boot/dts/
```

修改 system-user.dtsi

示例代码 修改后的 system-user.dtsi

```
1  ///include/ "system-conf.dtsi"
2  #include <dt-bindings/gpio/gpio.h>
3  #include <dt-bindings/input/input.h>
4  #include <dt-bindings/media/xilinx-vip.h>
5  #include <dt-bindings/phy/phy.h>
6
7  / {
8    model = "Alientek Navigator Zynq Development Board";
9    compatible = "xlnx,zynq-zc702", "xlnx,zynq-7000";
10
11   chosen {
12     bootargs = "console=ttyPS0,115200 earlycon root=/dev/mmcblk0p2 rw rootwait";
13     stdout-path = "serial0:115200n8";
14   };
15
16 };
17
18 &uart0 {
19   u-boot,dm-pre-reloc;
20   status = "okay";
21 };
22
23 &sdhci0 {
24   u-boot,dm-pre-reloc;
25   status = "okay";
26 };
27
28 &gem0 {
29   local-mac-address = [00 0a 35 00 8b 87];
30
31   phy-handle = <&ethernet_phy>;
32   ethernet_phy: ethernet-phy@7 {  /* yt8521 */
33     reg = <0x7>;
34     device_type = "ethernet-phy";
35   };
36 };
```

修改 arch/arm/boot/dts 下的 Makefile:

```
1200        wm8850-w70v2.dtb
1201 dtb-$(CONFIG_ARCH_ZYNQ) += \
1202        zynq-cc108.dtb \
1203        zynq-microzed.dtb \
1204        zynq-parallella.dtb \
1205        zynq-zc702.dtb \
1206        zynq-zc706.dtb \
1207        zynq-zc770-xm010.dtb \
1208        zynq-zc770-xm011.dtb \
1209        zynq-zc770-xm012.dtb \
1210        zynq-zc770-xm013.dtb \
1211        zynq-zed.dtb \
1212        zynq-zturn.dtb \
1213        zynq-zybo.dtb \
1214        zynq-zybo-z7.dtb \
1215        system-top.dtb
1216 dtb-$(CONFIG_MACH_ARMADA_370) += \
1217        armada-370-db.dtb \
```

添加 设备树.

defconfig { SDK环境变量设置 (1.4)

```
make xilinx_zynq_defconfig
```

内核配置.

menuconfig 默认

内核编译: make

内核编译完成后会在 arch/arm/boot 目录下生成内核镜像文件 zImage，在 arch/arm/boot/dts 目录下生成设备树镜像文件 system-top.dtb，如图 11.3.8 所示。

额外提一下，如果内核源码中只修改了设备树，可以只编译设备树。在内核源码根目录下输入如下命令编译设备树：

```
make dtbs
```

命令“make dtbs”将.dts 编译成.dtb，编译完成后就可以使用新的设备树镜像文件。

根文件系统编译:

进入 11.1 小节的 petalinux 工程中，输入如下命令配置根文件系统：

```
petalinux-config -c rootfs
```

这里我们使用默认配置，不做修改。

退出配置界面，输入如下命令编译根文件系统：

```
petalinux-build -c rootfs
```

```
rootfs.cpio.gz  rootfs.tar.gz          u-boot.elf  zynq_fsbl.elf
zy@zy-virtual-machine:~/petalinux/Navigator_7010_v3/ALIENTEK-ZYNQ-driver/images/
linux$
```

图 11.4.2 生成根文件系统压缩包

这里我们使用 rootfs.tar.gz 文件。将 rootfs.tar.gz 解压到 SD 卡 ext4 分区即可。

cpio.gz 格式也可以.

使用脚本文件制作SD启动卡:

从本章开始，我们使用 BOOT.BIN，boot.scr，system.bit，zImage，system.dtb 和根文件系统这六个文件启动开发板。

① 2个分区 FAT ext4

② 第一个分区

将前面过程当中生成的各种镜像文件拷贝到 SD 启动卡的 FAT 分区，包括 11.3 小节内核源码目录下 zImage 和 system-top.dtb 镜像文件，11.1 小节 petalinux 工程下 system.bit、BOOT.BIN 和 boot.scr 文件。大家根据自己前面步骤当中文件存放的目录去找到相应的镜像文件，注意复制之前确保 FAT 分区没有上述文件。

复制完成后，看看我们的 FAT 分区有哪些文件：

```
zy@zy-virtual-machine:~$ ls -l /media/zy/boot/
总用量 6976
-rw-r--r-- 1 zy zy  846796 6月   6 13:27 BOOT.BIN
-rw-r--r-- 1 zy zy    2010 6月   6 13:59 boot.scr
-rw-r--r-- 1 zy zy 2083850 6月   6 13:27 system.bit
-rw-r--r-- 1 zy zy   17338 6月   6 13:35 system-top.dtb
-rw-r--r-- 1 zy zy 4183488 6月   6 13:27 zImage
```

图 11.5.2 SD 卡 fat 分区中的文件

然后修改设备树文件名，运行如下命令，将 system-top.dtb 重命名为 system.dtb：

```
mv system-top.dtb system.dtb
```

结果如下图所示：

```
zy@zy-virtual-machine:/media/zy/boot$ ls
BOOT.BIN  boot.scr  system.bit  system-top.dtb  zImage
zy@zy-virtual-machine:/media/zy/boot$ mv system-top.dtb system.dtb
zy@zy-virtual-machine:/media/zy/boot$ ls
BOOT.BIN  boot.scr  system.bit  system.dtb  zImage
zy@zy-virtual-machine:/media/zy/boot$
```

第二个分区

接下来我们需要将 11.4 小节编译的根文件系统压缩包解压到 SD 启动卡的 EXT4 分区，这里笔者使用 rootfs.tar.gz 压缩包文件，进入到 rootfs.tar.gz 压缩包文件所在目录，执行解压命令（解压之前确保 EXT4 分区没有根文件系统）：

```
sudo tar -xzf rootfs.tar.gz -C /media/zy/rootfs
```

结果如下图所示：


```
zy@zy-virtual-machine:~/petalinux/Navigator_7010_v3/ALIENTEK-ZYNQ-driver/images/l
inux$ sudo tar -xzf rootfs.tar.gz -C /media/zy/rootfs
zy@zy-virtual-machine:~/petalinux/Navigator_7010_v3/ALIENTEK-ZYNQ-driver/images/l
inux$ sync
zy@zy-virtual-machine:~/petalinux/Navigator_7010_v3/ALIENTEK-ZYNQ-driver/images/l
inux$ ls /media/zy/rootfs/
bin  boot  dev  etc  home  lib  media  mnt  proc  run  sbin  sys  tmp  usr  var
zy@zy-virtual-machine:~/petalinux/Navigator_7010_v3/ALIENTEK-ZYNQ-driver/images/l
inux$ umount /dev/sdb*
umount: /dev/sdb: not mounted.
```

图 11.5.4 根文件系统解压到 ext4 分区

/media/zy/rootfs 是笔者 SD 启动卡对应的 ext4 分区的挂载点，解压时需要使用 sudo，也就是要以 root 权限进行解压。解压完成之后执行 sync 命令将数据同步到 SD 卡中，之后卸载 SD 启动卡。

还可以通过 NFS 方式挂载根文件系统.