MIPS CPU 测试文档

杜家驹[1] 袁星驰[2] 王硕[3]

2017年1月

[1]电子邮件：dujj14@mails.tsinghua.edu.cn，学号：2014010257
[2]电子邮件：yuanxc13@mails.tsinghua.edu.cn，学号：2013011665
[3]电子邮件：w-s14@mails.tsinghua.edu.cn，学号：2014012276

目录

1 测试步骤与目的

本项目中测试主要分为两类。一类是CPU与总线逻辑测试，另一类是外设测试。

逻辑测试主要在仿真中完成，又分为单元测试和集成测试。单元测试是针对一类指令的测试，集成测试是直接仿真操作系统从启动到进入shell的全过程。在这些测试中，需要对访存设备进行虚拟化，即虚拟出SRAM，Flash和串口。

外设测试主要是SRAM，Flash和串口的测试。

2 逻辑测试

2.1 单元测试

2.1.1 基本流水线测试

代码：

```
.global __start
.set noat
__start:
    lui $2, 0x0404
    ori $2, $2, 0x0404
    ori $7, $0, 0x7
    ori $5, $0, 0x5
    ori $8, $0, 0x8
    nop
    sll $2, $2, 8
    sllv $2, $2, $7
    srl $2, $2, 8
    srlv $2, $2, $5
    sll $2, $2, 19
    nop
    sra $2, $2, 16
    srav $2, $2, $8
```

直接观察波形

2.1.2 算术运算指令测试

代码：

```
.global __start
.set noat
__start:
    ori $3, $0, 0x0
    addiu $2, $0, 0x1234
    subu $3, $2, $3
    mult $2, $3
    slt $4, $3, $1
    sltu $5, $1, $3
    lui $1, 0x0000
    lui $2, 0xffff
    lui $3, 0x0505
    lui $4, 0x0000
    mthi $0
    mthi $2
    mthi $3
    mfhi $4
    mtlo $3
    mtlo $2
    mtlo $1
    mflo $4
```

直接观察波形测试

2.1.3 跳转指令测试

代码：

```
.global __start
.set noat
__start:
    lui $2, 0xbfc0
    ori $2, 0x20
    jalr $2
    jal func
```

```
    j next
func:
    jr $31
end:
    j end
next:
    lui $1, 0xffff
    ori $1, 0xfffd
    lui $2, 0x0
loop:
    addiu $1, $1, 0x1
    beq $1, $2, end
    j loop
```

直接观察波形测试

2.1.4 访存指令测试

代码：

```
.global __start
.set noat
__start:
    lui $3, 0xbfc0
    ori $3, $3, 0x100
    ori $1, $0, 0x1234
    sw $1, 0x0($3)
    ori $2, $0, 0x1234
    ori $1, $0, 0x0
    lw $1, 0x0($3)
    beq $1, $2, label
    ori $1, $0, 4567
    nop
label:
    ori $1, $0, 0x89ab
    nop
loop:
    j loop
```

直接观察波形进行测试

2.1.5 Helloworld测试

代码：

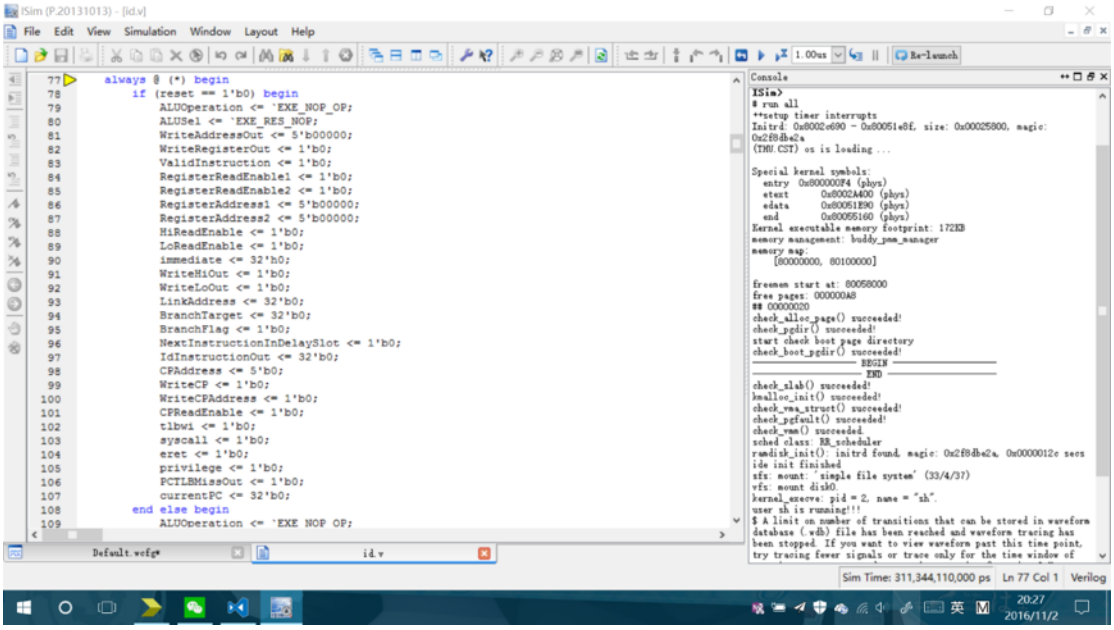
```
.global __start
.set noat
__start:
    lui $1, 0x9fd0
    ori $1, 0x03f8
    ori $2, $0, 0x68
    sw $2, 0x0($1)
    ori $2, $0, 0x65
    sw $2, 0x0($1)
    ori $2, $0, 0x6c
    sw $2, 0x0($1)
    ori $2, $0, 0x6c
    sw $2, 0x0($1)
    ori $2, $0, 0x6f
    sw $2, 0x0($1)
    ori $2, $0, 0x20
    sw $2, 0x0($1)
    ori $2, $0, 0x77
    sw $2, 0x0($1)
    ori $2, $0, 0x6f
    sw $2, 0x0($1)
    ori $2, $0, 0x72
    sw $2, 0x0($1)
    ori $2, $0, 0x6c
    sw $2, 0x0($1)
    ori $2, $0, 0x64
    sw $2, 0x0($1)
    ori $2, $0, 0x21
    sw $2, 0x0($1)
    ori $2, $0, 0x0a
    sw $2, 0x0($1)
loop:
```

b loop

应该输出hello world

2.2 集成测试

对操作系统进行仿真。把操作系统编译之后的二进制文件转换为仿真器能够识别的格式，并读入到虚拟设备中。操作系统应当能够进入用户态，并有类似于下图的输出。



3 外设测试

外设测试在电路板上进行，需要首先完成自检工具。完成自检工具后，可以通过python读写某一块内存或Flash。测试的主要方式为随机产生若干个读写序列，完成操作后与预期结果进行比对。

4 测试效果

逻辑测试效果良好，仿真通过后在电路板上没有遇到逻辑问题。 外设测试检测出了SRAM的竞争与冒险问题。这一问题导致操作系统随机崩溃，解决这一问题耗时2周。