

2  20

龙芯杯全国大学生计算机系统能力培养大赛

复旦大学 二队

韩晓宇 于凡奇 力维辰 马逸君

汇报时间：2020.08.20





01 / 项目概括

02 / CPU实现

03 / 系统实现

04 / 附录



PART ONE

1

项目概览

开发日程

Development schedule



7.9

前期分工协商



7.21

通过功能测试



7.22

通过性能测试



7.22~8.5

优化性能&尝试双发射



8.6~8.14

添加额外指令



8.14~8.17

启动操作系统

项目内容

Project Information

性能分	IPC比值	28.690
	CPU频率	95MHz

本项目主要分为两个部分，一个是CPU部分，一个是系统软件部分。

在CPU部分，我们设计了一个MIPS架构的五级流水线单发射CPU，同时支持Cache和TLB，能够通过大赛方给的全部测试，其中，初赛提交版本性能分数为48.9分，决赛提交版本频率为95MHz，IPC分数为28.690分。

在系统软件部分，我们将我们的CPU与官方给的文件相结合，成功搭建了一个soc，成功启动了Pmon，并运行至Ucore的debug界面及linux的版本信息部分。

序号	测试程序	myCPU			gs132	IPC _{mycpu} /IPC _{gs132}
		上板计时(16进制)		CPU count*2 : SoC count	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		95MHz : 100MHz		-	50MHz : 100MHz	-
1	bitcount	295d0	57550	0.947264839	4E3DD2	30.26509822
2	bubble_sort	dee10	01d68d2	0.947309765	1EF74EA	35.56777652
3	coremark	26bda6	051c9a9	0.947350216	43399B0	27.76400971
4	crc32	15ddb9	02e2991	0.947356243	2A86A88	31.11739219
5	dhystone	6e186	0e873e	0.947248578	7F000A	18.45677348
6	quick_sort	e8a83	01eb2c3	0.947353248	1C65821	31.2456045
7	select_sort	e181a	01dc13f	0.947350661	1B7FFF2	31.21861826
8	sha	e2bc3	01deacc	0.947345067	1D2E296	32.94664733
9	stream_copy	1121b	242d0	0.947130439	214F0D	31.10864887
10	stringsearch	ea5fb	1eecbc	0.947355384	14286C6	22.01792093



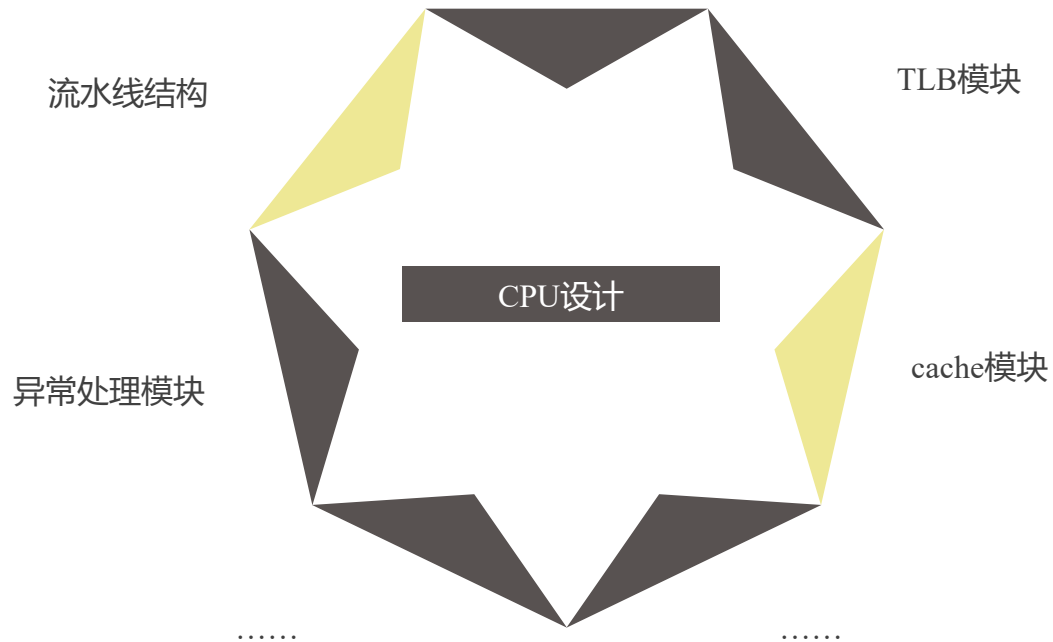
PART TWO

2

CPU实现

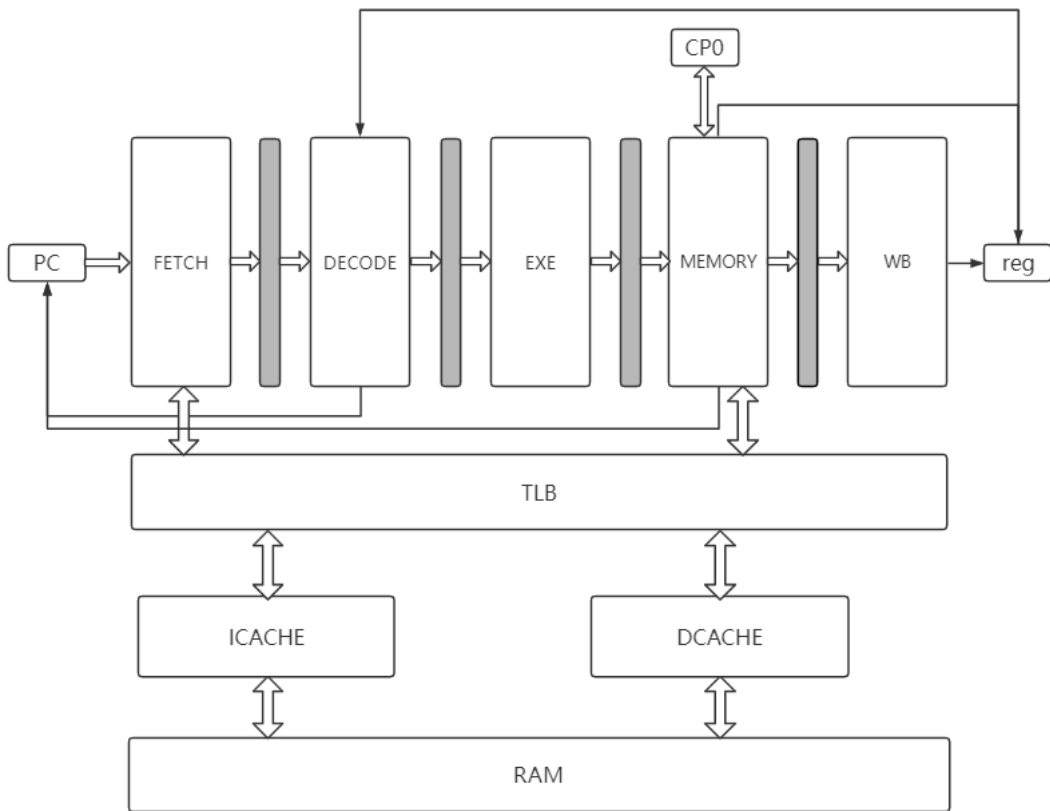
CPU设计概览

CPU design overview



流水线结构

Pipeline structure



经典五级流水线单发射结构

取指, 译码, 执行,
访存, 写回

冲突处理: 数据旁路, 暂停

结构特点

Strengths of structure



1

一级取指，降低分支预测错误的代价：准确的分支预测器虽然能够在很多情况下降低多级取指的代价，但对于jr指令等情况无能为力

2

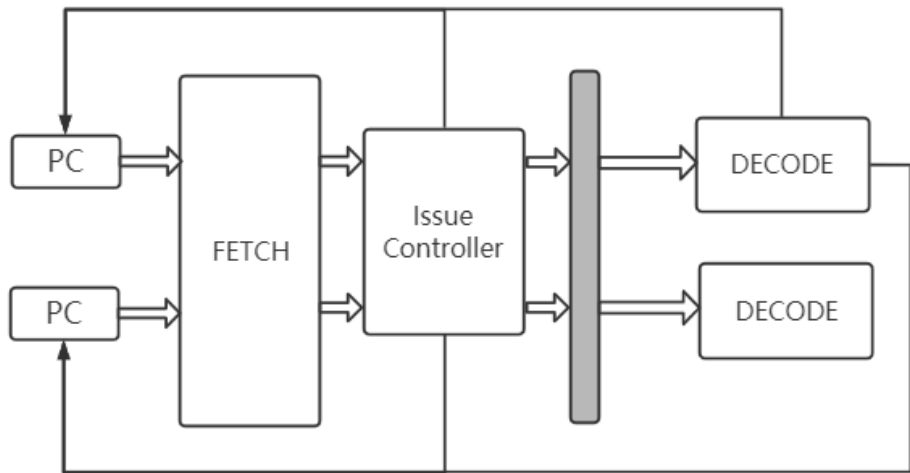
采用“分支总是跳转”的策略，在70%的情况下能够减少一拍的分支等待时间

3

将结构功能较为复杂但使用频率较低的CP0独立于主流流水线之外，提高主流流水线运行的效率

双发射

Dual issue



- ◆ 分为主流水线 and 次流水线
- ◆ Fetch阶段每次取两条指令，再利用issue controller判断是否可以进行双发射，再根据判断情况发射
- ◆ 利用buffer解决寄存器比对问题

缺陷：

1. 在未分多级流水的情况下主频较低
2. 并非任意两条指令都能双发，实际不少指令都是单发
3. 对于分支指令处理起来比较麻烦

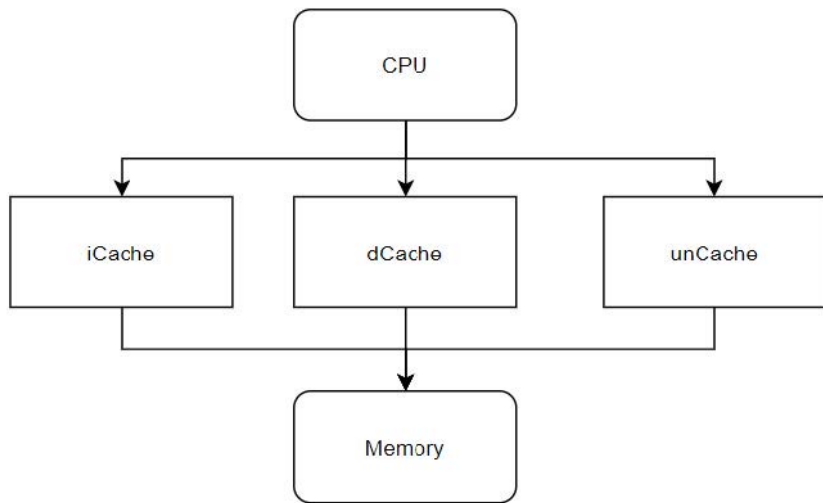
异常处理模块

Exception handler

- ◆ 支持处理多种例外，同时实现了相关的cp0寄存器
- ◆ 异常处理模块较为独立，容易控制且不会对数据通路产生过大的影响

cache模块设计

Design of cache



icache

四路组相联

行大小16B

总大小16KB

可进行参数化配置

dcache

四路组相联

行大小32B

总大小16KB

采用Write-back策略

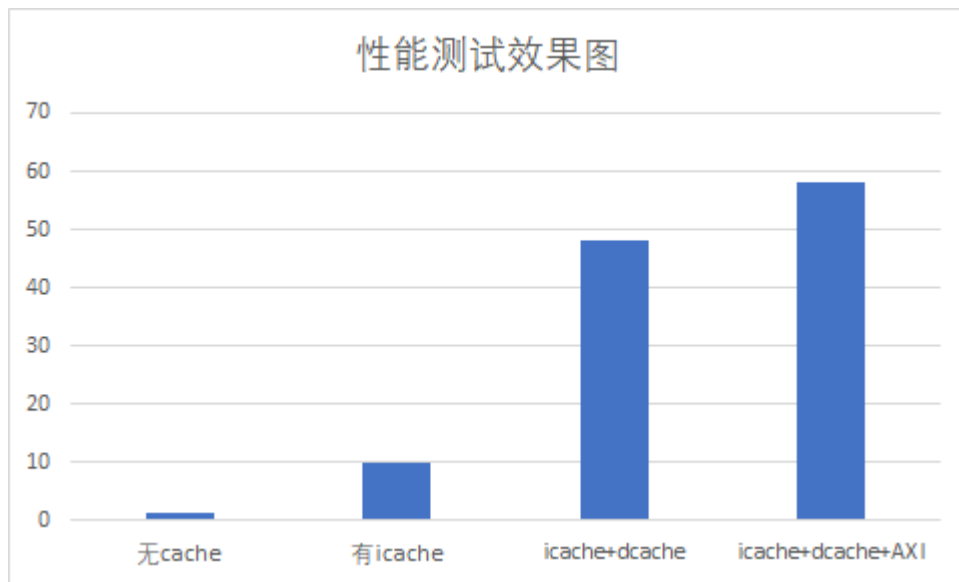
可进行参数化配置

均采用AXI burst, 一次读写一整行的数据。

Cache命中时单周期即可返回数据。

对于cache, ram的划分分为valid ram、dirty ram、tag ram和data ram。每次reset仅重置valid ram和dirty ram。对于tag ram, 通过调用DRAM IP核提升资源利用率。将data ram独立, 可以达到并行访问tag ram和data ram的效果, 避免串行访问, 从而提高性能。

鉴于访存延时较高，处理得当的cache可以显著提升性能



TLB模块设计

Design of TLB

- ◆ 32项TLB
- ◆ 多周期查询
- ◆ 第一周期确定地址是否能够直接翻译，能够直接翻译的当周期返回，不能直接翻译的进入第二周期
- ◆ 第二周期根据要求返回目标地址或者TLB异常



PART THREE

3

系统实现

运行Pmon

Run Pmon

准备工作



更多的指令

如CLO,CLZ,MADD,BEQL等



更多的模块

如TLB模块



更强的功能

完善了cp0寄存器,
异常处理模块

Pmon启动界面

```
serial-com5 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com5 x
NANDFlash info:
erasesize 131072 B
writesize 2048 B
oobsize 64 B
PMON> mtd_erase /dev/mtd0r
ERASE the device:"/dev/mtd0r",DON'T skip bad-blocks
mtd_erase working:
009e0000
mtd_erase work done!
PMON> mtd_erase /dev/mtd1r
ERASE the device:"/dev/mtd1r",DON'T skip bad-blocks
mtd_erase working:
07fe0000
mtd_erase work done!
PMON> ifconfig dmfe0 169.254.37.111
rx ring 70c4f00
tx ring 70c4f80
DE4X5_BMR= fe000000
DE4X5_TPD= 0
DE4X5_RRBA= 70c4f00
DE4X5_TRBA= 70c4f80
DE4X5_STS= f0660004
DE4X5_OMR= 32002242
TX_error status2 = 0x00000000
After setup
DE4X5_BMR= fe000000
DE4X5_TPD= 0
DE4X5_RRBA= 70c4f00
DE4X5_TRBA= 70c4f80
DE4X5_STS= f0660004
DE4X5_OMR= 32002242
PMON> load tftp://169.254.37.69/ucore-kernel-initrd
Loading file: tftp://169.254.37.69/ucore-kernel-initrd (elf)
0x80000000/344464 + 0x80084ed0/13072(z) + 302 syms/
Entry address is 80000000
PMON>
```


运行Pmon

Run Pmon

测试过的pmon命令

- ◆ devls
- ◆ env
- ◆ sleep
- ◆ set myvar
- ◆ eset myvar
- ◆ hi
- ◆ ifconfig
- ◆ ping
- ◆ load
- ◆ mtd_erase

```
serial-com5 x
Environment
env display variable
unset unset variable(s)
RAYS Commands for PMON 2000
PMON> mtdparts
device <nand-flash>,#parts = 2

#: name      size      offset      mask_flags
0: kernel    0x00a00000(10m) 0x00000000(0m) 3
1: rootfs    0x07600000(118m) 0x00a00000(10m) 0

mtdparts INFO:
now-active: mtdparts=nand-flash:10M@0(kernel)ro,-(rootfs)
<NOTE>:you can use command(CMD: unset mtdparts ) to become the default !!!

PMON> devls
Device name  Type
dmfe0        IFNET
loopdev0     DISK
PMON> env
ethaddr = 00:00:00:00:00:00
pll_reg0 = 0x00000000
pll_reg1 = 0x00000000
memsize = 128
highmemsize = 0
cpuclock = 99999999
busclock = 99999999
systype = FCR
brkcmd = "l -r @cpc 1"
datasize = -b [-b -h -w]
dlecho = off [off on lfeed]
dlproto = none [none xonxoff EtxAck]
bootp = no [no sec pri save]
hostport = tty0
inalpha = hex [hex symbol]
inbase = 16 [auto 8 10 16]
moresz = 10
prompt = "PMON> "
regstyle = sw [hw sw]
rptcmd = trace [off on trace]
trabort = ^K
ulcr = cr [cr lf crlf]
uleof = %
showsym = yes [no yes]
pfmft = both [both double single none]
fpdis = yes [no yes]
mtdparts = nand-flash:10M@0(kernel)ro,-(rootfs)
bootdelay = 3
PMON> sleep 1000
PMON> set myvar 666
warning! NVRAM checksum fail. Reset!
PMON> set myvar
myvar = 666
PMON> eset myvar
MYVAR=2020
PMON> set myvar
myvar = 2020
PMON> hi
13 hi
12 set myvar
11 eset myvar
10 set myvar
9 set myvar 666
8 sleep 1000
7 env
6 devls
5 mtdparts
4 h
more... █
```

Ucore启动界面 (仅进入debug模式)

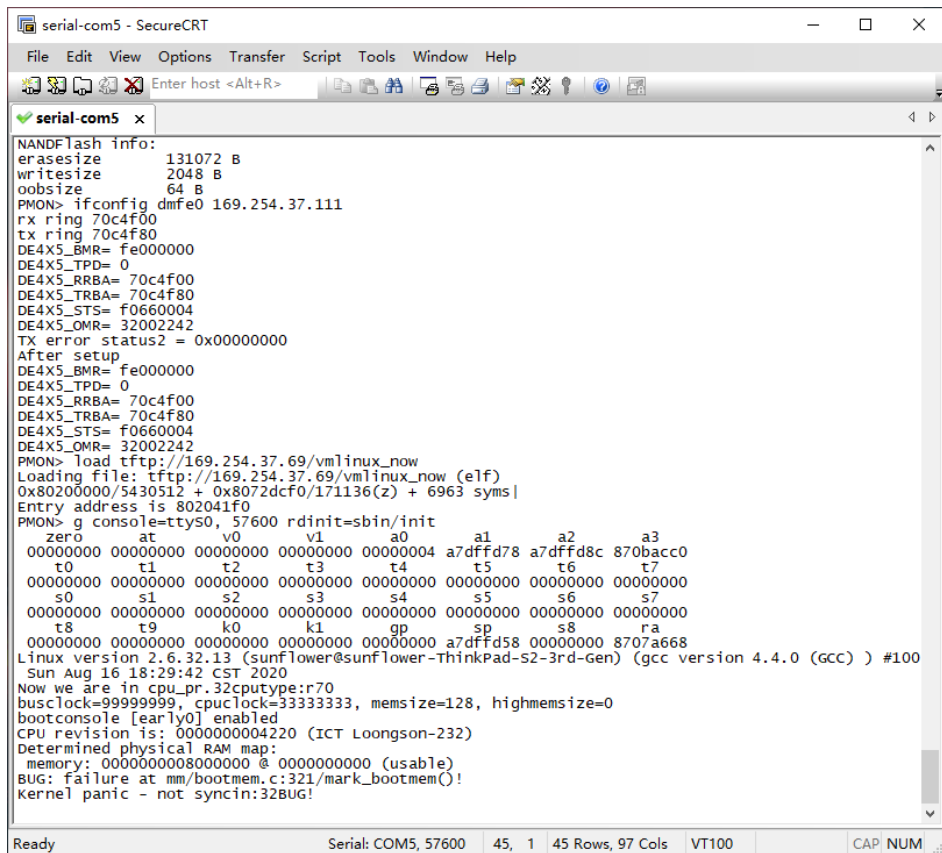
```
serial-com5 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com5 x
DE4X5_TPD= 0
DE4X5_RRBA= 70c4f00
DE4X5_TRBA= 70c4f80
DE4X5_STS= f0660004
DE4X5_OMR= 32002242
PMON> load tftp://169.254.37.69/ucore-kernel-initrd-2
Loading file: tftp://169.254.37.69/ucore-kernel-initrd-2 (elf)
0x80000000/556192 + 0x80087ca0/13072(z) + 302 syms\
Entry address is 80000000
PMON> g console=ttys0, 57600 rdinit=sbin/init
zero at v0 a1 a2 a3
00000000 00000000 00000000 00000004 a7dff78 a7dff8c 870bacc0
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 gp k0 sp s8 ra
00000000 00000000 00000000 00000000 00000000 a7dff58 00000000 8707a668
tlb invalidated
console is initrd
++setup timer interrupts
Initrd: 0x8002c6d0 - 0x80083ecf, size: 0x00057800, magic: 0x2f8dbe2a
(THU.CST) os is loading ...

Special kernel symbols:
entry 0x80000104 (phys)
etext 0x80024400 (phys)
edata 0x80087ca0 (phys)
end 0x8008af80 (phys)
Kernel executable memory footprint: 387KB
memory management: buddy_pmm_manager
memory map:
[80000000, 82000000]

freemem start at: 800c8000
free pages: 00001f35
## 00000020
check_alloc_page() succeeded!
check_pgdir() succeeded!
kernel panic at kern/mm/pmm.c:412:
assertion failed: *(int*)0x100 == 0x1234
welcome to the kernel debug monitor!!
Type 'help' for a list of commands.
help - Display this list of commands.
kerninfo - Display information about the kernel.
Special kernel symbols:
entry 0x80000104 (phys)
etext 0x80024400 (phys)
edata 0x80087ca0 (phys)
end 0x8008af80 (phys)
Kernel executable memory footprint: 387KB
K>
```

Ready Serial: COM5, 57600 52, 4 52 Rows, 97 Cols VT100 CAP NUM

Linux启动界面 (仅打印出相关信息)



```
serial-com5 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com5 x
NANDFlash info:
erasesize      131072 B
writesize      2048 B
oobsize        64 B
PMON> ifconfig dmfe0 169.254.37.111
rx ring 70c4f00
tx ring 70c4f80
DE4X5_BMR= fe000000
DE4X5_TPD= 0
DE4X5_RRBA= 70c4f00
DE4X5_TRBA= 70c4f80
DE4X5_STS= f0660004
DE4X5_OMR= 32002242
TX error status2 = 0x00000000
After setup
DE4X5_BMR= fe000000
DE4X5_TPD= 0
DE4X5_RRBA= 70c4f00
DE4X5_TRBA= 70c4f80
DE4X5_STS= f0660004
DE4X5_OMR= 32002242
PMON> load tftp://169.254.37.69/vmlinux_now
Loading file: tftp://169.254.37.69/vmlinux_now (elf)
0x80200000/5430512 + 0x8072dcf0/171136(z) + 6963 syms|
Entry address is 802041f0
PMON> g console=ttys0, 57600 rdinit=sbin/init
      zero      at      v0      v1      a0      a1      a2      a3
00000000 00000000 00000000 00000000 00000004 a7dfd78 a7dfd8c 870bacc0
      t0      t1      t2      t3      t4      t5      t6      t7
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      s0      s1      s2      s3      s4      s5      s6      s7
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      t8      t9      k0      k1      gp      sp      s8      ra
00000000 00000000 00000000 00000000 00000000 00000000 a7dfd58 00000000 8707a668
Linux version 2.6.32.13 (sunflower@sunflower-ThinkPad-S2-3rd-Gen) (gcc version 4.4.0 (GCC) ) #100
Sun Aug 16 18:29:42 CST 2020
Now we are in cpu_pr.32cputype:r70
busclock=99999999, cpuclock=33333333, memsize=128, highmemsize=0
bootconsole [early0] enabled
CPU revision is: 000000004220 (ICT Loongson-232)
Determined physical RAM map:
memory: 000000008000000 @ 0000000000 (usable)
BUG: failure at mm/bootmem.c:321/mark_bootmem()!
Kernel panic - not syncing:32BUG!
```

◆ 实现了cache指令



PART FOUR

4

附录

实现的cp0寄存器

Implementation of cp0

- ✓ Index
- ✓ Random
- ✓ EntryLo0
- ✓ EntryLo1
- ✓ Context
- ✓ PageMask
- ✓ Wired
- ✓ BadVAddr
- ✓ Count
- ✓ EntryHi
- ✓ Compare
- ✓ Status
- ✓ Cause
- ✓ EPC
- ✓ PRId
- ✓ Config0
- ✓ Config1

实现的例外

Implementation of exception

- ✔ Interrupt (0x0) (including Timer Interrupt)
- ✔ TLB Modify (0x1)
- ✔ TLB Load (0x2)
- ✔ TLB Store (0x3)
- ✔ Address Error Load (0x4)
- ✔ Address Error Store (0x5)
- ✔ System Call (0x8)
- ✔ Break (0x9)
- ✔ Reserved Instruction(0xa)
- ✔ Overflow (0xc)

指令集

Instruction set

✓ ADD	✓ DIV	✓ CLZ	✓ SLL	✓ BLTZ	✓ MFLO	✓ SB	✓ MFC0
✓ ADDI	✓ DIVU	✓ AND	✓ SRAV	✓ BGEZAL	✓ MTHI	✓ SH	✓ MTC0
✓ ADDU	✓ MUL	✓ ANDI	✓ SRA	✓ BLTZAL	✓ MTLO	✓ SW	✓ TLBP
✓ ADDIU	✓ MULT	✓ LUI	✓ SRLV	✓ BEQL	✓ MOVN	✓ LWL	✓ TLBR
✓ SUB	✓ MULTU	✓ NOR	✓ SRL	✓ BNEL	✓ MOVZ	✓ LWR	✓ TLBWI
✓ SUBU	✓ MADD	✓ OR	✓ BEQ	✓ J	✓ LB	✓ SWL	✓ TLBWR
✓ SLT	✓ MADDU	✓ ORI	✓ BNE	✓ JR	✓ LBU	✓ SWR	✓ WAIT
✓ SLTI	✓ MSUB	✓ XOR	✓ BGEZ	✓ JAL	✓ LH	✓ BREAK	✓ PREF
✓ SLTU	✓ MSUBU	✓ XORI	✓ BGTZ	✓ JALR	✓ LHU	✓ SYSCALL	✓ SYNC
✓ SLTIU	✓ CLO	✓ SLLV	✓ BLEZ	✓ MFHI	✓ LW	✓ ERET	✓ LL/SC

2  20

汇报完毕 感谢您的观看

FDU1.2队 2020.08.20

