

《计算机组成与设计》

课程设计报告

课程设计题目：基于微指令的简单模型机

班级：21 计机学堂

姓名：曹智宁

学号：202122460179

完成时间：2023 年 5 月 16 日

目录

1 指令集	3
2 总体结构与数据通路框图	3
2.1 ALU.....	4
2.2 寄存器组.....	5
2.3 数据选择器.....	6
3 控制部件 CU.....	6
3.1 CU 设计	6
3.2 指令执行流程.....	7
3.3 微指令格式.....	9
3.4 RAM 程序	10
4 课程设计总结 CU	11
鸣谢	11

1 指令集

本模型机设计主要包含了存取值、加减乘除法、逻辑运算、数位运算、跳转、比较指令，为 8 位指令系统，前 4 位为操作码，后 4 位为具体控制，具体指令如下：

指令名称	8 位指令	后续指令	意义
取指	00000000		取下一条指令
取数至 R0	00010000	数据	指令后续 8 位数字存至 R0
取数至 R1	00100000	数据	指令后续 8 位数字存至 R1
保存 R0 至 RAM	01100000	地址	保存 R0 数据至 RAM 的后续指令给出的地址中
保存 R1 至 RAM	01000000	地址	保存 R1 数据至 RAM 的后续指令给出的地址中
加法	00110000		$R0+R1 \rightarrow R1$
减法	01010000		$R0-R1 \rightarrow R1$
乘法	10000000		$R0 \times R1 \rightarrow R1$
除法	10010000		$R0 \div R1 \rightarrow R1$
与运算	10100000		$R0 \& R1 \rightarrow R1$
或运算	10110000		$R0 R1 \rightarrow R1$
异或	11000000		$R0 \wedge R1 \rightarrow R1$
非运算	11010000		$\sim R1 \rightarrow R1$
判断小于	01110001	地址	如果 $R0 < R1$ 则 PC 跳转至后续地址
判断等于	01110010	地址	如果 $R0 = R1$ 则 PC 跳转至后续地址
判断大于	01110100	地址	如果 $R0 > R1$ 则 PC 跳转至后续地址
备用	11110000		

在本指令系统中，我们支持共 16 种指令，同时还预留了一种指令位置以备用。在此套指令系统内可以完成当前计算机计算时所需要指令的基本功能。

2 总体结构与数据通路框图

此模型机模型设计基于课件中给出的设计模型，其中蓝色代表数据流，绿色代表控制流，一切执行都有 μ PC 控制下的 ROM 来控制。除微程序部分其余部件均链接在总线中，接受 ALU 产生的结果。

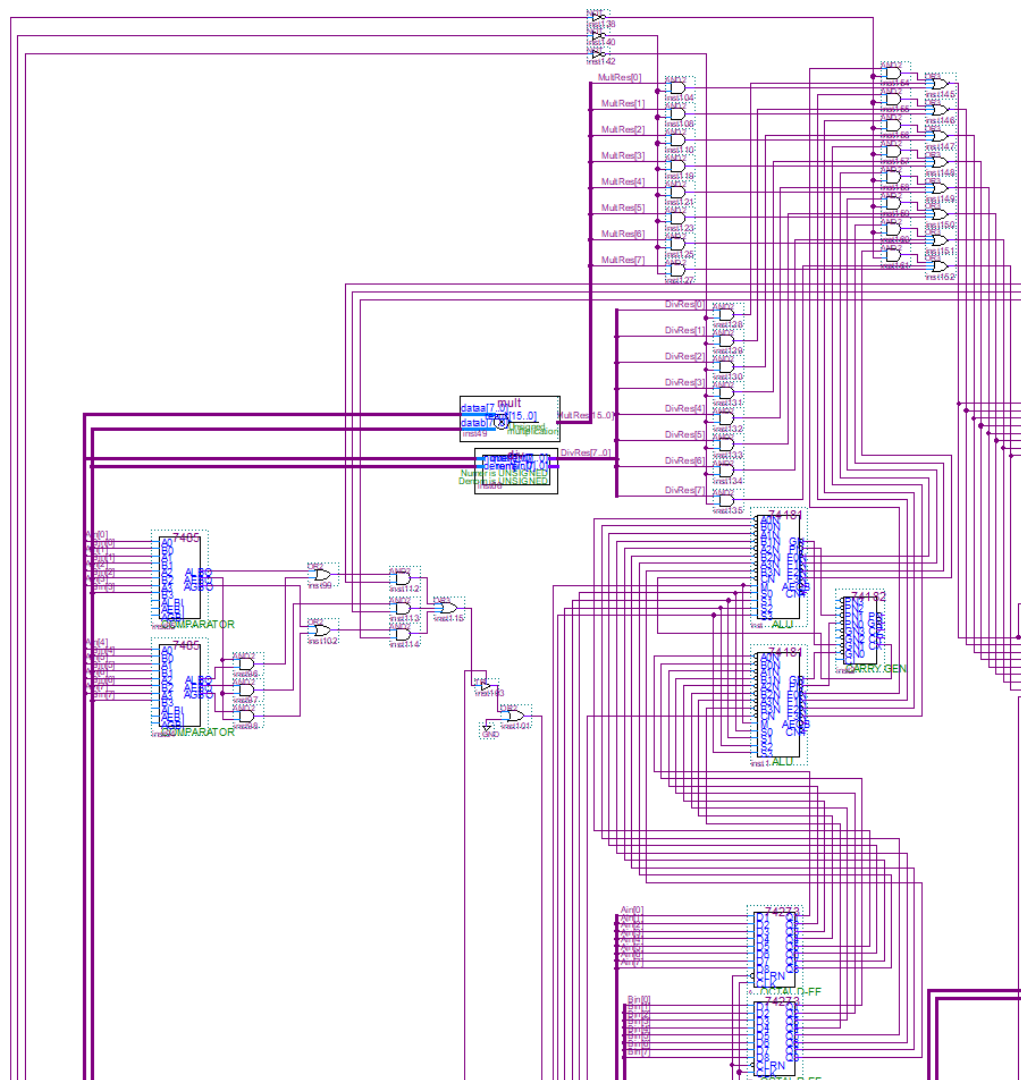
在前期实验一与实验二中，时序问题是问题主要所在，在整体框图中，我们也对时序进行了考虑，依据数据流动的次序分别使用系统时间的上升沿、下降沿、上升沿等等以此类推。

下图为模型机的总体原理图设计，部件设计相对位置与前图的相对位置相同。接下来，我将分别介绍除 CU 外部件的原理、功能和作用。

及控制信号，最后输出乘法或者除法结果。

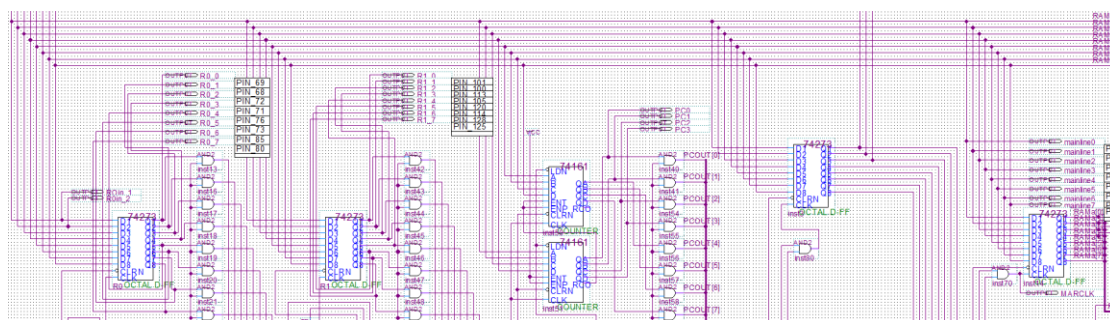
右侧包含了两个 71481 和一个 74182 组成的计算单元，支持加法、减法、逻辑运算和位运算，因为此部分包含多种计算功能，在本模型机中我们仅选择了加法、减法、异或、非运算、与运算和或运算。

三种运算器最后在 CU 的控制之下通过选择器最终只有一个结果流至总线。



2.2 寄存器组

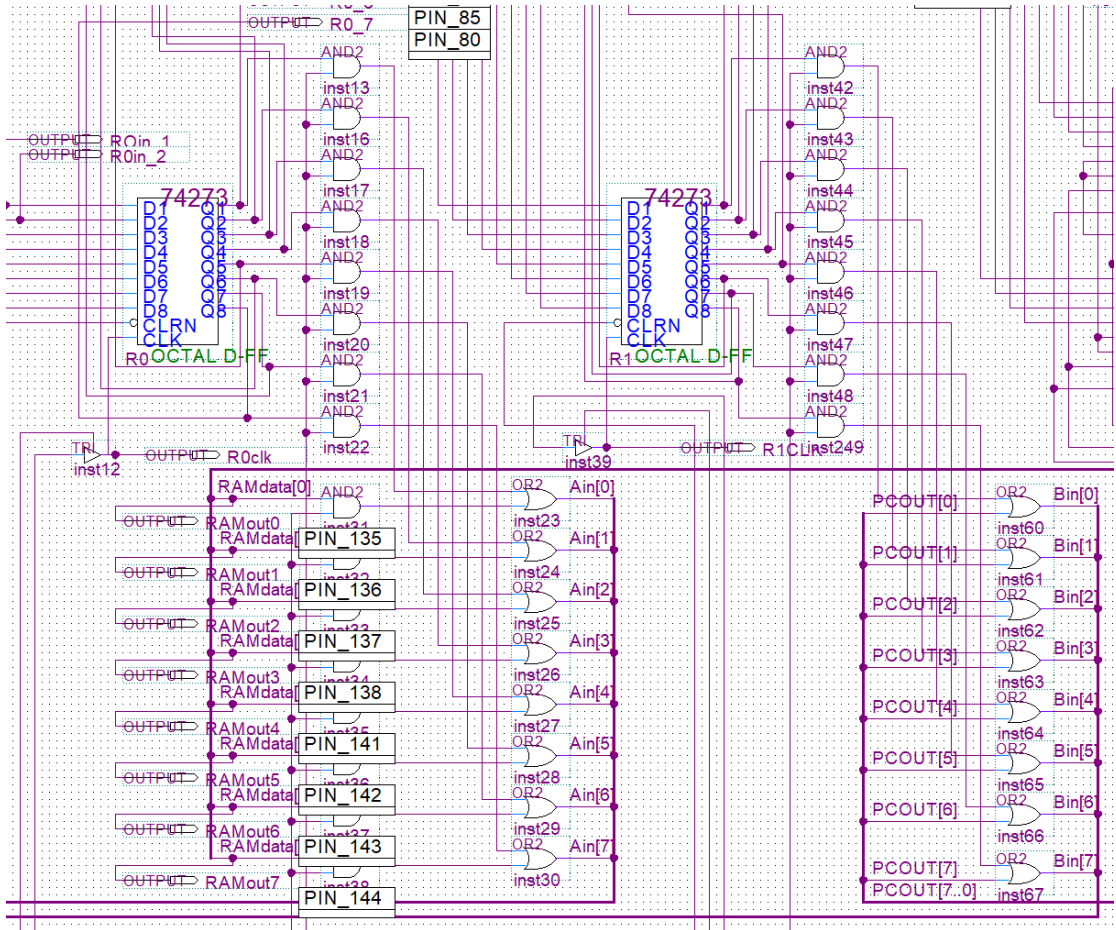
寄存器包含了 R0, R1, PC, IR 和 MAR 寄存器，他们都链接在总线上，可以在 CU 控制之下接受总线上的数据，以此来实现数据的更新，在下图中，五种寄存器依次排列。



除 PC 外，每个寄存器均采用了 74273 八位寄存器，PC 采用了两个 74161 自增一计数器，可通过简单信号实现快速 PC 自增一操作。

2.3 数据选择器

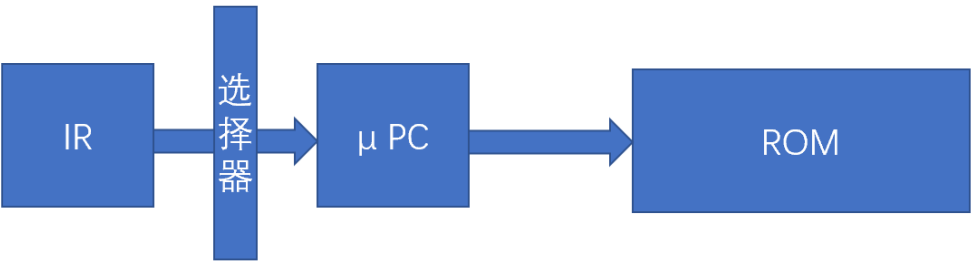
在寄存器 A 和 B 输入时，每个寄存器均包含了两种模数，寄存器 A 支持接受 R0 和 RAM 的数据，寄存器 B 支持接受 R1 和 PC 的数据，当时他们只能接受两种数据来源方式的一种，他们收到 PC 的控制，下图为数据选择器在模型机中的具体实现。



3 控制部件 CU

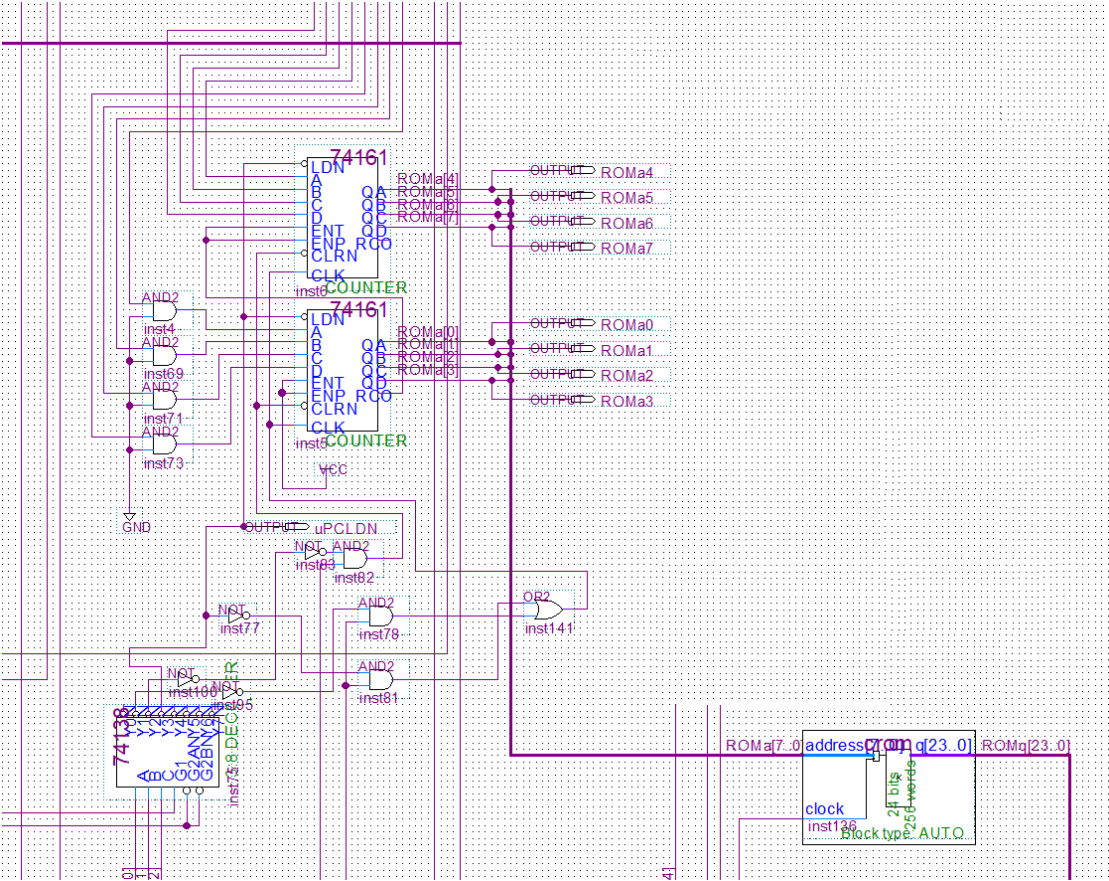
此部分讲介绍模型机的控制部件 CU，接下来将进一步展开介绍控制部件 CU。

3.1 CU 设计



CU 包含了 μ PC 和 ROM， μ PC 控制访问 ROM 的地址，ROM 存储了微指令， μ PC 在一定情况下会被初始化为指令入口地址， μ PC 接受了 IR 存储的指令经过选择器之后形成的地址。

下图为 PC 部分实际设计的原理图， μ PC 由两个 74161 自增一计数器构成， μ PC 连接着 ROM 的数据输入线，因此控制着 ROM。在上方引出的 8 位数据线为 IR 输出的地址，经过下方判断控制后传入 μ PC，在一定的信号下可以初始化 μ PC 的地址，从而达到进入不同微程序的功能。



3.2 指令执行流程

在此部分，将介绍指令的执行流程。

微操作	微地址	地址范围											十六进制代码		
		23, 22	21..16	15, 14	13, 12	11..9	8..7	6	5	4	3	2..0	CROM		
取指															
RAM→IR	00	00	111110	10	01	100	00	0	0	0	0	001	3E	98	01
PC+1→PC	01	00	000000	00	00	000	00	0	1	0	0	001	00	00	21
QJP	02	00	000000	00	00	000	00	0	0	0	0	011	00	00	03
存数 R0															
PC→MAR	10	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01

PC+1→PC	11	00	000000	00	00	000	00	0	1	0	0	001	00	00	21
RAM→R0	12	00	111110	10	00	001	00	0	0	0	0	001	3E	82	01
PC→MAR	13	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01
JP	14	00	000000	00	00	000	00	0	0	0	0	010	00	00	02
存数 R1															
PC→MAR	20	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01
PC+1→PC	21	00	000000	00	00	000	00	0	1	0	0	001	00	00	21
RAM→R1	22	00	111110	10	00	010	00	0	0	0	0	001	3E	84	01
PC→MAR	23	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01
JP	24	00	000000	00	00	000	00	0	0	0	0	010	00	00	02
加法															
R0+R1→R1	30	00	010011	01	10	010	00	0	0	0	0	001	13	64	01
PC→MAR	31	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01
JP	32	00	000000	00	00	000	00	0	0	0	0	010	00	00	02
减法															
R0-R1→R1	50	00	001100	01	10	010	00	0	0	0	0	001	13	64	01
PC→MAR	51	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01
JP	52	00	000000	00	00	000	00	0	0	0	0	010	00	00	02
乘法															
R0×R1-R1	80	01	000000	01	10	010	00	0	0	0	0	001	40	64	01
PC→MAR	81	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01
JP	82	00	000000	00	00	000	00	0	0	0	0	010	00	00	02
除法															
R0÷R1-R1	90	10	000000	01	10	010	00	0	0	0	0	001	80	64	01
PC→MAR	91	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01
JP	92	00	000000	00	00	000	00	0	0	0	0	010	00	00	02
与运算															
R0&R1→R1	A0	00		01	10	010	00	0	0	0	0	001		64	01
PC→MAR	A1	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01
JP	A2	00	000000	00	00	000	00	0	0	0	0	010	00	00	02
或运算															
R0 R1→R1	B0	00		01	10	010	00	0	0	0	0	001		64	01
PC→MAR	B1	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01
JP	B2	00	000000	00	00	000	00	0	0	0	0	010	00	00	02

异或															
$R0 \wedge R1 \rightarrow R1$	C0	00	101100	01	10	010	00	0	0	0	0	001	2C	64	01
$PC \rightarrow MAR$	C1	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01
JP	C2	00	000000	00	00	000	00	0	0	0	0	010	00	00	02
非运算															
$\sim R1 \rightarrow R1$	D0	00	101010	01	10	010	00	0	0	0	0	001	2A	64	01
$PC \rightarrow MAR$	D1	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01
JP	D2	00	000000	00	00	000	00	0	0	0	0	010	00	00	02
保存 R0															
$PC \rightarrow MAR$	60	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01
$PC+1 \rightarrow PC$	61	00	000000	00	00	000	00	0	1	0	0	001	00	00	21
$RAM \rightarrow MAR$	62	00	111110	10	00	101	00	0	0	0	0	001	3E	8A	01
$R0 \rightarrow RAM$	63	00	110100	01	00	000	00	0	0	1	0	001	34	40	11
$PC \rightarrow MAR$	64	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01
JP	65	00	000000	00	00	000	00	0	0	0	0	010	00	00	02
保存 R1															
$PC \rightarrow MAR$	40	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01
$PC+1 \rightarrow PC$	41	00	000000	00	00	000	00	0	1	0	0	001	00	00	21
$RAM \rightarrow MAR$	42	00	111110	10	00	101	00	0	0	0	0	001	3E	8A	01
$R1 \rightarrow RAM$	43	00	110100	00	10	000	00	0	0	1	0	001	34	20	11
$PC \rightarrow MAR$	44	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01
JP	45	00	000000	00	00	000	00	0	0	0	0	010	00	00	02
判断比较															
CMP	70	00	000000	01	10	000	00	0	0	0	0	001	00	60	01
$PC \rightarrow MAR$	71	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01
$MAR \rightarrow PC$	72	00	111110	10	00	011	00	1	0	0	0	001	3E	86	41
$MAR \rightarrow PC$	73	00	111110	10	00	011	00	1	0	0	0	001	3E	86	41
$PC \rightarrow MAR$	74	00	110100	00	01	101	00	0	0	0	0	001	34	1A	01
JP	75	00	000000	00	00	000	00	0	0	0	0	010	00	00	02

3.3 微指令格式

本模型机微指令包含 24 位，下面介绍每一位具体含义：

23-22 位：控制总线输出内容，00 为输出两片 74181 和一片 74182 构成的运算器内容，支持加减法、逻辑运算和位运算，01 则输出乘法器内容，10 则输出除法器内容。

21-16 位：控制两片 74181 和一片 74182 构成的运算器内容，分别对应 M，

S3, S2, S1, S0 和 CN.

15-14 位: 控制寄存器 A 输入内容, 10 为 RAM 数据, 01 为 R0 内容, 其余为全低电位。

13-12 位: 控制寄存器 B 输入内容, 01 为 PC 数据, 10 为 R1 内容, 其余为全低电位。

11-9 位: 控制链接在总线上的寄存器是否接受总线数据, 001 为 R0 接受, 010 为 R1 接受, 011 为 PC 接受, 100 为 IR 接受, 101 为 MAR 接受。

8-7 位: 空, 备用。

6 位: 比较器是否启用, 0 为不启用, 1 为启用。

5 位: PC+1→PC, 0 为不进行, 1 为进行。

4 位: WREN, 控制 RAM 读写。

3 位: 空, 备用。

2-0 位: 控制 μ PC, 001 为 μ PC+1, 010 为 μ PC 清零, 011 为 μ PC 接受 IR 输入进入微程序。

3.4 RAM 应用程序

汇编语言:

```
MOV R0, 4
MOV R1, 2
ADD R0, R1, R1
MOV (30), R1
MUL R0, R1, R1
MOV (32), R1
SUB R0, R1, R1
MOV (34), R1
DIV R0, R1, R1
MOD (36), R1
CMP R0, R1, <, 14
MOV (38), R0
AND R0, R1, R1
MOV (3A), R1
OR R0, R1, R1
MOV (3C), R1
XOR R0, R1, R1
MOV (3E), R1
NOT R1
MOV (40), R1
```

机器语言:

```
10 04
20 02
30
40 30
80
```

40 32
50
40 34
90
40 36
71 14
60 38
A0
40 3A
B0
40 3C
C0
40 3E
D0
40 40

4 课程设计总结

在本次课程学习中，我们先是通过前一学期部分器件的学习掌握了部分器件的设计方法，又通过上学期理论课程的学习掌握了计算机组成原理，在本学期的课程中，通过实验一、实验二的导入，对于实验三完成完整模型机逐渐有了设计的方法，最后也成功设计出了自我设计指令集的模型机。

在本学期的课程中，我遇到最大的困难莫过于时序的适配与调整，在张老师的帮助下，我逐步修改了我部件在数据流下的时序设计，通过使用三态门和不同器件的不同时序，最终成功解决了时序的问题。

总体而言，此门课程实验设计让我对于 CPU 设计和计算机组成原理有了更加深入的理解和学习，也希望以后能够借助此门课程带给我的知识进一步学习，不断丰富自我。

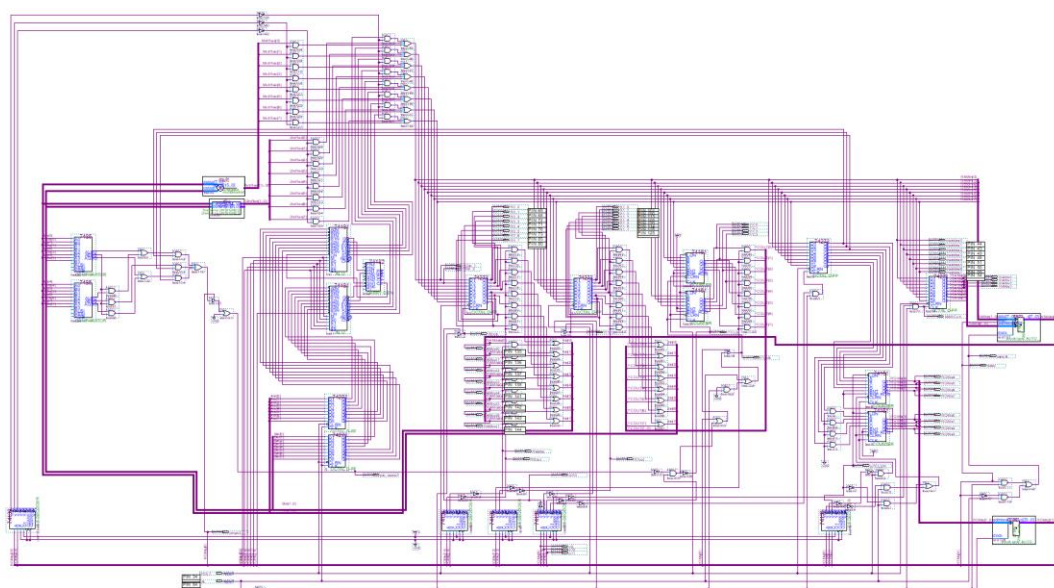
鸣谢

感谢张老师在此门课程的讲解和对我的帮助。

感谢同学们在实验课中对于我的帮助。

附件

附件一



附件二

见第 3 章

附件三

见 3.2 节

附件四

见 3.3 节