

Guide for zCore with PCI and GPU based on framebuffer

移植zCore

- 将 [zCore2Hifive](#) 克隆到本地，打开zCore
- 在 `zCore/src/platform/riscv/consts.rs` 中将物理地址的起始位置修改为 `0x8000_0000`：

```
// zCore/src/platform/riscv/consts.rs
if #[cfg(feature = "board_fu740")] {
    pub const KERNEL_OFFSET: usize = 0xFFFF_FFFF_8020_0000;
    pub const PHYS_MEMORY_BASE: usize = 0x8000_0000;
    pub const PHYS_MEMORY_END: usize = 0xC000_0000;
}
```

并添加对应编译feature：

```
# Makefile
ifeq ($(PLATFORM), fu740)
    features += board_fu740 link_user_img
```

- 在 `zCore/src/platform/riscv/entry.rs` 中添加boot entry：

```
// zCore/src/platform/riscv/entry.rs
#[cfg(feature = "board_fu740")]
global_asm!(include_str!("boot/boot_fu740.asm"));
```

- 在 `zCore/src/platform/riscv/boot` 中添加 `boot_fu740.asm`，内容如下：

```
# zCore/src/platform/riscv/boot/boot_fu740.asm
.equ PHY_MEM_OFS, 0xffffffff00000000

.section .data
.align 12 #12位对齐
boot_page_table_sv39:
    #1G的一个大页：0x00000000_80000000 --> 0x80000000
    #1G的一个大页：0xffffffff_80000000 --> 0x80000000

    #前510项置0
    .zero 8
    .zero 8
    .quad (0x80000 << 10) | 0xef #0x80000000 --> 0x80000000

    .zero 8 * 507
    #倒数第二项，PPN=0x80000(当转换为物理地址时还需左移12位)，标志位DAG_XWRV置1
    .quad (0x80000 << 10) | 0xef
    .zero 8
```

在官网下载编译好的[freedom-u-sdk](#)，windows上可以用rufus将其装入SD卡中

- 执行以下命令生成镜像：

```
make riscv-image
cd zCore
make MODE=release LINUX=1 ARCH=riscv64 PLATFORM=fu740
```

- 在官网下载编译好的[freedom-u-sdk](#)，windows上可以用rufus将其装入SD卡中，或直接将对应镜像复制其中：打开SD卡，将制作好的zCore镜像 zcore-fu740 复制到/boot文件夹下，修改/boot中的配置文件extlinux/extlinux.conf，将默认的形象.gz改为zCore镜像的文件名 zcore-fu740

```
> tree -a
.
├── extlinux
│   └── extlinux.conf
├── hifive-unmatched-a00.dtb
├── Image.gz
├── ucore
└── zcore-fu740

1 directory, 5 files
```

- 将SD卡插入主机，连接串口（串口相关参见中文文档）后启动并进入u-boot，执行 `printenv` 查看 `boot_targets`，`mmc0`为SD卡启动，执行 `setenv boot_targets mmc0`，再执行 `boot` 即可启动zCore

```
=> boot
switch to partitions #0, OK
mmc0 is current device
Scanning mmc 0:3...
Found /extlinux/extlinux.conf
Retrieving file: /extlinux/extlinux.conf
207 bytes read in 11 ms (17.6 KiB/s)
1:   OpenEmbedded-SiFive-HiFive-Unmatched
Retrieving file: /zcore-fu740
13961003 bytes read in 12070 ms (1.1 MiB/s)
append: root=/dev/mmcblk0p4 rootfstype=ext4 rootwait console=ttySIF0,115200 earlycon
Retrieving file: /hifive-unmatched-a00.dtb
10529 bytes read in 17 ms (604.5 KiB/s)
## Booting kernel from Legacy Image at 84000000 ...
   Image Name:   zCore-fu740
   Image Type:   RISC-V Linux Kernel Image (gzip compressed)
   Data Size:    13960939 Bytes = 13.3 MiB
   Load Address: 80200000
   Entry Point:  80200000
   Verifying Checksum ... OK
## Flattened Device Tree blob at 88000000
   Booting using the fdt blob at 0x88000000
   Uncompressing Kernel Image
   Using Device Tree in place at 0000000088000000, end 0000000088005920

Starting kernel ...

zCore rust_main(hartid: 2, device_tree_paddr: 0x88000000)
[ 3.029470 WARN 0 0:0 zcore_drivers::builder::dt] device-tree: failed to parsing node "pmic@58": InvalidParam
[ 3.032891 WARN 0 0:0 linux_object::loader] elf relocate Err:".rela.dyn not found", base 10000
[ 3.033755 WARN 0 0:0 linux_syscall] getuid: unimplemented
[ 3.034387 WARN 0 0:0 linux_syscall] brk: unimplemented
[ 3.035022 WARN 0 0:0 linux_syscall] brk: unimplemented
[ 3.035666 WARN 0 0:0 linux_syscall] getpgid: unimplemented
[ 3.036311 WARN 0 0:0 linux_syscall] setpgid: unimplemented
[ 3.036971 WARN 0 0:0 linux_object::fs::stdio] stdout TCGETS | TIOCSPPGRP, pretend to be tty.
[ 3.037862 WARN 0 0:0 linux_object::fs::stdio] stdin TCGETS | TIOCSPPGRP, pretend to be tty.
[ 3.038743 WARN 0 0:0 linux_syscall] brk: unimplemented
[ 3.039381 WARN 0 0:0 linux_syscall] geteuid: unimplemented
[ 3.040039 WARN 0 0:0 linux_syscall] geteuid: unimplemented
/ #
```

PCI与显示

与PCI和显示相关的内容在 `drivers/src` 中，实现了PCI驱动以及基于PCI的framebuffer图像显示。

模拟PCI与framebuffer显示

模拟PCI总线可以实现在qemu上运行的显示界面

- 在linux系统下，设置显示输出：

```
export DISPLAY=:0.0
```

- 制作镜像：

```
cd zCore #zCore ROOT DIR
make rootfs # make file system
make image # build image for rootfs
cd zCore # zCore kernel dir
make run MODE=release LINUX=1 GRAPHIC=on
```

此时可以进入qemu模拟的framebuffer显示界面，使用zCore命令行。

PCI and Graphic on fu740

在 `HiFive Unmatched fu740` 上，PCI驱动的使用方式需要参考中文文档以及设备树信息。

具体使用方式类似于移植zCore的过程，需要在生成镜像部分的指令更改为：

```
make riscv-image
cd zCore
make MODE=release LINUX=1 ARCH=riscv64 PLATFORM=fu740 GRAPHIC=on
```

之后将镜像装入sd卡即可启动。

后续相关

PCI的总线结构在zCore中会被进一步的重构，可以兼容更多外设与总线架构，目前处于重构阶段，具体文档详见<https://github.com/rcore-os/zCore>