# 果壳 FPGA 平台

陈煦豪

测试软件包括 Vivado 和 minicom。
Vivado 软件负责与目标机的 JTAG 接口交互，顺序完成三部分工作：
·向目标机的 JTAG 接口传输 bitstream，完成 FPGA 上逻辑的烧写
·将 linux.bin 通过 JTAG 接口传输给 DDR4 SODIMM 存储
·通过 JTAG 接口配置 vio 软复位，解除 FPGA 上核(CPU)的复位状态
minicom 软件负责与目标机的 UART 接口交互，其功能为，在核执行目标代码时，输出串口打印的信息。
工程资源：https://github.com/ssdfghhhhhhh/NutShell_U250.git
环境：32 服务器

## 一：生成 Bitstream

1:需要：verilog 代码，位于/NutShell_U250/build。在文件夹中替换 Topmain 后需要在 vivado 中更新文件。

配置：修改 Makefile 文件

```
DATAWIDTH ?= 64
BOARD ?= pynq  # sim  pynq  axu3cg
CORE  ?= inorder  # inorder  ooo  embedded
```

修改\src\main\scala\top\Settings.scala

```
object PynqSettings {
  def apply() = Map(
    "FPGAPlatform" -> true,
    "NrExtIntr" -> 3,
    "ResetVector" -> 0x80000000L,
    "MemMapBase" -> 0x0000000080000000L,
    "MemMapRegionBits" -> 28,
    "MMIOBase" -> 0x0000000040600000L,
    "MMIOSize" -> 0x0000000001000000L
  )
}
```

2:Generate Bitstream

3:最终需要两个文件：位于

path/NutShell_U250/fpga/board/pynq/build/cxhU250_pynq/cxhU250_pynq.runs/impl_1/
（1）System_top_wrapper.bit
（2）System_top_wrapper.ltx

4:使用 vivado 内 hardware manager 打开一块 U250 板卡，右键 xcu250 选择 program device，烧写上述两个文件。注：避免使用 21330409X021A 板卡，这张卡有用户在使用。

## 二：编译程序

1：目标机上运行的目标代码为*.bin。编译*.bin 需要使用果壳可用的 Riscv 工具链。

2：编辑 DDR4 装载的数据 data.txt

目标机上运行的目标代码为*.bin，需要通过 JTAG 接口传输到 DDR4 SODIMM 中，在 FPGA 平台的逻辑设计中，包括了 jtag_axi 的 IP，其连接 DDR4 MIG IP，主机根据写地址和写数据信息发送 jtag_axi 的写事务请求，完成 DDR4 SODIMM 数据的装载。

具体的实现为：使用 bin2fpgadata，将.bin 转为.txt

./bin2fpgadata -i a.bin -o a.txt

使用方法　　Plain Text　复制代码

```
usage: ./bin2fpgadata --in=string [options] ...
options:
  -i, --in            input binary file (string)
  -o, --out           output data text file (string [=data.txt])
  -b, --burst         burst length (unsigned int [=1024])
      --noeof         disable end of file char 0a
      --tcl           generate tcl script
      --offset        address offset (unsigned int [=0])
      --hole-begin    address hole begin (unsigned int [=0])
      --hole-end      address hole end (unsigned int [=0])
  -?, --help          print this message
```

例子：`./bin2fpgadata -i a.bin -o a.txt`

图 1：bin2fpgadata 使用方法

注：bin2fpgadata 默认以 0 为基地址，需在 cpp 文件中修改“offset”为 80000000 后重新编译，与工程的内存映射相匹配。

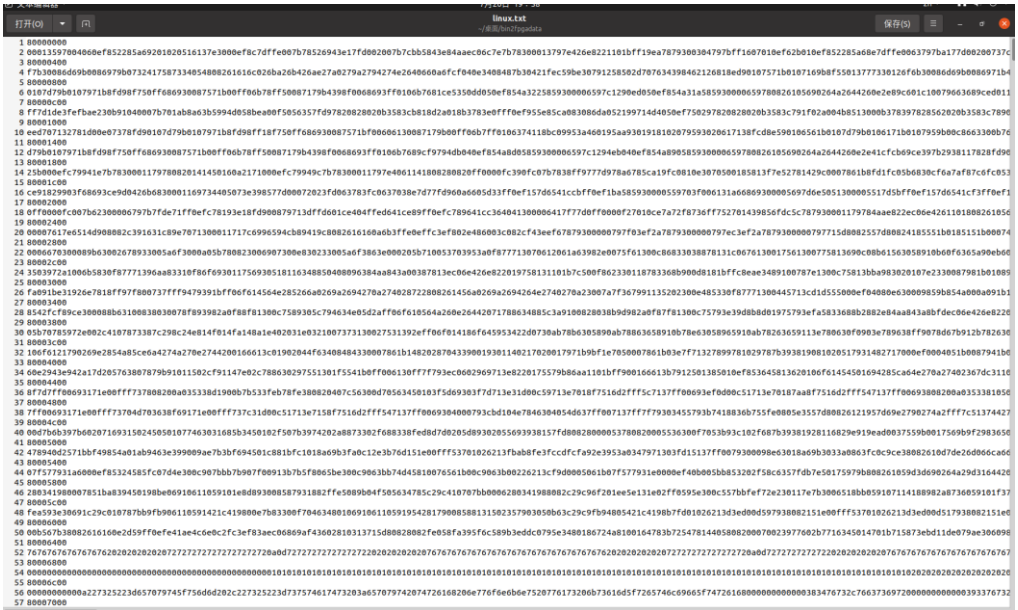

图 2：data.txt 内容

## 三：写入 data，复位 CPU，观察串口

1：命令行 sudo minicom 打开串口观察工具，调整至 fpga 板卡对应接口（21330409X021A 对应 ttyUSB2），波特率 115200


图 3：minicom 界面

2：通过 jtag 向 DDR 写入 data.txt

jtagmv.tcl 功能为提取 data.txt 内地址数据信息，写入 DDR4 SODIMM。

在 vivado 的 tcl 窗口下输入：

source /path/jtagmv.tcl

注：完整的*.bin 的数据写入 DDR4 后，会报告 ERROR：bad hexStr for Cse_NumberUtil::hexStrBinStr，如图 4。如果需要再一次用 jtag 传输*.bin 的内容到 DDR4 ，则需要先在 tcl 窗口输入一条命令 delete_hw_axi_txn wr_txn，这是因为上一条 ERROR 指示使得建立的 wr_txn 事务未重置，需要手动删除该事务信息。

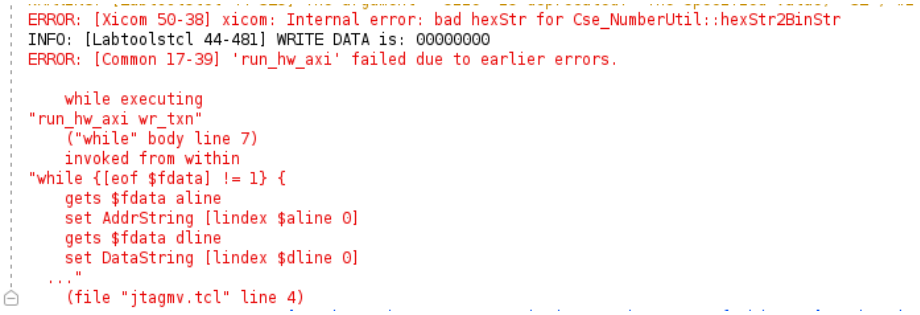


图 4：Jtag 向 DDR 写入完成后的提示信息

3：用 vio 设置 CPU 复位状态，解复位后 CPU 启动，串口输出信息。如图 5&6

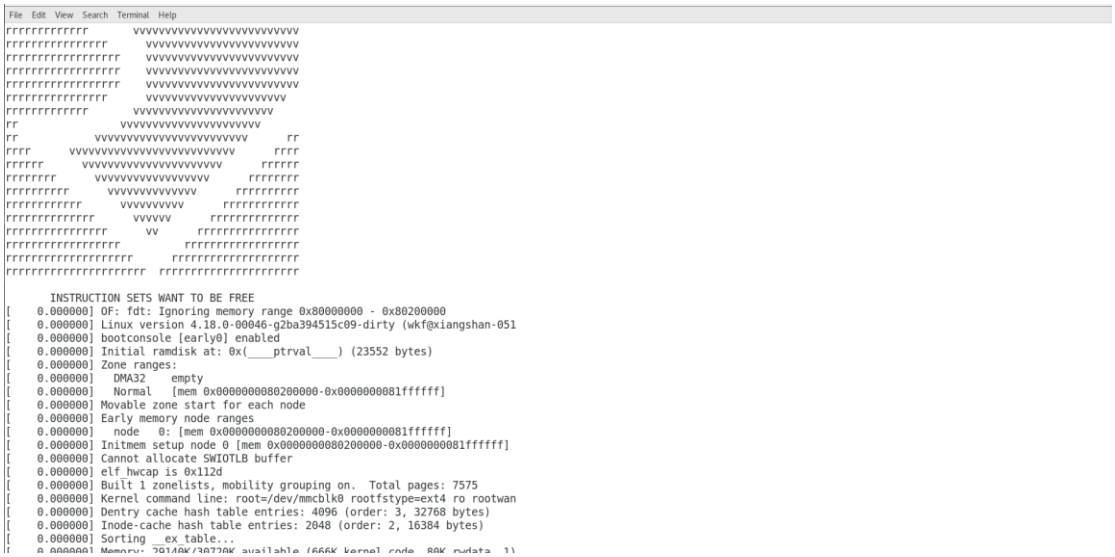


图 5、6：启动 linux 系统