

存储内容

预期输出: b-a-d-c-e-f

指令:

travel:

addi $sp, $sp, -8 #修/改堆栈指针，准备入栈

sw $ra, 4($sp) #返回地址入栈

sw $a0, 0($sp) #参数入栈

beq $a0, $zero, retn #节点为空，出栈；不为空，其左节点继续执行函数

lw $t0, -1($a0) #将$a0存储的字母的左子树的地址存入$t0

addi $a0, $t0, 0 #参数存入$a0

jal travel #继续执行travel

sll $zero, $zero, 0

lw $ra, 4($sp) #恢复本轮调用的返回地址至$ra

lw $a0, 0($sp) #将栈指针指向的字母指针存入a0

lw $t0, 0($a0) #将a0指向的字母值存入t0

sw $t0, 0($t1) #把字母存入主存某位置

addi $t1, $t1, 4 #存字母的主存地址位置+4

lw $t0, 1($a0) #将$a0存储的字母的右子树的地址存入$t0

addi $a0, $t0, 0 #参数存入$a0

jal travel #继续执行travel

sll $zero, $zero, 0

lw $ra, 4($sp) #恢复本轮调用的返回地址至$ra

retn:

addi $sp, $sp, 8

jr $ra

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| Add | 000000 | rs | rt | Rd(rs+rt) | 00000 | 100000 |
| Or | 000000 | rs | rt | Rd(rs|rt) | 00000 | 100101 |
| Sub | 000000 | rs | rt | Rd(rs-rt) | 00000 | 100010 |
| xor | 000000 | rs | rt | Rd(rs^rt) | 00000 | 100110 |
| Addi | 001000 | rs | | rt | | Imm16 |
| Andi | 001100 | rs | | rt | | Imm16 |
| xori | 001110 | rs | | rt | | Imm16 |
| Bne | 000101 | rs | rt | Offset16 | | |
| Beq | 000100 | rs | rt | Offset16 | | |
| Sw | 101011 | rs | rt | Offset16 | | |
| Lw | 100011 | rs | rt | Offset16 | | |
| J | 000010 | Target26 | | | | |
| Jal | 000011 | Target26 | | | | |
| Jr | 000000 | rs | 000000000000000(15) | | | 001000 |
| Sll | 000000 | 00000 | rt | rd | Shamt5 | 000000 |
| Srl | 000000 | 00000 | rt | rd | Shamt5 | 000010 |

Imm: immediate

信号表：

|  |  |  |
| --- | --- | --- |
| ALUCon | 000 | 加法 |
|  | 001 | 左移 |
|  | 010 | 右移 |
|  | 011 | 异或 |
|  | 100 | 与 |
|  | 101 | 或 |
|  | 110 | 减 |
| PCsrc | 00 | pc来自A寄存器（对应jr指令） |
|  | 01 | pc来自pc-extend（对应无条件转移指令j和jar） |
|  | 10 | pc直接来自运算器结果（对应pc = pc + 1） |
|  | 11 | pc来自ALUOut寄存器（对应条件转移指令beq和bne） |
| PCWC1 | 0 | / |
|  | 1 | 对应相等则转移（beq） |
| PCWC2 | 0 | / |
|  | 1 | 对应不相等则转移（bne） |
| PCW | 0 | / |
|  | 1 | 对应无条件转移(j jal jr pc+1) |
| Regsrc | 00 | Wd来自MDR（对应读内存指令lw） |
|  | 01 | Wd来自pc（对应jal指令） |
|  | 10 | Wd来自ALUOut（对应所有计算性结果） |
| ALUsrca | 00 | A操作数来自pc （对应beq bne pc + 1 ） |
|  | 01 | A操作数来自A寄存器 (branch也用) |
|  | 10 | A操作数来自IR[10:6]（对应移位指令） |
| ALUsrcb | 00 | B操作数来自B寄存器（branch也用） |
|  | 01 | B操作数来自sign-extend（对应bne beq lw sw \*\*\*i） |
|  | 10 | B操作数是1（pc = pc + 1） |
| RegWrite | 0 | / |
|  | 1 | Regfile可以写 |
| RegDst | 00 | Wa来自IR[20:16] |
|  | 01 | Wa来自IR[15:11] (对应and or sll srl) |
|  | 10 | Wa为31（对应jal存pc当前值到$ra） |
| MEMW | 0 | / |
|  | 1 | Memory可以写 |
| MEMR | 0 | / |
|  | 1 | Memory可以读 |
| IorD | 0 | 访存地址来自PC（对应取指周期） |
|  | 1 | 访存地址来自ALUOut（对应sw） |
| IRWrite | 0 | / |
|  | 1 | IR可以写（对应取指周期） |