**关于测试单周期CPU的简单方法**

**（特别说明：本表每个同学都需建立，检查实验时，必须提供！）**

1. 测试程序段

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 地址 | 汇编程序 | op（6） | ALU运算结果  10进制 | ALU运算结果  2进制 | ALU运算结果16进制 | instruction[31:0] | PC[31:0] | next\_PC[31:0] | DM输出结果（LW） memreaddata[31:0] |
|  | #算术运算 |  |  |  |  |  |  |  |  |
| 0x00000000 | add $7,$0,$8 | 000000 | 8 | 1000 | 8 | 00083820 | 00000000 | 00000004 | x |
| 0x00000004 | sub $2,$3,$2 | 001000 | 1 | 1 | 1 | 00621022 | 00000004 | 00000008 | x |
| 0x00000008 | addi $7,$7,8 | 000000 | 16 | 10000 | 10 | 20e70008 | 00000008 | 0000000c | x |
| 0x0000000C | addu $7,$0,$8 | 000000 | 8 | 1000 | 8 | 00083821 | 0000000c | 00000010 | x |
|  | #位运算 |  |  | 0 | 0 |  |  |  |  |
| 0x00000010 | and $4,$5,$2 | 000000 | 6 | 110 | 6 | 00a22024 | 00000010 | 00000014 | x |
| 0x00000014 | or $8,$4,$2 | 000000 | 7 | 111 | 7 | 00824025 | 00000014 | 00000018 | x |
| 0x00000018 | ori $2,$0,2 | 001101 | 2 | 10 | 2 | 34020002 | 00000018 | 0000001c | x |
| 0x0000001C | sll $8,$8,1 | 000000 | 14 | 1110 | E | 00084040 | 0000001c | 00000020 | x |
| 0x00000020 | srl $7,$7,1 | 000000 | 4 | 100 | 4 | 00073842 | 00000020 | 00000024 | x |
| 0x00000024 | lui $1,255 000000ff | 001111 | 16711680 | #NUM! | FF0000 | 3c0100ff | 00000024 | 00000028 | x |
|  |  |  |  |  |  |  |  |  |  |
|  | #分支指令 |  |  |  |  |  |  |  |  |
| 0x00000028 | addi $1,$0,2 | 001000 | 2 | 10 | 2 | 20010002 | 00000028 | 0000002c | x |
| 0x0000002C | addi $2,$0,2 | 001000 | 2 | 10 | 2 | 20020002 | 0000002c | 00000030 | x |
| 0x00000030 | beq $1,$2,lable1 | 000100 | 0 | 0 | 0 | 10220002 | 00000030 | 0000003c | x |
|  |  |  |  | 0 | 0 |  |  |  |  |
| 0x00000034 | addi $1,$0,2#指令无效 |  |  | 0 | 0 |  |  |  |  |
| 0x00000038 | sll $8,$8,1#指令无效 |  |  | 0 | 0 |  |  |  |  |
|  |  |  |  | 0 | 0 |  |  |  |  |
| 0x0000003C | lable1: addi $2,$0,1 | 001000 | 1 | 1 | 1 | 20020001 | 0000003c | 00000040 | x |
| 0x00000040 | bne $1,$2,lable2 | 000101 | 1 | 1 | 1 | 14220002 | 00000040 | 0000004c | x |
|  |  |  |  | 0 | 0 |  |  |  |  |
| 0x00000044 | addi $1,$0,2#指令无效 |  |  | 0 | 0 |  |  |  |  |
| 0x00000048 | sll $8,$8,1#指令无效 |  |  | 0 | 0 |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | #lw/sw指令 |  |  |  |  |  |  |  |  |
| 0x0000004C | lable2: sw $2,4($0) #2号寄存器原本存入1，现在我们在第一个data memory写入1 | 101011 | 4 | 100 | 4 | ac020004 | 0000004c | 00000050 | x |
| 0x00000050 | lw $9,4($0) | 100011 | 4 | 100 | 4 | 8c090004 | 00000050 | 00000054 | 1 |
|  |  |  |  |  |  |  |  |  |  |
|  | #跳转指令（注意汇编指令顺序和右侧指令并非一一对应） | |  |  |  |  |  |  |  |
| 0x00000054 | jal label4 | 000011 | 16 | 10000 | 10 | 0c100018 | 00000054 | 00000060 | x |
| 0x00000058 | addi $1,$0,2 | 001000 | 88 | 1011000 | 58 | 03e00008 | 00000060 | 00000058 | x |
| 0x0000005C | j fin | 000010 | 90 | 1011010 | 5A | 20010002 | 00000058 | 0000005c | x |
| 0x00000060 | label4:jr $ra | 000000 | 104 | 1101000 | 68 | 08100019 | 0000005c | 00000064 | x |
|  |  |  |  |  |  |  |  |  | x |
| 0x00000064 | fin:slt $3,$9,$2  #$9==1,$2==1 | 000000 | 0 | 0 | 0 | 0122182a | 00000064 | 00000068 | x |
|  | #半字，字节指令 |  |  |  |  |  |  |  |  |
| 0x00000068 | sb $7,1($0) #0x00000020 $7是4 | 101000 | 89 | 1011001 | 59 | a0070001 | 68 | 0000006c | x |
| 0x0000006C | lb $7,1($0) | 100000 | 89 | 1011001 | 59 | 80070001 | 0000006c | 70 | 4 |
| 0x00000070 | sh $7,2($0) | 101001 | 90 | 1011010 | 5A | a4070002 | 70 | 74 | x |
| 0x00000074 | lh $7,2($0) | 100001 | 90 | 1011010 | 5A | 84070002 | 74 | 78 | 4 |

指引：

1、将**指令代码初始化到指令存储器**中，直接写入。

1. 初始化PC的值，也就是以上程序段首地址PC=**0x00000000**，以上程序段从**0x00000000**地址开始存放。
2. 运行Xilinx Vivado进行仿真，看波形。

扩展指令测试程序段，可根据自己实现的功能进行修改。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 指令地址 | 汇编指令 | 二进制 | 十六进制 | 运行结果 | |
| 0x00000000 | addiu $1,$0,8 | 00001000 00000001 00000000 00001000 | 08010008 | 8 | 1 |
| 0x00000004 | ori $2,$0,2 | 01001000 00000010 00000000 00000010 | 48020002 | 2 | 2 |
| 0x00000008 | xori $3,$2,8 | 01001100 01000011 00000000 00001000 | 4c430008 | 10 | 3 |
| 0x0000000C | sub $4,$3,$1 | 00000100 01100001 00100000 00000000 | 4612000 | 2 | 4 |
| 0x00000010 | and $5,$4,$2 | 01000000 10000010 00101000 00000000 | 40822800 | 2 | 5 |
| 0x00000014 | sll $5,$5,2 | 01100000 00000101 00101000 10000000 | 60052880 | 8 | 6 |
| 0x00000018 | beq $5,$1,-2(=,转14) | 11010000 10100001 11111111 11111110 | d0a1fffe | 8 | 7 |
| 0x00000014 | sll $5,$5,2 | 01100000 00000101 00101000 10000000 | 60052880 | 32 | 8 |
| 0x00000018 | beq $5,$1,-2(=,转14) | 11010000 10100001 11111111 11111110 | d0a1fffe | 32 | 9 |
| 0x0000001C | jal 0x0000050 | 11101000 00000000 00000000 00010100 | e8000014 |  | 10 |
| 0x00000050 | sw $2,4($1) | 11000000 00100010 00000000 00000100 | c0220004 |  | 11 |
| 0x00000054 | lw $13,4($1) | 11000100 00101101 00000000 00000100 | c42d0004 |  | 12 |
| 0x00000058 | jr $31 | 11100111 11100000 00000000 00000000 | e7e00000 |  | 13 |
| 0x00000020 | slt $8,$13,$1 | 10011101 10100001 01000000 00000000 | 9da14000 | 1 | 14 |
| 0x00000024 | addiu $14,$0,-2 | 00001000 00001110 11111111 11111110 | 80efffe | -2 | 15 |
| 0x00000028 | slt $9,$8,$14 | 10011101 00001110 01001000 00000000 | 9d0e4800 | 0 | 16 |
| 0x0000002C | slti $10,$9,2 | 10011001 00101010 00000000 00000010 | 992a0002 | 1 | 17 |
| 0x00000030 | slti $11,$10,0 | 10011001 01001011 00000000 00000000 | 994b0000 | 0 | 18 |
| 0x00000034 | add $11,$11,$10 | 00000001 01101010 01011000 00000000 | 16a5800 | 1 | 19 |
| 0x00000038 | bne $11,$2,-2 (≠,转34) | 11010101 01100010 11111111 11111110 | d562fffe | 1 | 20 |
| 0x00000034 | add $11,$11,$10 | 00000001 01101010 01011000 00000000 | 16a5800 | 2 | 21 |
| 0x00000038 | bne $11,$2,-2 (≠,转34) | 11010101 01100010 11111111 11111110 | d562fffe | 2 | 22 |
| 0x0000003C | addiu $12,$0,-2 | 00001000 00001100 11111111 11111110 | 80cfffe | 0 | 23 |
| 0x00000040 | addiu $12,$12,1 | 00001001 10001100 00000000 00000001 | 98c0001 | 1 | 24 |
| 0x00000044 | bltz $12,-2 (<0,转40) | 11011001 10000000 11111111 11111110 | d980fffe | 1 | 25 |
| 0x00000040 | addiu $12,$12,1 | 00001001 10001100 00000000 00000001 | 98c0001 | 0 | 26 |
| 0x00000044 | bltz $12,-2 (<0,转40) | 11011001 10000000 11111111 11111110 | d980fffe | 0 | 27 |
| 0x00000048 | andi $12,$2,2 | 01000100 01001100 00000000 00000010 | 444c0002 |  | 28 |
| 0x0000004C | j 0x000005C | 11100000 00000000 00000000 00010111 | e0000017 |  | 29 |
| 0x0000005C | halt | 11111100 00000000 00000000 00000000 | FC000000 |  | 30 |