中国科学技术大学计算机学院

《计算机组成原理实验》报告

实验题目：单周期 CPU 设计

学生姓名：黄瑞轩

学生学号：PB20111686

完成日期：2022.4.8

计算机实验教学中心制

2020 年 09 月

实验题目

单周期 CPU 设计

实验目的

- 理解单周期 CPU 的结构和工作原理
- 掌握单周期 CPU 的设计和调试方法
- 熟练掌握数据通路和控制器的设计和描述方法

实验环境

- Nexys4-DDR
- Vivado 2019.1

实验 1: 单周期 CPU 设计、CPU+PDU 下载和指令逐条测试

● 先导器件设计

按理论部分的单周期 CPU 设计数据通路，在本实验中需要一些先导器件。PC 模块见附件 pc_calc.v；数据存储器见附件 mem_data_c.v；指令存储器见附件 mem_inst_c.v；寄存器文件模块见附件 reg_file.v；ALU 模块见附件 alu.v；立即数扩展模块见附件 imm.v；控制器模块见附件 controller.v；分支模块见附件 brah.v。

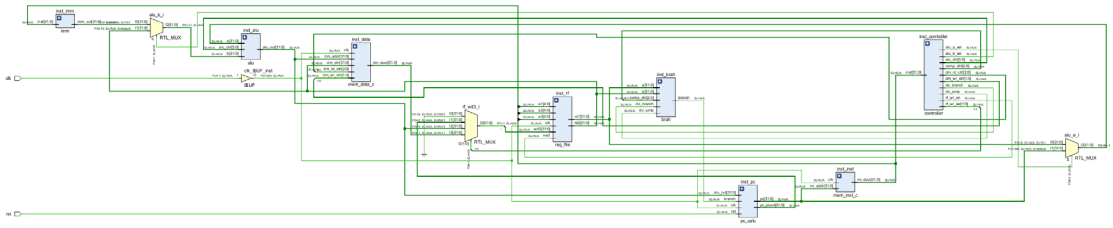
● 单周期 CPU 设计

➔ *要求实现的功能*

可执行 add, addi, sub, auipc, lw, sw, beq, blt, jal, jalr 这 10 条指令。并且能配合外设和调试单元 PDU，实现对 CPU 的下载测试。

➔ *CPU 设计和核心数据通路*

设计采用 COD5 教材单周期 CPU 方案，RTL 电路图如下。

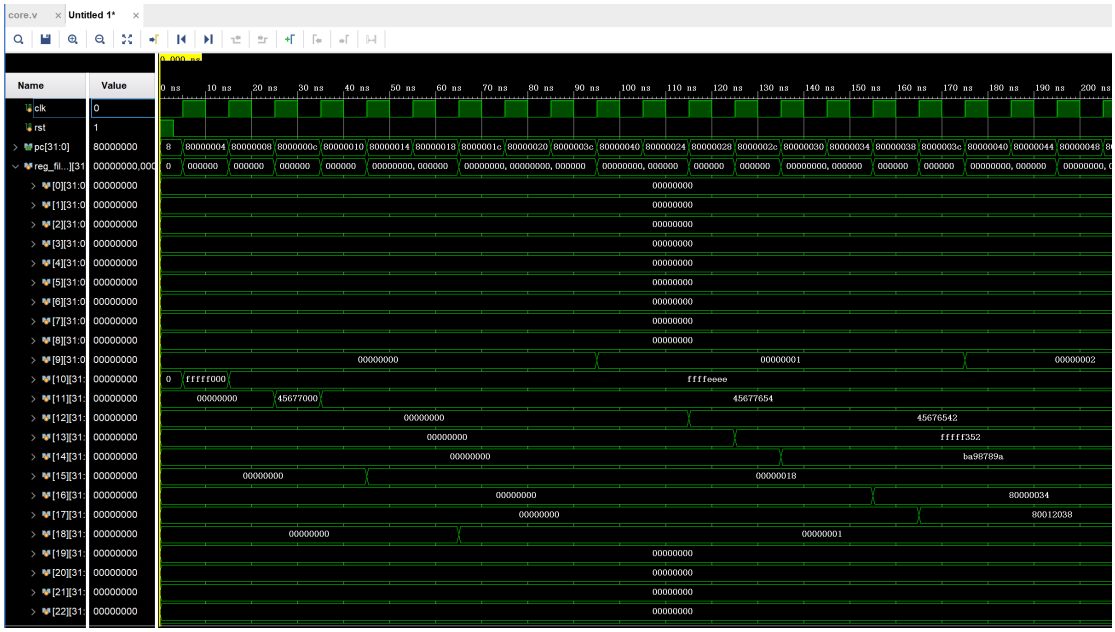


➔ *CPU 核心代码*

见附件 core.v。

➔ *CPU 仿真测试*

指令存储器加载 LabH3 step1.coe，仿真文件见附件 tb.v，仿真结果如下。



与 LabH3 step1.asm 单步调试结果相比，内容和时序都完全正确。

➔ ***配合外设 PDU 模块所作的设计***

为了配合外设，需要对 CPU 的数据存储器空间做划分：

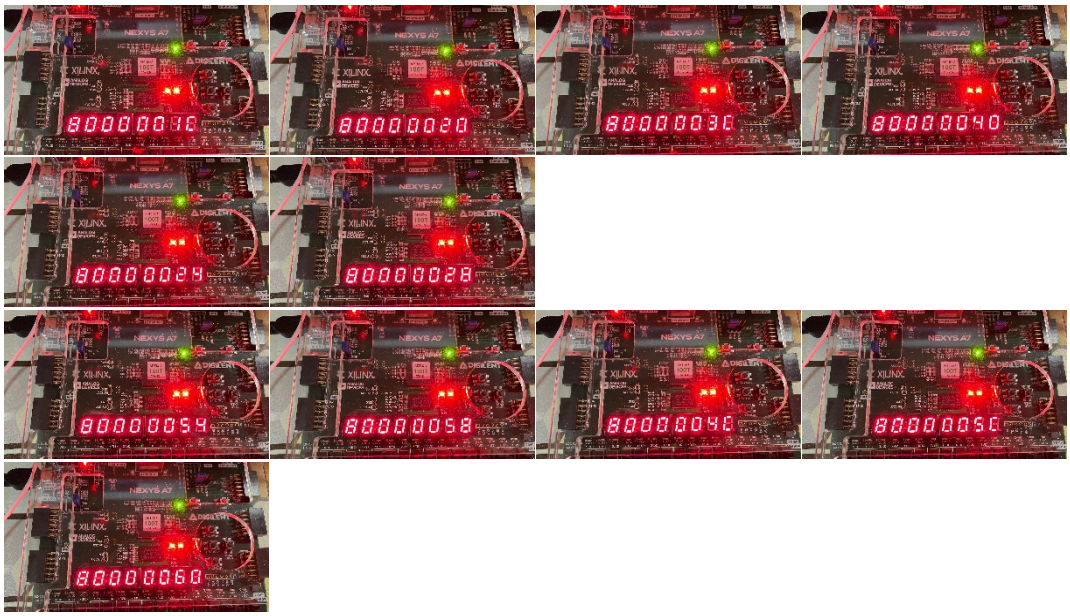
- 全空间：0x0000_0000 ~ 0x0000_0fff
  - 数据段：0x0000_0000 ~ 0x0000_03bf
  - MMIO 段：0x0000_03c0 ~ 0x0000_03ff
    - 0x0000_03c0 ~ 0x0000_03c3 led15-0
    - 0x0000_03c4 ~ 0x0000_03c7 btn, sw15-0
    - 0x0000_03c8 ~ 0x0000_03cb 数码管准备好
    - 0x0000_03cc ~ 0x0000_03cf 数码管输出数据
    - 0x0000_03d0 ~ 0x0000_03d3 开关输入有效
    - 0x0000_03d4 ~ 0x0000_03d7 开关输入数据
    - 0x0000_03d8 ~ 0x0000_03db 计数器数据

➔ ***CPU+PDU 上板测试***

顶层模块代码见 cpu.v。生成比特流后烧写到开发板上，根据仿真结果来比对结果。

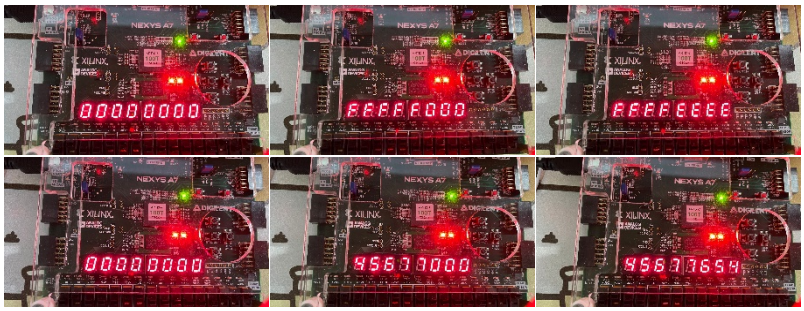
（1）PC 两次跳转：

上板情况（只看当前 PC，连续的照片之间是按了一次 step）：



（2）RF 寄存器内容变化：

上板情况：



实验 2: 排序程序测试和 CPU+PDU 下载

➔ *排序程序预仿真*

更改 COE 文件为排序程序的 COE 文件。

[77][31:0]	7ffff869
[76][31:0]	7ffff946
[75][31:0]	7fffa23
[74][31:0]	7ffffb00
[73][31:0]	7ffffbdd
[72][31:0]	7ffffcba
[71][31:0]	7ffffd97
[70][31:0]	7ffffe74
[69][31:0]	7fffff51
[68][31:0]	8000002e
[67][31:0]	8000010b
[66][31:0]	800001e8
[65][31:0]	800002c5
[64][31:0]	800003a2

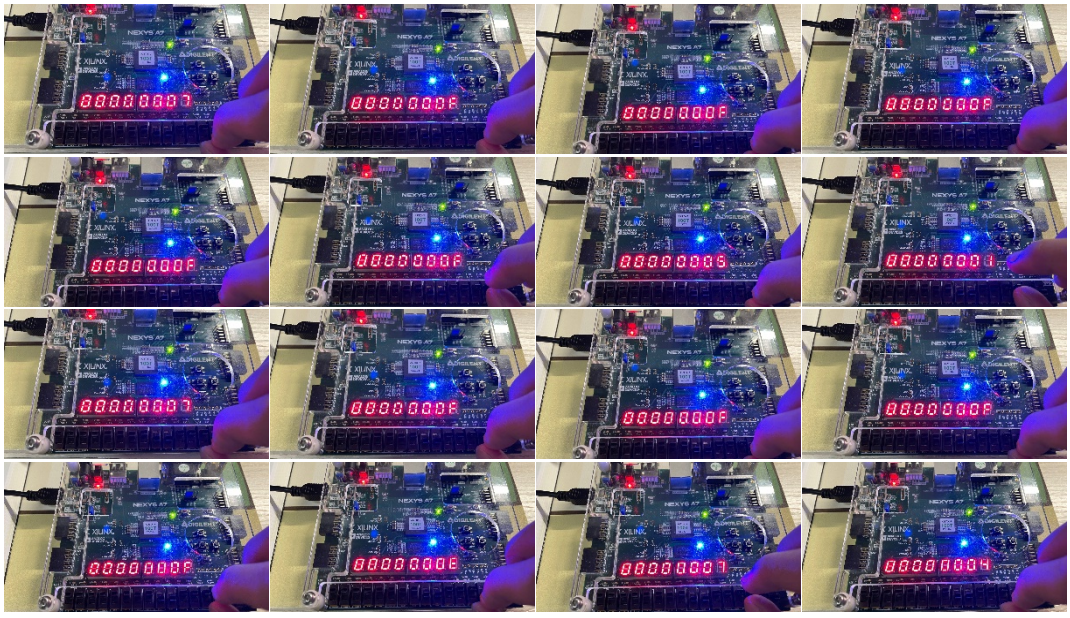
排序前

[77][31:0]	8000002e
[76][31:0]	8000010b
[75][31:0]	800001e8
[74][31:0]	800002c5
[73][31:0]	800003a2
[72][31:0]	7ffff869
[71][31:0]	7ffff946
[70][31:0]	7fffa23
[69][31:0]	7ffffb00
[68][31:0]	7ffffbdd
[67][31:0]	7ffffcba
[66][31:0]	7ffffd97
[65][31:0]	7ffffe74
[64][31:0]	7fffff51

排序后

➔ *排序程序下载测试*

生成比特流后烧写到开发板上，根据显示器（数码管）上内容判断正确与否。这里以线下检查内容为主，这里只展示前 16 个数字输出，应当是排序后的前两个数字：0x7fffff51、0x7ffffe74。



➔ *电路资源和性能报告*

Name	Slice LUTs (63400)	Slice Registers (126800)	F7 Muxes (31700)	Slice (15850)	LUT as Logic (63400)	LUT as Memory (19000)	Block RAM Tile (135)	Bonded IPADs (2)	BUFIO (24)
cpu	2089	504	260	694	1505	584	504	58	7
inst_core (core)	1727	147	256	532	1143	584	0	0	0
inst_pdu (pdu)	362	357	4	249	362	0	0	0	0

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.348 ns	Worst Hold Slack (WHS):	0.094 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	55	Total Number of Endpoints:	55	Total Number of Endpoints:	37

All user specified timing constraints are met.

实验选项: 扩展指令设计及其下载测试

- **测试 led[15:0]**

➔ *测试目的*

编写汇编程序，生成 COE 文件载入 CPU 指令存储器，用于测试的程序让 led：

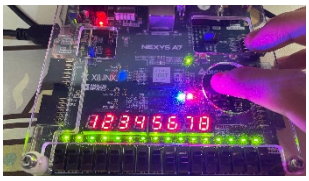
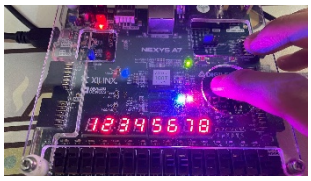
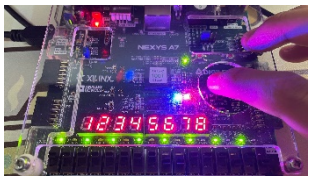
① 全亮

② 全灭

③ 奇数亮、偶数灭

➔ *上板测试*

由于 led 没有“准备好”的机制，这里采取单步调试的办法。

- **测试拓展指令**

前面设计的 CPU 已经实现了 RISC-V 基础指令集全部 37 条指令，前面的排序测试中，也用到了 and, andi, srai, slli, bge, blt, lb 等指令，结果正确，故这里不再对这些指令做测试。

步骤 1 要求的指令：add, addi, sub, auipc, lw, sw, beq, blt, jal, jalr；

步骤 2 使用到的拓展指令：and, andi, srai, slli, bge, lb；

还需要测试的指令： lui, bne, bltu, bgeu, lh, lbu, lhu, sb, sh, slti, sltiu, xori, ori, srli, sll, slt, sltu, xor, srl, sra, or。

汇编程序见附件 step_choose.asm，COE 文件见 step_choose.coe。

利用 RARS 测试，应得结果如下：

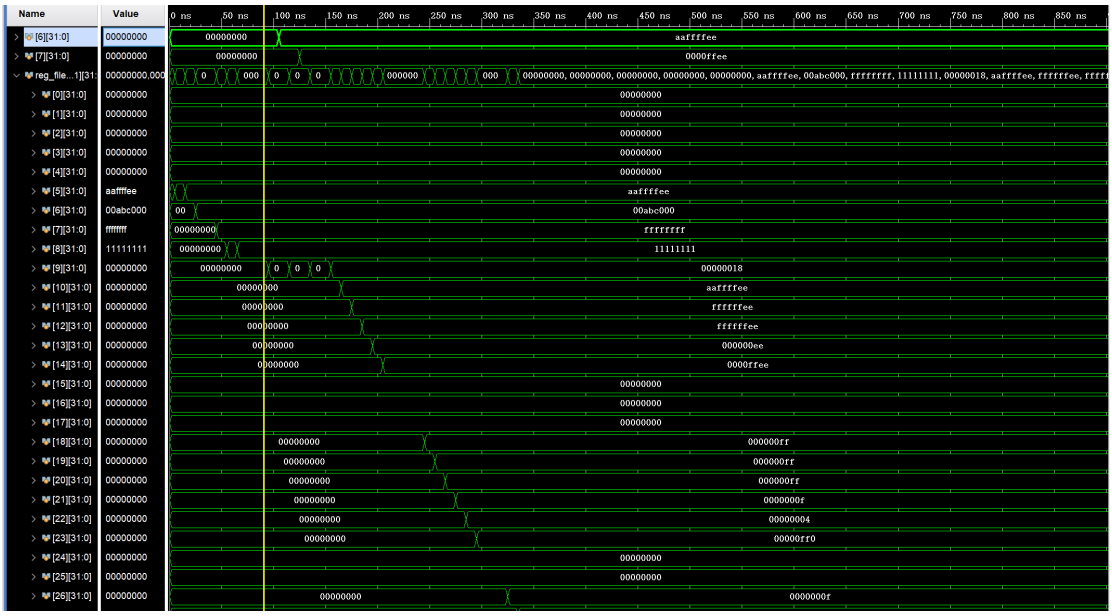
Registers | Floating Point | Control and Status

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x00002ffc
gp	3	0x00001800
tp	4	0x00000000
t0	5	0xaaffffee
t1	6	0x00abc000
t2	7	0xffffffff
s0	8	0x11111111
s1	9	0x00000018
a0	10	0xaaffffee
a1	11	0xffffffee
a2	12	0xffffffee
a3	13	0x000000ee
a4	14	0x0000ffee
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x000000ff
s3	19	0x000000ff
s4	20	0x000000ff
s5	21	0x0000000f
s6	22	0x00000004
s7	23	0x00000ff0
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x0000000f
s11	27	0x0000000f
t3	28	0x00000000

Data Segment

Address	Value (+0)	...	...	...	...	...	Value (+18)	Value (+1c)
0x00000000	0x00000000	. .	. .	. .	. .	. .	0xaaffffee	0x0000ffee
0x00000020	0x000000ee	. .	. .	. .	. .	. .	0x00000000	0x00000000

Vivado 仿真情况如下：



结果正确。

总结篇

- **收获**

“一阵微风吹过，树叶沙沙作响，仿佛是在唱着一首忧伤的歌曲，让人听着心中感到非常悲凉。一道清瘦的身影从远方慢慢向着这边走来，他穿着一身黑色长袍，脸上带着一副银质面具，手里拿着一根木棍，看起来就像是一个古代的侠客。一直来到近前，那名男子停下脚步，将手里的木棍扔在地上，随后伸出右手抓住左耳旁的耳环，轻轻的摇晃两下。只听“咔擦”声响起，面前的虚空突然打开了一条裂缝，一股奇异的香气从中散发而出，闻之令人神智清晰，整个人都变得舒服起来。”

- **建议**

（1） I/O 总线部分，给一些详细的文档，PPT 上的对做实验基本没有帮助。

（2） 建议老师提前做好 PDU，这样大家做实验就比较舒服，不用担心是否有 bug。