|  |
| --- |
|  |
| 7指令源代码说明 |
|  |
|  |
|  |

# 设计说明

1. 处理器应支持MIPS-Lite1指令集。
   1. MIPS-Lite1＝{addu，subu，ori，lw，sw，beq，jal}。
   2. 所有运算类指令均可以不支持溢出。
2. 处理器为多周期设计。

## 文件结构说明

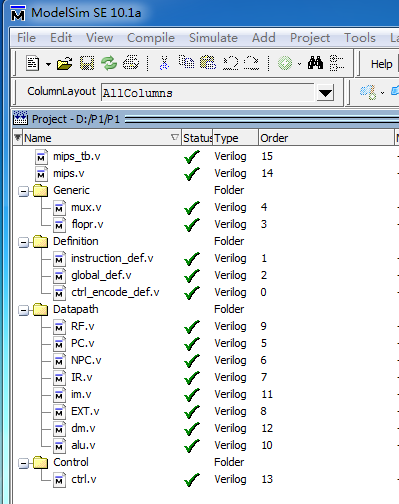


图1工程的文件结构示意图

目录P1中仅给出了基础模块文件，你需要创建自己的工程（project），导入这些文件，形成类似于图1所示的工程结构。其中，Generic内文件为通用文件，Definition内文件为宏定义文件，Datapath包含数据通路文件，Control包含控制器文件。内部的模块定义见下一章节。

## 模块定义

### 2.1. flopr（异步复位触发器）

#### 2.1.1.基本描述

flopr主要异步复位触发器，可实例化为ALUOut、DR等无写使能信号的寄存器。当实例化flopr时，可使用#(XXX)，实例化位宽是XXX的触发器。

#### 模块接口

表格1

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| rst | I | 复位信号 |
| d | I | 输入数据 |
| q | O | 输出数据 |

### 2.2. mux(多路选择器)

#### 2.2.1. 基本描述

mux主要功能是多路选择器。mux.v文件包含二选一、四选一、八选一、十六选一4中多路选择器。实例化多路选择器时，可使用#(XXX)，实例化位宽为XXX的多路选择器。

### 2.2.2. 模块接口

表格2

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| d0、d1、d2... | I | 供选择数据（d0、d1） |
| s | I | 片选信号 |
| y | O | 片选后的数据 |

### 2.3.RF（寄存器文件）

#### 2.3.1.基本描述

RF主要功能是保存寄存器文件，并支持对通用寄存器的访问。

#### 2.3.2. 模块接口

表格3

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| A1[4:0] | I | 需要读的寄存器1的地址 |
| A2[4:0] | I | 需要读的寄存器2的地址 |
| A3[4:0] | I | 需要写的寄存器的地址 |
| WD[31:0] | I | 需要写的寄存器的数据 |
| RFWr | I | 寄存器写使能端  0：寄存器不写  1：寄存器写 |
| clk | I | 时钟信号 |
| RD1 [31:0] | O | 需要读的寄存器1的数据 |
| RD2 [31:0] | O | 需要读的寄存器2的数据 |

#### 2.3.3. 功能定义

表格4

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 读取通用寄存器 | 根据输入的RS、RT域的值，输出相应通用寄存器所存储的数据。 |
| 2 | 写回通用寄存器 | 当RF写使能有效时，将待写的数据写入给定地址的通用寄存器中。 |

### 2.3. ALU（算术逻辑运算单元）

#### 2.3.1.基本描述

ALU主要功能是完成对输入数据的算数逻辑计算，包括加法、减法、按位或运算以及判断两个操作数是否相等。

#### 2.3.2. 模块接口

表格5

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| A[31:0] | I | 操作数A |
| B[31:0] | I | 操作数B |
| ALUOp[1:0] | I | 需要进行的运算  00：加法  01：减法  10：或运算 |
| Zero | O | 两操作数是否相等 |
| C[31:0] | O | 运算结果 |

#### 2.3.3. 功能定义

表格6

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 加法 | 执行加法运算 |
| 2 | 减法 | 执行减法运算 |
| 3 | 或运算 | 执行或运算 |
| 4 | 判断两个操作数是否相等 | 若相等，Zero输出为1；否则，Zero为0 |

### 2.4. EXT（扩展单元）

#### 2.4.1.基本描述

EXT主要功能是将16位的数据扩展为32位数据。

#### 2.4.2. 模块接口

表格7

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| Imm16[15:0] | I | 需要进行扩展的数据 |
| EXTOp[1:0] | I | 扩展方式的控制信号  00：0扩展  01：符号扩展  10：将立即数扩展到高位 |
| Imm32[31:0] | O | 扩展结果 |

#### 2.4.3. 功能定义

表格8

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 32位扩展 | 对16位立即数根据方式进行扩展 |

### 2.5. DM（数据存储器）

#### 2.5.1.基本描述

DM是数据存储器。

#### 2.5.2. 模块接口

表格9

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| din [31:0] | I | 需要写回的数据 |
| DMWr | I | 读写操作的写使能端  0：禁止写  1：允许写 |
| addr[11:2] | I | 访问地址 |
| dout[31:0] | O | 读出的数据 |

#### 2.5.3. 功能定义

表格10

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 读数据存储器 | 输出地址所对应的数据 |
| 2 | 写数据存储器 | 当写使能有效时，将待写数据写入对应地址 |

### 2.6. IM（指令存储器）

#### 2.6.1.基本描述

IM是指令存储器。

#### 2.6.2. 模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| addr[11:2] | I | 访问地址 |
| dout[31:0] | O | 读出的指令 |

#### 2.6.3. 功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 读指令存储器 | 输出地址所对应的指令 |

### 2.7. Ctrl（控制器）

#### 2.7.1.基本描述

#### 2.7.2. 模块接口

表格11

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| rst | I | 复位信号 |
| OP[5:0] | I | 指令格式中的OPCODE域 |
| Funct[5:0] | I | 指令格式中的FUNCT域 |
| Zero | I | ALU输出信号  0：ALU两操作数不等  1：ALU两操作数相等 |
| BSel | O | ALU第二操作数的片选信号  0：RT域对应的数据  1：16位立即数通过EXT模块扩张后的数据 |
| WDSel[1:0] | O | 寄存器写入数据的片选信号  0：ALU计算结果  1：DM读出数据  2：PC（针对JAL指令） |
| RFWr | O | RF的写使能信号  0：禁止写  1：允许写 |
| DMWr | O | 数据存储器写使能信号  0：禁止写  1：允许写 |
| NPCOp[1:0] | O | NPC的片选信号  00：选择PC+4  01：选择分支地址  10：选择跳转地址 |
| EXTOp[1:0] | O | 数据扩展模式选择信号  00：零扩展  01：符号扩展  10：16位立即数扩展到高位 |
| ALUOp[1:0] | O | 传送给ALU的运算控制信号。  00：加法  01：减法  10：或运算 |
| PCWr | O | PC写使能，控制PC写入下一条地址  0：禁止写  1：允许写 |
| IRWr | O | IR写使能，控制IR写入下一条指令  0：禁止写  1：允许写 |
| GPRSel[1:0] | O | RF的写回寄存器的地址片选信号  00：指令格式中的Rd域  01：指令格式中的Rs域  10：$31寄存器（针对JAL） |

#### 2.7.3. 功能定义

表格12

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 控制RF回写寄存器地址的选择 | 选择正确的RF回写地址 |
| 2 | 控制ALU第二操作数的选择 | 选择正确的ALU第二操作数 |
| 3 | 控制RF回写数据的选择 | 选择正确的RF回写数据 |
| 4 | 控制RF写使能 | 控制RF的写使能 |
| 5 | 控制数据存储器的写使能 | 控制数据存储器的写使能端 |
| 6 | 控制下一条指令地址的选择 | 00：PC+4  01：分支地址  10：跳转地址 |
| 7 | 控制EXT单元的扩展方式 | 控制正确的立即数扩展方式 |
| 8 | 控制ALU执行的运算操作 | 控制ALU执行的运算操作。 |

### 2.8. 宏定义文件

#### 2.8.1 global\_def.v

定义DEBUG宏，控制是否调试。

#### 2.8.2 instruction\_def.v

定义7条指令的OPCODE/FUNCT域的。

#### 2.8.3 ctrl\_encode\_def.v

相关控制信号的宏。其中使用Verilog描述控制器的控制信号时，需与该宏保持一致。也可根据情况增加控制信号的宏。

### 2.9. mips\_tb.v

激励文件。初始化时钟周期等信号。

## 测试

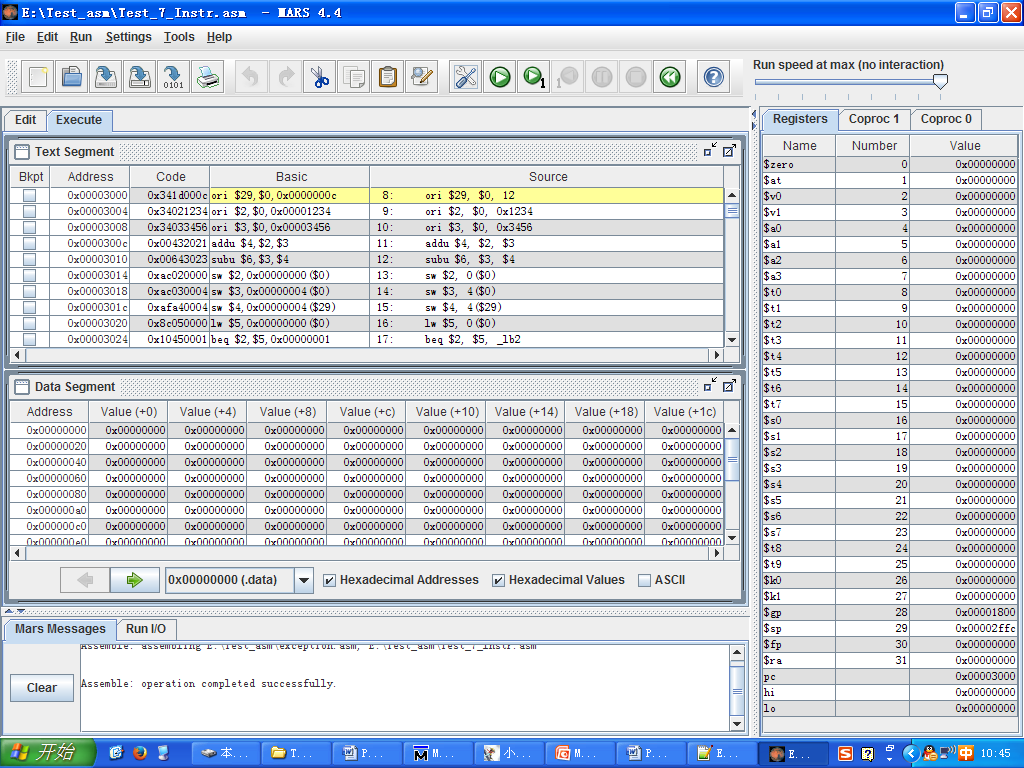
### 测试指令

见Test\_7\_Instr.asm文件。

### 机器码文件生成

打开Mars4\_4.jar→ 打开Test\_7\_Instr.asm→ 点击工具栏的如下图标可执行当前汇编指令  
2.png

点击后，界面如下：



其中，工具栏的按钮说明如下：下图1按钮可直接执行到程序末；2按钮可单步执行；3按钮重新执行上一条指令；4按钮重新执行所有指令。



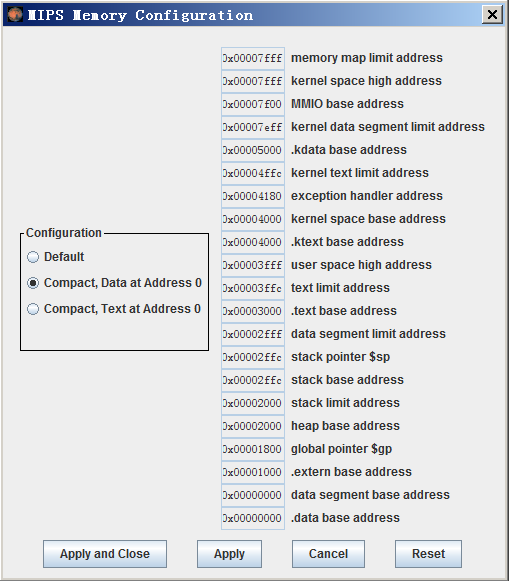
3

4

2

1

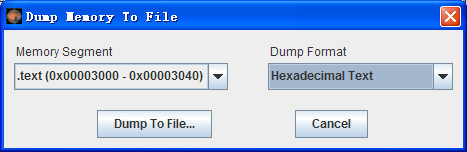
再运行指令前，需要保证Memory Configuration与处理器设计保持一致。可通过Settings → Memory Configuration设置。设置为如下模式后应用，即可执行汇编指令。



针对当前汇编指令，可单击工具栏中如下按钮生成二进制文件：

4.png

单击后，显示为下图



选择Dump Format为Hex Text，生成16进制，单击Dump To File...按钮后选择路径，即可生成机器码作为多周期处理器测试文件。