如何用EGO-XZ7通过SD卡载入Linux

环境设置

OS: Ubuntu18.04LTS
官方工具: Vivado SDK 2018.3、Vivado 2018.3
串口连接工具: putty
硬件连接: 一台交换机，通过网线分别连接到Ubuntu主机以及EGO-XZ7

Zynq7000系列启动流程

1. STAGE0 上电执行BootROM中的代码，不可修改，无操作
2. STAGE1 执行程序代码First Stage Boot Loader(FSBL)，通过SDK可以自动生成fsbl.elf，用户可修改
3. STAGE2 裸机执行OS引导程序，对嵌入式Linux，即uboot
4. uboot启动Linux 使用tftp工具将设备树、Linux内核镜像、文件系统加载到内存，使用bootm工具启动

详细步骤

注1：make过程中部分报错是因为系统中缺乏相应工具，如bison、flex，百度可解
注2：tftp装载镜像为方案之一，可以通过烧录SD卡装载Linux系统

I. 创建目录并下载Xilinx官方提供的uboot、Linux内核

```
mkdir ~/BootLinux
git clone https://github.com/Xilinx/u-boot-xlnx.git //用于生成uboot
git clone https://github.com/Xilinx/linux-xlnx.git //用于创建Linux镜像
cd u-boot-xlnx
git checkout xilinx-v2018.3 //此版本中有对应板子的配置文件
```

II. 使用VIvado SDK 2018.3创建FSBL程序

1. 启动Vivado

```
cd ${Xilinxdir}/Vivado/2018.3/bin
./vivado
```

2. 创建新工程，File->Project->New，在Default Part页面选择EGO-XZ7 Board，完成创建
3. Create Block Design，Add IP，选择ZYNQ7 Processing System，其他按需配置
4. 右击block design(.bd)，选择Generate Output Products，选择Create HDL Wrapper
5. Generate Bitstream(可选)
6. File->Export->Export Hardware
7. File->Launch SDK，进入SDK
8. File->New->Application Project，Project Name指定为FSBL，注意Hardware Platform应是由之前vivado生成的Hardware文件目录

9. Next->Zynq FSBL->Finish
10. 将生成的FSBL.elf文件拷至BootLinux目录

III. 生成uboot程序

1. 设置交叉编译环境，需要用到arm-linux-gnueabihf，该工具Vivado SDK已提供，可以直接使用

```
source ${Xilinxdir}/SDK/2018.3/settings64.sh
```

2. 开始编译uboot

```
cd ~/BootLinux/u-boot-xlnx
make CROSS_COMPILE=arm-linux-gnueabihf- zynq_zc702_defconfig //按板子对make进行配置，此处直接借用官方套件的配置，有能力的同学可以针对EGO-EZ7进行修改
make CROSS_COMPILE=arm-linux-gnueabihf- tools  //编译开发所需要的工具
make CROSS_COMPILE=arm-linux-gnueabihf-  //正式编译uboot
```

3. 为生成的u-boot文件增加后缀.elf，拷至BootLinux目录

IV. 生成Boot镜像

1. 进入SDK，Xilinx->Create Boot Image
2. Output BIF file path指定为BootLinux
3. Boot image partitions-> Add->File Path， 选中~/BootLinux/FSBL.elf并添加
4. 同3添加II-5中生成的比特流文件.bit(可选)
5. 同3添加~/BootLinux/u-boot.elf
6. Create Image，可以在BootLinux文件夹中看到生成了.bif文件和BOOT.bin

V. 生成Linux镜像uImage

1. 将uboot中生成的mkimage工具添加到环境变量

```
export PATH=~/BootLinux/u-boot-xlnx/tools:$PATH
```

2. 编译Linux内核镜像

```
cd ~/BootLinux/linux-xlnx
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- xilinx_zynq_defconfig //对make配置
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- prepare scripts //编译开发所需要的工具
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- UIMAGE_LOADADDR=0x8000 uImage //编译内核，且生成uImage
```

3. 将uImage拷至BootLinux目录

VI. 生成设备树.dtb文件

注：一般设备树文件需要通过SDK载入device-tree-xlnx项目，然后根据板子配置生成相应设备树，由于操作过程中碰到了一些问题，未能解决，故暂选用官方套件zynq-zc702的默认设备树，目前尚未遇到问题，仅供参考

1. 生成设备树文件

```
cd ~/BootLinux/linux-xlnx
make ARCH=arm zynq-zc702.dtb
```

在arch/arm/boot/dts目录下会生成zynq_zc702.dtb文件，将其拷至BootLinux目录并重命名为devicetree.dtb

VII. 下载并生成根文件系统

1. 点击下载官网构建好的文件系统镜像arm_ramdisk.image.gz
2. 使用mkimage工具生成可用的镜像

```
mkimage -A arm -T ramdisk -C gzip -d arm_ramdisk.image.gz uarm_ramdisk.image.gz
```

3. 将uarm_ramdisk.image.gz拷至BootLinux目录
4. 至此，所需文件已在BootLinux目录下就绪，分别是BOOT.bin、devicetree.dtb、uImage、uarm_ramdisk.image.gz

VIII. 部署tftp服务器

1. 安装tftp服务器

```
sudo apt-get install tftpd-hpa
```

2. 配置服务

```
sudo vim /etc/default/tftpd-hpa
TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/home/${username}/BootLinux"
TFTP_ADDRESS="0.0.0.0:69"
TFTP_OPTIONS="-l -c -s"
```

3. 重启服务

```
sudo service tftpd-hpa restart
```

IX. 设备配置

1. SD卡接入Ubuntu，将BOOT.bin拷入SD卡，将SD卡插到EGP-XZ7卡槽中
2. 将SW8中的3、4位向下扳，置为1，其余位保持0
3. 连接电源，连接串口USB线，连接网线，使EGO-XZ7与主机位于同一子网

X. 装载Linux镜像并启动

1. 启动putty

```
sudo putty
```

2. 选择协议Serial，Serial Line中填串口对应的文件，本实验中是/dev/ttyUSB1，速率设为115200，Open
3. 启动EGO-XZ7电源，可以看到putty中打印出加载uboot的内容，最后停留在命令行：

```
Zynq>
```

4. 查看启动时输出的信息，可以看到EGO-XZ7通过dhcp获取到了ip。如果没有ip，可以通过setenv手动获取。检查与主机的连接是否畅通

```
ping ${your-host-ip}
```

5. 设置tftp服务主机

```
setenv serverip ${your-host-ip}
```

6. 通过tftp获取必要文件，并写入内存地址

```
tftpboot 0x10000000 uImage
tftpboot 0x11000000 devicetree.dtb
tftpboot 0x12000000 uarm_ramdisk.image.gz
```

7. 启动镜像

```
bootm 0x10000000 0x12000000 0x11000000
```

8. 使用Linux，可以看到文件系统，至此，Linux装载与启动完毕

```
ls
```