

第七届“龙芯杯”全国大学生计算机系统能力培养大赛

CoreMIPS i11 CPU

项目设计报告

西北工业大学 2 队

冯宜湑

yixu-nwpu@mail.nwpu.edu.cn

陈世杰

csj314159@mail.nwpu.edu.cn

蒋晓天

timba@mail.nwpu.edu.cn

李旭辉

lixuhui123@mail.nwpu.edu.cn

2023 年 8 月

目录

第一部分 概述.....	3
1.1 项目简介	3
第二部分 CPU.....	3
2.1 流水线设计	3
2.1.1 总体架构.....	3
2.1.2 预取指（pre_IF）级 & 取指（IF）级	4
2.1.4 译码（ID）级	4
2.1.5 执行（EXE）级 & 预访存（MEMO）级.....	5
2.1.6 访存（MEM）级.....	5
2.1.7 写回（WR）级.....	5
2.2 指令集	5
2.3 协处理器 CPO，中断与异常	5
2.4 内存管理（TLB&MMU）	6
2.5 缓存设计	7
2.6 分支预测设计.....	8
2.7 性能优化	9
第三部分 测试与系统.....	9
3.1 功能测试与性能测试.....	9
3.2 系统与外设	10
3.2.1 系统测试.....	10
3.2.2 Pmon	11
3.2.3 Ucore	14
3.2.4 Linux	16
3.2.5 外设支持.....	17
第四部分 开发支持工具.....	17
4.1 Verilator+NEMU	17
第五部分 附录.....	20
4.1 参考资料	20
4.2 参考仓库	20
4.2.1 17 年参赛作品	20
4.2.2 19 年参赛作品	20
4.2.3 21 年参赛作品	20
4.2.4 其他.....	20
4.3 参考文献	21
第五部分 致谢.....	21

第一部分 概述

1.1 项目简介

CoreMIPSi11 CPU 是一个实现在龙芯教学实验平台（Artix-7 XC7A200T）的基于 MIPS 32 Rev 1 指令集架构的处理器。其包含指令和数据缓存，其频率达到约 90MHz，可正常运行系统测试。

1.1.1 设计参数

CoreMIPSi11 CPU 目前采用单发射七级流水线架构，可正确运行 MIPS32 Rev 1 中的 98 条指令，19 个 CP0 寄存器，支持两个软件中断 (SW0~SW1)，六个硬件中断 (HW0~HW5)，一个计时器中断，计时器中断复用 HW5 硬件中断。CPU 对外通过 2 个接口进行通信，分别是 Cache 的指令和数据接口（AXI4 协议）。

1.1.2 系统支持

CoreMIPSi11 CPU 可正确运行系统测试，同时启动龙芯发布包内的 Pmon 系统和 Ucore 系统。

1.1.3 外设支持

CoreMIPSi11 CPU 修改了官方 SOC 中的部分设计，目前成功支持 SPI Flash 板载引导芯片, MAC 网络控制器, DDR3 内存控制器, NAND Flash 板载储存, Uart 串口通信(16550), 共 5 种外设。

第二部分 CPU

2.1 流水线设计

2.1.1 总体架构

CoreMIPS i11 目前采用七级流水顺序单发射结构，七级流水线分别为：预取指(IF)，取指 (IF)，译码(ID)，执行(EXE)，预访存(MEM0)，访存(MEM)，写回(WB)。具体结构见图 2.1.1.1 与 2.1.1.2。

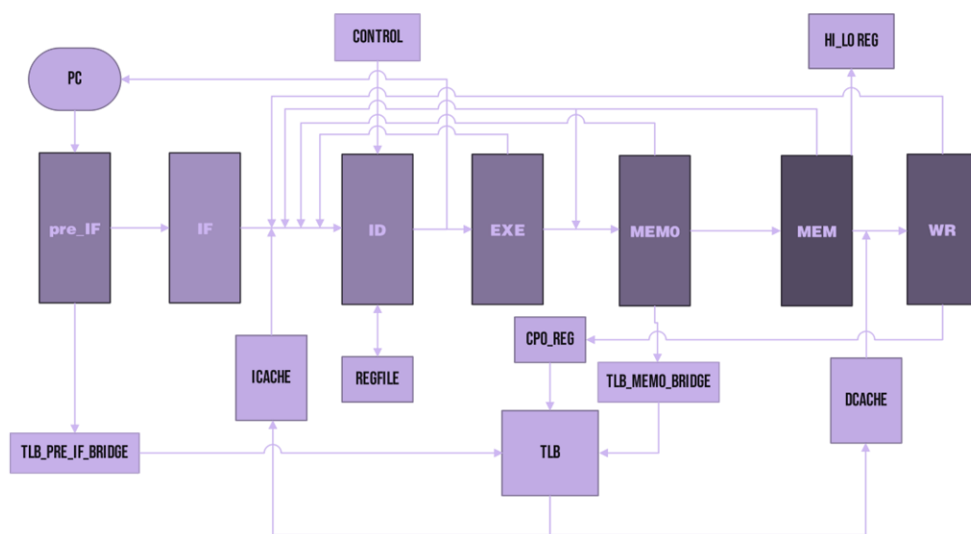


图 2.1.1.1: 流水线结构示意图

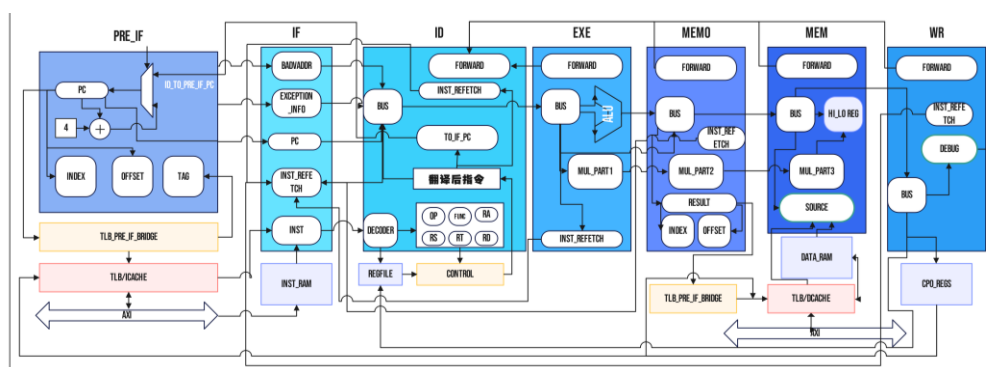


图 2.1.1.2: 流水级的具体功能

2.1.2 预取指（pre_IF）级 & 取指（IF）级

进行地址转换，访问 Icache 或者存储器取得指令；：即读取 PC 中存储的指令地址；拆分为 Tag, Index, Offset 段；TLB_bridge 虚实地址转换。如果命中，锁存所有的访存信息，如果缺失，则向 AXI 总线发起访存请求，重填后提供相应的数据。

2.1.4 译码（ID）级

主要工作为：使用 decoder 对指令进行译码；读取寄存器堆；检测数据冒险，并用前递和阻塞来解决。静态分支预测，进行跳转。

2.1.5 执行（EXE）级 & 预访存（MEM0）级

主要包括 ALU 计算；三周期华莱士树，多周期除法器。

2.1.6 访存（MEM）级

DCache 模块, 访问命中,则在 MEM 级获得访问的数据。未命中, 则阻塞流水线, 向 AXI 总线发起访存请求。同时将乘除法计算结果存入 hi_lo 寄存器。

2.1.7 写回（WR）级

WR: 异常处理, 与 CP0 交互, 同时写回寄存器堆。

2.2 指令集

CoreMIPS11 CPU 实现了与计算, 分支, 访存, 异常等指令共 98 条, 具体如下:

计算指令: ADD, ADDI, ADDU, ADDIU, SUB, SLT, SUBU, AND, ANDI, OR, ORI, XOR, NOR, XORI, SLTI, SLTIU, SLTU, MUL, MULT, MULTU, DIV, DIVU, SLL, SLLV, SRA, SRAV, SRL, SRLV, (28)

分支指令: BEQ, BNE, BGEZ, BGTZ, BLEZ, BLTZ, BLTZAL, BGEZAL, J, JAL, JL, JALR, JR (12)

访存指令: LB, LW, LH, SB, SW, SH, LBU, LHU, (8)

特权指令: ERET, SYSCALL, BREAK, SYNC, PREF, WAIT, MOVFT, CACHE (8)

无条件移动指令: LUI, MTC0, MFC0, MFHI, MFLO, MTLO, MTHI (6)

TLB 指令: TBLWI, TLBP, TLBR, TLBWR (4)

TRAP 指令: TGE, TGEI, TGEU, TGEIU, TLT, TLTI, TLTU, TLTIU, TEQ, TEQI, TNE, TNEI (12)

系统指令: CLO, CLZ, MOVN, MOVZ, LWR, LWL, SWL, SWR, COP1, LWC1, LDC1, SWC1, SDC1, BEQ1, BNE1 (15)

HILO 寄存器: MADD, MADDU, MSUB, MSUBU (4)

2.3 协处理器 CP0, 中断与异常

为了实现精确异常, 我们将异常处理模块放在 WR 级, 在 WR 级检测到异常并刷新流水线, 将 CPU 权限交给操作系统。根据 MIPS 文档中的异常的优先级处理检测到的异常。可支持的异常有:

1. TLB refill exception
2. TLB invalid exception
3. TLB modified exception
4. overflow exception

- 5. Systemcall exception
- 6. Breakpoint exception
- 7. Reserved Instruction exception
- 8. Trap exception
- 9. Address Error exception
- 10.ADEL_if exception
- 11.ADEL_mem exception
- 12.ADES exception
- 13. interrupt exception
- 14.CpU exception

6 个硬件中断：HW0 ~ HW5

2 个软件中断：SW0 ~ SW1

相关寄存器有：

Index	(CP0 Register 0, Select 0)
Random	(CP0 Register 1, Select 0)
EntryLo0	(CP0 Register 2, Select 0)
EntryLo1	(CP0 Register 3, Select 0)
Context	(CP0 Register 4, Select 0)
PageMask	(CP0 Register 5, Select 0)
Wired	(CP0 Register 6, Select 0)
BadVaddr	(CP0 Register 8, Select 0)
Count	(CP0 Register 9, Select 0)
EntryHi	(CP0 Register 10, Select 0)
Compare	(CP0 Register 11, Select 0)
Status	(CP0 Register 12, Select 0)
Cause	(CP0 Register 13, Select 0)
EPC	(CP0 Register 14, Select 0)
PRID	(CP0 Register 15, Select 0)
Ebase	(CP0 Register 15, Select 1)
Config	(CP0 Register 16, Select 0)
Config1	(CP0 Register 16, Select 1)
TagLo	(CP0 Register 28, Select 0)

2.4 内存管理 (TLB&MMU)

8 项全相联 TLB，设置两个查找接口，一个读一个写接口。预取指阶段将 PC 根据 ICache 的需求拆分为 Tag,Index,Offset 段，直接将 Index 信号连入 Data Ram 和 Tagv Ram，Tagv Ram 当拍读出数据，Data Ram 下一拍读出数据。在 PRE_IF 与 EXE 级有对应的 bridge 进行虚实地址转换，并通过 bridge 控制 TLB 相关查找信号，可以执行 tlbw、tlbr、tlbwi、tlbwr 四条指令。对于 tlbr、tlbwi、tlbwr 写后读冲突，这里采用指令重取报假例外来处理，而对于 tlbw 与 mtc0 的写后读冲突，我们采用 EXE 阶段阻塞方式解决。

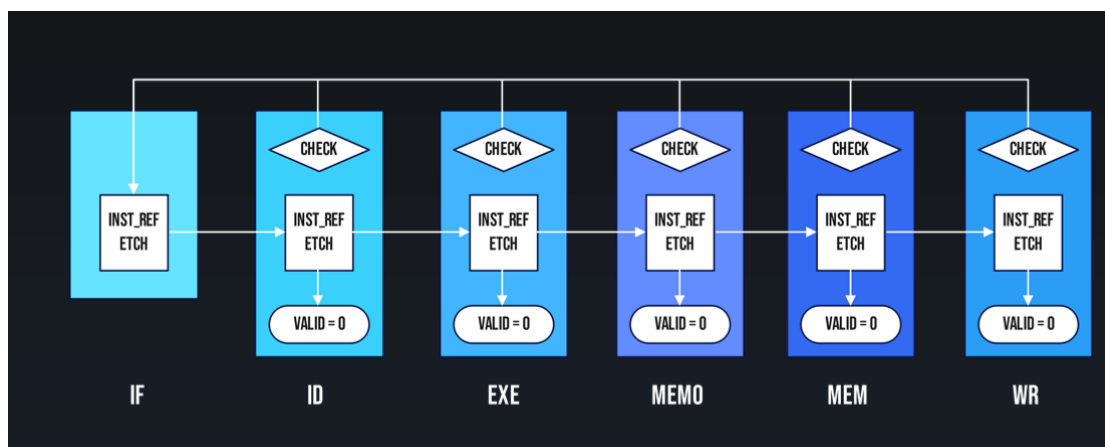


图 2.4.1: TLB 模块

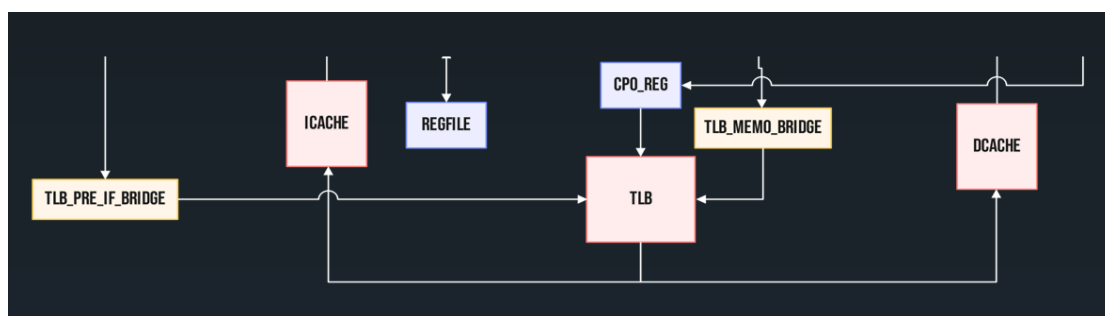


图 2.4.2: TLB 与流水线设计

2.5 缓存设计

CPU 实现了 2 路组相联的 ICache (4KB) 和 DCache (4KB)，分别响应 CPU 的取指和访存请求。tag 比对方式为 VIPT，其中 tag 20 位，index 8 位，每一路大小 128 字节，共 256 组。我们的 ICache、DCache 以及 CPU 与 AXI 交互的转接桥都是由 CHISEL 编写，能够避免一些常见的 Verilog 设计失误如未连接线网，latch 等。替换策略利用了 CHISEL 中的 LFSR 移位寄存器实现了伪随机替换。

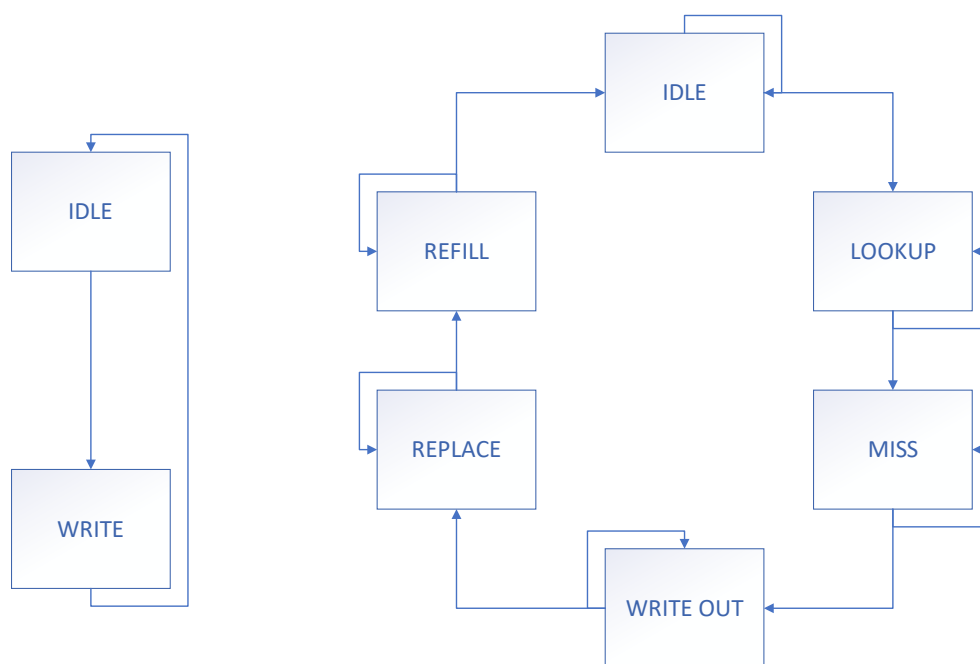


图 2.5.1: cache 状态图转化

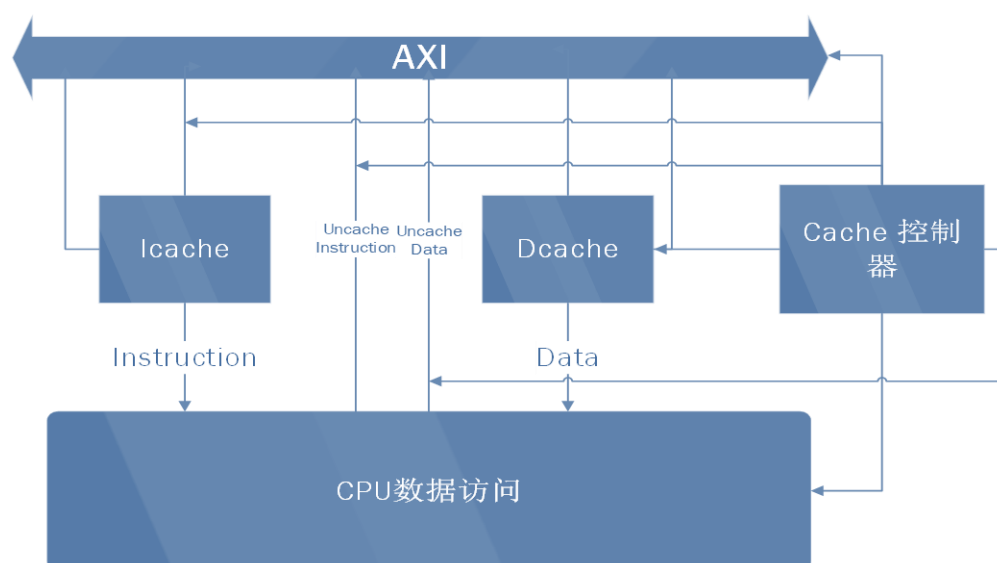


图 2.5.2: cache 与 AXI 交互

在性能测试环境下，ICache 的命中率在 99%以上，DCache 的命中率在 98%以上。具体见分支预测命中率部分。

2.6 分支预测设计

在分支预测的设计上，我们采用 BTB 表的结构，其中 BTB 为 256 项的历史分支查找表。下图为我们的分支预测命中率（包含 cache 命中率），平均在 90%以上。

BTB 使用 Vivado 双端口 XPM 实现，在 EXE 级更新 BTB 表，如果分支预测错误，则恢复 PC，正确则进入如下状态转化过程：弱命中(WEAKLY_T)，弱不命中(WEAKLY_NT)，强命中(STRONGLY_NT)，强不命中(STRONGLY_T)中四种情况。

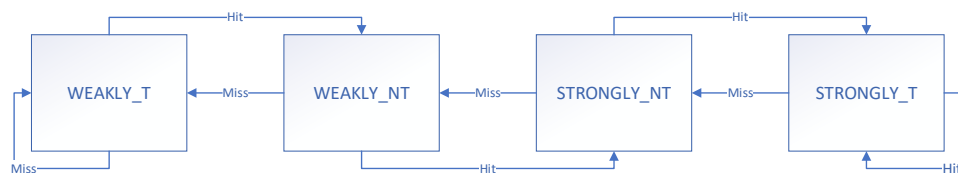


图 2.6.1 BTB 状态图转换

观察 10 个测例，我们发现了大量的循环，RAS 用处不大，因此我们删除了 RAS。

程序	icache			dcache			分支预测		
	cached	miss	命中率	cached	miss	命中率	branch	miss	命中率
bitcount	37482	147	99.61%	3977	43	98.92%	5767	1220	78.85%
bubble_sort	186935	161	99.91%	61983	176	99.72%	25003	7397	70.42%
coremark	450944	225	99.95%	83938	2437	97.10%	78374	11040	85.91%
crc32	309425	140	99.95%	54623	108	99.80%	53274	2679	94.97%
dhystone	62755	294	99.53%	11269	131	98.84%	14464	794	94.51%
quick_sort	713331	433	99.94%	38898	799	97.95%	32318	5836	81.94%
select_sort	225572	151	99.93%	21982	176	99.20%	51646	1950	96.22%
sha	198521	364	99.82%	40447	229	99.43%	23571	573	97.57%
stream_copy	16775	124	99.26%	4675	531	88.64%	2246	101	95.50%
stringsearch	145342	176	99.88%	36682	306	99.17%	28856	1942	93.27%
	2347082	2215	99.91%	358474	4936	98.62%	315519	33532	89.37%

图 2.6.2 Cache 命中率与分支预测命中率

2.7 性能优化

在性能优化上，我们主要做了以下工作：

1. 设计分支预测，提高预测准确率。
 2. 频率提升。
 3. 优化关键路径，增加寄存器，修改调整状态机等方法。
 4. 对于 AXI Bridge，当读写数据不相关时，采用非阻塞方式向 AXI 发送请求
- 在优化过程中，主要使用寄存器平衡，消除代码优先级，关键信号后移等方法。

第三部分 测试与系统

3.1 功能测试与性能测试

CoreMIPSi11 CPU 在进行功能测试时满分通过，在系统测试时遇到了一定的问题，比如设置 global 和 out of context, 我们全部设置为了 Global，导致最终测出来的结果偏高，我们重新调整设置，最终回到了正确结果，并重新修改代码，调整关键路径实现了性能的优化，最终的频率提升到了 90MHz，IPC 在 24.5 左右。如下为功能测试和性能测试的结果：

myCPU接口类型	89个功能点通过数	记忆游戏	系统测试
sram接口	0	-	-
axi接口	89	1	1

表 3.1.1: 功能测试结果

序号	测试程序	myCPU			gs132	IPC _{myCPU} /IPC _{gs132}
		上板计时(16进制)		CPU count*2 : SoC cout	上板(16进制)	
		数码管显示 (CPU count) (最左开关拨下)	数码管显示 (SoC count) (最左开关拨上)		数码管显示 (CPU count)	
cpu_clk : sys_clk		50MHz : 100MHz		-	50MHz : 100MHz	-
1	bitcount	2de81	6606b	0.899896865	4E3DD2	27.26986221
2	bubble_sort	123a6e	28823b	0.899966438	1EF74EA	27.180719
3	coremark	345fc4	7463a3	0.899981816	43399B0	20.53688103
4	crc32	1cc446	3fed72	0.899978375	2A86A88	23.65291467
5	dhystone	57334	c1d7b	0.899701377	7F000A	23.3027281
6	quick_sort	14bf9a	2e1bf4	0.899969952	1C65821	21.89775109
7	select_sort	117351	26c7c3	0.89996447	1B7FFF2	25.21423044
8	sha	1124c3	261934	0.899963153	1D2E296	27.23378338
9	stream_copy	16bb2	32863	0.899805264	214F0D	23.4455889
10	stringsearch	c41bb	1b3d1f	0.899950087	14286C6	26.31417015

性能分	IPC比值	24.497
	CPU频率	90MHz

表 3.1.2：性能测试结果

3.2 系统与外设

3.2.1 系统测试

CoreMIPSi11 CPU 在进行系统测试的过程中，主要按照系统测试的说明进行系统测试程序和 R 指令和 D & U 指令测试，下图是我们正确进入系统测试的截图（包含系统测试）：

```
D:\jingsai\group\nccsc2022-group-mips\nccsc2022-group-mips\system_t
MONITOR for MIPS32 - initialized.
>> g
>>addr: 0x8000380c
elapsed time: 0.992s
>> g
>>addr: 0x8000383c
elapsed time: 0.673s
>> g
>>addr: 0x800038c4
elapsed time: 3.923s
>> g
>>addr: 0x8000315c
OK
elapsed time: 0.816s
>> g
>>addr: 0x80003180
elapsed time: 10.505s
>> g
>>addr: 0x800031b4
elapsed time: 6.349s
>> g
>>addr: 0x800031fc
elapsed time: 11.786s
>> g
>>addr: 0x80003228
elapsed time: 16.327s
```

图 3.1.1：6 个系统测试程序运行成功截图

```

>> r
R1 (AT)      = 0x00000000
R2 (v0)      = 0x0000001e
R3 (v1)      = 0x80505d38
R4 (a0)      = 0x0000004b
R5 (a1)      = 0x485af164
R6 (a2)      = 0xbae4174e
R7 (a3)      = 0x00100000
R8 (t0)      = 0x00000000
R9 (t1)      = 0x00000000
R10(t2)      = 0x00000002
R11(t3)      = 0x00000003
R12(t4)      = 0x8041bf80
R13(t5)      = 0x1005c000
R14(t6)      = 0x1005c000
R15(t7)      = 0x00000000
R16(s0)      = 0x00000000
R17(s1)      = 0x00000000
R18(s2)      = 0x00000000
R19(s3)      = 0x00000000
R20(s4)      = 0x00000000
R21(s5)      = 0x00000000
R22(s6)      = 0x00000000
R23(s7)      = 0x00000000
R24(t8)      = 0x00000000
R25(t9/jp)   = 0x00000000
R26(k0)      = 0x00000000
R27(k1)      = 0x00000000
R28(gp)      = 0x00000000
R29(sp)      = 0x807f0000
R30(fp/s8)   = 0x807f0000

>> d
>>addr: 0xbfc00000
>>num: 16
0xbfc00000: 0x3c088000
0xbfc00004: 0x25081000
0xbfc00008: 0x3c098000
0xbfc0000c: 0x25291190
>> u
>>addr: 0xbfc00000
>>num: 100
0xbfc00000: lui t0,0x8000
0xbfc00004: addiu t0,t0,4096
0xbfc00008: lui t1,0x8000
0xbfc0000c: addiu t1,t1,4496
0xbfc00010: lui t2,0xa000
0xbfc00014: or t0,t0,t2
0xbfc00018: or t1,t1,t2
0xbfc0001c: lui t2,0xbfc0
0xbfc00020: or t2,t2,t0
0xbfc00024: lw t3,0(t2)
0xbfc00028: addiu t2,t2,4
0xbfc0002c: sw t3,0(t0)
0xbfc00030: addiu t0,t0,4
0xbfc00034: bne t0,t1,0xbfc00024
0xbfc00038: nop
0xbfc0003c: lui t0,0x8000
0xbfc00040: addiu t0,t0,8192
0xbfc00044: lui t1,0x8000
0xbfc00048: addiu t1,t1,10244
0xbfc0004c: lui t2,0xa000
0xbfc00050: or t0,t0,t2
0xbfc00054: or t1,t1,t2
0xbfc00058: lui t2,0xbfc0
0xbfc0005c: or t2,t2,t0
0xbfc00060: lw t3,0(t2)

```

图 3.1.2：R 指令运行结果（左），D 指令 & U 指令运行结果（右）

本次测试中主要遇到以下两个问题（初赛）：

问题 1：由于之前功能测试和在性能测试中未使用硬件中断功能，导致 CPU 无法响应外界的串口中断和 flash 中断导致的系统测试卡住，最终我们通过上板抓取信号，在线调试抓取信号，锁定问题为 CPU 无法识别中断信号。

问题 2：由于本次系统测试涉及到与串口和 Flash 芯片的数据交换，我们没有考虑到 cache 访问的数据长度问题，导致访问握手无法成功，解决方案是修改监控程序本身问题的汇编代码，重新编译定位到问题点。

3.2.2 Pmon

在我们重新对 PMON 进行编译的过程中，我们选择 ls_1b 做为目标设备，编译出可以在实验板上正常运行的 PMON，同时通过在源代码中加入诸多调试信息，以及在线调试和反汇编，最终正常运行了 PMON。我们在 PMON 启动时遇到了很多问题：

1. awvalid,awready 信号握手失败，后将 awvalid 保持一拍后握手成功。
2. 当 PMON 正常加载的时候，发现命令行显示如图 3.2.2.1 的 eraseblock 坏块。经 IDA 工具反汇编发现未实现 cache 指令。在实现 cache 指令之后，pmon 正常进入命令行。

```
Bad eraseblock 1016 at 0x07f00000
Bad eraseblock 1017 at 0x07f20000
Bad eraseblock 1018 at 0x07f40000
Bad eraseblock 1019 at 0x07f60000
Bad eraseblock 1020 at 0x07f80000
Bad eraseblock 1021 at 0x07fa0000
Bad eraseblock 1022 at 0x07fc0000
Bad eraseblock 1023 at 0x07fe0000
NANDFlash info:
erasesize      131072 B
writesize      2048 B
oobsize        64 B
```

图 3.2.2.1 坏块产生位置

```
1 unsigned int sub_D88()
2 {
3     unsigned int result; // $v0
4
5     result = 0x80000000;
6     asm
7     {
8         mtc0 $zero, TagLo # Cache Tag register
9         mtc0 $zero, TagHi # Cache Tag register
10    }
11    do
12    {
13        asm
14        {
15            cache 9, 0($v0) # Index Store Tag (primary data or unified primary cache)
16            cache 9, 0x1000($v0) # Index Store Tag (primary data or unified primary cache)
17            cache 9, 0x2000($v0) # Index Store Tag (primary data or unified primary cache)
18            cache 9, 0x3000($v0) # Index Store Tag (primary data or unified primary cache)
19            cache 8, 0($v0) # Index Store Tag (primary instruction cache)
20            cache 8, 0x1000($v0) # Index Store Tag (primary instruction cache)
21            cache 8, 0x2000($v0) # Index Store Tag (primary instruction cache)
22            cache 8, 0x3000($v0) # Index Store Tag (primary instruction cache)
23        }
24        result += 32;
25    }
26    while ( result != -2147479552 );
27    return result;
28 }
```

图 3.2.2.2 IDA 反汇编工具

```
if (ret) {
    this->bbt[i >> 3] |= 0x03 << (i & 0x6);
    printk(KERN_WARNING "Bad eraseblock %d at 0x%08x\n",
           i >> 1, (unsigned int)from);
    mtd->ecc_stats.badblocks++;
}
```

图 3.2.2.3 Pmon 坏块 printk 源代码

3. Load 原本 awsize 固定为 2（即一次事务写 4 字节），spi 外设时进入死循环，后阅读 soc 源码发现 soc 上 spi 在接收 axi 请求时只接受 awsize 为 0 或 1 的信号，于是将 awsize 由固定为 2 改为根据 store 类型推断解决。

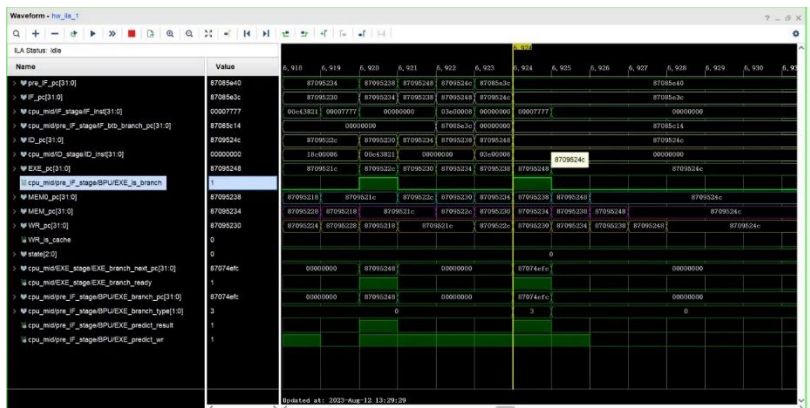


图 3.2.2.4 ILA 抓取波形进入死循环截图

4. 建立网口时无法与外界 ping 通，经修改 Pmon 源码以及测试《CPU 设计实战》的 cache_test 等 debug 方式后发现流水线中一个控制信号实现错误，使得 icache 取回的数据被流水线丢弃。

```
PMON> load tftp://192.168.127.13/ucore
Enter nload
Reach open
Enter open, fname:tftp://192.168.127.13/ucore, dname:tftp://192.168.127.13/ucore
Reach string compare between dname and "/dev/"
Enter strpat
Enter __try_open, fname:tftp://192.168.127.13/ucore, dname:netEnter open, fname:/dev/socket, dname:/dev/socket
Reach string compare between dname and "/dev/"
Enter __try_open, fname:/dev/socket, dname:socket
```

图 3.2.2.5 load Ucore 失败

```
DE4X5_TRBA= 70b8f80
DE4X5_STS= f0660004
DE4X5_OMR= 32002242
PMON> load tftp://192.168.127.13/ucore
Enter nload
Reach open
Enter open, fname:tftp://192.168.127.13/ucore, dname:tftp://192.168.127.13/ucore
Enter strpat
Enter __try_open, fname:tftp://192.168.127.13/ucore, dname:tftp://192.168.127.13/ucore
route1, fsp->open:0x0x8706eea8
Enter open, fname:/dev/socket, dname:/dev/socket
Enter __try_open, fname:/dev/socket, dname:socket
route1, fsp->open:0x0x00000000
Return from __try_open
Return from __try_open
Reach free(fname)
Reach dl_initialize
Loading file: tftp://192.168.127.13/ucore (elf)
0x80000000/544464 + 0x80084ed0/13072(z) + 302 syms/
Entry address is 80000000
PMON> █
```

图 3.2.2.6 load 建立网口连接成功

最终我们可以正常启动 pmon，并运行其中的全部指令。

```
PMON2000 MIPS INITIALIZING. Standby...
ERRSRC=00000000 CONFID=8000008
pid=0004220
init caches... do nothing...
gddont caches found
init caches done. cfp = 80000083

Copy PMON to execute location...
start = 0x0790000
so = 0x3020000
_start = 0x794010
_end = 0x794f028
a300000
copy text section done.
Copy PMON to execute location done.
ip=079f000
uncompressing bios
...executing bios...
OK booting bios
FREE
RTC time invalid, reset to epoch.
FREE
DONE
DEV2
ENV2
MAP2
In envinit
mram=0f00000
wram=0f00000
wram is invalid!
wram=0f00000
STD
BULL0000: memory between 82fff800-83000000 is already been allocated, heap is already above this point
0800
0200
--gpio: gpio status 0
NET
RTCL
in before configure
mainbus0 (front)
localbus at mainbus0
dfe0 at localbus0: address 00:98:76:64:32:19
in if attach
loopdev0 at mainbus0out configure
--before init ps/2 kb
devconf done.
ifinit done.
domaininit done.
init_apmc...
HST
ENV2
S000

# PMON2000 professional +
Configuration [rch.tl.NET]
version PMON2000 2.1 [tl] #363: 2023年05月17日 09:00:00 CST commit: eac6b23ec368d6e948938d7a234b19fa0eb291c Author: qiaochung <qiaochung@loongson.cn> Date: Thu Apr 13 10:13:59 2017 +0800 .
Supported loaders (srec, elf, bin)
Supported filesystems (ind, fat, fs, offsets, fat, fs, disk, socket, tty, ram)
This software may be redistributed under the BSD copyright.
Copyright 2000-2003, digicon AB, Sweden.
Copyright 2005, IGT CAS.
CPU: GE6500 @ 99.99 MHz / Bus @ 99.99 MHz
Memory size 128 MB (128 MB Low memory, 0 MB high memory) .
Primary instruction cache size 16KB (32 line, 2 way)
Primary data cache size 16KB (32 line, 2 way)

BEV1
BEV2
BEV3
BEV4
BEV5
BEV6
BEV7
BEV8
BEV9
BEV10
BEV11
BEV12
BEV13
BEV14
BEV15
BEV16
BEV17
BEV18
BEV19
BEV20
BEV21
BEV22
BEV23
BEV24
BEV25
BEV26
BEV27
BEV28
BEV29
BEV30
BEV31
BEV32
BEV33
BEV34
BEV35
BEV36
BEV37
BEV38
BEV39
BEV40
BEV41
BEV42
BEV43
BEV44
BEV45
BEV46
BEV47
BEV48
BEV49
BEV50
BEV51
BEV52
BEV53
BEV54
BEV55
BEV56
BEV57
BEV58
BEV59
BEV60
BEV61
BEV62
BEV63
BEV64
BEV65
BEV66
BEV67
BEV68
BEV69
BEV70
BEV71
BEV72
BEV73
BEV74
BEV75
BEV76
BEV77
BEV78
BEV79
BEV80
BEV81
BEV82
BEV83
BEV84
BEV85
BEV86
BEV87
BEV88
BEV89
BEV90
BEV91
BEV92
BEV93
BEV94
BEV95
BEV96
BEV97
BEV98
BEV99
BEV100
BEV101
BEV102
BEV103
BEV104
BEV105
BEV106
BEV107
BEV108
BEV109
BEV110
BEV111
BEV112
BEV113
BEV114
BEV115
BEV116
BEV117
BEV118
BEV119
BEV120
BEV121
BEV122
BEV123
BEV124
BEV125
BEV126
BEV127
BEV128
BEV129
BEV130
BEV131
BEV132
BEV133
BEV134
BEV135
BEV136
BEV137
BEV138
BEV139
BEV140
BEV141
BEV142
BEV143
BEV144
BEV145
BEV146
BEV147
BEV148
BEV149
BEV150
BEV151
BEV152
BEV153
BEV154
BEV155
BEV156
BEV157
BEV158
BEV159
BEV160
BEV161
BEV162
BEV163
BEV164
BEV165
BEV166
BEV167
BEV168
BEV169
BEV170
BEV171
BEV172
BEV173
BEV174
BEV175
BEV176
BEV177
BEV178
BEV179
BEV180
BEV181
BEV182
BEV183
BEV184
BEV185
BEV186
BEV187
BEV188
BEV189
BEV190
BEV191
BEV192
BEV193
BEV194
BEV195
BEV196
BEV197
BEV198
BEV199
BEV200
BEV201
BEV202
BEV203
BEV204
BEV205
BEV206
BEV207
BEV208
BEV209
BEV210
BEV211
BEV212
BEV213
BEV214
BEV215
BEV216
BEV217
BEV218
BEV219
BEV220
BEV221
BEV222
BEV223
BEV224
BEV225
BEV226
BEV227
BEV228
BEV229
BEV230
BEV231
BEV232
BEV233
BEV234
BEV235
BEV236
BEV237
BEV238
BEV239
BEV240
BEV241
BEV242
BEV243
BEV244
BEV245
BEV246
BEV247
BEV248
BEV249
BEV250
BEV251
BEV252
BEV253
BEV254
BEV255
BEV256
BEV257
BEV258
BEV259
BEV260
BEV261
BEV262
BEV263
BEV264
BEV265
BEV266
BEV267
BEV268
BEV269
BEV270
BEV271
BEV272
BEV273
BEV274
BEV275
BEV276
BEV277
BEV278
BEV279
BEV280
BEV281
BEV282
BEV283
BEV284
BEV285
BEV286
BEV287
BEV288
BEV289
BEV290
BEV291
BEV292
BEV293
BEV294
BEV295
BEV296
BEV297
BEV298
BEV299
BEV300
BEV301
BEV302
BEV303
BEV304
BEV305
BEV306
BEV307
BEV308
BEV309
BEV310
BEV311
BEV312
BEV313
BEV314
BEV315
BEV316
BEV317
BEV318
BEV319
BEV320
BEV321
BEV322
BEV323
BEV324
BEV325
BEV326
BEV327
BEV328
BEV329
BEV330
BEV331
BEV332
BEV333
BEV334
BEV335
BEV336
BEV337
BEV338
BEV339
BEV340
BEV341
BEV342
BEV343
BEV344
BEV345
BEV346
BEV347
BEV348
BEV349
BEV350
BEV351
BEV352
BEV353
BEV354
BEV355
BEV356
BEV357
BEV358
BEV359
BEV360
BEV361
BEV362
BEV363
BEV364
BEV365
BEV366
BEV367
BEV368
BEV369
BEV370
BEV371
BEV372
BEV373
BEV374
BEV375
BEV376
BEV377
BEV378
BEV379
BEV380
BEV381
BEV382
BEV383
BEV384
BEV385
BEV386
BEV387
BEV388
BEV389
BEV390
BEV391
BEV392
BEV393
BEV394
BEV395
BEV396
BEV397
BEV398
BEV399
BEV400
BEV401
BEV402
BEV403
BEV404
BEV405
BEV406
BEV407
BEV408
BEV409
BEV410
BEV411
BEV412
BEV413
BEV414
BEV415
BEV416
BEV417
BEV418
BEV419
BEV420
BEV421
BEV422
BEV423
BEV424
BEV425
BEV426
BEV427
BEV428
BEV429
BEV430
BEV431
BEV432
BEV433
BEV434
BEV435
BEV436
BEV437
BEV438
BEV439
BEV440
BEV441
BEV442
BEV443
BEV444
BEV445
BEV446
BEV447
BEV448
BEV449
BEV450
BEV451
BEV452
BEV453
BEV454
BEV455
BEV456
BEV457
BEV458
BEV459
BEV460
BEV461
BEV462
BEV463
BEV464
BEV465
BEV466
BEV467
BEV468
BEV469
BEV470
BEV471
BEV472
BEV473
BEV474
BEV475
BEV476
BEV477
BEV478
BEV479
BEV480
BEV481
BEV482
BEV483
BEV484
BEV485
BEV486
BEV487
BEV488
BEV489
BEV490
BEV491
BEV492
BEV493
BEV494
BEV495
BEV496
BEV497
BEV498
BEV499
BEV500
BEV501
BEV502
BEV503
BEV504
BEV505
BEV506
BEV507
BEV508
BEV509
BEV510
BEV511
BEV512
BEV513
BEV514
BEV515
BEV516
BEV517
BEV518
BEV519
BEV520
BEV521
BEV522
BEV523
BEV524
BEV525
BEV526
BEV527
BEV528
BEV529
BEV530
BEV531
BEV532
BEV533
BEV534
BEV535
BEV536
BEV537
BEV538
BEV539
BEV540
BEV541
BEV542
BEV543
BEV544
BEV545
BEV546
BEV547
BEV548
BEV549
BEV550
BEV551
BEV552
BEV553
BEV554
BEV555
BEV556
BEV557
BEV558
BEV559
BEV560
BEV561
BEV562
BEV563
BEV564
BEV565
BEV566
BEV567
BEV568
BEV569
BEV570
BEV571
BEV572
BEV573
BEV574
BEV575
BEV576
BEV577
BEV578
BEV579
BEV580
BEV581
BEV582
BEV583
BEV584
BEV585
BEV586
BEV587
BEV588
BEV589
BEV590
BEV591
BEV592
BEV593
BEV594
BEV595
BEV596
BEV597
BEV598
BEV599
BEV600
BEV601
BEV602
BEV603
BEV604
BEV605
BEV606
BEV607
BEV608
BEV609
BEV610
BEV611
BEV612
BEV613
BEV614
BEV615
BEV616
BEV617
BEV618
BEV619
BEV620
BEV621
BEV622
BEV623
BEV624
BEV625
BEV626
BEV627
BEV628
BEV629
BEV630
BEV631
BEV632
BEV633
BEV634
BEV635
BEV636
BEV637
BEV638
BEV639
BEV640
BEV641
BEV642
BEV643
BEV644
BEV645
BEV646
BEV647
BEV648
BEV649
BEV650
BEV651
BEV652
BEV653
BEV654
BEV655
BEV656
BEV657
BEV658
BEV659
BEV660
BEV661
BEV662
BEV663
BEV664
BEV665
BEV666
BEV667
BEV668
BEV669
BEV670
BEV671
BEV672
BEV673
BEV674
BEV675
BEV676
BEV677
BEV678
BEV679
BEV680
BEV681
BEV682
BEV683
BEV684
BEV685
BEV686
BEV687
BEV688
BEV689
BEV690
BEV691
BEV692
BEV693
BEV694
BEV695
BEV696
BEV697
BEV698
BEV699
BEV700
BEV701
BEV702
BEV703
BEV704
BEV705
BEV706
BEV707
BEV708
BEV709
BEV710
BEV711
BEV712
BEV713
BEV714
BEV715
BEV716
BEV717
BEV718
BEV719
BEV720
BEV721
BEV722
BEV723
BEV724
BEV725
BEV726
BEV727
BEV728
BEV729
BEV730
BEV731
BEV732
BEV733
BEV734
BEV735
BEV736
BEV737
BEV738
BEV739
BEV740
BEV741
BEV742
BEV743
BEV744
BEV745
BEV746
BEV747
BEV748
BEV749
BEV750
BEV751
BEV752
BEV753
BEV754
BEV755
BEV756
BEV757
BEV758
BEV759
BEV760
BEV761
BEV762
BEV763
BEV764
BEV765
BEV766
BEV767
BEV768
BEV769
BEV770
BEV771
BEV772
BEV773
BEV774
BEV775
BEV776
BEV777
BEV778
BEV779
BEV780
BEV781
BEV782
BEV783
BEV784
BEV785
BEV786
BEV787
BEV788
BEV789
BEV790
BEV791
BEV792
BEV793
BEV794
BEV795
BEV796
BEV797
BEV798
BEV799
BEV800
BEV801
BEV802
BEV803
BEV804
BEV805
BEV806
BEV807
BEV808
BEV809
BEV810
BEV811
BEV812
BEV813
BEV814
BEV815
BEV816
BEV817
BEV818
BEV819
BEV820
BEV821
BEV822
BEV823
BEV824
BEV825
BEV826
BEV827
BEV828
BEV829
BEV830
BEV831
BEV832
BEV833
BEV834
BEV835
BEV836
BEV837
BEV838
BEV839
BEV840
BEV841
BEV842
BEV843
BEV844
BEV845
BEV846
BEV847
BEV848
BEV849
BEV850
BEV851
BEV852
BEV853
BEV854
BEV855
BEV856
BEV857
BEV858
BEV859
BEV860
BEV861
BEV862
BEV863
BEV864
BEV865
BEV866
BEV867
BEV868
BEV869
BEV870
BEV871
BEV872
BEV873
BEV874
BEV875
BEV876
BEV877
BEV878
BEV879
BEV880
BEV881
BEV882
BEV883
BEV884
BEV885
BEV886
BEV887
BEV888
BEV889
BEV890
BEV891
BEV892
BEV893
BEV894
BEV895
BEV896
BEV897
BEV898
BEV899
BEV900
BEV901
BEV902
BEV903
BEV904
BEV905
BEV906
BEV907
BEV908
BEV909
BEV910
BEV911
BEV912
BEV913
BEV914
BEV915
BEV916
BEV917
BEV918
BEV919
BEV920
BEV921
BEV922
BEV923
BEV924
BEV925
BEV926
BEV927
BEV928
BEV929
BEV930
BEV931
BEV932
BEV933
BEV934
BEV935
BEV936
BEV937
BEV938
BEV939
BEV940
BEV941
BEV942
BEV943
BEV944
BEV945
BEV946
BEV947
BEV948
BEV949
BEV950
BEV951
BEV952
BEV953
BEV954
BEV955
BEV956
BEV957
BEV958
BEV959
BEV960
BEV961
BEV962
BEV963
BEV964
BEV965
BEV966
BEV967
BEV968
BEV969
BEV970
BEV971
BEV972
BEV973
BEV974
BEV975
BEV976
BEV977
BEV978
BEV979
BEV980
BEV981
BEV982
BEV983
BEV984
BEV985
BEV986
BEV987
BEV988
BEV989
BEV990
BEV991
BEV992
BEV993
BEV994
BEV995
BEV996
BEV997
BEV998
BEV999
BEV1000
BEV1001
BEV1002
BEV1003
BEV1004
BEV1005
BEV1006
BEV1007
BEV1008
BEV1009
BEV1010
BEV1011
BEV1012
BEV1013
BEV1014
BEV1015
BEV1016
BEV1017
BEV1018
BEV1019
BEV1020
BEV1021
BEV1022
BEV1023
BEV1024
BEV1025
BEV1026
BEV1027
BEV1028
BEV1029
BEV1030
BEV1031
BEV1032
BEV1033
BEV1034
BEV1035
BEV1036
BEV1037
BEV1038
BEV1039
BEV1040
BEV1041
BEV1042
BEV1043
BEV1044
BEV1045
BEV1046
BEV1047
BEV1048
BEV1049
BEV1050
BEV1051
BEV1052
BEV1053
BEV1054
BEV1055
BEV1056
BEV1057
BEV1058
BEV1059
BEV1060
BEV1061
BEV1062
BEV1063
BEV1064
BEV1065
BEV1066
BEV1067
BEV1068
BEV1069
BEV1070
BEV1071
BEV1072
BEV1073
BEV1074
BEV1075
BEV1076
BEV1077
BEV1078
BEV1079
BEV1080
BEV1081
BEV1082
BEV1083
BEV1084
BEV1085
BEV1086
BEV1087
BEV1088
BEV1089
BEV1090
BEV1091
BEV1092
BEV1093
BEV1094
BEV1095
BEV1096
BEV1097
BEV1098
BEV1099
BEV1100
BEV1101
BEV1102
BEV1103
BEV1104
BEV1105
BEV1106
BEV1107
BEV1108
BEV1109
BEV1110
BEV1111
BEV1112
BEV1113
BEV1114
BEV1115
BEV1116
BEV1117
BEV1118
BEV1119
BEV1120
BEV1121
BEV1122
BEV1123
BEV1124
BEV1125
BEV1126
BEV1127
BEV1128
BEV1129
BEV1130
BEV1131
BEV1132
BEV1133
BEV1134
BEV1135
BEV1136
BEV1137
BEV1138
BEV1139
BEV1140
BEV1141
BEV1142
BEV1143
BEV1144
BEV1145
BEV1146
BEV1147
BEV1148
BEV1149
BEV1150
BEV1151
BEV1152
BEV1153
BEV1154
BEV1155
BEV1156
BEV1157
BEV1158
BEV1159
BEV1160
BEV1161
BEV1162
BEV1163
BEV1164
BEV1165
BEV1166
BEV1167
BEV1168
BEV1169
BEV1170
BEV1171
BEV1172
BEV1173
BEV1174
BEV1175
BEV1176
BEV1177
BEV1178
BEV1179
BEV1180
BEV1181
BEV1182
BEV1183
BEV1184
BEV1185
BEV1186
BEV1187
BEV1188
BEV1189
BEV1190
BEV1191
BEV1192
BEV1193
BEV1194
BEV1195
BEV1196
BEV1197
BEV1198
BEV1199
BEV1200
BEV1201
BEV1202
BEV1203
BEV1204
BEV1205
BEV1206
BEV1207
BEV1208
BEV1209
BEV1210
BEV1211
BEV1212
BEV1213
BEV1214
BEV1215
BEV1216
BEV1217
BEV1218
BEV1219
BEV1220
BEV1221
BEV1222
BEV1223
BEV1224
BEV1225
BEV1226
BEV1227
BEV1228
BEV1229
BEV1230
BEV1231
BEV1232
BEV1233
BEV1234
BEV1235
BEV1236
BEV1237
BEV1238
BEV1239
BEV1240
BEV1241
BEV1242
BEV1243
BEV1244
BEV1245
BEV1246
BEV1247
BEV1248
BEV1249
BEV1250
BEV1251
BEV1252
BEV1253
BEV1254
BEV1255
BEV1256
BEV1257
BEV1258
BEV1259
BEV1260
BEV1261
BEV1262
BEV1263
BEV1264
BEV1265
BEV1266
BEV1267
BEV1268
BEV1269
BEV1270
BEV1271
BEV1272
BEV1273
BEV1274
BEV1275
BEV1276
BEV1277
BEV1278
BEV1279
BEV1280
BEV1281
BEV1282
BEV1283
BEV1284
BEV1285
BEV1286
BEV1287
BEV1288
BEV1289
BEV1290
BEV1291
BEV1292
BEV1293
BEV1294
BEV1295
BEV1296
BEV1297
BEV1298
BEV1299
BEV1300
BEV1301
BEV1302
BEV1303
BEV1304
BEV1305
BEV1306
BEV1307
BEV1308
BEV1309
BEV1310
BEV1311
BEV1312
BEV1313
BEV1314
BEV1315
BEV1316
BEV1317
BEV1318
BEV1319
BEV1320
BEV1321
BEV1322
BEV1323
BEV1324
BEV1325
BEV1326
BEV1327
BEV1328
BEV1329
BEV1330
BEV1331
BEV1332
BEV1333
BEV1334
BEV1335
BEV1336
BEV1337
BEV1338
BEV1339
BEV1340
BEV1341
BEV1342
BEV1343
BEV1344
BEV1345
BEV1346
BEV1347
BEV1348
BEV1349
BEV1350
BEV1351
BEV1352
BEV1353
BEV1354
BEV1355
BEV1356
BEV1357
BEV1358
BEV1359
BEV1360
BEV1361
BEV1362
BEV1363
BEV1364
BEV1365
BEV1366
BEV1367
BEV1368
BEV1369
BEV1370
BEV1371
BEV1372
BEV1373
BEV1374
BEV1375
BEV1376
BEV1377
BEV1378
BEV1379
BEV1380
BEV1381
BEV1382
BEV1383
BEV1384
BEV1385
BEV1386
BEV1387
BEV1388
BEV1389
BEV1390
BEV1391
BEV1392
BEV1393
BEV1394
BEV1395
BEV1396
BEV1397
BEV1398
BEV1399
BEV1400
BEV1401
BEV1402
BEV1403
BEV1404
BEV1405
BEV1406
BEV1407
BEV1408
BEV1409
BEV1410
BEV1411
BEV1412
BEV1413
BEV1414
BEV1415
BEV1416
BEV1417
BEV1418
BEV1419
BEV1420
BEV1421
BEV1422
BEV1423
BEV1424
BEV1425
BEV1426
BEV1427
BEV1428
BEV1429
BEV1430
BEV1431
BEV1432
BEV1433
BEV1434
BEV1435
BEV1436
BEV1437
BEV1438
BEV1439
BEV1440
BEV1441
BEV1442
BEV1443
BEV1444
BEV1445
BEV1446
BEV1447
BEV1448
BEV1449
BEV1450
BEV1451
BEV1452
BEV1453
BEV1454
BEV1455
BEV1456
BEV1457
BEV1458
BEV1459
BEV1460
BEV1461
BEV1462
BEV1463
BEV1464
BEV1465
BEV1466
BEV1467
BEV1468
BEV1469
BEV1470
BEV1471
BEV1472
BEV1473
BEV1474
BEV1475
BEV1476
BEV1477
BEV1478
BEV1479
BEV1480
BEV1481
BEV1482
BEV1483
BEV1484
BEV1485
BEV1486
BEV1487
BEV1488
BEV1489
BEV1490
BEV1491
BEV1492
BEV1493
BEV1494
BEV1495
BEV1496
BEV1497
BEV1498
BEV1499
BEV1500
BEV1501
BEV1502
BEV1503
BEV1504
BEV1505
BEV1506
BEV1507
BEV1508
BEV1509
BEV1510
BEV1511
BEV1512
BEV1513
BEV1514
BEV1515
BEV1516
BEV1517
BEV1518
BEV1519
BEV1520
BEV1521
BEV1522
BEV1523
BEV1524
BEV1525
BEV1526
BEV1527
BEV1528
BEV1529
BEV1530
BEV1531
BEV1532
BEV1533
BEV1534
BEV1535
BEV1536
BEV1537
BEV1538
BEV1539
BEV1540
BEV1541
BEV1542
BEV1543
BEV1544
BEV1545
BEV1546
BEV1547
BEV1548
BEV1549
BEV1550
BEV1551
BEV1552
BEV1553
BEV1554
BEV1555
BEV1556
BEV1557
BEV1558
BEV1559
BEV1560
BEV1561
BEV1562
BEV1563
BEV1564
BEV1565
BEV1566
BEV1567
BEV1568
BEV1569
BEV1570
BEV1571
BEV1572
BEV1573
BEV1574
BEV1575
BEV1576
BEV1577
BEV1578
BEV1579
BEV1580
BEV1581
BEV1582
BEV1583
BEV1584
BEV1585
BEV1586
BEV1587
BEV1588
BEV1589
BEV1590
BEV1591
BEV1592
BEV1593
BEV1594
BEV1595
BEV1596
BEV1597
BEV1598
BEV1599
BEV1600
BEV1601
BEV1602
BEV1603
BEV1604
BEV1605
BEV1606
BEV1607
BEV1608
BEV1609
BEV1610
BEV1611
BEV1612
BEV1613
BEV1614
BEV1615
BEV1616
BEV1617
BEV1618
BEV1619
BEV1620
BEV1621
BEV1622
BEV1623
BEV1624
BEV1625
BEV1626
BEV1627
BEV1628
BEV1629
BEV1630
BEV1631
BEV1632
BEV1633
BEV1634
BEV1635
BEV1636
BEV1637
BEV1638
BEV1639
BEV1640
BEV1641
BEV1642
BEV1643
BEV1644
BEV1645
BEV1646
BEV1647
BEV1648
BEV1649
BEV1650
BEV1651
BEV1652
BEV1653
BEV1654
BEV1655
BEV1656
BEV1657
BEV1658
BEV1659
BEV1660
BEV1661
BEV1662
BEV1663
BEV1664
BEV1665
BEV1666
BEV1667
BEV1668
BEV1669
BEV1670
BEV1671
BEV1672
BEV1673
BEV1674
BEV1675
BEV1676
BEV1677
BEV1678
BEV1679
BEV1680
BEV1681
BEV1682
BEV1683
BEV1684
BEV1685
BEV1686
BEV1687
BEV1688
BEV1689
BEV1690
BEV1691
BEV1692
BEV1693
BEV1694
BEV1695
BEV1696
BEV1697
BEV1698
BEV1699
BEV1700
BEV1701
BEV1702
BEV1703
BEV1704
BEV1705
BEV1706
BEV1707
BEV1708
BEV1709
BEV1710
BEV1711
BEV1712
BEV1713
BEV1714
BEV1715
BEV1716
BEV1717
BEV1718
BEV1719
BEV1720
BEV1721
BEV1722
BEV1723
BEV1724
BEV1725
BEV1726
BEV1727
BEV1728
BEV1729
BEV1730
BEV1731
BEV1732
BEV1733
BEV1734
BEV1735
BEV1736
BEV1737
BEV1738
BEV1739
BEV1740
BEV1741
BEV1742
BEV1743
BEV1744
BEV1745
BEV1746
BEV1747
BEV1748
BEV1749
BEV1750
BEV1751
BEV1752
BEV1753
BEV1754
BEV1755
BEV1756
BEV1757
BEV1758
BEV1759
BEV1760
BEV1761
BEV1762
BEV1763
BEV1764
BEV1765
BEV1766
BEV1767
BEV1768
BEV1769
BEV1770
BEV1771
BEV1772
BEV1773
BEV1774
BEV1775
BEV1776
BEV1777
BEV1778
BEV1779
BEV1780
BEV1781
BEV1782
BEV1783
BEV1784
BEV1785
BEV1786
BEV1787
BEV1788
BEV1789
BEV1790
BEV1791
BEV1792
BEV1793
BEV1794
BEV1795
BEV1796
BEV1797
BEV1798
BEV1799
BEV1800
BEV1801
BEV1802
BEV1803
BEV1804
BEV1805
BEV1806
BEV1807
BEV1808
BEV1809
BEV1810
BEV1811
BEV1812
BEV1813
BEV1814
BEV1815
```



```

kernel_execve: pid = 2, name = "sh".
not valid addr c000006c, and can not find it in vma
trapframe at BFFFFFF38
$1 : 00000000
$2 : 10004000
$3 : 81F0E007
$4 : 00000000
$5 : 10004000
$6 : 0007C35F
$7 : 0007C39F
$8 : FFFFFFFE4
$9 : 00008800
$10 : 100040A0
$11 : 00000000
$12 : 00000000
$13 : 00000000
$14 : 00000000
$15 : 00000000
$16 : 0007C35F
$17 : 10004000
$18 : 00000000
$19 : 00000000
$20 : 00000000
$21 : 00000000
$22 : 00000000
$23 : 00000000
$24 : 00000000
$25 : 800074A8
$26 : 4000800C
$27 : BFFFFFFCC
$28 : 00000000
$29 : BFFFFFFCC
$30 : 10003EB4
$ra : 800071F0
BadVA : C000006C
Status : 00008802
Cause : 4000800C
EPC : 80007228
Trap in kernel: TLB miss on store
kernel panic at kern/trap/trap.c:206:
unhandled pgfault
welcome to the kernel debug monitor!!
Type 'help' for a list of commands.
K> ojnnnnnnnnnnnkgvkgvkuhgvukvgugvuuuuuuu

```

图 3.2.3.1 ucore 进入 trap in kernel

2. 未实现用户态导致进入时报 tlb 异常。我们发现在 2021 年参赛组 GenshinCPU 中遇到过相同的问题，于是我们实现了 Status 寄存器的 um、cu0 域，而后 Ucore 启动成功。下图是我们正确进入 Ucore 的截图：

```

[00000000, 02000000]
freemem start at: 800C9000
free pages: 00001F37
## 00000020
check_alloc_page() succeeded!
check_pgdir() succeeded!
check_boot_pgdir() succeeded!
----- BEGIN -----
----- END -----
check_slab() succeeded!
kmalloc_init() succeeded!
check_vma_struct() succeeded!
check_pgfault() succeeded!
check_vmm() succeeded.
sched class: RR_scheduler
ramdisk_init(): initrd found, magic: 0x2f8dbe2a, 0x000002bc secs
sfs: mount: 'simple file system' (81/6/87)
vfs: mount disk0.
kernel_execve: pid = 2, name = "sh".
user sh is running!!!
$ ls
@ is [directory] 2(hlinks) 6(blocks) 1536(bytes) : @'.'
[d] 2(h) 6(b) 1536(s) .
[d] 2(h) 6(b) 1536(s) ..
[-] 1(h) 22(b) 89501(s) sh
[-] 1(h) 21(b) 85360(s) ls
[-] 1(h) 21(b) 85234(s) cat
[-] 1(h) 1(b) 21(s) test.txt
lsdir: step 4
$ cat test.txt
hello world! Haha...
$ █

```

图 3.2.3.2 Ucore 成功进入

```

PHON: g
zero at v0 v1 a0 a1 a2 a3
00000000 00000000 00000000 00000001 a7dfda8 a7dfdb0 870af5b0
t0 t1 t2 t3 t4 t5 t6 t7
00000000 00000000 00000000 00000000 00000000 00000000 00000000
s0 s1 s2 s3 s4 s5 s6 s7
00000000 00000000 00000000 00000000 00000000 00000000 00000000
t8 t9 k0 s8 r4
00000000 00000000 00000000 00000000 a7dfda8 00000000 8707ad48
tlb invalidated
console is initrd
++setup timer interrupts
initrd: 0x8002b400 - 0x80084edf, size: 0x00057800, magic: 0x2f8db2ea
(THU.CST) os is loading ...

Special kernel symbols:
entry 0x80000108 (phys)
etext 0x8002b400 (phys)
edata 0x80084ed0 (phys)
end 0x80088100 (phys)
kernel executable memory footprint: 372KB
enter pmm_init
memory management: buddy_pmm_manager
memory map:
[80000000, 82000000]

freemem start at: 800c9000
free pages: 00001f37
## 0000020
check_alloc_page() succeeded!
check_pgdir() succeeded!
reach check_boot_pgdir
check_boot_pgdir() succeeded!
out check_boot_pgdir----- BEGIN -----
check_slab() succeeded!
kalloc_init() succeeded!
enter vmm_init
check_vma_struct() succeeded!
check_pgfault() succeeded!
check_vmm() succeeded.
enter sched_init
sched class: RR_scheduler
enter proc_init
ramdisk_init(): initrd found, magic: 0x2f8db2ea, 0x000002bc secs
sfs: mount: simple file system (81/6/87)
vfs: mount disk0.

IN P U O O N S E S E E
WE L C O M E T O U N I V E R S I T Y

enter idle
kernel_execve: pid = 2, name = "sh".
user sh is running!!!
$ ls
0 is [directory] 2(hlinks) 6(blocks) 1536(bytes) : 0'.
[d] 2(h) 6(b) 1536(s) .
[d] 2(h) 6(b) 1536(s) ..
[.] 1(h) 22(b) 89501(s) sh
[.] 1(h) 21(b) 85360(s) ls
[.] 1(h) 21(b) 85234(s) cat
[.] 1(h) 1(b) 21(s) test.txt
lsdir: step 4
$ cat test.txt.
hello world! Haha...
$ █

```

图 3.2.3.1 加入 Logo 的 Ucore 启动截图

3.2.4 Linux

在启动 linux 时，我们遇到许多问题，主要是 kernel panic 与我们的代码设计有关，需要修改，但是因为时间有限，很遗憾最终我们没能完成。下图是我们 Linux 启动的最新进展。

```

CPU 0 Unable to handle kernel paging request at virtual address 40000004, epc == 802b4af8, ra == 802b5de4
oops[#1]:
Cpu 0
$ 0 : 00000000 10000000 874015d4 874015d4
$ 4 : 40000000 8781ed24 87401594 00000000
$ 8 : 0000002f 00000000 00000001 8781e01c
$12 : 8780a380 ffffffff 8780a388 8780a390
$16 : 87401594 8781ed24 87813a00 80285790
$20 : 8781bf40 00000000 00000000 00000000
$24 : 00000000 00000000
$28 : 8781a080 8781be98 00000000 802b5de4
Hi : 00000000
Lo : 00000000
epc : 802b4af8 __d_instantiate+0x10/0x84
Not tainted
ra : 802b5de4 d_alloc_root+0x48/0x7c
Status: 10000003 KERNEL EXL IE
Cause: 4000800c
BadVA : 40000004
PrId : 00004220 (ICT Loongson-232)
Modules linked in:
Process (pid: -2021688F) threadinfo=8781a000, task=87803508, tls=8781a080)
Stack : 87813a00 87802400 8781bf40 87813a00 80285790 87802400 8781ed24 802858f4
8781bf40 80285790 8781bf40 00000000 00400000 87813a00 8781bf40 802a2be8
87813c00 00000000 802a1bb0 80598c44 00400000 8780d5a0 8055d0d0 80598c44
00400000 802a190c 806a0000 00000000 00000000 802a1940 8780d5a0 87802680
00000000 805a0000 805c3c60 00000000 00000000 805b02ac 805c3c60 80590000

Call Trace:
[<802b4af8>] __d_instantiate+0x10/0x84
[<802b5de4>] d_alloc_root+0x48/0x7c
[<802858f4>] shmem_fill_super+0x164/0x1a8
[<802a2be8>] get_sb_single+0xd0/0xfc
[<802a190c>] vfs_kern_mount+0x5c/0x100
[<805b02ac>] devtmpfs_init+0x50/0xc4
[<805b01f4>] driver_init+0x10/0x4c
[<805a0318>] kernel_init+0x94/0x160
[<80209e74>] kernel_thread_helper+0x10/0x18

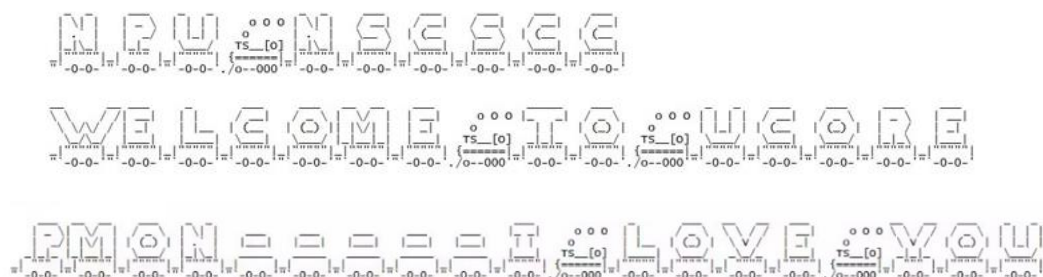
Code: 00803021 8ca40018 24c20040 <ac820004> acc40040 aca20018 8cc30018 acc5000c 8c63000c
Disabling lock debugging due to kernel taint
Kernel panic - not syncing: Fatal exception in interrupt

```

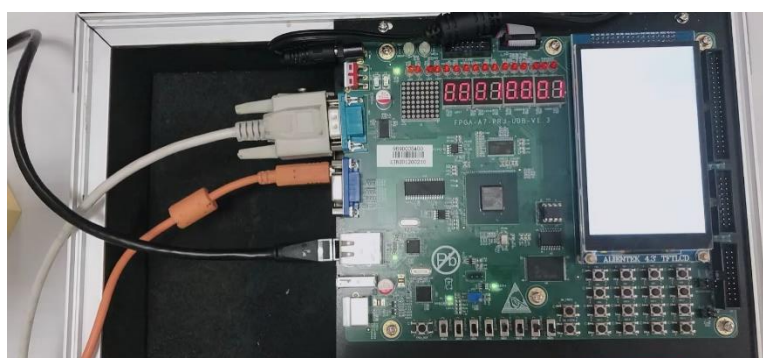

3.2.5 外设支持

通过修改 PMON, UCORE 内核, 添加自己的 logo 和特殊含义的字符画, 比如 PMON, UCORE 相关的字符画, 将图片转为 coe 格式实现图片在试验箱上的加载。

运行效果如下图所示:



开机之后我们可以支持显示屏, 这里借鉴国科大的 LCD 实现方式, 数码管成功显示说明可以加载图片, 但是由于实验箱故障, 我们的 logo 无法正常加载。



第四部分 开发支持工具

4.1 Verilator+NEMU

Verilator 是一个开源的周期精确 Verilog/SystemVerilog 仿真器, 速度相较 Vivado 要快十几倍。在 Vivado 中性能测试的 Coremark 需要运行十分钟以上才能完成, 而在 Verilator 中在仿真模型编译完成的情况下只需要数秒。

NEMU 是南京大学开源的一项全系统模拟器, 可以在指令级别上模拟处理器的指令过程。我们此次沿用的 NEMU 版本是一生一芯第五期过程中 PA 实验所完成的 RISC-V 版本 NEMU 扩展而来, 具备一个简单调试器 (SDB)、itrace、ftrace 等调试功能, 且可以通过 C 语言的接口对外用于差分测试。

我们将原本的 CPU 代码稍作修改即可用于 Verilator 仿真，主要将一些在 Vivado 中使用的 ip 核替换为 Verilog 编写的模块，如 SRAM 用 FF 模拟、乘法器和除法器直接使用 * 和 / 进行功能上的验证等。用于存放指令和数据的 RAM 则通过 C++ 中的数组进行模拟，然后在 Verilog 中使用 DPI-C 对其进行读写，从而模拟访存行为。

一生一芯提供的 NAVY-APPS 和 AM 里有许多测例和应用，NEMU 的差分测试和基础设施可以很方便地帮助我们定位到 CPU 功能上的错误。

```

/home/seven7/Documents/学业/Kurui111/npc/csrc/monitor/init.cpp:56 init_npc] npc initialization finish.
(npc) c
/home/seven7/Documents/学业/Kurui111/npc/csrc/cpu/gpr.cpp:30 diff_test_checkregs] DiffTest found v0 value is 0x0, should be 0xaabbcc00
No.  Name
0   $0: 0000000000000000
1   at: 0000000000000000
2   v0: 0000000000000000
3   v1: 0000000000000000
4   a0: 00000000fc00318
5   a1: 0000000000000000
6   a2: 0000000000000000
7   a3: 0000000000000000
8   t0: 0000000000000000
9   t1: 0000000000000000
10  t2: 0000000000000000
11  t3: 0000000000000000
12  t4: 0000000000000000
13  t5: 0000000000000000
14  t6: 0000000000000000
15  t7: 0000000000000000
16  s0: 00000000fc0033c
17  s1: 0000000000000000
18  s2: 00000000aabbccdd
19  s3: 00000000fc000000
20  s4: 000000005443323
21  s5: 00000000fc000000
22  s6: 0000000000000000
23  s7: 0000000000000000
24  t8: 0000000000000000
25  t9: 0000000000000000
26  k0: 0000000000000000
27  k1: 0000000000000000
28  gp: 0000000000000000
29  sp: 00000000fc08fb8
30  s8: 0000000000000000
31  ra: 00000000fc00308
/home/seven7/Documents/学业/Kurui111/npc/csrc/cpu-exec.cpp:124 execute] Recieve 130 bubbles in pipeline running
NPC ABORT at pc = bfc00280
Itrace Info : q
    0xbfc0025c::afb0014 sw      $16, 0x14($sp)
    0xbfc00260::afb002c sw      $ra, 0x2c($sp)
    0xbfc00264::2411004 addiu    $17, $zero, 0x4
    0xbfc00268::257003c addiu    $16, $19, 0x33c
    0xbfc0026c::34150fc0 lui     $21, 0xbfc0
    0xbfc00270::3452ccdd ori     $18, $18, 0xccdd
    0xbfc00274::24943323 addiu    $20, $20, 0x3323
    0xbfc00278::aa12006 swl      $18, 0x6($16)
    0xbfc0027c::ba12003 swr      $18, 0x3($16)
--> 0xbfc00280::8a02006 lwl      $2, 0x6($16)
Itrace Info End
(npc) []

```

图 4.1 NEMU 差分测试

```

lower-case load-store select-sort mov-c
[ dummy] PASS!
[ movsx] PASS!
[ recursion] PASS!
[ fib] PASS!
[ matrix-mul] PASS!
[ wanshu] PASS!
[ goldbach] PASS!
[ sum] PASS!
[ min3] FAIL!
[ sub-longlong] PASS!
[ bubble-sort] PASS!
[ quick-sort] PASS!
[ leap-year] PASS!
[ bit] PASS!
[ add] PASS!
[ switch] PASS!
[ add-longlong] PASS!
[ div] PASS!
[ shuixianhua] PASS!
[ pascal] PASS!
[ unalign] FAIL!
[ string] PASS!
[ hello-str] FAIL!
[ fact] PASS!
[ prime] PASS!
[ to-lower-case] PASS!
[ load-store] PASS!
[ select-sort] FAIL!
[ mov-c] PASS!
[ if-else] FAIL!
[ mul-longlong] PASS!
[ shift] PASS!
[ max] FAIL!

```

```

Welcome to mips32-NEMU!
For help, type "help"
log:/home/seven7/Documents/学业/KuruiI11/nemu/build/nemu-log.txt
(nemu) c
#
SHABI PHON MIPS Initializing. Standby...
ERRORPC=00000000 CONFIG=00000000
PRID=00004220
Init caches... do noting...
godson1 caches found
Init caches done, cfg = 00000000

Copy PMON to execute location...
start = 0x87900000
s0 = 0x38300000
_edata = 0x8794e010
_end = 0x8794f028
a7940000
copy text section done.
Copy PMON to execute location done.
sp=878fc000
Uncompressing Bios
.....
OK,Booting Bios
FREQ
RTC time invalid, reset to epoch.
FREI
DONE
DEVI
ENVI
MAPV
in envinit
nvram=bfc00000
NVRAM is invalid!
NVRAM@bfc00000
STDV
80100000: memory between 82fff800-83000000 is already been allocated,heap is already above this point
SBD0
DINI
==gpio: gpio status 0
NETI
RTCL
===before configure
in configure
mainbus0 (root)
localbus0 at mainbus0
dmfe0 at localbus0: address 00:98:76:64:32:19
in if attach
loopdev0 at mainbus0out configure

```

图 4.2 NEMU 运行一生一芯测例、PMON

除了运行功能测试以外，NEMU 还能够运行部分系统测试。我们在 NEMU 上模拟了实验箱 SOC 当中的串口外设，实现了 UART_DAT 和 UART_LSR 寄存器，当往 CPU 往 UART_DAT 写时即调用 printf 将其输出，达到模拟串口输出的效果。当 CPU 读取 UART_LSR 寄存器的值时固定返回 0x20，代表此时串口的 TX_FIFO 内无内容，可以发送。此外，我们还实现一个 C 语言的函数 uart_write 用于模拟串口向 CPU 发送数据，输入的数据先存放在一个队列内，当 CPU 读取时将其弹出。

NEMU 还可以运行 PMON 到初始化结束进入命令行前的阶段。由于不知道有关 NAND FLASH 控制的具体方法，因此需要将 PMON 内初始化 NAND 的函数注释才能够继续运行。

```

====before init ps/2 kbd
devconfig done.
ifinit done.
domaininit done.
init_proc....
HSTI
SWRt
SBD0

* PMON2000 Professional *
Configuration [FCR,EL,NET]
Version: PMON2000 2.1 (ls1b) #363: 2023000 00000 13000 000000000 05:09:00 CST commit each743ec368d0e948938d7a234b19fa0eb291c Author: QiaoChong <qiaochong@openwrt.org>
Supported loaders [srec, elf, bin]
Supported filesystems [mtd, net, fs/yaffs2, fat, fs, disk, socket, tty, ram]
This software may be redistributed under the BSD copyright.
Copyright 2000-2002, Opsyscon AB, Sweden.
Copyright 2005, ICT CAS.
CPU G0pSON1 @ 15.82 MHz / Bus @ 24.34 MHz
Memory size 128 MB (128 MB Low memory, 0 MB High memory) .
Primary Instruction cache size 8Kb (2 line, 1 way)
Primary Data cache size 8Kb (2 line, 1 way)

BEV1
BEV2
BEV3
BEV0
BEV in SR set to zero.

```

图 4.3 NEMU 运行 PMON

由于我们实现的 NEMU 中的 TLB 实现还存在 bug，因此未能直接在 NEMU 中启动 ucore，但也能够通过跳过 TLB 指令的比对并将 CPU 的执行结果拷贝到 NEMU 中来对其余指令的执行进行差分测试。

第五部分 附录

4.1 参考资料

MIPS[®] Architecture For Programmers Volume I- A: Introduction to the MIPS32[™] Architecture, rev3.02

MIPS[®] Architecture For Programmers Volume II- A: The MIPS32[™] Instruction Set

MIPS[®] Architecture For Programmers Volume III: The MIPS32[™] and microMIPS32[™] Privileged Resource Architecture

Xilinx IP: AXI Crossbar (2.1) LogiCORE IP Product Guide

4.2 参考仓库

4.2.1 17 年参赛作品

清华大学 NaiveMIPS
<https://github.com/z4yx/NaiveMIPS-HDL>

4.2.2 19 年参赛作品

中国科学院大学 2 队
<https://github.com/nscscc2019ucas/nscscc2019ucas>

4.2.3 21 年参赛作品

西北工业大学一队 GenshinCPU
<https://gitee.com/seddon2001/genshin-cpu/tree/master>

4.2.4 其他

一生一芯 NutShell
<https://github.com/OSCPU/NutShell>

4.3 参考文献

- [1] 戴维 A. 帕特森. 计算机组成与设计：硬件软件接口. 机械工业出版社. 2015.
- [2] Sarah L. Harris, David Harris. Digital Design and Computer Architecture RISC-V Edition. Morgan Kaufmann. 2021.
- [3] 汪文祥, 邢金璋. CPU 设计实战. 机械工业出版社. 2022.

第五部分 致谢

衷心感谢王党辉，安建峰，张萌三位老师的辛勤指导，感谢我们的学长学魏天昊，陈思源在项目过程中提供的大力帮助。

感谢 NSCSCC 群友给与我们调试过程中的建议与帮助。

感谢一生一芯项目对我们调试工作带来的启发。

感谢西北工业大学 loongarch 一队队员和康继昌智能系统班老师和同学给予我们的激励与帮助。