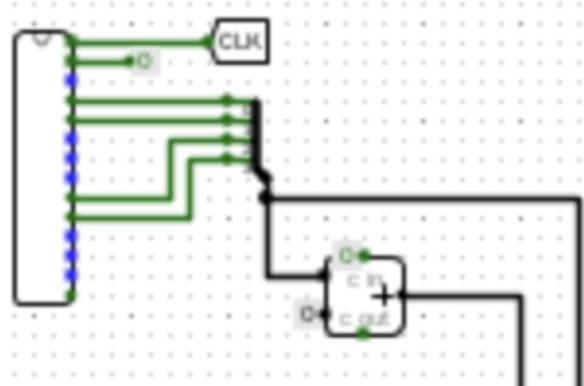
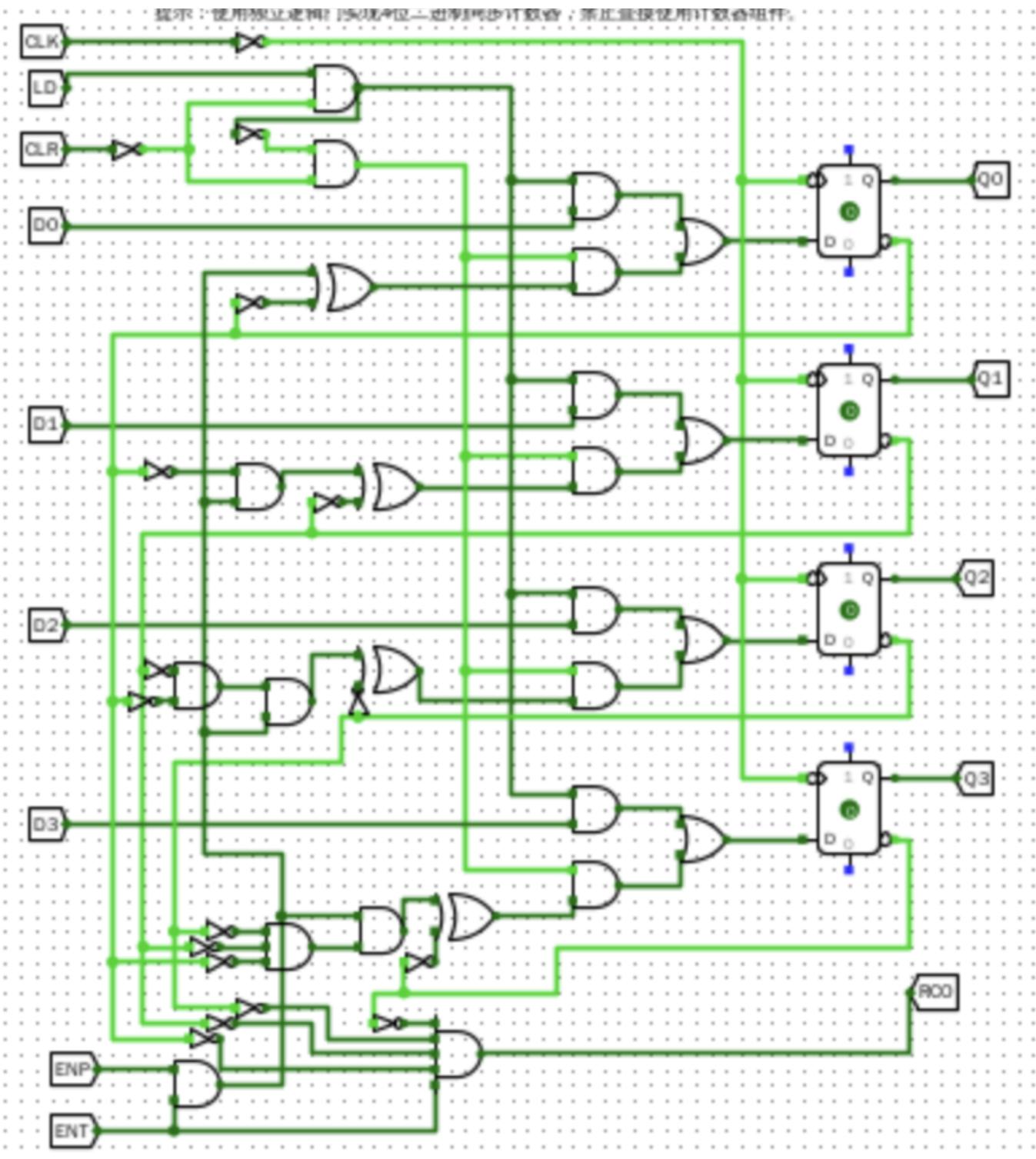


实验报告

思考题

如何利用 ROM 实验实现滚动显示的功能，在 3 个 LED 点阵矩阵中，左右滚动显示 5 个 ASCII 字符，如“NJUCS”。

将code与一个时序逻辑电路相连，在对应的时钟周期输出对应的code信号。得到一个滚动显示的LED矩阵。实现如下：



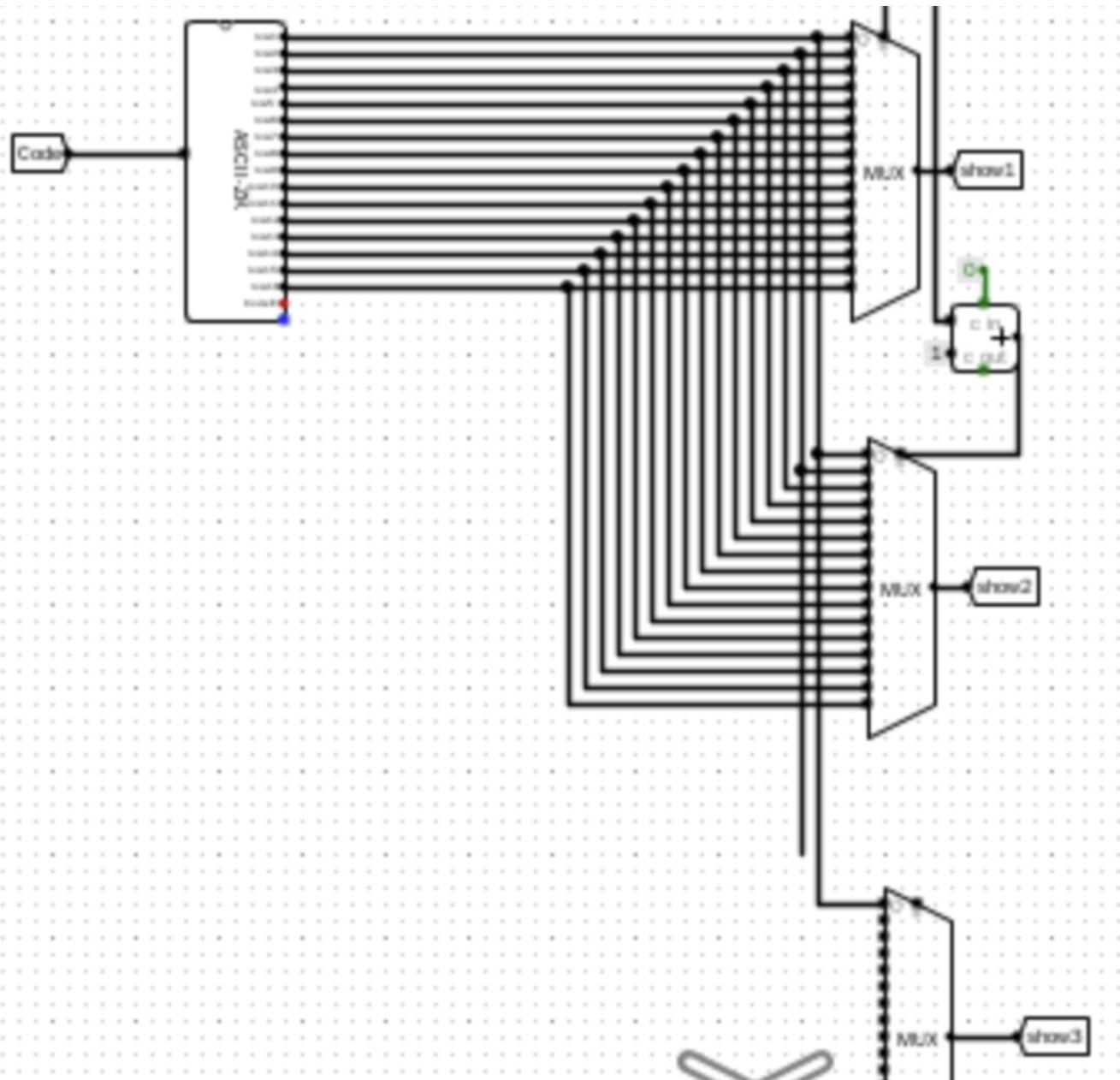


图 思考题实现

2. 分析说明如果 PC 寄存器、寄存器堆和数据存储器写入数据的时钟信号触发边沿不一致，则对程序执行结果有什么影响？

如果一条指令包括读和写，可能导致运算结果没有正确生成的时候，寄存器堆和存储器的写入使能端就被激活了，导致写入错误信号，进而读取的信号发生错误。同时，PC寄存器可能跳转到错误的地址，产生程序流错误。

3. 在 CPU 启动执行后，如何实现停止执行功能

可以设计专门的停止指令，如果执行该指令则将PC的写使能端关闭，PC不会更新，cpu不会继续运行

3. 利用 4 位移位寄存器设计 8 位二进制伪随机序列电路，写出输出序列值。

基本信息

实验名称 存储器及数据通路设计 实验人：241220071葛家韬

实验目的

1. 掌握利用 ROM 组件实现指令存储器的设计方法。
2. 掌握 ASCII 点阵字库的使用方法。
3. 掌握利用 RAM 组件实现按字节存取的数据存储器的设计方法。
4. 掌握实现 RV32I 存取指令的设计方法。
5. 掌握 RV32I 下地址逻辑部件的设计方法
6. 掌握 RV32I 指令译码、反汇编的实现方法
7. 掌握 RV32I 立即数扩展器的设计方式
8. 根据 RV32I 指令执行过程，掌握 RV32I 运算类指令数据通路的设计方法
9. 掌握跳转控制器的设计方法

实验环境

Logisim 2.16

实验内容

1. 只读存储器实验

(1) 实验整体方案设计

要求：说明本次实验的顶层设计模块图，对每个子模块进行详细描述，定义输入输出引脚，数据及控制信号的传输通道等。

利用 Logisim 的 ROM 组件加载 ASCII 点阵字库文件 ascii8-16.zk，通过输入 ASCII 码值计算字符点阵起始地址（ASCII 码值减 0x20 后左移 4 位），使用 16 个 ROM 并行读取 16 行点阵数据，同步输出至 8×16 的 LED 点阵显示字符。若输入码值超出可见范围（0x20-0x7F），则输出错误标志 CodeErr。



图 实验电路封装图

(2) 实验原理图和电路图

要求：给出每个子模块的原理图和 Logisim 中的电路图，定义子模块的外观图。如果对实验指导讲义中的内容提出优化或改进，需要此说明原因、方法和效果。实验电路无模块化设计，不需要原理图。

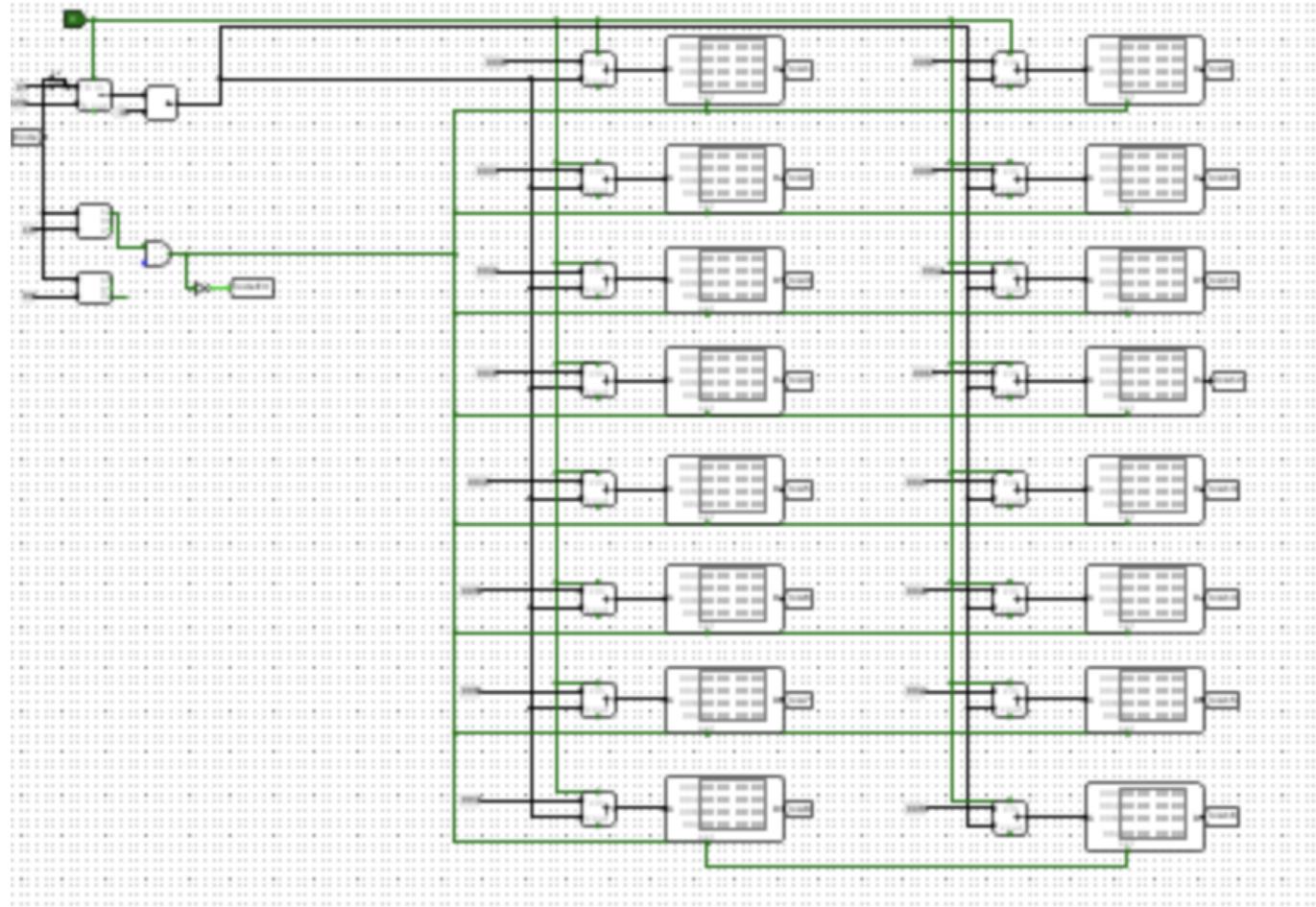


图 电路图

(3) 实验数据仿真测试图

要求：根据实验要求，输入测试数据，选择单步时钟执行，截取仿真运行时的电路图，分析电路状态是否满足设计需求。说明子模块的功能，列出子模块的功能表。

预期输出													实际输出													展示原始输出			
Cnt	ASCII	row1	row2	row3	row4	row5	row6	row7	row8	row9	row10	row11	row12	ro	Cnt	ASCII	row1	row2	row3	row4	row5	row6	row7	row8	row9	row10	row11	row12	ro
00	00	xx	xx	xx	xx	00	00	xx	xx	xx																			
01	01	xx	xx	xx	xx	01	01	xx	xx	xx																			
02	02	xx	xx	xx	xx	02	02	xx	xx	xx																			
03	03	xx	xx	xx	xx	03	03	xx	xx	xx																			
04	04	xx	xx	xx	xx	04	04	xx	xx	xx																			
05	05	xx	xx	xx	xx	05	05	xx	xx	xx																			
06	06	xx	xx	xx	xx	06	06	xx	xx	xx																			
07	07	xx	xx	xx	xx	07	07	xx	xx	xx																			
08	08	xx	xx	xx	xx	08	08	xx	xx	xx																			
09	09	xx	xx	xx	xx	09	09	xx	xx	xx																			
0a	0a	xx	xx	xx	xx	0a	0a	xx	xx	xx																			
0b	0b	xx	xx	xx	xx	0b	0b	xx	xx	xx																			
0c	0c	xx	xx	xx	xx	0c	0c	xx	xx	xx																			
0d	0d	xx	xx	xx	xx	0d	0d	xx	xx	xx																			
0e	0e	xx	xx	xx	xx	0e	0e	xx	xx	xx																			
0f	0f	xx	xx	xx	xx	0f	0f	xx	xx	xx																			
10	10	xx	xx	xx	xx	10	10	xx	xx	xx																			
11	11	xx	xx	xx	xx	11	11	xx	xx	xx																			
12	12	xx	xx	xx	xx	12	12	xx	xx	xx																			
13	13	xx	xx	xx	xx	13	13	xx	xx	xx																			
14	14	xx	xx	xx	xx	14	14	xx	xx	xx																			
15	15	xx	xx	xx	xx	15	15	xx	xx	xx																			
16	16	xx	xx	xx	xx	16	16	xx	xx	xx																			
17	17	xx	xx	xx	xx	17	17	xx	xx	xx																			
18	18	xx	xx	xx	xx	18	18	xx	xx	xx																			
19	19	xx	xx	xx	xx	19	19	xx	xx	xx																			
1a	1a	xx	xx	xx	xx	1a	1a	xx	xx	xx																			

图 仿真测试图

(4) 错误现象及分析

要求：在电路设计、连接和仿真运行时，遇到的任何错误，都需要截屏放置到实验报告中，并分析错误原因和解决办法。

若 LED 点阵无显示，可能是 ROM 地址计算错误（如未正确减去 0x20 或左移位数不足），或片选信号未设置为高电平有效；若显示乱码，可能是字库文件加载错误或 ROM 数据位宽未设为 8 位。

2.数据存储器实验

(1) 实验整体方案设计

要求：说明本次实验的顶层设计模块图，对每个子模块进行详细描述，定义输入输出引脚，数据及控制信号的传输通道等。

使用 4 片 8 位 RAM (RAM3-RAM0) 级联实现 32 位数据存储，通过 MemOp 信号和地址低两位 (Addr [1:0]) 控制片选信号 SEL3-SEL0，支持字节、半字、字的存取操作。读取时根据 MemOp 类型进行 0 扩展或符号扩展，地址未对齐或 MemOp 无效时输出 AlignErr 错误。

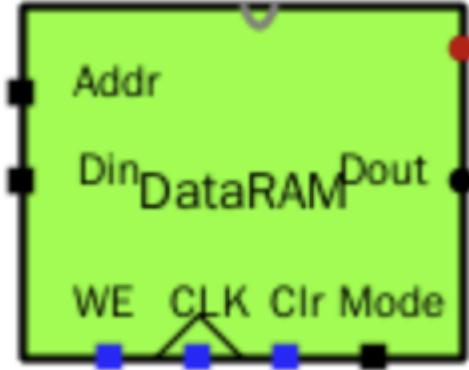


图 实验电路封装图

(2) 实验原理图和电路图

要求：给出每个子模块的原理图和 Logisim 中的电路图，定义子模块的外观图。如果对实验指导讲义中的内容提出优化或改进，需要此说明原因、方法和效果。

MemOp	指令	含义
000	lw,sw	按字存取，4 字节
001	lbu	按字节读取，1 个字节，0 扩展到 4 字节
010	lhu	按半字读取，2 字节，0 扩展到 4 字节
101	lb,sb	按字节存取，1 个字节，在读取时，按符号位扩展到 4 字节
110	lh,sh	按半字存取，2 个字节，在读取时，按符号位扩展到 4 字节

图 实验原理

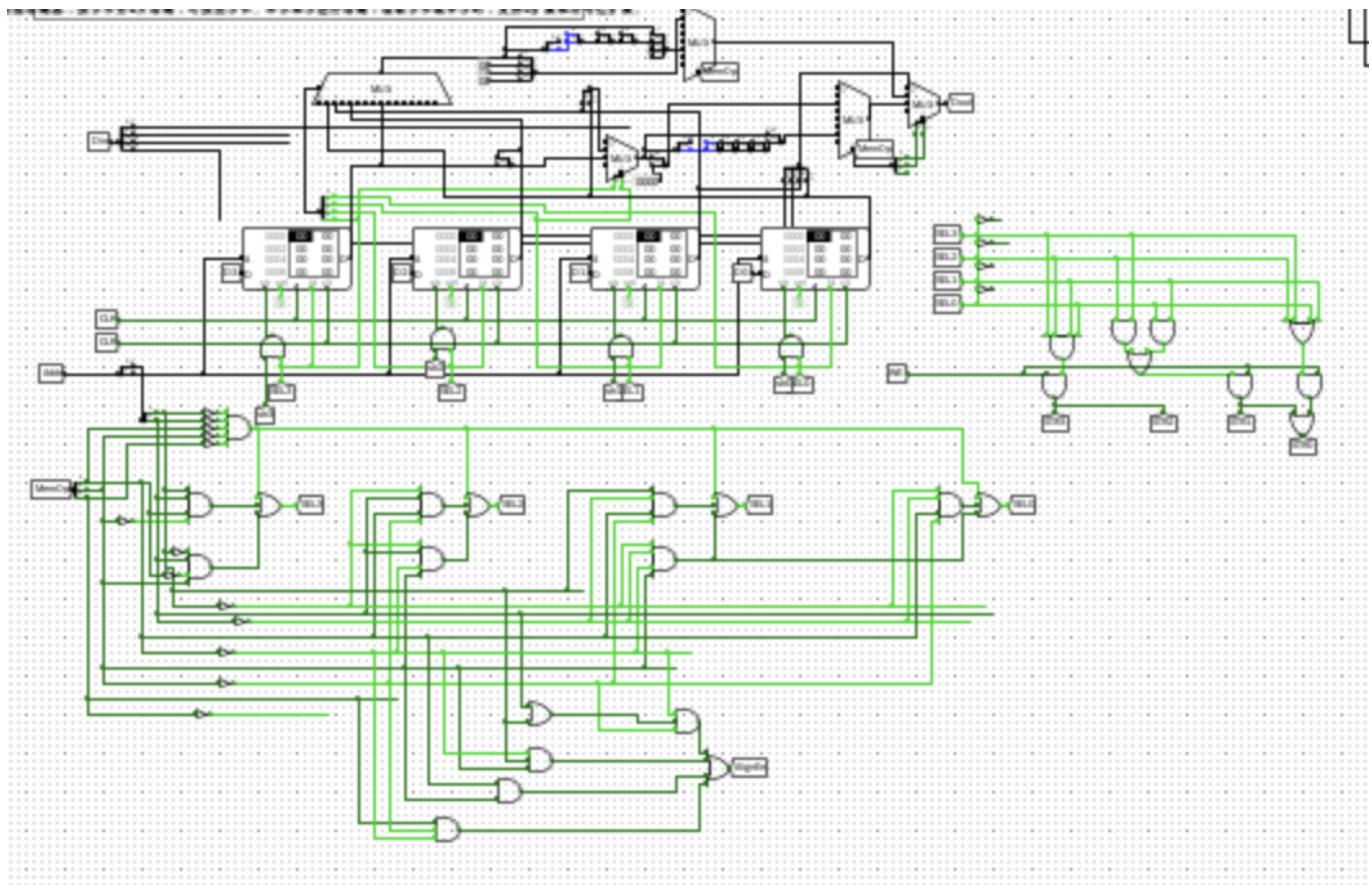


图 电路图

(3) 实验数据仿真测试图

要求：根据实验要求，输入测试数据，选择单步时钟执行，截取仿真运行时的电路图，分析电路状态是否满足设计需求。说明子模块的功能，列出子模块的功能表。

— 预期输出 —								— 实际输出 —								展示原始输出
Cnt	Addr	WE	CLR	MemOp	Din	Dout	AlignE	Cnt	Addr	WE	CLR	MemOp	Din	Dout	AlignE	
00	00000	1	0	0	00000001	00000000	0	00	00000	1	0	0	00000001	00000000	0	
01	00000	0	0	0	005dece	00000001	0	01	00000	0	0	0	005dece	00000001	0	
02	00010	1	0	5	b61488df	00000000	0	02	00010	1	0	5	b61488df	00000000	0	
03	00011	1	0	5	f4111591	00000000	0	03	00011	1	0	5	f4111591	00000000	0	
04	00012	1	0	5	023eaf12	00000000	0	04	00012	1	0	5	023eaf12	00000000	0	
05	00013	1	0	5	b578f6a	00000000	0	05	00013	1	0	5	b578f6a	00000000	0	
06	00010	0	0	0	d38a8b1c	6a1291df	0	06	00010	0	0	0	d38a8b1c	6a1291df	0	
07	00100	1	0	6	f5d50649	00000000	0	07	00100	1	0	6	f5d50649	00000000	0	
08	00102	1	0	6	202e3c08	00000000	0	08	00102	1	0	6	202e3c08	00000000	0	
09	00100	0	0	0	812fba12	3c088649	0	09	00100	0	0	0	812fba12	3c088649	0	
0a	00100	0	0	1	6755ebab	00000049	0	0a	00100	0	0	1	6755ebab	00000049	0	
0b	00101	0	0	1	f6e7cab1	00000006	0	0b	00101	0	0	1	f6e7cab1	00000006	0	
0c	00102	0	0	1	e6bc21b9	00000008	0	0c	00102	0	0	1	e6bc21b9	00000008	0	
0d	00103	0	0	1	a6c5abc6	0000003c	0	0d	00103	0	0	1	a6c5abc6	0000003c	0	
0e	00100	0	0	2	58228a26	00000649	0	0e	00100	0	0	2	58228a26	00000649	0	
0f	00102	0	0	2	9dfc1362	00000308	0	0f	00102	0	0	2	9dfc1362	00000308	0	
10	01000	1	0	0	ac5b2754	00000000	0	10	01000	1	0	0	ac5b2754	00000000	0	
11	01000	0	0	5	2f43454c	00000054	0	11	01000	0	0	5	2f43454c	00000054	0	
12	01001	0	0	5	18c3dc9c	00000027	0	12	01001	0	0	5	18c3dc9c	00000027	0	
13	01002	0	0	5	1abb8d8c	0000005b	0	13	01002	0	0	5	1abb8d8c	0000005b	0	
14	01003	0	0	5	c1b1b519a	fffffffac	0	14	01003	0	0	5	c1b1b519a	fffffffac	0	
15	01000	0	0	3	0f672c6a	xxxxxxxx	1 Error!	15	01000	0	0	3	0f672c6a	xxxxxxxx	1 Error!	
16	01000	0	0	4	e37362d1	xxxxxxxx	1 Error!	16	01000	0	0	4	e37362d1	xxxxxxxx	1 Error!	
17	01000	0	0	7	39eae841	xxxxxxxx	1 Error!	17	01000	0	0	7	39eae841	xxxxxxxx	1 Error!	
18	01001	0	0	6	95c7a81f	xxxxxxxx	1 Error!	18	01001	0	0	6	95c7a81f	xxxxxxxx	1 Error!	
19	01003	0	0	6	9b8ab32e	xxxxxxxx	1 Error!	19	01003	0	0	6	9b8ab32e	xxxxxxxx	1 Error!	
1b	01002	0	0	0	72f86d97	xxxxxxxx	1 Error!	1b	01002	0	0	0	72f86d97	xxxxxxxx	1 Error!	
1c	01003	0	0	0	52130e8e	xxxxxxxx	1 Error!	1c	01003	0	0	0	52130e8e	xxxxxxxx	1 Error!	
1d	01000	0	0	0	1ee9d827	ac5b2754	0	1d	01000	0	0	0	1ee9d827	ac5b2754	0	
1e	01000	0	1	0	9056d989	00000000	0	1e	01000	0	1	0	9056d989	00000000	0	
1f	01000	0	0	0	dc896b07	00000000	0	1f	01000	0	0	0	dc896b07	00000000	0	
20	1f000	1	0	0	913171e3	00000000	0	20	1f000	1	0	0	913171e3	00000000	0	
21	1f000	0	0	0	6a830aaa	913171e3	0	21	1f000	0	0	0	6a830aaa	913171e3	0	
22	1f010	1	0	5	ebccdf55	00000000	0	22	1f010	1	0	5	ebccdf55	00000000	0	
23	1f011	1	0	5	1bead7a1	00000000	0	23	1f011	1	0	5	1bead7a1	00000000	0	
24	1f012	1	0	5	70ce89af	00000000	0	24	1f012	1	0	5	70ce89af	00000000	0	
25	1f013	1	0	5	d6c6f30f	00000000	0	25	1f013	1	0	5	d6c6f30f	00000000	0	
26	1f010	0	0	0	4f2f15f6	0faf155	0	26	1f010	0	0	0	4f2f15f6	0faf155	0	
27	1f100	1	0	6	0bfd3217	00000000	0	27	1f100	1	0	6	0bfd3217	00000000	0	
28	1f102	1	0	6	74276d58	00000000	0	28	1f102	1	0	6	74276d58	00000000	0	
29	1f100	0	0	0	c6947ef2	6d583217	0	29	1f100	0	0	0	c6947ef2	6d583217	0	
2a	1f100	0	0	5	e86e0a71	00000017	0	2a	1f100	0	0	5	e86e0a71	00000017	0	
2b	1f101	0	0	5	e71009ff	00000032	0	2b	1f101	0	0	5	e71009ff	00000032	0	
2c	1f102	0	0	5	d4203d58	00000058	0	2c	1f102	0	0	5	d4203d58	00000058	0	
2d	1f103	0	0	5	bdb817173	0000006d	0	2d	1f103	0	0	5	bdb817173	0000006d	0	
2e	1f100	0	0	6	64d7cb03	00003217	0	2e	1f100	0	0	6	64d7cb03	00003217	0	
2f	1f102	0	0	6	5847aa13	00006d58	0	2f	1f102	0	0	6	5847aa13	00006d58	0	
30	3ffffc	1	0	6	b183207a	00000000	0	30	3ffffc	1	0	6	b183207a	00000000	0	
31	3ffffe	1	0	6	5333b778	00000000	0	31	3ffffe	1	0	6	5333b778	00000000	0	
32	3ffffc	0	0	0	fad2465a	b778207a	0	32	3ffffc	0	0	0	fad2465a	b778207a	0	
33	3ffffc	0	0	1	608b2671	0000007a	0	33	3ffffc	0	0	1	608b2671	0000007a	0	
34	3ffffc	0	0	5	1229d521	0000007a	0	34	3ffffc	0	0	5	1229d521	0000007a	0	
35	3ffffd	0	0	1	bd93bfe7	00000020	0	35	3ffffd	0	0	1	bd93bfe7	00000020	0	
36	3ffffd	0	0	5	57945ed9	00000020	0	36	3ffffd	0	0	5	57945ed9	00000020	0	
37	3ffffe	0	0	1	149d0edc	00000078	0	37	3ffffe	0	0	1	149d0edc	00000078	0	
38	3ffffe	0	0	5	d4a71682	00000078	0	38	3ffffe	0	0	5	d4a71682	00000078	0	
39	3fffff	0	0	1	97e417ee	000000b7	0	39	3fffff	0	0	1	97e417ee	000000b7	0	
3a	3fffff	0	0	5	2594f1d9	fffffffffb7	0	3a	3fffff	0	0	5	2594f1d9	fffffffffb7	0	

图 仿真测试图

(4) 错误现象及分析

要求：在电路设计、连接和仿真运行时，遇到的任何错误，都需要截屏放置到实验报告中，并分析错误原因和解决办法。

写入数据后读取异常，可能是片选信号逻辑错误（如 SEL 与 Addr [1:0] 不匹配）或写使能信号 str 未在时钟上升沿有效；扩展结果错误可能是 MemOp 编码与扩展逻辑不匹配，如 lbu 未正确执行零扩展。

3. 取指令部件实验

(1) 实验整体方案设计

要求：说明本次实验的顶层设计模块图，对每个子模块进行详细描述，定义输入输出引脚，数据及控制信号的传输通道等。

通过程序计数器 PC 加载初始地址，利用加法器和多路选择器计算下一条指令地址（NPC），支持顺序执行（PC+4）、无条件跳转（jal/jalr）和条件分支（beq/bne 等）。PC 寄存器上升沿触发，指令存储器地址需舍弃 PC 最低两位以匹配 Logisim 按字编址规则。

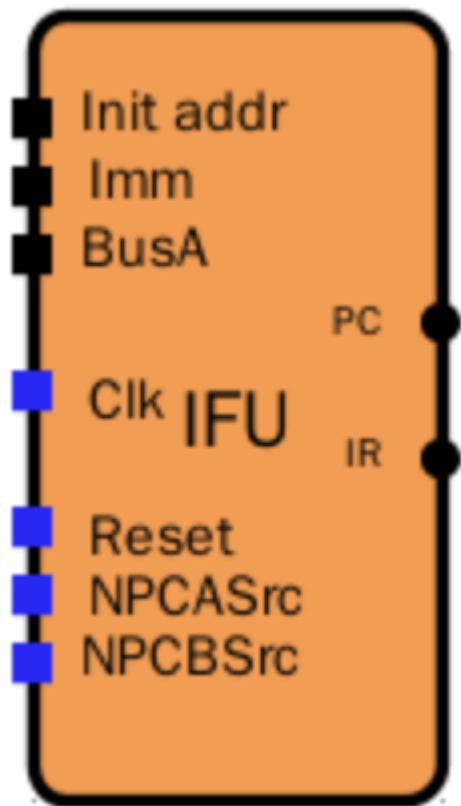


图 实验电路封装图

(2) 实验原理图和电路图

要求：给出每个子模块的原理图和 Logisim 中的电路图，定义子模块的外观图。如果对实验指导讲义中的内容提出优化或改进，需要此说明原因、方法和效果。

指令类型	NPCASrc	NPCBSrc	下条指令地址
顺序指令	0	0	$PC=PC+4$
无条件跳转 jal	0	1	$PC=PC+Imm$
无条件跳转 jalr	1	1	$PC=R[rs1]+Imm$
分支跳转指令	0	1	条件成立 $PC=PC+Imm$, 否则 $PC=PC+4$

图 实验原理

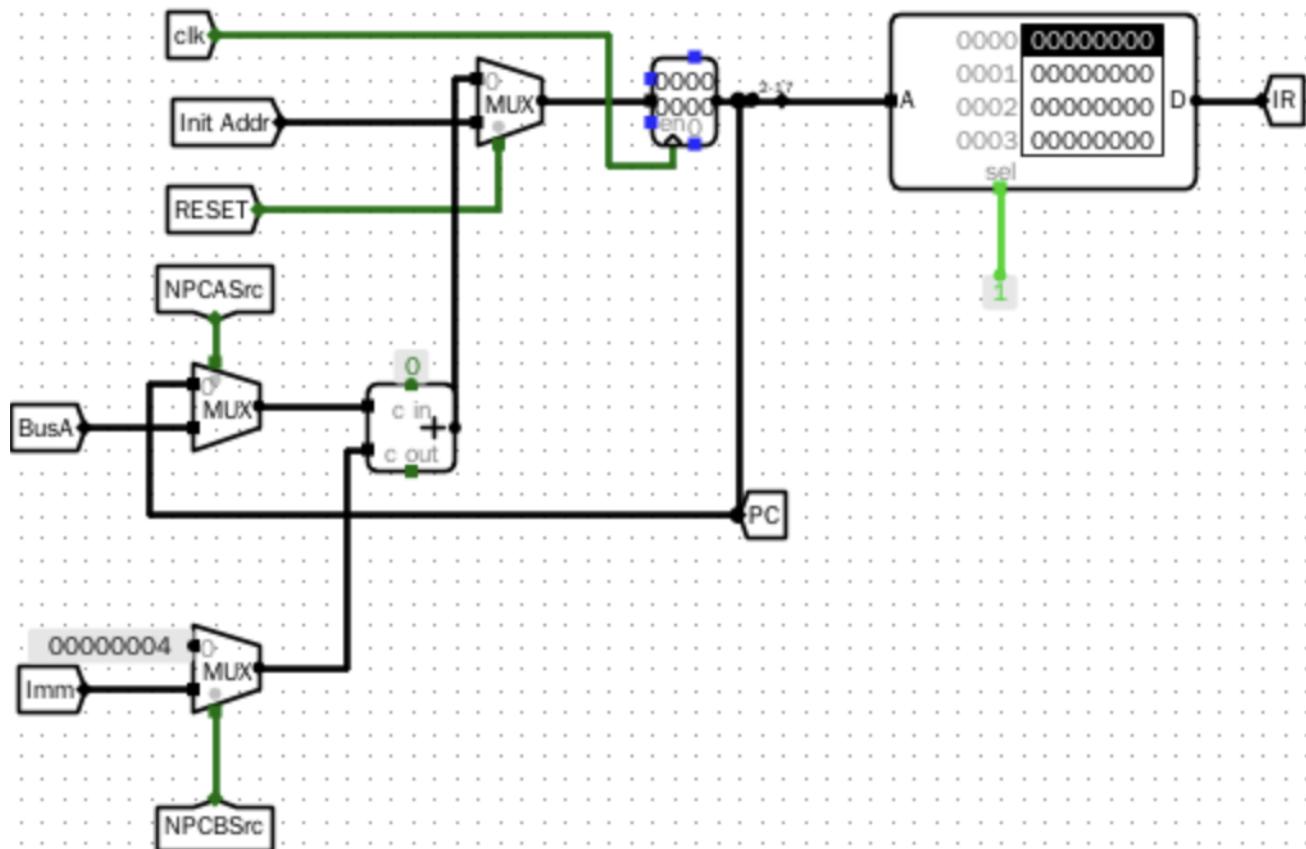


图3 电路图

(3) 实验数据仿真测试图

要求：根据实验要求，输入测试数据，选择单步时钟执行，截取仿真运行时的电路图，分析电路状态是否满足设计需求。说明子模块的功能，列出子模块的功能表。

Cnt	InitAddr	Reset	NPCASr	NPCBSr	Imm	BusA	PC	IR	Cnt	InitAddr	Reset	NPCASr	NPCBSr	Imm	BusA	PC	IR	
0	00000400	1	0	0	00000000	00000000	00000000	00000000	0	00000400	1	0	0	00000000	00000000	00000000	00000000	00000000
1	00000000	0	0	0	00000000	00000000	00000400	00002083	1	00000000	0	0	0	00000000	00000000	00000400	00002083	
2	00000000	0	0	0	00000000	00000000	00000404	00008133	2	00000000	0	0	0	00000000	00000000	00000404	00008133	
3	00000000	0	0	1	000007f8	00000000	00000408	00106213	3	00000000	0	0	1	000007f8	00000000	00000408	00106213	
4	00000000	0	0	1	0000020c	00000000	00000c00	00000413	4	00000000	0	0	1	0000020c	00000000	00000c00	00000413	
5	00001500	1	0	0	00000000	00000000	00000e0c	deadbeef	5	00001500	1	0	0	00000000	00000000	00000e0c	deadbeef	
6	00000000	0	0	0	00000000	00000000	00001500	00000413	6	00000000	0	0	0	00000000	00000000	00001500	00000413	
7	00000000	0	0	0	00000000	00000000	00001504	00009117	7	00000000	0	0	0	00000000	00000000	00001504	00009117	
8	00000000	0	1	1	00000000	0003fb00	00001508	ff10113	8	00000000	0	1	1	00000000	0003fb00	00001508	ff10113	
9	00000000	0	0	0	00000000	00000000	0003fb00	00000413	9	00000000	0	0	0	00000000	0003fb00	00000413	00000413	
a	00000000	0	0	1	000001fc	00000000	0003fb04	00009117	a	00000000	0	0	1	000001fc	00000000	0003fb04	00009117	
b	00000000	0	1	1	00000000	00000150c	0003fdc0	00040513	b	00000000	0	1	1	00000000	00000150c	0003fdc0	00040513	
c	00000000	0	1	0	00000000	00000000	0000150c	39c000ef	c	00000000	0	1	1	00000000	00000000	0000150c	39c000ef	
d	00000000	0	0	0	00000000	00000000	00000c00	00000413	d	00000000	0	0	0	00000000	00000000	00000c00	00000413	
e	00000000	0	0	0	00000000	00000000	00000c04	00009117	e	00000000	0	0	0	00000000	00000000	00000c04	00009117	
f	00000000	0	0	1	fffff9c8	00000000	00000c08	fffc10113	f	00000000	0	0	1	fffff9c8	00000000	00000c08	fffc10113	

图 仿真测试图

(4) 错误现象及分析

要求：在电路设计、连接和仿真运行时，遇到的任何错误，都需要截屏放置到实验报告中，并分析错误原因和解决办法。

PC 值未更新可能是时钟触发边沿设置错误（如误设为下降沿）；指令读取错误可能是地址映射错误（未舍弃 PC 低两位）或 ROM 未加载正确的镜像文件 lab5.3.hex。

4. 取操作数部件 IDU 实验

(1) 实验整体方案设计

要求：说明本次实验的顶层设计模块图，对每个子模块进行详细描述，定义输入输出引脚，数据及控制信号的传输通道等。

通过分线器解析指令字段 (opcode、funct3 等)，立即数扩展器根据 ExtOp 信号生成 5 种类型 (I/U/S/B/J) 的 32 位立即数，寄存器堆读取 rs1 和 rs2 数据，多路选择器根据 ALUASrc/ALUBSrc 选择操作数输入 ALU。

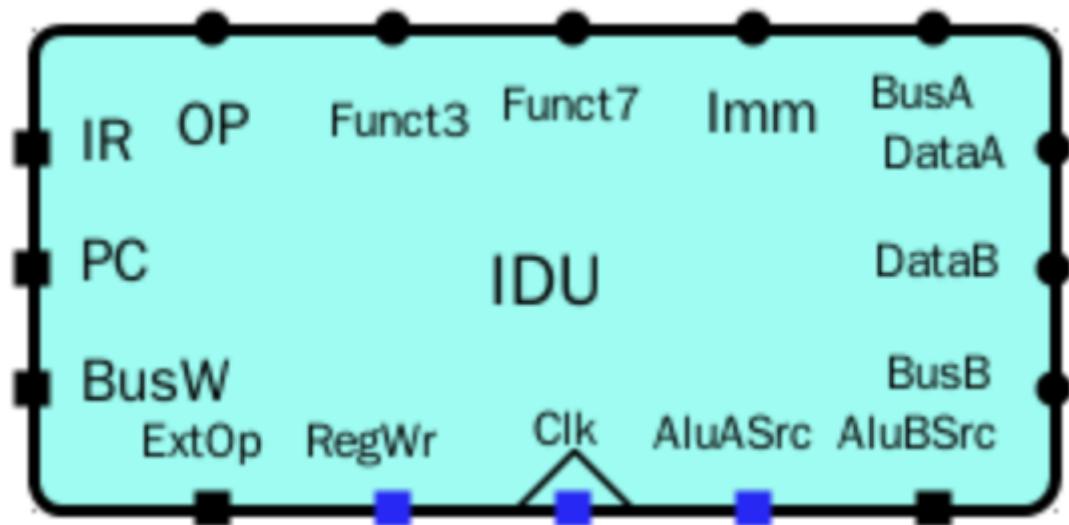


图 实验电路封装图

(2) 实验原理图和电路图

要求：给出每个子模块的原理图和 Logisim 中的电路图，定义子模块的外观图。如果对实验指导讲义中的内容提出优化或改进，需要此说明原因、方法和效果。

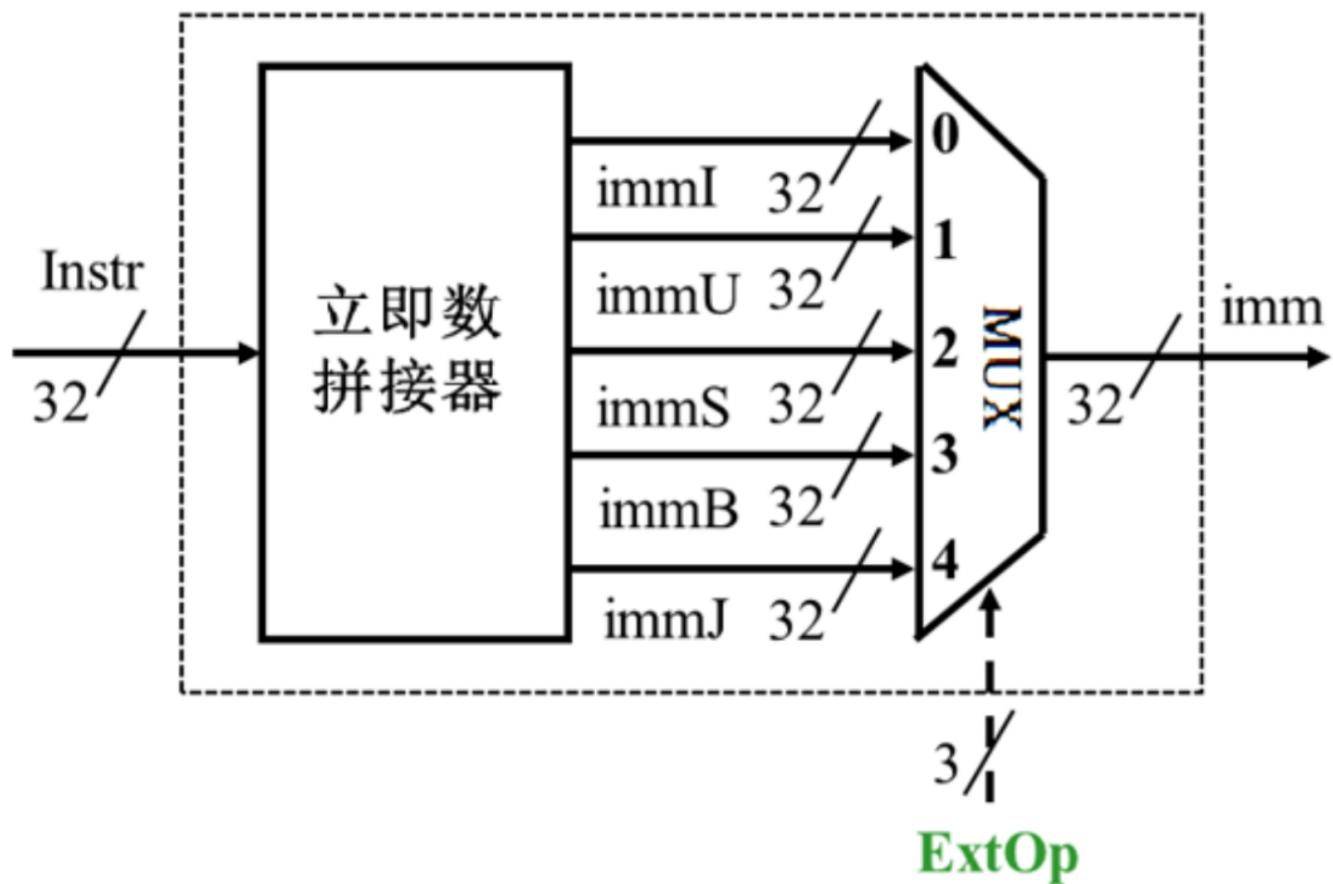


图 原理图

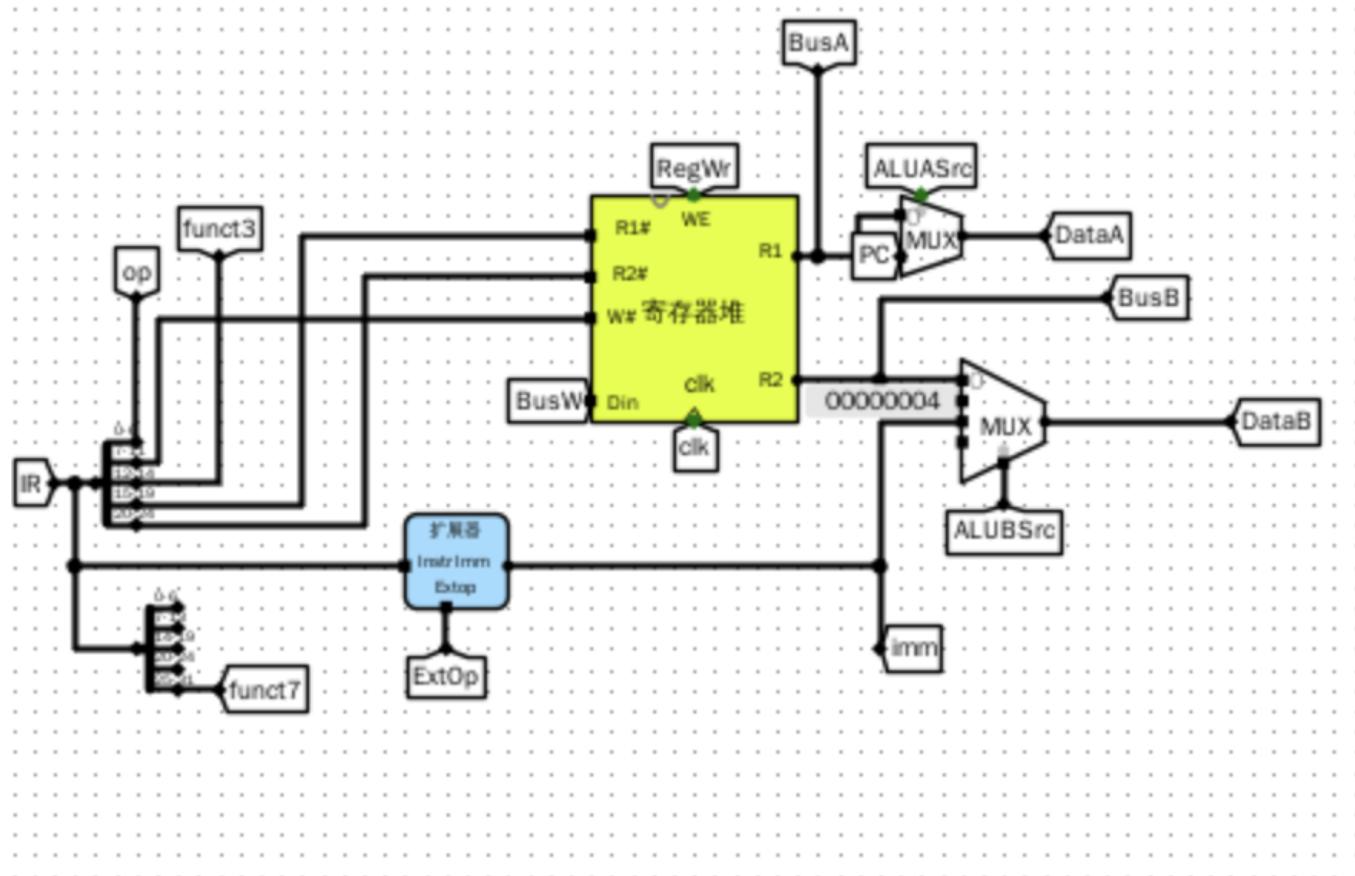


图 电路图

(3) 实验数据仿真测试图

要求：根据实验要求，输入测试数据，选择单步时钟执行，截取仿真运行时的电路图，分析电路状态是否满足设计需求。说明子模块的功能，列出子模块的功能表。

Cnt	IR	BusW	PC	ExtOp	RegNr	ALUASr	ALUBSr	DataA	DataB	BusS	Cnt	IR	BusW	PC	ExtOp	RegNr	ALUASr	ALUBSr	DataA	DataB	BusA
00	fedca2b7	fedca000	00000000	1	1	0	2	00000000	fedca000	00000000	00	fedca2b7	fedca000	00000000	1	1	0	2	00000000	fedca000	00000000
01	f9c28293	fedc9f9c	00000004	0	1	0	2	fedca000	ffffff9c	fedc:	01	f9c28293	fedc9f9c	00000004	0	1	0	2	fedca000	ffffff9c	fedc
02	01006013	00000010	00000008	0	1	0	2	00000000	00000010	00000000	02	01006013	00000008	00000008	0	1	0	2	00000000	00000010	0000
03	06502223	00000064	0000000c	2	0	0	2	00000000	00000064	00000000	03	06502223	00000064	0000000c	2	0	0	2	00000000	00000064	0000
04	06400303	ffffff9c	00000010	0	1	0	2	00000000	00000064	00000000	04	06400303	ffffff9c	00000010	0	1	0	2	00000000	00000064	0000
05	06601423	00000068	00000014	2	0	0	2	00000000	00000068	00000000	05	06601423	00000068	00000014	2	0	0	2	00000000	00000068	0000
06	06405383	00000f9c	00000018	0	1	0	2	00000000	00000064	00000000	06	06405383	00000f9c	00000018	0	1	0	2	00000000	00000064	0000
07	06701623	0000006c	0000001c	2	0	0	2	00000000	0000006c	00000000	07	06701623	0000006c	0000001c	2	0	0	2	00000000	0000006c	0000
08	4042d413	ffedc9f9	00000020	0	1	0	2	fedc9f9c	00000404	fedc:	08	4042d413	ffedc9f9	00000020	0	1	0	2	ffedc9f9	00000404	fedc
09	006444b3	00123665	00000024	0	1	0	0	ffedc9f9	ffffff9c	ffedc:	09	006444b3	00123665	00000024	0	1	0	0	ffedc9f9	ffffff9c	ffedc
0a	00649533	50000000	00000028	0	1	0	0	00123665	ffffff9c	0012:	0a	00649533	50000000	00000028	0	1	0	0	00123665	ffffff9c	0012
0b	008505b3	4fedc9f9	0000002c	0	1	0	0	50000000	ffedc9f9	50000000	0b	008505b3	4Fedc9f9	0000002c	0	1	0	0	50000000	ffedc9f9	5000
0c	00b2a633	00000001	00000030	0	1	0	0	fedc9f9c	4fedc9f9	fedc:	0c	00b2a633	00000001	00000030	0	1	0	0	fedc9f9c	4Fedc9f9	fedc
0d	00b2b6b3	00000000	00000034	0	1	0	0	fedc9f9c	4fedc9f9	fedc:	0d	00b2b6b3	00000000	00000034	0	1	0	0	fedc9f9c	4Fedc9f9	fedc
0e	40b287b3	aeed5a3	00000038	0	1	0	0	fedc9f9c	4fedc9f9	fedc:	0e	40b287b3	aeed5a3	00000038	0	1	0	0	fedc9f9c	4Fedc9f9	fedc
0f	06f02823	00000070	0000003c	2	0	0	2	00000000	00000070	00000000	0f	06f02823	00000070	0000003c	2	0	0	2	00000000	00000070	0000
10	0067c263	00000001	00000040	3	0	0	0	aeed5a3	ffffff9c	aeee:	10	0067c263	00000001	00000040	3	0	0	0	aeed5a3	ffffff9c	aeee
11	0067d263	00000001	00000044	3	0	0	0	aeed5a3	ffffff9c	aeee:	11	0067d263	00000001	00000044	3	0	0	0	aeed5a3	ffffff9c	aeee
12	0040086f	0000004c	00000048	4	1	1	1	00000048	00000044	00000000	12	0040086f	0000004c	00000048	4	1	1	1	00000048	00000044	0000
13	004080e7	00000050	0000004c	0	1	1	1	0000004c	00000044	00000000	13	004080e7	00000050	0000004c	0	1	1	1	0000004c	00000044	0000
14	00001917	000001050	00000050	1	1	1	2	00000050	000001000	00000000	14	00001917	000001050	00000050	1	1	1	2	00000050	000001000	0000
15	00000000	00000000	00000000	0	0	0	0	00000000	00000000	00000000	15	00000000	00000000	00000000	0	0	0	0	00000000	00000000	0000

图 仿真测试图

(4) 错误现象及分析

要求：在电路设计、连接和仿真运行时，遇到的任何错误，都需要截屏放置到实验报告中，并分析错误原因和解决办法。

立即数扩展错误可能是 ExtOp 编码与扩展逻辑不对应（如 ExtOp=2 时未生成 immS）；寄存器堆输出为 0 可能是 0 号寄存器硬连接未正确设置，或 rs1/rs2 索引超出范围。

5. 数据通路实验

(1) 实验整体方案设计

要求：说明本次实验的顶层设计模块图，对每个子模块进行详细描述，定义输入输出引脚，数据及控制信号的传输通道等。

整合 IFU、IDU、ALU、数据存储器和跳转控制器，形成单周期 CPU 数据通路。跳转控制器根据 BandJ 信号和 ALU 结果 (Zero/Result [0]) 生成 NPC 控制信号，支持 jal、jalr、beq 等指令的地址计算，数据存储器与寄存器堆通过时钟边沿同步写入。

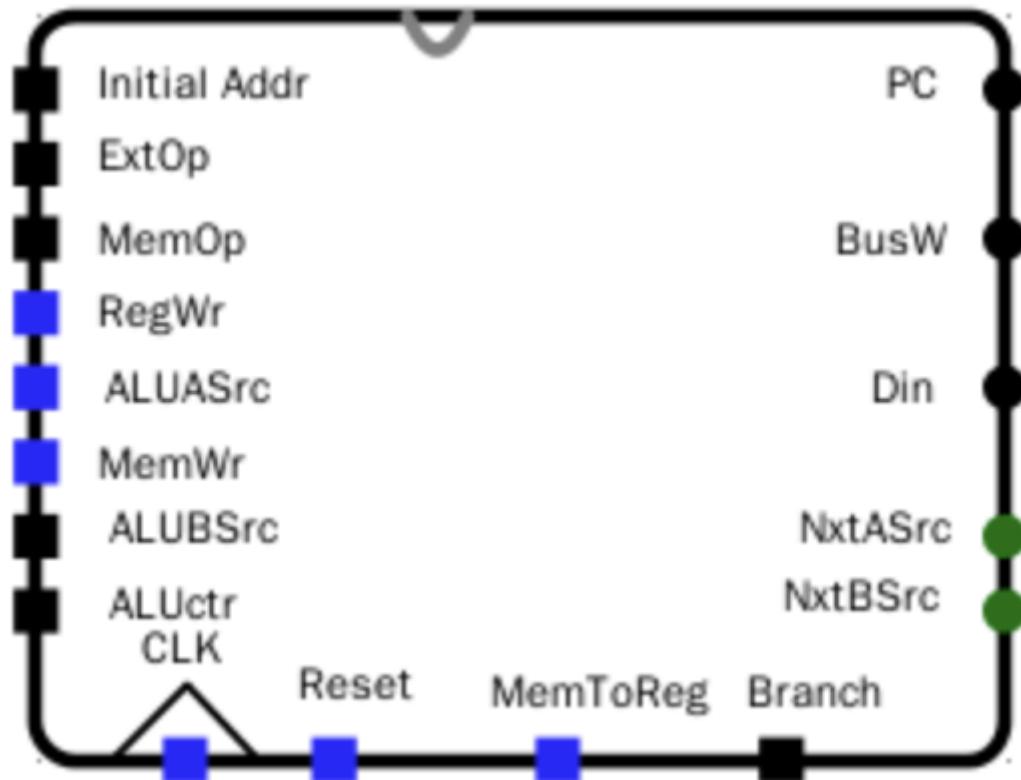


图 实验电路封装图

(2) 实验原理图和电路图

要求：给出每个子模块的原理图和 Logisim 中的电路图，定义子模块的外观图。如果对实验指导讲义中的内容提出优化或改进，需要此说明原因、方法和效果。

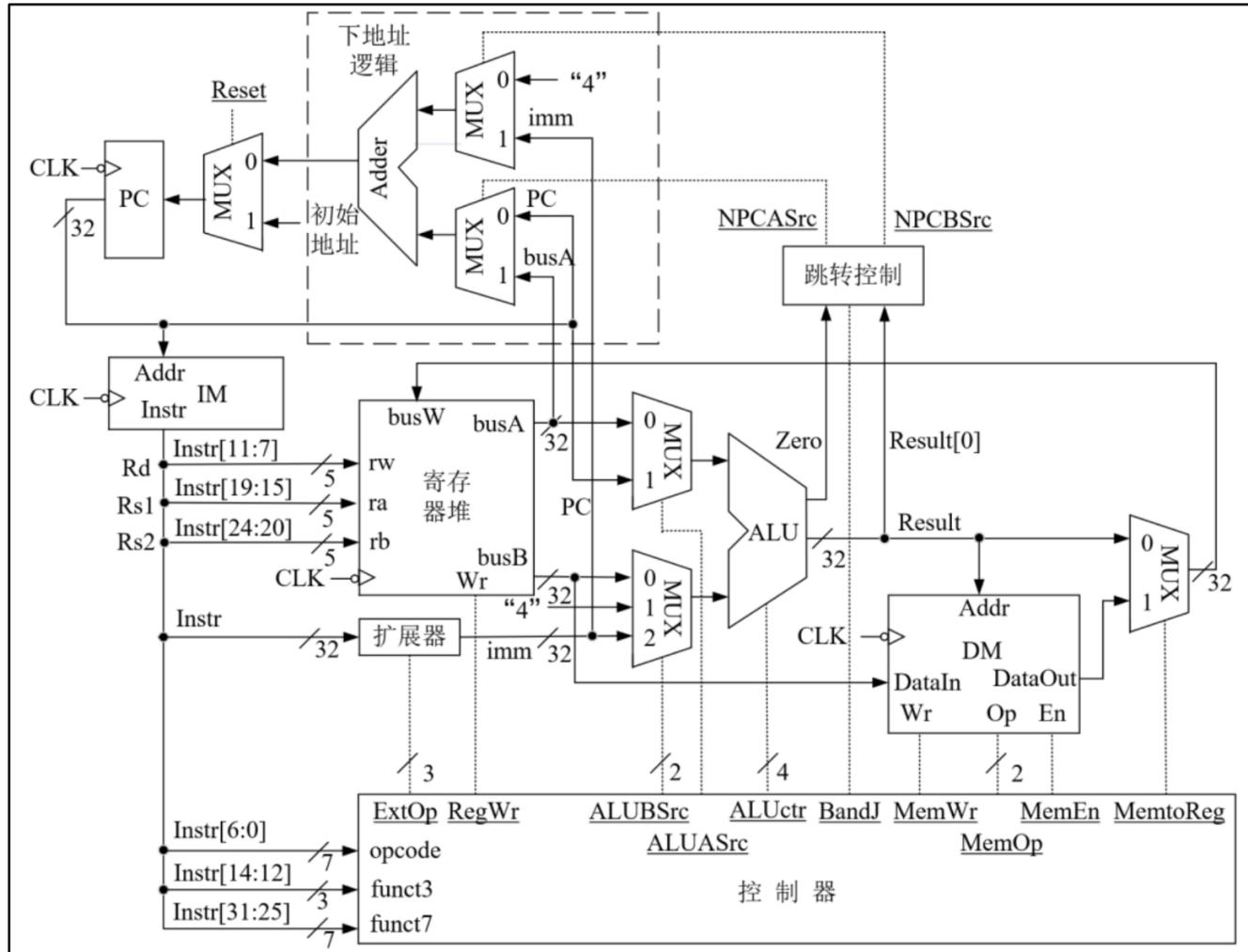


图 原理图

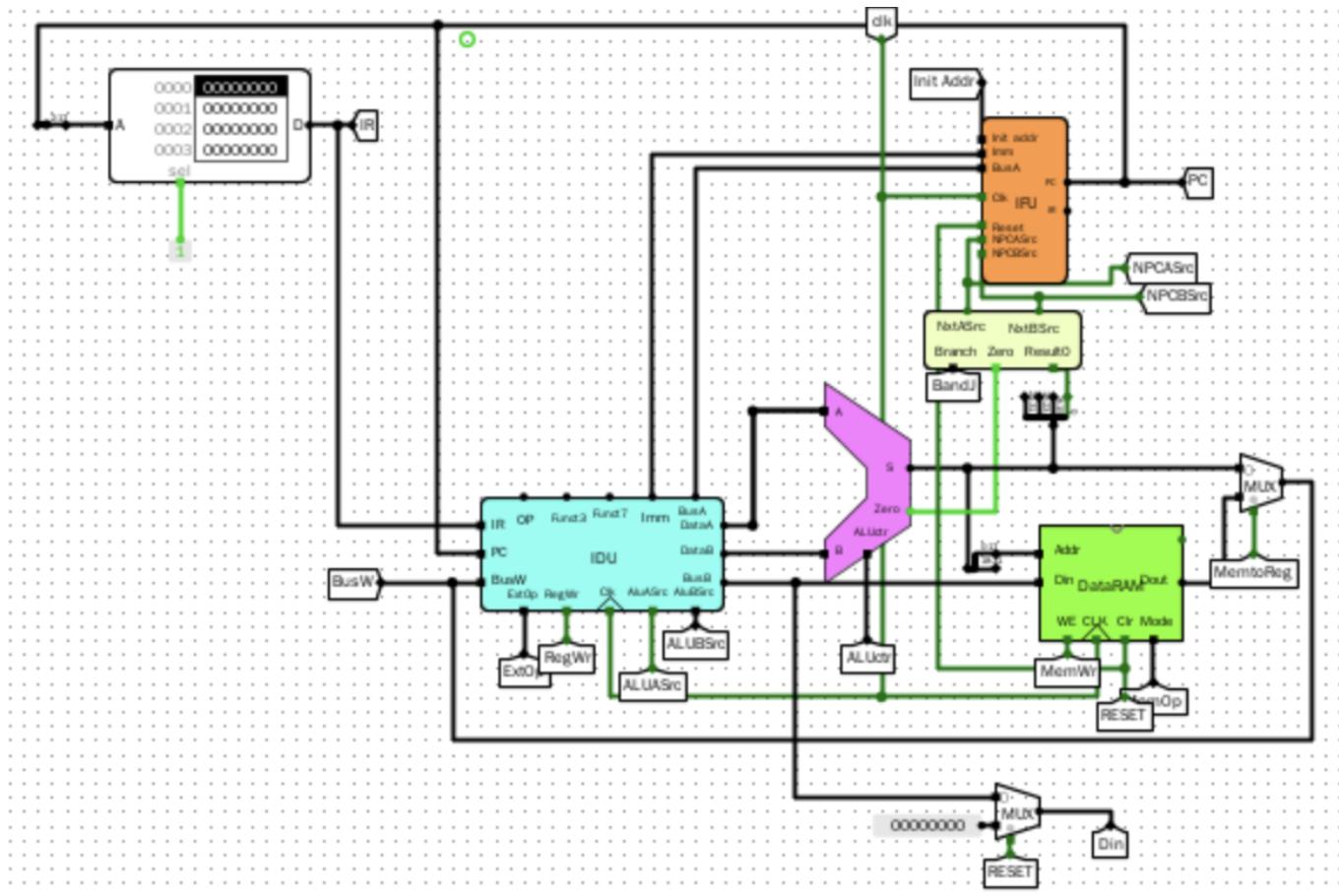


图 电路图

(3) 实验数据仿真测试图

要求：根据实验要求，输入测试数据，选择单步时钟执行，截取仿真运行时的电路图，分析电路状态是否满足设计需求。说明子模块的功能，列出子模块的功能表。

Cnt	InitialAddress	ReSet	ExtOp	MemOp	RegWr	ALUASr	ALUBSr	ALUctr	MemWr	MemToRB	Band1	PC	BusW	Cnt	InitialAddress	ReSet	ExtOp	MemOp	RegWr	ALUASr	ALUBSr	ALUctr	MemWr	MemToRB	Band1	PC	BusW	
00	00000000	1	0	0	0	0	0	0	0	0	0	00000000	00000000	00	00000000	1	0	0	0	0	0	0	0	0	0	00000000	00000000	
01	00000000	0	1	0	1	0	2	f	0	0	0	00000000	fedca000	01	00000000	0	1	0	1	0	2	f	0	0	0	0	00000000	fedca000
02	00000000	0	0	0	1	0	2	0	0	0	0	00000000	fedc9f90	02	00000000	0	0	0	1	0	2	0	0	0	0	0	00000000	fedc9f90
03	00000000	0	0	0	1	0	2	6	0	0	0	00000000	00000001	03	00000000	0	0	0	1	0	2	6	0	0	0	0	00000000	00000001
04	00000000	0	2	0	0	0	2	0	1	0	0	00000000	00000000	04	00000000	0	2	0	0	0	2	0	1	0	0	0	00000000	00000006
05	00000000	0	0	5	1	0	2	0	0	1	0	00000000	00000010	05	00000000	0	0	5	1	0	2	0	0	1	0	0	00000000	fffffff9
06	00000000	0	2	6	0	0	2	0	1	0	0	00000000	00000014	06	00000000	0	2	6	0	0	2	0	1	0	0	0	00000000	00000061
07	00000000	0	0	2	1	0	2	0	0	1	0	00000000	00000018	07	00000000	0	0	2	1	0	2	0	0	1	0	0	00000000	00000018
08	00000000	0	2	6	0	0	2	0	1	0	0	00000000	0000001c	08	00000000	0	2	6	0	0	2	0	1	0	0	0	00000000	00000061
09	00000000	0	0	0	1	0	2	d	0	0	0	00000000	00000020	09	00000000	0	0	0	1	0	2	d	0	0	0	0	00000000	ffedc9f1
0a	00000000	0	0	0	1	0	0	4	0	0	0	00000000	00000024	0a	00000000	0	0	0	1	0	0	4	0	0	0	0	00000000	00123661
0b	00000000	0	0	0	1	0	0	1	0	0	0	00000000	00000028	0b	00000000	0	0	0	1	0	0	1	0	0	0	0	00000000	50000000
0c	00000000	0	0	0	1	0	0	0	0	0	0	00000000	00000042	0c	00000000	0	0	0	1	0	0	0	0	0	0	0	00000000	4fedc9f1
0d	00000000	0	0	0	1	0	0	2	0	0	0	00000000	00000048	0d	00000000	0	0	0	1	0	0	2	0	0	0	0	00000000	00000048
0e	00000000	0	0	0	1	0	0	3	0	0	0	00000000	00000044	0e	00000000	0	0	0	1	0	0	3	0	0	0	0	00000000	00000044
0f	00000000	0	0	0	1	0	0	8	0	0	0	00000000	00000048	0f	00000000	0	0	0	1	0	0	8	0	0	0	0	00000000	aeed5a1
10	00000000	0	2	0	0	0	2	0	1	0	0	00000000	00000043	10	00000000	0	2	0	0	0	2	0	1	0	0	0	00000000	00000071
11	00000000	0	3	0	0	0	0	2	0	0	6	00000000	00000040	11	00000000	0	3	0	0	0	0	2	0	0	6	0	00000000	00000000
12	00000000	0	3	0	0	0	0	2	0	0	7	00000000	00000044	12	00000000	0	3	0	0	0	0	2	0	0	7	0	00000000	00000000
13	00000000	0	4	0	1	1	1	0	0	0	1	00000000	00000048	13	00000000	0	4	0	1	1	1	0	0	0	1	0	00000000	00000044
14	00000000	0	0	0	1	1	1	0	0	0	2	00000000	0000004c	14	00000000	0	0	0	1	1	0	0	0	2	0	0	00000000	00000045
15	00000000	0	1	0	1	1	2	0	0	0	0	00000000	00000045	15	00000000	0	1	0	1	1	2	0	0	0	0	0	00000000	00000145
16	00000000	0	0	0	0	0	0	0	0	0	0	00000000	00000044	16	00000000	0	0	0	0	0	0	0	0	0	0	0	00000000	00000045

图 仿真测试图

序号	指令 PC=0	RV32 汇编语句	控制信号赋值, 未列出的控制信号值为 0	输出信号值 (32 位, 16 进制数)
1	008000ef	jal x1, 0x8	ExtOp=4, RegWr=1, ALUBSrc=1, Branch=1	x1: 00000004
2	ffdff06f	jal x0, 0xffffffffc	ExtOp=4, RegWr=1, ALUBSrc=1, Branch=1	x0: 00000000
3	00100013	addi x0, x0, 1	RegWr=1, ALUBSrc=2	x0: 00000000
4	deadc2b7	lui x5, deadc	ExtOp=1, RegWr=1, ALUBSrc=2, ALUCtr=15	x5: deadc000
5	eef28293	addi x5, x5, ffffffe	RegWr=1, ALUBSrc=2	x5: deadbeef
6	0052d333	srl x6, x5, x5	RegWr=1, ALUCtr=5	x6: 0001bd5b
7	00135313	sili x6, x6, 1	RegWr=1, ALUBSrc=2, ALUCtr=5	x6: 0000dead
8	4052d3b3	sra x7, x5, x5	RegWr=1, ALUCtr=13	x7: fffffbd5b
9	00529e33	sll x28, x5, x5	RegWr=1, ALUctr=1	x28: df778000
10	001e1e13	slli x28, x28, 1	RegWr=1, ALUBSrc=2, ALUctr=1	x28: beef0000
11	006e0eb3	add x29, x28, x6	RegWr=1	x29: beafdead
12	01000513	addi x10, x0, 10	RegWr=1, ALUBSrc=2	x10: 10
13	005500a3	sb x5, 1(x10)	ExtOp=2, ALUBSrc=2, MemWr=1, MemOp=5	M[0x10]: 0000ef00
14	00551223	sh x5, 4(x10)	ExtOp=2, ALUBSrc=2, MemWr=1, MemOp=6	M[0x14]: beef
15	00551323	sh x5, 6(x10)	ExtOp=2, ALUBSrc=2, MemWr=1, MemOp=6	M[0x14]: beefbeef
16	00552423	sw x5, 8(x10)	ExtOp=2, ALUBSrc=2, MemWr=1	M[0x18]: deadbeef
17	00150483	lb x9, 1(x10)	RegWr=1, ALUBSrc=2, MemOp=5, MemToReg=1	x9: ffffffef
18	00154903	lbu x18, 1(x10)	RegWr=1, ALUBSrc=2, MemOp=1, MemToReg=1	x18: 0xef
19	00451983	lh x19, 4(x10)	RegWr=1, ALUBSrc=2, MemOp=6, MemToReg=1	x19: ffffbeef
20	00655a03	lhu x20, 6(x10)	RegWr=1, ALUBSrc=2, MemOp=2, MemToReg=1	x20: beef
21	00452a83	lw x21, 4(x10)	RegWr=1, ALUBSrc=2, MemToReg=1	x21: beefbeef
22	00a55b03	lhu x22, 10(x10)	RegWr=1, ALUBSrc=2, MemOp=2, MemToReg=1	x22: dead
23	ffff0bb7	lui x23, ffff0	ExtOp=1, RegWr=1, ALUBSrc=2, ALUCtr=15	x23: ffff0000
24	017afc33	and x24, x21, x23	RegWr=1, ALUCtr=7	x24: beef0000
25	016c6cb3	or x25, x24, x22	RegWr=1, ALUCtr=6	x25: beefdead
26	fb9e92e3	bne x29, x25, ffffffa4	ExtOp=3, ALUCtr=2, Branch=5	PC: 68
27	00000073			PC: 6c

图 测评表

(4) 错误现象及分析

要求：在电路设计、连接和仿真运行时，遇到的任何错误，都需要截屏放置到实验报告中，并分析错误原因和解决办法。

程序执行顺序混乱可能是跳转控制器逻辑错误（如 BandJ 编码与 NPCASrc/NPCBSrc 映射错误）；数据写入异常可能是时钟触发边沿不一致（如 PC 用上升沿而存储器用下降沿），导致时序冲突。

思考题在报告开头处