**1.实验介绍**

SC7-RVfpga是基于framework-wd-riscv-sdk提供的nexys\_a7\_eh1，在NEXYS A7开发版上运行的RV32架构的操作系统内核。支持线程管理，线程切换的功能

NEXYS A7提供的是fpga部件，本身不是任何架构的。而nexys\_a7\_eh1提供了可以在NEXYS A7上使用的bitstream文件，写入后能在NEXYS A7上构建一个RV32的软核。之后就可以运行riscv架构的32位程序了。更多内容可以参考RVfpga\_入门指南.pdf



图片来自(RVfpga\_入门指南.pdf page21)

SC7-RVfpga是SC7内核的移植版本。原本的SC7内核是支持RV64和Loongarch架构的，支持内核态和用户态的内核。由于nexys\_a7\_eh1支持的限制，裁剪了SC7的大部分内容

nexys\_a7\_eh1提供的RV32的软核支持ICM扩展，没有特权指令扩展，只在M态运行，不支持特权指令和特权寄存器。而内核态和用户态都需要特权指令。此外，虚拟内存也是不支持的。Riscv架构的页表配置在satp特权寄存器，读写satp是特权指令。

**2.内核**

项目仓库 https://github.com/BandieraRosse/SC7-RVfpga

下面的内容参照第5次提交，SHA是7ce7503的版本。

内核有1个.S文件，8个.c文件，7个.h文件。其中SC7\_start\_kernel.c包括内核的主函数。其它c文件都有对应的头文件

内核的核心文件是：

SC7\_start\_kernel.c

process.c

switch.S

print.c

这四个文件实现了线程创建，线程切换和必要的print功能。

剩下的几个文件提供辅助功能

cpu.c事实上只提供访问当前运行线程的接口，proc\_t\* myproc(void)。如果在process.c中写这个函数，删除cpu.c也没有问题。

string.c提供的memset函数在process.c中有使用

figlet.c扩展了输出艺术字体的功能

test.c提供了两个测试printf的函数

此外print.h还提供了颜色打印的宏

**3.运行**

连接NEXYS A7开发板后，先给对应串口打上zadig驱动。

然后在platformio.ini中修改monitor\_port属性为正确的串口号(在作者本机是COM8)。

随后使用vscode的platformio插件和Serial Monitor插件，

首先用Serial Monito监视对应串口，

然后先执行Upload Bitstream，再执行Upload，即可运行并看到输出。



（Serial Monito 参考）

**4.难点**

移植时最大的问题是从RV64到RV32的切换和对内核的裁剪。SC7在具有线程切换功能时，已经包含了很多特权指令，支持用户态了。这一部分是要全部删除的。

nexys\_a7\_eh1核无法执行特权指令集，虽然编译时可以通过，但是执行特权指令如csrw sstatus(修改sstatus)时，会导致Cpu复位。

关于spinlock锁的功能也全部要删去，加锁时用到了关中断intr\_off，会使用特权指令。

最后是swtch.S的更改，原本的swtch.S是64位版本的，存取指令使用sd，ld，处理64位数据。换到32位要变成sw,lw。此外对应的context结构体的偏移也要变，具体来说就是都除以2.

原本的Swtch.S:

.globl swtch

swtch: #14个寄存器

sd ra, 0(a0)

sd sp, 8(a0)

sd s0, 16(a0)

sd s1, 24(a0)

sd s2, 32(a0)

sd s3, 40(a0)

sd s4, 48(a0)

sd s5, 56(a0)

sd s6, 64(a0)

sd s7, 72(a0)

sd s8, 80(a0)

sd s9, 88(a0)

sd s10, 96(a0)

sd s11, 104(a0)

ld ra, 0(a1)

ld sp, 8(a1)

ld s0, 16(a1)

ld s1, 24(a1)

ld s2, 32(a1)

ld s3, 40(a1)

ld s4, 48(a1)

ld s5, 56(a1)

ld s6, 64(a1)

ld s7, 72(a1)

ld s8, 80(a1)

ld s9, 88(a1)

ld s10, 96(a1)

ld s11, 104(a1)

ret

移植后的swtch.S

.globl swtch

swtch: #14个寄存器

sw ra, 0(a0)

sw sp, 4(a0)

sw s0, 8(a0)

sw s1, 12(a0)

sw s2, 16(a0)

sw s3, 20(a0)

sw s4, 24(a0)

sw s5, 28(a0)

sw s6, 32(a0)

sw s7, 36(a0)

sw s8, 40(a0)

sw s9, 44(a0)

sw s10, 48(a0)

sw s11, 52(a0)

lw ra, 0(a1)

lw sp, 4(a1)

lw s0, 8(a1)

lw s1, 12(a1)

lw s2, 16(a1)

lw s3, 20(a1)

lw s4, 24(a1)

lw s5, 28(a1)

lw s6, 32(a1)

lw s7, 36(a1)

lw s8, 40(a1)

lw s9, 44(a1)

lw s10, 48(a1)

lw s11, 52(a1)

ret