**关于测试单周期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** | **指令代码** | | | | | | **运行结果(16进制)** |
| **op（6）** | **rs(5)** | **rt(5)** | **rd(5)/immediate (16)** | **16进制数代码** | |
| **0x00000000** | addiu $1,$0,8 | **001001** | **00000** | **00001** | **0000 0000 0000 1000** | **=** | **24010008** | $1 = 8 |
| **0x00000004** | ori $2,$0,2 | **001101** | **00000** | **00010** | **0000 0000 0000 0010** | **=** | **34020002** | $2 = 2 |
| **0x00000008** | add $3,$2,$1 | **000000** | **00010** | **00001** | **0001 1000 0010 0000** | **=** | **00411820** | $3 = a |
| **0x0000000C** | sub $5,$3,$2 | **000000** | **00011** | **00010** | **0010 1000 0010 0010** | **=** | **00622822** | $5 = 8 |
| **0x00000010** | and $4,$5,$2 | **000000** | **00101** | **00010** | **0010 0000 0010 0100** | **=** | **00a22024** | $4 = 0 |
| **0x00000014** | or $8,$4,$2 | **000000** | **00100** | **00010** | **0100 0000 0010 0101** | **=** | **00824025** | $8 = 2 |
| **0x00000018** | sll $8,$8,1 | **000000** | **00000** | **00100** | **0010 0000 0010 0000** | **=** | **00084040** | 第一次：  $8 = 4  第二次：  $8 = 8 |
| **0x0000001C** | **bne $8,$1,-2 (≠,转18)** | **000101** | **01000** | **00001** | **1111 1111 1111 1110** | **=** | **1501fffe** | 第一次：  跳转到0x18  第二次:  不跳转，执行下一条指令 |
| **0x00000020** | slti $6,$2,4 | **001010** | **00010** | **00110** | **0000 0000 0000 0100** | **=** | **28460004** | $6 = 1 |
| **0x00000024** | slti $7,$6,0 | **001010** | **00110** | **00111** | **0000 0000 0000 0000** | **=** | **28c70000** | $7 = 0 |
| **0x00000028** | addiu $7,$7,8 | **001001** | **00111** | **00111** | **0000 0000 0000 1000** | **=** | **24e70008** | 第一次：  $7 = 8  第二次：  $7 = 16 |
| **0x0000002C** | **beq $7,$1,-2 (=,转28)** | **000100** | **00111** | **00001** | **1111 1111 1111 1110** | **=** | **10e1fffe** | 第一次：  跳转到0x28  第二次：  不跳转，执行下一条指令 |
| **0x00000030** | sw $2,4($1) | **101011** | **00001** | **00010** | **0000 0000 0000 0100** | **=** | **ac220004** | 将$2中的值存入0x000c |
| **0x00000034** | lw $9,4($1) | **100011** | **00001** | **01001** | **0000 0000 0000 0100** | **=** | **8c290004** | $9 = 2 |
| **0x00000038** | addiu $10,$0,-2 | **001001** | **00000** | **01010** | **1111 1111 1111 1110** | **=** | **240afffe** | $10 = -2 |
| **0x0000003C** | addiu $10,$10,1 | **001001** | **01010** | **01010** | **0000 0000 0000 0001** | **=** | **254a0001** | $10 = -1 |
| **0x00000040** | **bltz $10,-2(<0,转3C)** | **000001** | **01010** | **00000** | **1111 1111 1111 1110** | **=** | **0540fffe** | 第一次：跳转到0x3C  第二次：不跳转，执行下一条指令 |
| **0x00000044** | **andi $11,$2,2** | **001100** | **00010** | **01011** | **0000 0000 0000 0010** | **=** | **304b0002** | $11 = 2 |
| **0x00000048** | **xori $1, $1, 1** | **001110** | **00001** | **00001** | **0000 0000 0000 0001** | **=** | **38210001** | $1 = 9 |
| **0x0000004C** | **sltiu $2, $2, 2** | **001011** | **00010** | **00010** | **0000 0000 0000 0010** | **=** | **2c420002** | $2 = 0 |
| **0x00000050** | **nor $3, $3, $1** | **000000** | **00011** | **00001** | **0001 1000 0010 0111** | **=** | **00611827** | $3 = -12 |
| **0x00000054** | **srl $8, $8, 2** | **000000** | **00000** | **01000** | **0100 0000 1000 0010** | **=** | **00084082** | $8 = 2 |
| **0x00000058** | **lui $4, 1** | **001111** | **00000** | **00100** | **0000 0000 0000 0001** | **=** | **3c040001** | $4 =  0x00010000 |
| **0x0000005C** | **j 0x00000064** | **000010** | **00000** | **00000** | **0000 0000 0010 0100** | **=** | **08000019** | 跳转到  0x0050 |
| **0x00000060** | or $8,$4,$2 | **000000** | **00100** | **00010** | **0100 0000 0010 0101** | **=** | **00824025** | 未执行 |
| **0x00000064** | **halt** | **111111** | **00000** | **00000** | **0000000000000000** | **=** | **FC000000** | PC停止 |

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。