

武汉大学计算机学院 本科生课程设计报告

RISC-V CPU 设计

专业名称：计算机科学与技术(弘毅学堂)

课程名称：计算机系统综合设计

指导教师：内切圆 副教授

学生学号：2020400004071

学生姓名：外接圆

二〇二三年七月

郑重声明

本人呈交的实验报告，是在指导老师的指导下，独立进行实验工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本实验报告不包含他人享有著作权的内容。对本实验报告做出贡献的其他个人和集体，均已在文中以明确的方式标明。本实验报告的知识产权归属于培养单位。

本人签名：_____

日期：_____

摘 要

本实验的实验目的是完成基于 \LaTeX 的武汉大学计算机学院实验报告设计，使其支持绝大多数课程要求（要求交 word 的课程除外）。

实验设计主要遵循【计算机组成与设计课程设计】[\[1\]](#) 的实验报告格式，其他课程大同小异。

实验内容主要包括：各个标题、段落、格式的生成，并可插入代码段，本报告使用 Verilog 语言。

实验结论为： \LaTeX 可以高度复现武汉大学（计算机学院）实验报告。

仓库地址：<https://github.com/wjy-yy/whucs-latex>

关键词：武汉大学；计算机学院； \LaTeX ；实验报告

目 录

1	引言	5
1.1	实验目的	5
1.2	国内外研究现状	5
2	实验环境介绍	6
2.1	L ^A T _E X	6
2.2	本地编译环境	6
3	概要设计 - 单周期	7
3.1	图片示例	7
3.2	表格示例	7
3.2.1	功能描述	7
3.2.2	模块接口	7
3.3	RF（寄存器文件）	8
3.3.1	功能描述	8
3.3.2	模块接口	8
4	详细设计 - 单周期	9
4.1	参考文献引用示例	9
4.2	代码段插入示例	9
4.3	PC（程序计数器）	10
5	概要设计 - 流水线	11
6	详细设计 - 流水线	11
7	测试及结果分析	12
7.1	分栏示例（此处为代码分栏）	12
7.2	仿真测试结果	12
7.2.1	引用图片标号示例	12
8	实验总结	12
8.1	实验总结	12
8.2	取得的收获	13

1 引言

1.1 实验目的

section 的大标题为黑体小二字号，subsection 的小标题为黑体小四字号。

1.2 国内外研究现状

目前暂未发现其他仓库上传了武汉大学计算机学院实验报告 L^AT_EX 模板。

2 实验环境介绍

2.1 L^AT_EX

L^AT_EX 是一个文档准备系统 (Document Preparing System), 它非常适用于生成高印刷质量的科技类和数学类文档。它也能够生成所有其他种类的文档, 小到简单的信件, 大到完整的书籍。L^AT_EX 使用 TEX [6] 作为它的排版引擎。

使用文档参见: <https://texdoc.org/serve/lshort-zh-cn.pdf/0>

2.2 本地编译环境

一般 macOS 推荐使用 TeX Live 作为编译器, VS Code 或 TeX Studio 作为编辑器。

3.3 RF（寄存器文件）

3.3.1 功能描述

控制 32 个寄存器的读写操作。在任意时刻都可以根据输入 $A1[4:0]$ 和 $A2[4:0]$ 的值读取指定编号的寄存器的值，并赋值给 $RD1[31:0]$ ， $RD2[31:0]$ 作为输出。当时钟处在上升沿时，并且写寄存器信号 $RFWr$ 有效，由 $A3[4:0]$ 的值决定向指定编号的寄存器写入 $WD[31:0]$ 的值。

3.3.2 模块接口

表略，参见 `mac-example/`。

4 详细设计 - 单周期

4.1 参考文献引用示例

依据教材《计算机组成与设计：软/硬件接口 RISC-V（第五版）》[1] 中单周期架构 CPU 各模块的输入、输出端口，功能信号的规定以及 RISC-V 指令集中各个指令对应的数据周转流程与对应信号。模块中所有连线基于该架构进行实现。

4.2 代码段插入示例

SCPU.v 设计如下：（省略连线，详见附件中.v 文件）

```
1  `include "ctrl_encode_def.v"
2  module SCPU(
3      input      clk,          // 时钟信号
4      input      reset,        // 复位信号
5      input [31:0] inst_in,     // 输入指令
6      input [31:0] Data_in,     // 来自 DM 的数据
7
8      output      mem_w,        // DM 写使能
9      output [31:0] PC_out,     // 输出 PC 地址 ( debug 用)
10
11     output [31:0] Addr_out,    // ALU 输出值 (一般用于计算地址)
12     output [31:0] Data_out,    // 输入 DM 的数据
13     output [2:0] DMType        // 选择访问 DM 的字节数
14 );
15     assign Addr_out=aluout;
16     assign B = (ALUSrc) ? immout : RD2; // ALU 第二个操作数的来源
17     assign Data_out = RD2;
18
19     assign iimm_shamt=inst_in[24:20]; //UJ 型指令
20     assign iimm=inst_in[31:20]; //I 型指令
21     assign simm={inst_in[31:25],inst_in[11:7]}; //S 型指令
22     assign bimm={inst_in[31],inst_in[7],inst_in[30:25],inst_in
        [11:8]}; //SB 型
23     assign uimm=inst_in[31:12]; //U 型
24     assign jimm={inst_in[31],inst_in[19:12],inst_in[20],inst_in
        [30:21]}; //J 型
25
26     assign Op = inst_in[6:0]; // opcode 部分
27     assign Funct7 = inst_in[31:25]; // funct7 部分
28     assign Funct3 = inst_in[14:12]; // funct3 部分
29     assign rs1 = inst_in[19:15]; // rs1 序号
30     assign rs2 = inst_in[24:20]; // rs2 序号
```

```

31     assign rd = inst_in[11:7]; // rd 序号
32     assign Imm12 = inst_in[31:20];
33     assign IMM = inst_in[31:12];
34
35     // 各模块的例化
36     ctrl U_ctrl(.Op(Op), .Funct7(Funct7), .Funct3(Funct3), .Zero(
        Zero), .RegWrite(RegWrite), .MemWrite(mem_w),
37         .EXTOp(EXTOp), .ALUOp(ALUOp), .NPCOp(NPCOp),
38         .ALUSrc(ALUSrc), .GPRSel(GPRSel), .WDSel(WDSel), .DMType(
        DMType) );
39     PC U_PC(.clk(clk), .rst(reset), .NPC(NPC), .PC(PC_out) );
40     NPC U_NPC(.PC(PC_out), .NPCOp(NPCOp), .IMM(immout), .NPC(NPC),
        .aluout(aluout) );
41     EXT U_EXT(.iimm_shamt(iimm_shamt), .iimm(iimm), .simm(simm), .
        bimm(bimm), .uimm(uimm), .jimm(jimm),
42         .EXTOp(EXTOp), .immout(immout) );
43     RF U_RF(.clk(clk), .rst(reset), .RFWr(RegWrite),
44         .A1(rs1), .A2(rs2), .A3(rd), //Read1, Read2, Write
45         .WD(WD), //Write data
46         .RD1(RD1), .RD2(RD2) //Read1 Read2
47     );
48     alu U_alu(.A(RD1), .B(B), .ALUOp(ALUOp), .C(aluout), .Zero(Zero
        ), .PC(PC_out) );
49 always @*
50 begin
51     case(WDSel)
52         `WDSel_FromALU: WD<=aluout;
53         `WDSel_FromMEM: WD<=Data_in;
54         `WDSel_FromPC: WD<=PC_out+4;
55     endcase
56 end
57 endmodule

```

4.3 PC（程序计数器）

代码略，参见 mac-example/。

5 概要设计 - 流水线

6 详细设计 - 流水线

7 测试及结果分析

7.1 分栏示例（此处为代码分栏）

```
1  addi x5, x0, 1000
2  addi x4, x0, 10
3  lui x6, 0xffff
4  sw x6, 4(x5)
5  lh x7, 4(x5)
6  sub x8, x7, x6
7  ori x9, x0, 111
8  andi x9, x9, 0
9  skip1:
10 xori x9, x9, 1
11 bne x0, x9, skip1
12 or x9, x4, x0
13 srli x10, x5, 10
14 skip2:
15 add x10, x9, x9
16 bge x9, x10, skip2
17 sw x9, 8(x5)
18 lw x28, 8(x5)
19 addi x29, x4, 0
20 jal x1, mut
21 sll x11, x10, x9
22 srai x14, x10, 2
23 sb x11 16(x5)
24 and x8, x7, x6
25 auipc x12, 4
26 lbu x13, 16(x5)
27 bgeu x8, x7, end
28
29 beq x0, x0, end
30
31 mut: #x28*x29
32 beq x29, x0, end4
33 addi x30, x0, 0
34 addi x31, x0, 0
35 loop:
36 add x31, x31, x28
37 addi x30, x30, 1
38 blt x30, x29, loop
39 end4:
40 jalr x0, x1, 0
41
42 end:
```

7.2 仿真测试结果

7.2.1 引用图片标号示例

下一周期时，第 6 行代码仍在 EX 阶段，第 5 行代码执行 WB 阶段，此时 MEM/WB 里已有 x7 的值，通过旁路将其传入 EX/MEM 阶段即可。

注意下一行“图”后面的数字可以点击。

如图2为第 8 周期的仿真图像。

此时还没有 x7 的数据 (0xffff0000) 出现，所以无法将其前递到 ALU 的操作数 a 处。

8 实验总结

8.1 实验总结

本模板如对你产生帮助，请在<https://github.com/wjy-yy/whucs-latex>进行 star★。

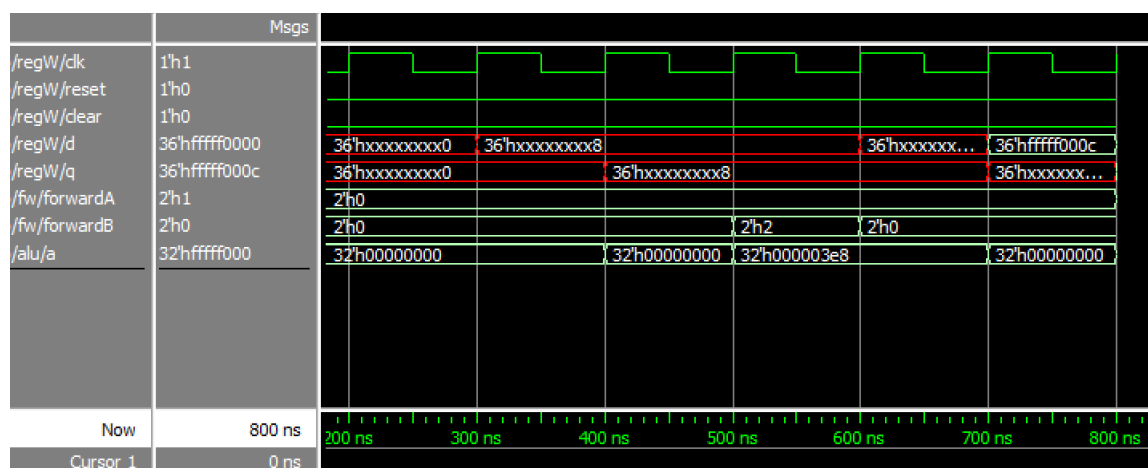


图 2: 出现数据冒险时相关线路的仿真（第 8 周期）

8.2 取得的收获

参考文献

- [1] David A. P, John L.H, 计算机组成与设计：硬件/软件接口 [M]. 易江芳, 刘先华. 北京：机械工业出版社，2020.
- [2] 芮雪, 王亮亮, 杨琴. 国产处理器研究与发展现状综述 [J]. 现代计算机 (专业版),2014(08):15-19.
- [3] 潘树朋, 刘有耀.RISC-V 微处理器以及商业 IP 的综述 [J]. 单片机与嵌入式系统应用,2020,20(06):5-8+12.
- [4] 全球首颗智能穿戴领域人工智能芯片发布 [J]. 智能城市,2019,5(10):191.
- [5] 雷思磊.RISC-V 架构的开源处理器及 SoC 研究综述 [J]. 单片机与嵌入式系统应用,2017,17(02):56-60+76.
- [6] 袁攀. 基于嵌入式 RISC-V 微处理器的流水线研究与设计 [D]. 长沙理工大学,2021.DOI:10.26985/d.cnki.gcsjc.2021.000811.
- [7] 李亚民. 计算机原理与设计：Verilog HDL 版 [M]. 北京：清华大学出版社，2011.

教师评语评分

评语: _____

评阅人: _____

年 月 日

(备注: 对该实验报告给予优点和不足的评价, 并给出百分制评分。)