**关于测试多周期CPU的简单方法**

**（特别说明：本表每个同学都必须建立，检查实验时，必须提供！）。**

1. 测试程序段

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 地址 | 汇编程序 | 答案 | 指令代码 | | | | | |
| op（6） | rs(5) | rt(5) | rd(5)/immediate (16) | 16进制数代码 | |
| 0x00000000 | addiu $1,$0,8 | $1=8 | 000010 | 00000 | 00001 | 0000 0000 0000 1000 | = | 08010008 |
| 0x00000004 | ori $2,$0,2 | $2=2 | 010010 | 00000 | 00010 | 0000 0000 0000 0010 | = | 48020002 |
| 0x00000008 | xori $3,$2,8 | $3=10 | 010011 | 00010 | 00011 | 0000 0000 0000 1000 | = | 4c430008 |
| 0x0000000C | sub $4,$3,$1 | $4=2 | 000001 | 00011 | 00001 | 00100 | = | 04612000 |
| 0x00000010 | and $5,$4,$2 | $5=2 | 010000 | 00100 | 00010 | 00101 | = | 40822800 |
| 0x00000014 | sll $5,$5,2 | $5=8,32 | 011000 | 00000 | 00101 | 00101 sa=00010 | = | 60052880 |
| 0x00000018 | beq $5,$1,-2(=,转14) | 先跳后不跳 | 110100 | 00101 | 00001 | 1111 1111 1111 1110 | = | d0a1fffe |
| 0x0000001C | jal 0x0000050 | 跳0x50 | 111010 | 00 0000 0000 0000 0000 0001 0100 | | | = | e8000014 |
| 0x00000020 | slt $8,$13,$1 | $8=2<8=1 | 100111 | 01101 | 00001 | 01000 | = | 9da14000 |
| 0x00000024 | addiu $14,$0,-2 | $14=-2 | 000010 | 00000 | 01110 | 1111 1111 1111 1110 | = | 080efffe |
| 0x00000028 | slt $9,$8,$14 | $9=1<-2=0 | 100111 | 01000 | 01110 | 01001 | = | 9d0e4800 |
| 0x0000002C | slti $10,$9,2 | $10=0<2=1 | 100110 | 01001 | 01010 | 0000 0000 0000 0010 | = | 992a0002 |
| 0x00000030 | slti $11,$10,0 | $11=1<0=0 | 100110 | 01010 | 01011 | 0000 0000 0000 0000 | = | 994b0000 |
| 0x00000034 | add $11,$11,$10 | $11=1,2 | 000000 | 01011 | 01010 | 01011 | = | 016a5800 |
| 0x00000038 | bne $11,$2,-2 (≠,转34) | 先跳后不跳 | 110101 | 01011 | 00010 | 1111 1111 1111 1110 | = | d562fffe |
| 0x0000003C | addiu $12,$0,-2 | $12=-2 | 000010 | 00000 | 01100 | 1111 1111 1111 1110 | = | 080cfffe |
| 0x00000040 | addiu $12,$12,1 | $12=-1,0 | 000010 | 01100 | 01100 | 0000 0000 0000 0001 | = | 098c0001 |
| 0x00000044 | bltz $12,-2 (<0,转40) | 先跳后不跳 | 110110 | 01100 | 00000 | 1111 1111 1111 1110 | = | d980fffe |
| 0x00000048 | andi $12,$2,2 | $12=2 | 010001 | 00010 | 01100 | 0000 0000 0000 0010 | = | 444c0002 |
| 0x0000004C | j 0x000005C |  | 111000 | 00 0000 0000 0000 0000 0001 0111 | | | = | e0000017 |
| 0x00000050 | sw $2,4($1) | [12]=2 | 110000 | 00001 | 00010 | 0000 0000 0000 0100 | = | c0220004 |
| 0x00000054 | lw $13,4($1) | $13=2 | 110001 | 00001 | 01101 | 0000 0000 0000 0100 | = | c42d0004 |
| 0x00000058 | jr $31 | 跳0x20 | 111001 | 11111 | 00000 | 0000 0000 0000 0000 | = | e7e00000 |
| 0x0000005C | halt |  | 111111 | 00000 | 00000 | 0000 0000 0000 0000 | = | fc000000 |

1. 将**指令代码初始化到指令存储器**中，直接写入。
2. 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。
3. 运行Xilinx Vivado进行仿真，看波形。