**DLX指令集**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | 31 26 | 25 21 | 20 16 | | 15 11 | 10 6 | | 5 0 |
| 加 | ADD | DR, | SR1, | SR2 | 000000 | SR1 | SR2 | | DR | 00000 | | 000001 |
| 加 | ADDI | DR, | SR1, | Imm/LABEL | 000001 | SR1 | DR | | Imm16 | | | |
| 减 | SUB | DR, | SR1, | SR2 | 000000 | SR1 | SR2 | | DR | 00000 | | 000011 |
| 减 | SUBI | DR, | SR1, | Imm/LABEL | 000011 | SR1 | DR | | Imm16 | | | |
| 按位与 | AND | DR, | SR1, | SR2 | 000000 | SR1 | SR2 | | DR | 00000 | | 001001 |
| 按位与 | ANDI | DR, | SR1, | Imm/LABEL | 001001 | SR1 | DR | | Imm16 | | | |
| 按位或 | OR | DR, | SR1, | SR2 | 000000 | SR1 | SR2 | | DR | 00000 | | 001010 |
| 按位或 | ORI | DR, | SR1, | Imm/LABEL | 001010 | SR1 | DR | | Imm16 | | | |
| 按位异或 | XOR | DR, | SR1, | SR2 | 000000 | SR1 | SR2 | | DR | 00000 | | 001011 |
| 按位异或 | XORI | DR, | SR1, | Imm/LABEL | 001011 | SR1 | DR | | Imm16 | | | |
| 加载高位 | LHI | DR, | Imm | | 001100 | 00000 | DR | | Imm16 | | | |
| 左移 | SLL | DR, | SR1, | SR2 | 000000 | SR1 | SR2 | | DR | 00000 | | 001101 |
| 左移 | SLLI | DR, | SR1, | Imm/LABEL | 001101 | SR1 | DR | | Imm16 | | | |
| 逻辑右移 | SRL | DR, | SR1, | SR2 | 000000 | SR1 | SR2 | | DR | 00000 | | 001110 |
| 逻辑右移 | SRLI | DR, | SR1, | Imm/LABEL | 001110 | SR1 | DR | | Imm16 | | | |
| 算数右移 | SRA | DR, | SR1, | SR2 | 000000 | SR1 | SR2 | | DR | 00000 | | 001111 |
| 算数右移 | SRAI | DR, | SR1, | Imm/LABEL | 001111 | SR1 | DR | | Imm16 | | | |
| 小于 | SLT | DR, | SR1, | SR2 | 000000 | SR1 | SR2 | | DR | 00000 | | 010000 |
| 小于 | SLTI | DR, | SR1, | Imm/LABEL | 010000 | SR1 | DR | | Imm16 | | | |
| 小于等于 | SLE | DR, | SR1, | SR2 | 000000 | SR1 | SR2 | | DR | 00000 | | 010010 |
| 小于等于 | SLEI | DR, | SR1, | Imm/LABEL | 010010 | SR1 | DR | | Imm16 | | | |
| 等于 | SEQ | DR, | SR1, | SR2 | 000000 | SR1 | SR2 | | DR | 00000 | | 010100 |
| 等于 | SEQI | DR, | SR1, | Imm/LABEL | 010100 | SR1 | DR | | Imm16 | | | |
| 加载字节 | LB | DR, | Imm/LABEL(SR1) | | 010110 | SR1 | DR | | Imm16 | | | |
| 存储字节 | SB | Imm/LABEL(SR1), | | DR | 010111 | SR1 | DR | | Imm16 | | | |
| 加载字 | LW | DR, | Imm/LABEL(SR1) | | 011100 | SR1 | DR | | Imm16 | | | |
| 存储字 | SW | Imm/LABEL(SR1), | | DR | 011101 | SR1 | DR | | Imm16 | | | |
| (授权)存储字 | MOVI2S | GPR | NUMBER | | 100010 | 00000 | GPR | | NUMBER | | 000 0000 0000 | |
| (授权)加载字 | MOVS2I | GPR | NUMBER | | 100011 | 00000 | GPR | | NUMBER | | 000 0000 0000 | |
| 条件假跳转 | BEQZ | SR1, | LABEL | | 101000 | SR1 | 00000 | | Imm16 | | | |
| 条件真跳转 | BNEZ | SR1, | LABEL | | 101001 | SR1 | 00000 | | Imm16 | | | |
| 无条件跳转 | J | LABEL | | | 101100 | PCOffset26 | | | | | | |
| 无条件跳转 | JR | SR1 | | | 101101 | SR1 | 0 0000 0000 0000 0000 0000 | | | | | |
| 从TRAP返回 | RET | (JR R31) | | | 101101 | 11111 | 0 0000 0000 0000 0000 0000 | | | | | |
| 链接 | JAL | LABEL | | | 101110 | PCOffset26 | | | | | | |
| 链接 | JALR | SR1 | | | 101111 | SR1 | | 0 0000 0000 0000 0000 0000 | | | | |
| 自陷 GETC | TRAP | x06 | | | 110000 | 00 0000 0000 0000 0000 0000 0110 | | | | | | |
| 自陷 OUT | TRAP | x07 | | | 110000 | 00 0000 0000 0000 0000 0000 0111 | | | | | | |
| 自陷 PUTS | TRAP | x08 | | | 110000 | 00 0000 0000 0000 0000 0000 1000 | | | | | | |
| 自陷 IN | TRAP | x09 | | | 110000 | 00 0000 0000 0000 0000 0000 1001 | | | | | | |
| 自陷 GETS | TRAP | x0A | | | 110000 | 00 0000 0000 0000 0000 0000 1010 | | | | | | |
| 自陷 HALT | TRAP | x00 | | | 110000 | 00 0000 0000 0000 0000 0000 0000 | | | | | | |
| (授权)恢复PC | RFE |  | | | 110001 | 00 0000 0000 0000 0000 0000 0000 | | | | | | |

**伪操作**

|  |  |
| --- | --- |
| **.data** address | DLX数据存放在数据区地址address。 |
| **.align** n | 将下面的数据或代码加载到二进制以n个0结尾的地址中。 |
| **.space** size | 在数据区留出size个连续存储单元。 |
| **.text** address | DLX指令存放在地址address中。 |
| **.global** label | 使标记label全局（使以后加载的其他文件也可以使用此标记）。  **必须有.global main，且执行从标记main的指令开始。** |
| **.extern** label | 使用其他文件的全局标记label。 |
| **.byte** byte1,byte2 … | 将字节1、字节2、…连续存储到存储单元中。 |
| **.word** word1,word2 … | 将字1、字2、…连续存储到存储单元中。 |
| **.ascii** “string1”, “string2” … | 将字符串1、字符串2、…连续存储到存储单元中。 |
| **.asciiz** “string1”, “string2” … | 分别在在字符串1、字符串2、…末尾添加x00后连续存储到存储单元中。 |

**设备寄存器映射地址**

|  |  |  |
| --- | --- | --- |
| **MCR** | 机器控制寄存器 | xFFFF 00F8 |
| **KBSR** | 键盘状态寄存器 | xFFFF 0000 |
| **KBDR** | 键盘数据寄存器 | xFFFF 0004 |
| **DSR** | 显示器状态寄存器 | xFFFF 0008 |
| **DDR** | 显示器数据寄存器 | xFFFF 000C |

**特殊寄存器编号**

|  |  |  |  |
| --- | --- | --- | --- |
| **SR** | 12 | x0C | 中断屏蔽，中断允许 |
| **CAUSE** | 13 | x0D | 未决中断位 |
| **EPC** | 14 | x0E | PC |

**DLX寄存器**

**GPR** **通用寄存器**

I/O寄存器：R4

局部变量寄存器：R16、R17、R18、R19、R20、R21、R22、R23

临时寄存器：R8、R9、R10、R11、R12、R13、R14、R15、R24、R25

全局指针：R28

栈指针：R29

帧指针（框架指针）：R30

PC寄存器：R31

HALT影响寄存器：R1、R2、R4、R31

不可改写（值为0）：R0

参数传递寄存器：R4、R5、R6、R7

返回值寄存器：R2、R3

**MCR 机器控制寄存器**

**KBSR 键盘状态寄存器：提供键盘键入字符的状态信息**

**KBDR 键盘数据寄存器：保存由键盘键入的字符的ASCII码**

**DSR 显示器状态寄存器：提供显示器显示的状态信息**

**DDR 显示器数据寄存器：保存将被显示在显示器上的内容的ASCII码**

**PC 程序计数器：指向下一条指令的地址**

**IR 指令寄存器：保存正在处理的指令**

**MAR 地址寄存器：用于寻址**

**MDR 数据寄存器：用于存取单元内容**

**CAUSE 原因寄存器：记录哪些设备发出中断信号（特权模式允许访问）**

**SR 状态寄存器：决定谁能中断处理器（特权模式允许访问）**

**EPC 中断时保存PC**

**DLX模拟器**

**1.配置：**dlx文件夹所在目录配置到环境变量path中。

**2.汇编：**dlxassembler \*.dlx 输出\*.link

**3.链接**：dlxlinker \*.link \*.bin 输出\*.bin，由模拟器运行。