

2022 春季学期

计算机组成原理

32  
位  
多  
周  
期  
CPU  
设  
计  
报  
告

学院：信息科学与工程学院

班级：计算机基地班

姓名：吴芑、袁雨辰、王研、李付彬、周恒

一、项目简述

1. 项目环境与级别

设计语言：Verilog 硬件描述语言  
仿真环境：Vivado  
设计级别：RTL 级  
指令集：MIPS

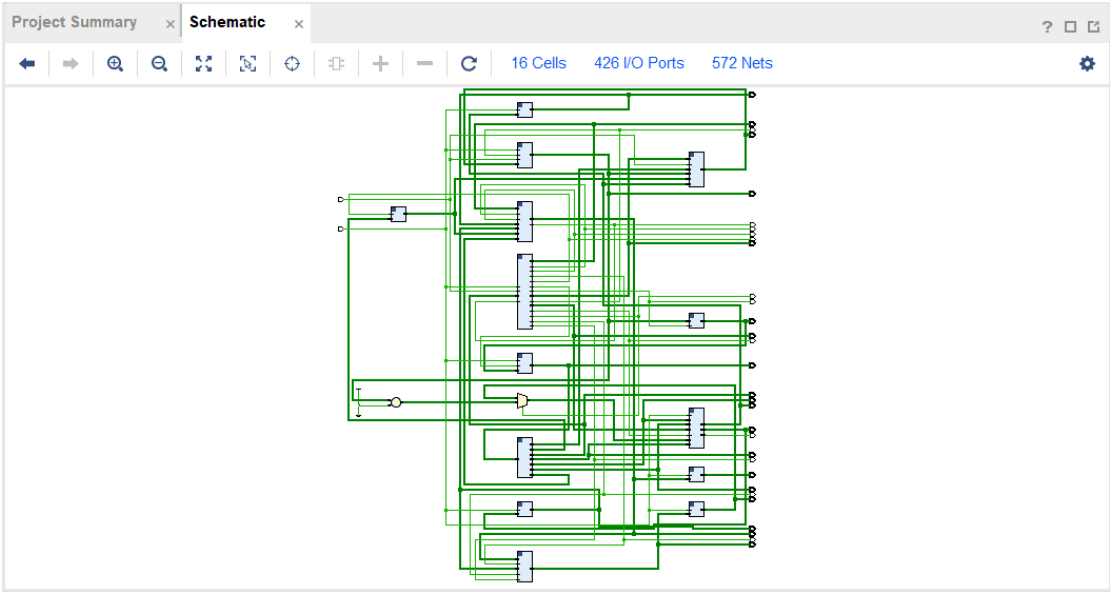
2. 任务目标

设计一个 MIPS 指令集的多周期 CPU。  
用汇编程序编译为二进制机器码来对多周期 CPU 进行测试。

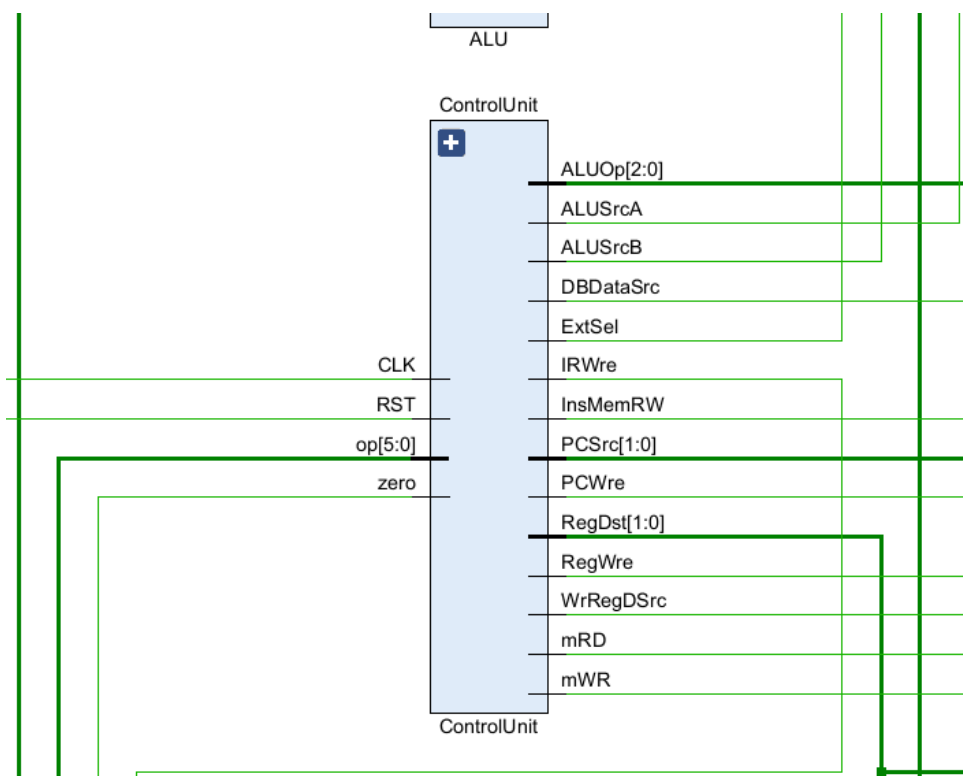
3. 实现的 MIPS 指令集

序号	指令名称	格式
1	add	add rd, rs, rt
2	sub	sub rd, rs, rt
3	addi	addi rt, rs, <b>immediate</b>
4	or	or rd, rs, rt
5	and	and rd, rs, rt
6	ori	ori rt, rs, <b>immediate</b>
7	sll	sll rd, rt, sa
8	slt	slt rd, rs, rt
9	sltiu	sltiu rt, rs, immediate
10	sw	sw rt, <b>immediate</b> (rs)
11	lw	lw rt, <b>immediate</b> (rs)
12	beq	beq rs, rt, <b>immediate</b>
13	bltz	bltz rs, <b>immediate</b>
14	j	j addr
15	jr	jr rs
16	jal	jal addr
17	halt	halt (停机指令)

二、电路图



具体模块示例：



### 三、各模块介绍

#### 1. pcAdd

模块功能：根据控制信号 PCSrc，计算获得下一个 pc 以及控制信号 Reset 重置。

实现思路：首先决定何时引起触发，将里面涉及的信号量作为敏感变量，主要是为了确保下一条 pc 能够正确得到。

#### 2. PC

模块功能：根据控制信号 PCWre，判断 pc 是否改变以及根据 Reset 信号判断是否重置。

实现思路：将时钟信号的上升沿和控制信号 Reset 作为敏感变量，使得 pc 在上升沿的时候发生改变或被重置。

#### 3. InsMEM

模块功能：依据当前 pc 和信号量 InsMemRW，读取指令寄存器中，相对应地址的指令。

实现思路：将 pc 的输入作为敏感变量，当 pc 发生改变的时候，则进行指令的读取，根据相关的地址，输出指令寄存器中相对应的指令。

#### 4. IR

模块功能：为了使指令代码保持稳定。

实现思路：将时钟上升沿作为敏感信号，同时依据信号量 IRWre，对 IR 寄存器进行写入。

#### 5. InstructionCut

模块功能：对指令进行分割，获得相对应的指令信息。

实现思路：根据各种类型的指令结构，将指令分割，得到相对应的信息。

#### 6. ControlUnit

模块功能：控制单元，依据指令的操作码（op）、标记符（ZERO）以及当前 CPU 状态，依据表三 控制信号、指令以及执行状态之间的相互关系，输出相匹配控制信号量。

实验思路：设计一个 CPU 的状态机，依据指令的操作码（op）、当前状态以及重置信号（RST），其状态在上升沿的时候发生，同时依据当前的状态、操作码（op）以及标记符（ZERO）修改并且输出控制信号。本模块为多周期 CPU 中最重要的模块，需要注意部分信号在特定的 CPU 状态才能输出相应的使能，否则将出现错误。

#### 7. RegisterFile

模块功能：寄存器组，通过控制单元输出的控制信号，进行相对应的读或写操作。

实现思路：当 ReadReg1 或者 ReadReg2 发生改变的时候，即对寄存器组进行数据读取。至于数据写入，则选择在时钟的下降沿时候进行操作，同时写入信号（RegWre）必须为 1 且写入寄存器的地址不能为 \$0。至于数据的写入地址则在任意信号发生改变则依据写入地址的控制信号（RegDst）进行修改，使得其在数据写回阶段确保为正确的写入地址。

#### 8. SignZeroExtend

模块功能：根据指令相关的控制信号 ExtSel，对立即数进行扩展。

实现思路：根据控制信号 ExtSel 判断是 0 扩展还是符号扩展，然后进行相对应的扩展

#### 9. ALU

模块功能：算术逻辑单元，对两个输入依据 ALUOp 进行相对应的运算。

实现思路：依据实验原理中的 ALU 运算功能表（表 2）完成操作码对应的操作，当 ReadDatat1、ReadDatat2、ALUSrcA、ALUSrcB、ALUOp 任意发生改变的时候，即进行运算。

#### 10. DataMEM

模块功能：数据存储器，通过控制信号，对数据寄存器进行读或者写操作，并且此处模块额外合并了输出 DB 的数据选择器，此模块同时输出写回寄存器组的数据 DB。

实现思路：将相关信号量作为敏感变量，假如读数据寄存器的信号量（mRD）为 1 的时候，则对数据寄存器进行读操作，假如写数据寄存器的信号量（mWR）为 1 的时候，则对数据寄存器进行写操作。

#### 11. 寄存器 ADR、BDR、ALUoutDR、DBDR

模块功能：切分数据通路，将大组合逻辑切分为若干个小组合逻辑，大延迟变为多个分段小延迟。

实现思路：在时钟上升沿将相应的数据写入寄存器中。四个寄存器功能相同，可以使用用一个模块实例得到。

#### 12. 顶层模块：MultiCycleCPU

实现思路：在顶层模块中将各个已实现的底层模块进行实例，并且用 verilog 语言将各个模块用线连接起来。

### 四、仿真测试

地址	汇编程序	指令代码					
		op (6)	rs (5)	rt (5)	rd (5)/immediate (16)	16 进制数代码	
0x00000000	addi \$1, \$0, 8	000010	00000	00001	0000 0000 0000 1000	=	08010008

0x00000004	ori \$2, \$0, 2	010010	00000	00010	0000 0000 0000 0010	=	48020002
0x00000008	or \$3, \$2, \$1	010000	00010	00001	0001 1000 0000 0000	=	40411800
0x0000000C	sub \$4, \$3, \$1	000001	00011	00001	0010 0000 0000 0000	=	04612000
0x00000010	and \$5, \$4, \$2	010001	00100	00010	0010 1000 0000 0000	=	44822800
0x00000014	sll \$5, \$5, 2	011000	00000	00101	0010 1000 1000 0000		60052880
0x00000018	beq \$5, \$1, -2 (=, 转 14)	110100	00101	00001	1111 1111 1111 1110		D0A1FFFE
0x0000001C	jal 0x00000040	111010	00000	00000	0000 0000 0001 0000		E8000010
0x00000020	slt \$8, \$12, \$1	100110	01100	00001	0100 0000 0000 0000		99814000
0x00000024	addi \$13, \$0, -2	000010	00000	01101	1111 1111 1111 1110		080DFFFE
0x00000028	slt \$9, \$8, \$13	100110	01000	01101	0100 1000 0000 0000		990D4800
0x0000002C	sltiu \$10, \$9, 2	100111	01001	01010	0000 0000 0000 0010		9D2A0002
0x00000030	sltiu \$11, \$10, 0	100111	01010	01011	0000 0000 0000 0000		9D4B0000
0x00000034	addi \$13, \$13, 1	000010	01101	01101	0000 0000 0000 0001		09AD0001
0x00000038	bltz \$13, -2 (<0, 转 34)	110110	01101	00000	1111 1111 1111 1110		D9A0FFFE
0x0000003C	j 0x0000004C	111000	00000	00000	0000 0000 0001 0011		E0000013
0x00000040	sw \$2, 4(\$1)	110000	00001	00010	0000 0000 0000 0100		C0220004
0x00000044	lw \$12, 4(\$1)	110001	00001	01100	0000 0000 0000 0100		C42C0004
0x00000048	jr \$31	111001	11111	00000	0000 0000 0000 0000		E7E00000
0x0000004C	halt	111111	00000	00000	0000000000000000	=	FC000000
0x00000050							
0x00000054							

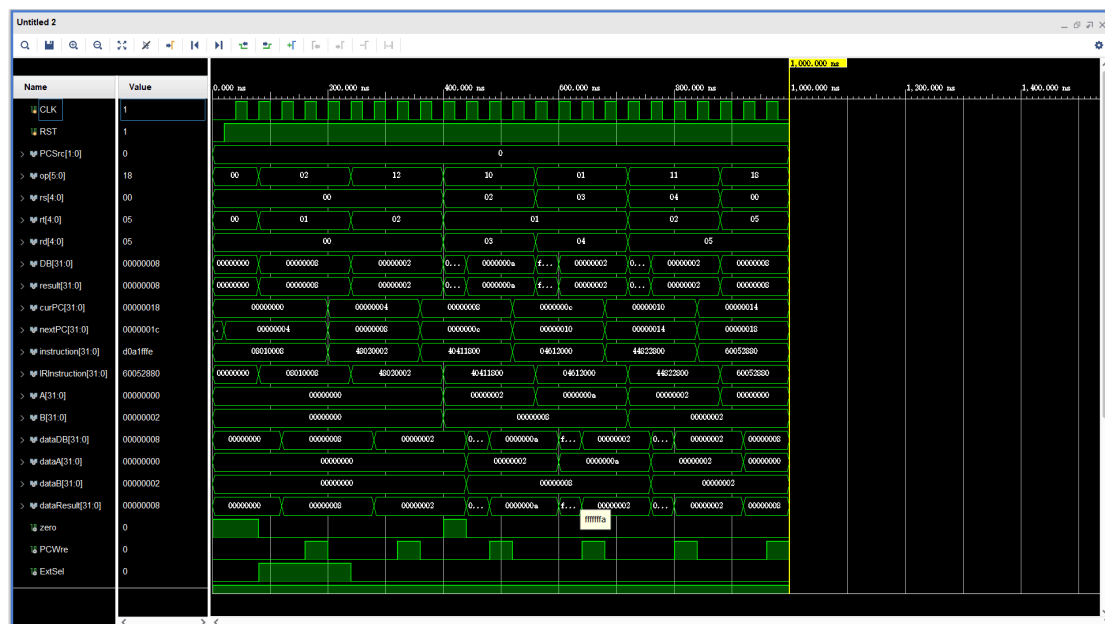
将以上指令编译为二进制机器指令：

```

08 01 00 08
48 02 00 02
40 41 18 00
04 61 20 00
44 82 28 00
60 05 28 80
D0 A1 FF FE
E8 00 00 10
99 81 40 00
08 0D FF FE
99 0D 48 00
9D 2A 00 02
9D 4B 00 00
09 AD 00 01
D9 A0 FF FE
E0 00 00 13
C0 22 00 04
C4 2C 00 04
E7 E0 00 00
FC 00 00 00

```

Behavioral simulation:



## 五、性能指标

Runs	Clock Networks	Methodology	Timing	Utilization
Clock Summary				
Name	Waveform	Period (ns)	Frequency (MHz)	
test	{0.000 5.000}	10.000	100.000	

由上图知：CPU 的频率为 100.0MHz

CPI：1.25

## 六、实验分工

吴芃：仿真测试和文档编写；

袁雨辰：CPU 底层模块代码实现；

王研：CPU 指令实现；

李付彬：性能测试和 PPT 编写；

周恒：监督与审稿。