

Project1 MIPS 单周期处理器

一、设计说明

1. 处理器应支持 MIPS-Lite1 指令集。
 - a) MIPS-Lite1={addu, subu, ori, lw, sw, beq, jal}。
 - b) 所有运算类指令均可以不支持溢出。
2. 处理器为单周期设计。给出了部分源代码，请补充完整实现支持 7 条指令的单周期 cpu。

1. 文件结构说明

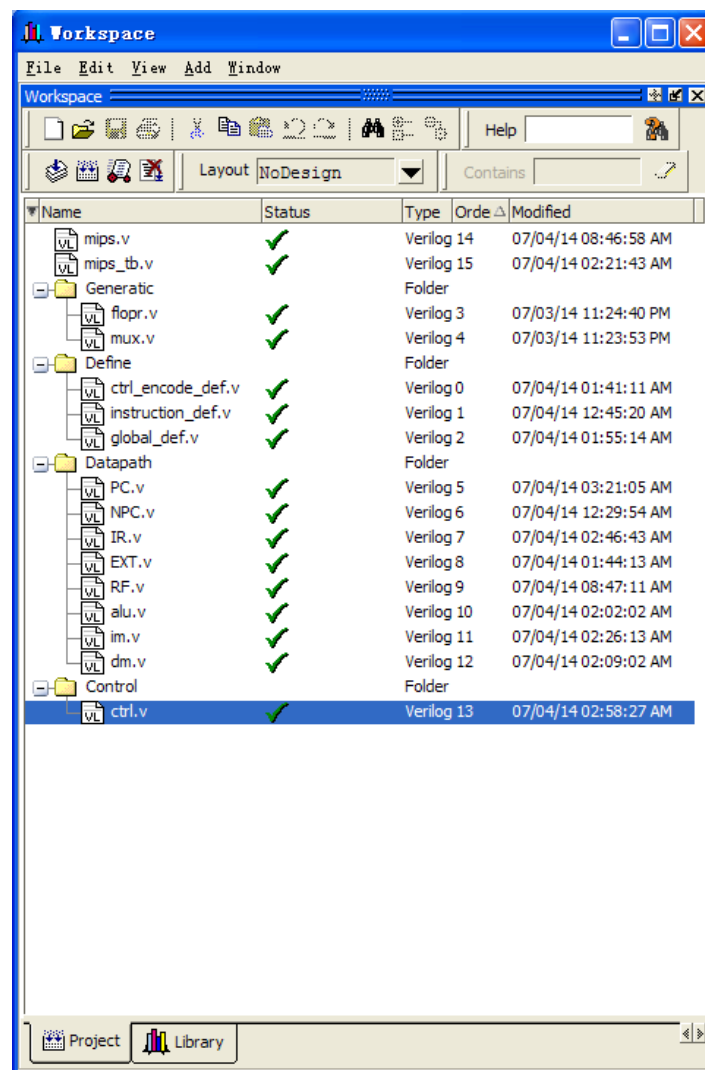


图 1 工程的文件结构示意图

其中，Generatic内文件为通用文件，Define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。

2. 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。

2.1.2. 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

ctrl 主要功能是产生根据输入指令控制数据通路各模块的控制信号。其中状态表如教材所示。

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。

3. 测试

3.1. 测试指令

见 Test_7_Instr.asm 文件

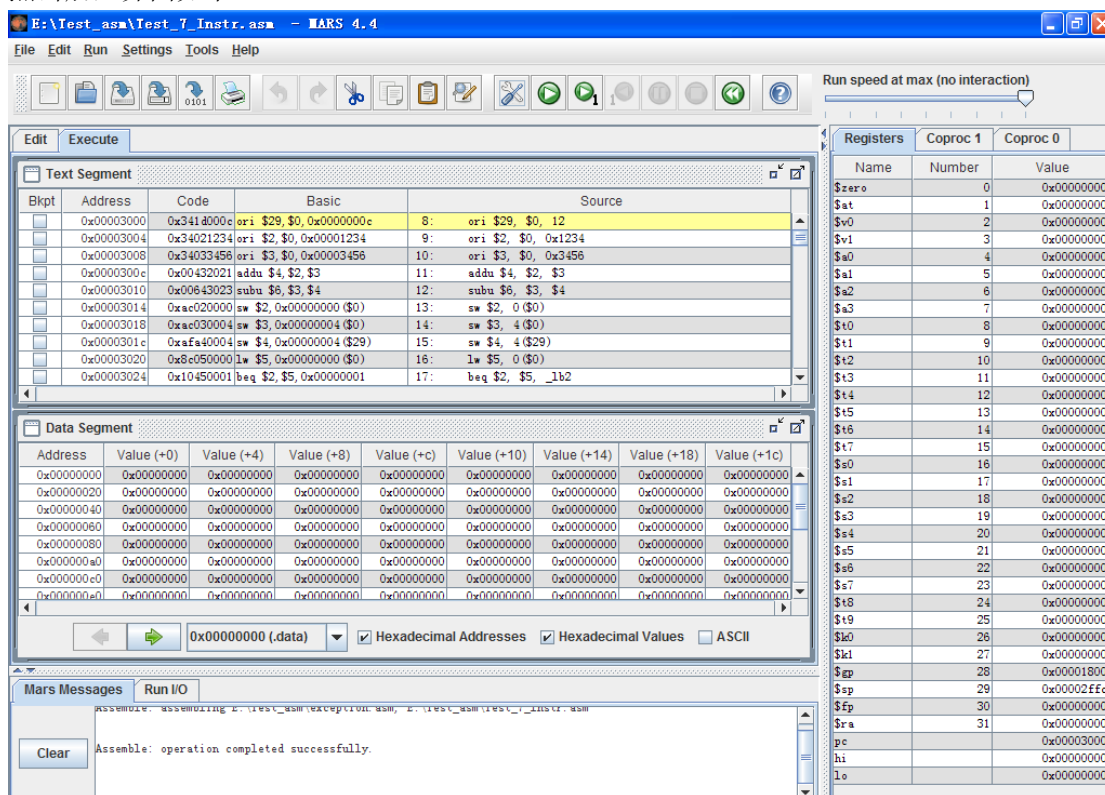
3.2. 机器码文件生成

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

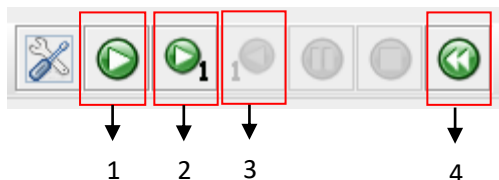




点击后，界面如下：



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。



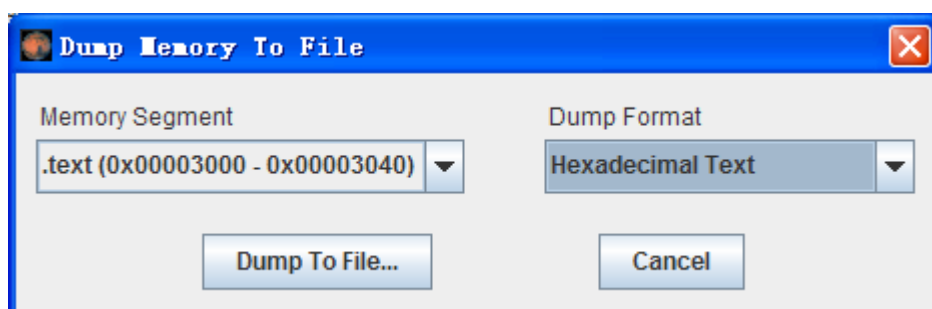
再运行指令前，需要保证 Memory Configuration 与处理器设计保持一致。可通过 Settings → Memory Configuration 设置。设置为如下模式后应用，即可执行汇编指令。



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：



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。