

8.4 实验四 31 条指令单周期 CPU 设计

1. 实验介绍

在本实验中，我们将使用 Verilog HDL 语言实现 31 条 MIPS 指令的 CPU 的设计，前仿真，后仿真和下板调试运行。

2. 实验目标

- 深入掌握 CPU 的构成及工作原理
- 设计 31 条指令的 CPU 的数据通路及控制器
- 使用 Verilog HDL 语言设计实现 31 条指令的 CPU 下板运行

3. 实验原理

(1) 需要实现的 31 条 MIPS 指令，如表 8.4.2 所示。各条指令格式及功能详细说明参阅 MIPS_Architecture_MIPS32_InstructionSet 手册。

(2) 单周期 CPU 数据通路设计

- 根据指令的功能，确定每条指令在执行过程中所用到的部件（包括取指）。
- 根据该指令所用的部件，用表格列出，并在表格中填入每个部件的数据输入来源。
- 根据表格所涉及部件和部件的数据输入来源，画出每条指令的数据通路。
- 最后将所有指令数据通路合成一个总的数据通路。

(3) 控制部件设计

- 根据每条指令功能，在已形成的数据通路下，画出每条指令从取指到执行过程的指令流程图。
- 根据指令流程图，编排指令取指到执行的操作时间表。
- 根据指令操作时间表，写出每个控制信号的逻辑表达式。
- 根据逻辑表达式，用门电路实现，完成控制部件设计。

(4) CPU 前仿真测试

- 单条指令的测试，对所设计的 31 条指令，一条一条指令进行验证。
- 指令边界数据测试，对于每条指令所对应的边界数据，一条一条指令进行验证。
- 随机指令序列测试，可以自行编写一些符合 MIPS 规范的指令序列。将这序列分别放到 CPU 仿真状态下和 MARS 上去执行，分别产生两个执行结果文件，比较执行结果文件来判断 CPU 执行指令是否正确。也可利用 Mips246.tongji.edu.cn 上提供的工具，自动生成随机指令测试序列。
- 程序测试，运行一个有意义的程序，观察运行的结果。本书的网站 Mips246.tongji.edu.cn 提供一些程序验证，类似快速排序和斐波拉契数列等程序供验证使用。

(5) 后仿真测试

后仿真分为两步，先做指令序列测试，再做程序测试。后仿真指令序列和程序测试和前仿真可以是一样的，但是要注意，后仿真时 CPU 中不可有不可综合语句，time, defparam, \$finish, fork, join, initial, delays, UDP, wait 等语句都是不可综合的。后仿真的测试指令通过以 IP 核方式实现的 RAM 来进行测试，其余操作与前仿真一样。

(6) 下板测试

由于我们自行编写的指令 RAM 用来初始化内存的 `initial` 指令是不可综合的, 无法在开发板上运行, 所以, 我们可以使用 Vivado 提供的 IP 核来替换我们的 ram, 其可以使用一个 `coe` 文件来初始化内存。

表 8.4.2 31 条 MIPS 指令表

Mnemonic Symbol	Format						Sample
Bit #	31..26	25..21	20..16	15..11	10..6	5..0	
R-type	op	rs	rt	rd	shamt	func	
add	000000	rs	rt	rd	0	100000	add \$1,\$2,\$3
addu	000000	rs	rt	rd	0	100001	addu \$1,\$2,\$3
sub	000000	rs	rt	rd	0	100010	sub \$1,\$2,\$3
subu	000000	rs	rt	rd	0	100011	subu \$1,\$2,\$3
and	000000	rs	rt	rd	0	100100	and \$1,\$2,\$3
or	000000	rs	rt	rd	0	100101	or \$1,\$2,\$3
xor	000000	rs	rt	rd	0	100110	xor \$1,\$2,\$3
nor	000000	rs	rt	rd	0	100111	nor \$1,\$2,\$3
slt	000000	rs	rt	rd	0	101010	slt \$1,\$2,\$3
sltu	000000	rs	rt	rd	0	101011	sltu \$1,\$2,\$3
sll	000000	0	rt	rd	shamt	000000	sll \$1,\$2,10
srl	000000	0	rt	rd	shamt	000010	srl \$1,\$2,10
sra	000000	0	rt	rd	shamt	000011	sra \$1,\$2,10
sllv	000000	rs	rt	rd	0	000100	sllv \$1,\$2,\$3
srlv	000000	rs	rt	rd	0	000110	srlv \$1,\$2,\$3
srav	000000	rs	rt	rd	0	000111	srav \$1,\$2,\$3
jr	000000	rs	0	0	0	001000	jr \$31
Bit #	31..26	25..21	20..16	15..0			
I-type	op	rs	rt	immediate			
addi	001000	rs	rt	Immediate(- ~ +)		addi \$1,\$2,100	
addiu	001001	rs	rt	Immediate(- ~ +)		addiu \$1,\$2,100	
andi	001100	rs	rt	Immediate(0 ~ +)		andi \$1,\$2,10	
ori	001101	rs	rt	Immediate(0 ~ +)		andi \$1,\$2,10	
xori	001110	rs	rt	Immediate(0 ~ +)		andi \$1,\$2,10	
lw	100011	rs	rt	Immediate(- ~ +)		lw \$1,10(\$2)	
sw	101011	rs	rt	Immediate(- ~ +)		sw \$1,10(\$2)	
beq	000100	rs	rt	Immediate(- ~ +)		beq \$1,\$2,10	
bne	000101	rs	rt	Immediate(- ~ +)		bne \$1,\$2,10	
slti	001010	rs	rt	Immediate(- ~ +)		slti \$1,\$2,10	
sltiu	001011	rs	rt	Immediate(- ~ +)		sltiu \$1,\$2,10	
lui	001111	00000	rt	Immediate(- ~ +)		Lui \$1, 10	
Bit #	31..26	25..0					
J-type	op	Index					
j	000010	address				j 10000	
jal	000011	address				jal 10000	

4. 实验步骤

1. 新建 Vivado 工程，编写各个模块；
2. 用 ModelSim 前仿真逐条测试所有指令；
3. 用 ModelSim 前仿真逐条测试所有指令边界数据；
4. 用 ModelSim 前仿真测试指令序列；
5. 用 ModelSim 前仿真运行测试程序；
6. 用 ModelSim 进行后仿真测试指令序列；
7. 用 ModelSim 进行后仿真运行测试程序；
8. 配置 XDC 文件，综合下板，并观察实验现象；
9. 按照要求书写实验报告。