

计算机组成与结构专题实验

实验报告

第六次 基本模型计算机设计与实现

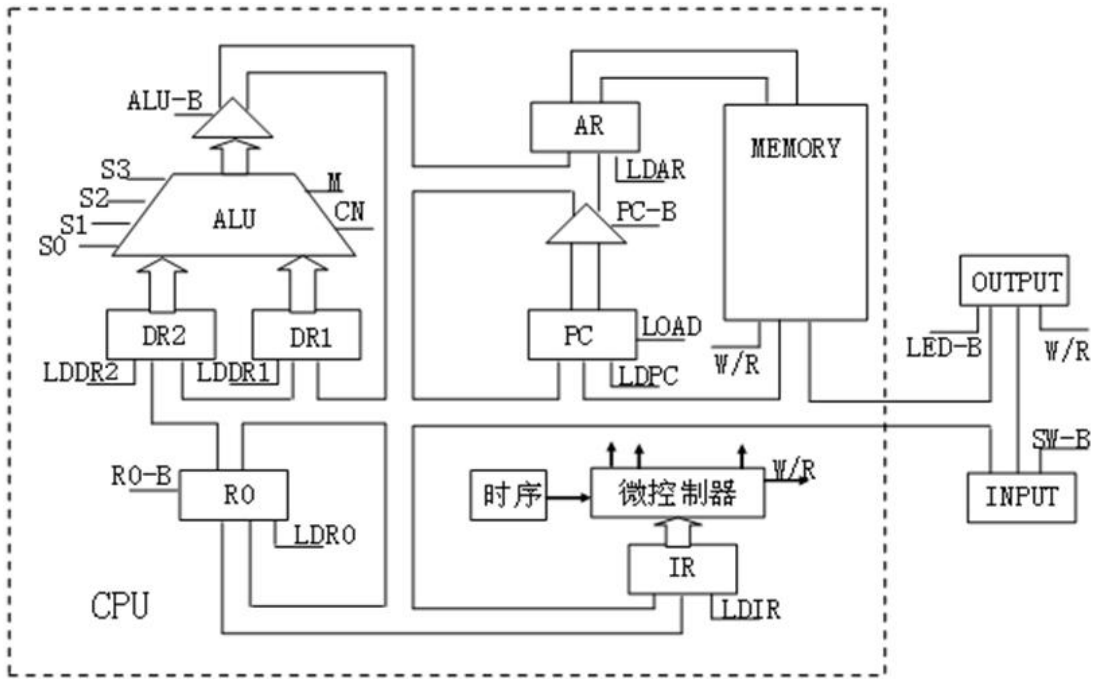
一、 实验目的

1. 深刻领会计算机基本模型的构成要素以及其具备的各项功能。
对计算机中各类典型指令的执行过程进行深入探究。
2. 学习微程序控制器的设计原理及相关技术，熟练掌握 LPM_ROM 的配置方式。
3. 在熟悉部件单元电路实验的基础上，进一步将单元电路组合成一个完整的系统，构建出一台基本模型计算机。
4. 设计五条机器指令，并为其编写相应的微程序，通过上机调试，深入理解计算机整机的概念。掌握微程序的设计方法，学会编写二进制微指令代码表。
5. 通过完整地设计计算机，全面了解并掌握采用微程序控制方式的计算机的设计方法。

二、 实验原理

该 8 位 CPU 的结构如下图所示，主要包括运算部件、寄存器组、指令寄存器、程序计数器、地址寄存器、标志寄存器、微命令产生部件以及时序系统等。在这个模型机中，每条指令都必须包含以下四个部分：操作码、操作数的地址、操作结果的存储地址以及下一条指令的地址，

具体结构如图所示。



8 位 CPU 的结构

位	7	6	5	4	32	10
功能	OP-CODE				rs	rd

指令的基本格式

rs 或 rd	选定的寄存器
00	R0
01	R1
10	R2

寄存器操作数

地址（16进制）		助记符	说明
00	00	IN	“INPUT”→ R0， 键盘输入数据
01	10	ADD [0AH]	[R0]+[0AH] → R0
02	0A		
03	20	STA [0BH]	[R0] → [0BH]
04	0B		
05	30	OUT [0BH]	[0BH] “OUTPUT”， 显示输出数据
06	0B		
07	40	JMP [08H]	[08H] → PC ， 以[08H]内容为转移地址
08	00		
09	00		
0A	34	DB 34H	被加数（自定）
0B	XX		求和结果

模型机指令及编码形式

24	23	22	21	20	19	18	17	16	15 14 13	12 11 10	9 8 7	6	5	4	3	2	1
S3	S2	S1	S0	M	Cn	WE	A9	A8	A	B	C	<u>uA5</u>	<u>uA4</u>	<u>uA3</u>	<u>uA2</u>	<u>uA1</u>	<u>uA0</u>
操作控制信号									译码器	译码器	译码器	下地址字段					

A9	A8	选择
0	0	SW_B
0	1	RAM_B
1	0	LED_B
1	1	无

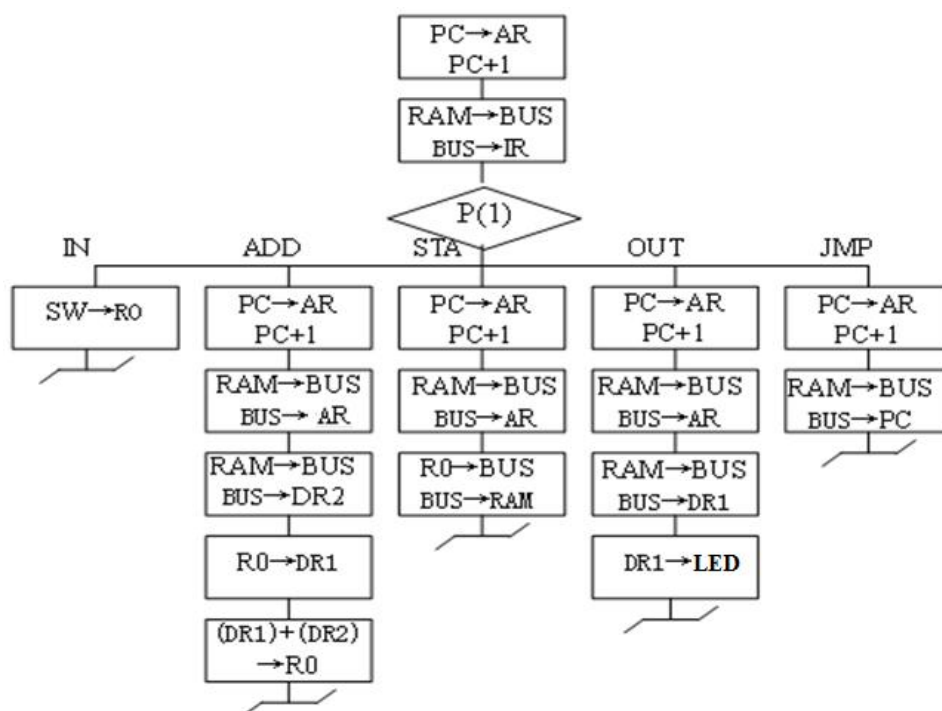
24 位微代码定义

A 字段				B 字段				C 字段			
15	14	13	选择	12	11	10	选择	9	8	7	选择
0	0	0		0	0	0		0	0	0	
0	0	1	LDRi	0	0	1	RS-B	0	0	1	P (1)
0	1	0	LDDR1	0	1	0	RD_B	0	1	0	P (2)
0	1	1	LDDR2	0	1	1	RJ_B	0	1	1	P (3)
1	0	0	LDIR	1	0	0	SFT_B	1	0	0	P (4)
1	0	1	LOAD	1	0	1	ALU-B	1	0	1	AR
1	1	0	LDAR	1	1	0	PC-B	1	1	0	LDPC

A、B、C 各字段功能说明

地址（16进制）	内容（16进制）	助记符	说明
00	00	IN	“INPUT”→ R0，键盘输入数据
01	10	ADD [0AH]	[R0]+[0AH] → R0
02	0A		
03	20	STA [0BH]	[R0] → [0BH]
04	0B		
05	30	OUT [0BH]	[0BH] “OUTPUT”，显示输出数据
06	0B		
07	40	JMP [08H]	[09H] → PC，以[08H]内容为转移地址
08	00		
09	00		
0A	34	DB 34H	被加数（自定）
0B	XX		求和结果

实验程序： 模型机指令及编码形式



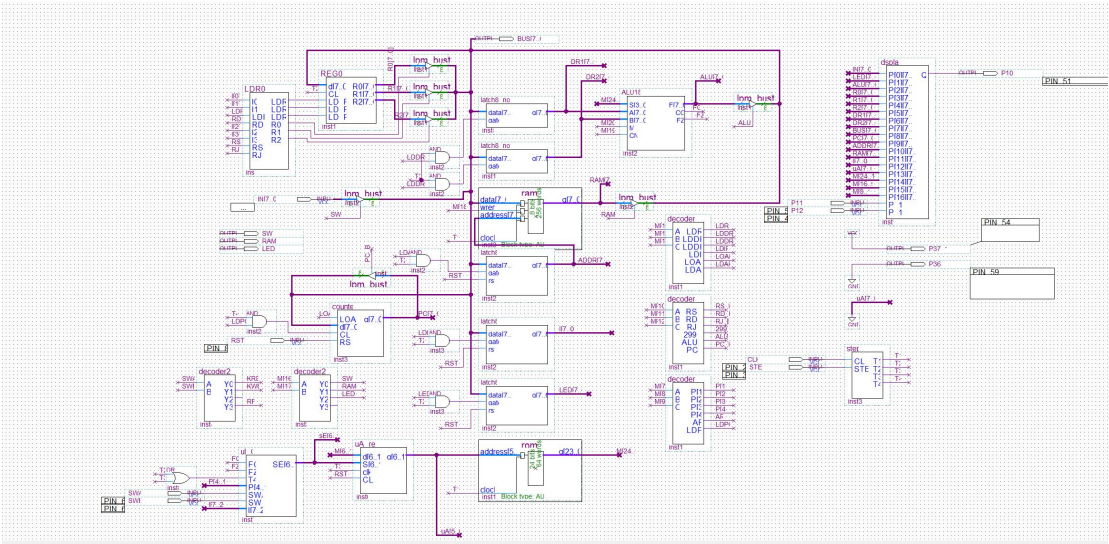
微程序流程图

三、实验任务

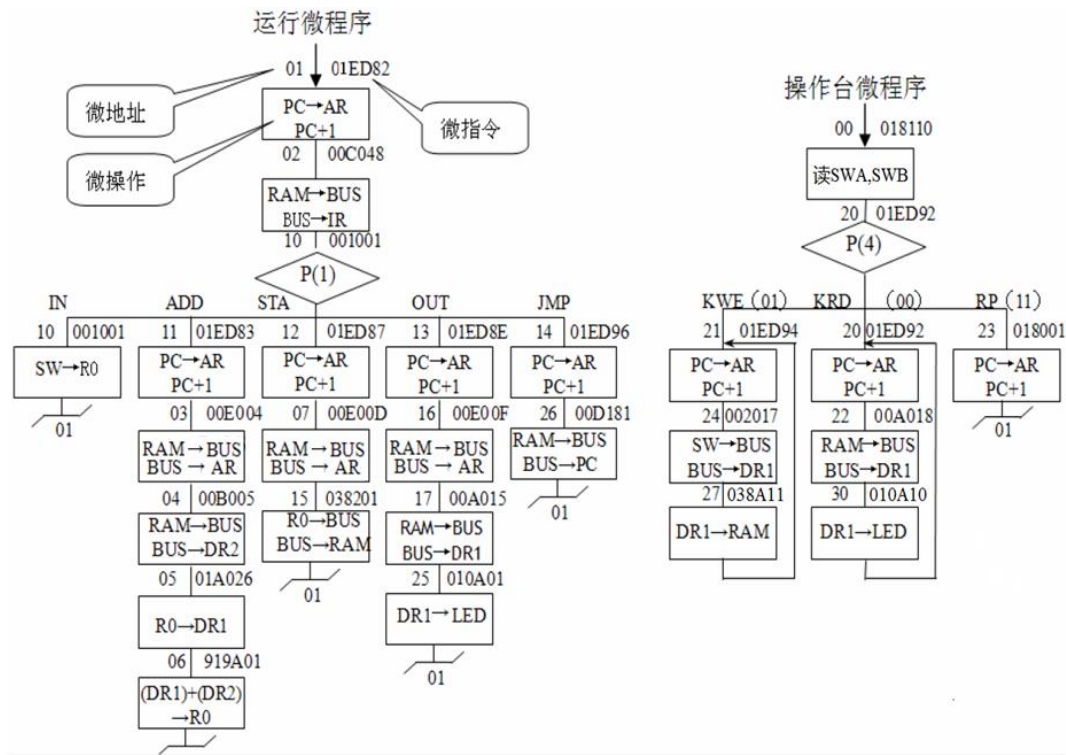
- (1) 用图形编辑工具设计模型 CPU 的顶层电路原理图。
- (2) 根据微程序的微操作，对于所需的控制信号，确定微指令，并确定微地址。
- (3) 微程序流程图按微指令格式转化为“二进制微代码表”。
- (4) 设计控制存储器 LPM_ROM。
- (5) 对模型 CPU 的整机硬件电路进行编译、波形仿真和调试。
- (6) 根据仿真波形，查找故障原因，排除故障，重新编译。
- (7) 将编译通过的电路和应用程序下载到实验台上的 FPGA 中，在实验台上单步跟踪微程序的执行过程。
- (8) 最终完成模型 CPU 的硬件电路设计和应用程序及微程序的设计和调试。

四、 实验步骤及结果

CPU 顶层设计



微指令流程图

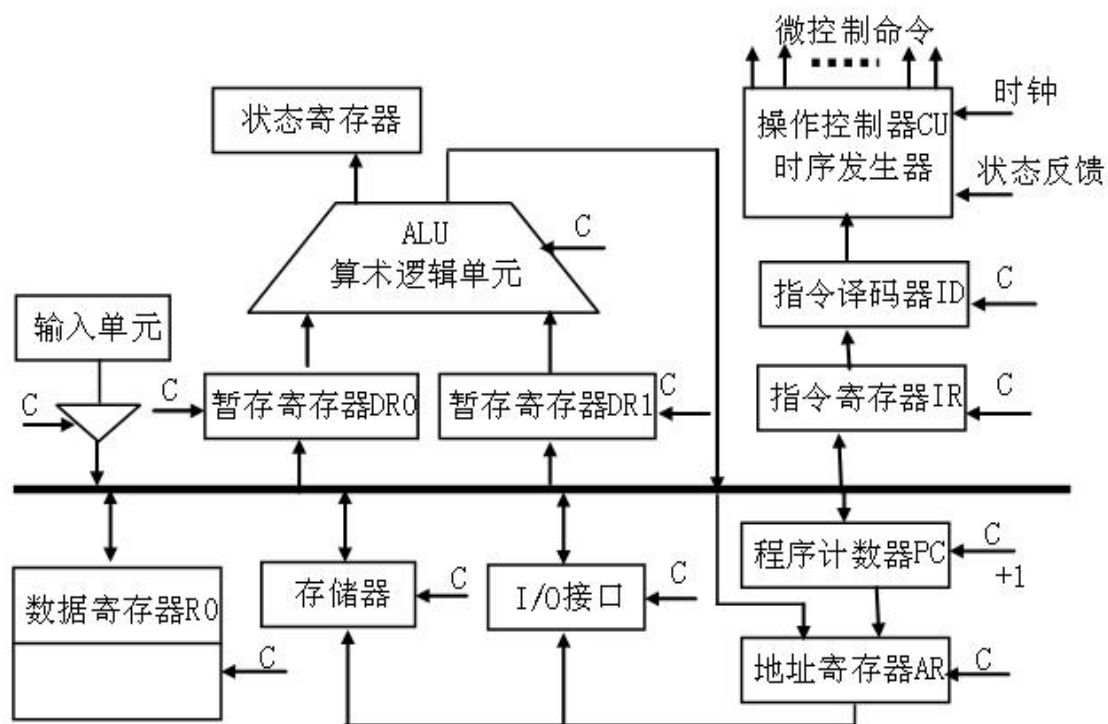


微代码表

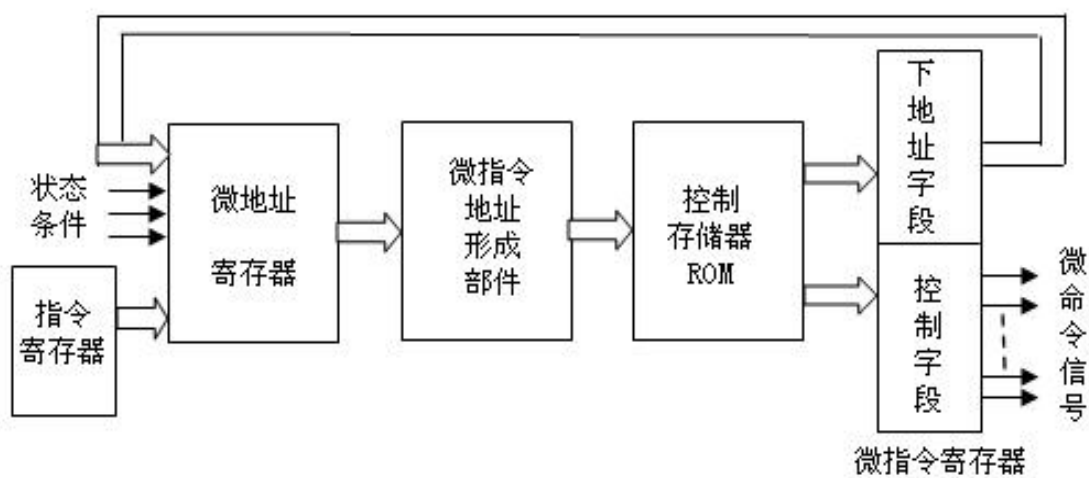
微地址	微指令	S3	S2	S1	S0	M	CN	WE	A9	A8	A	B	C	uA5-uA0
0 0	018110	0	0	0	0	0	0	0	1	1	000	000	100	010000
0 1	01ED82	0	0	0	0	0	0	0	1	1	110	110	110	000010
0 2	00C048	0	0	0	0	0	0	0	0	1	100	000	001	001000
0 3	00E004	0	0	0	0	0	0	0	0	1	110	000	000	000100
0 4	00B005	0	0	0	0	0	0	0	0	1	011	000	000	000101
0 5	01A206	0	0	0	0	0	0	0	1	1	010	001	000	000110
0 6	919A01	1	0	0	1	0	0	0	1	1	001	101	000	000001
0 7	00E00D	0	0	0	0	0	0	0	0	1	110	000	000	001101
1 0	001001	0	0	0	0	0	0	0	0	0	001	000	000	000001
1 1	01ED83	0	0	0	0	0	0	0	1	1	110	110	110	000011
1 2	01ED87	0	0	0	0	0	0	0	1	1	110	110	110	000111
1 3	01ED8E	0	0	0	0	0	0	0	1	1	110	110	110	001110
1 4	01ED96	0	0	0	0	0	0	0	1	1	110	110	110	010110
1 5	038201	0	0	0	0	0	0	1	1	1	000	001	000	000001
1 6	00E00F	0	0	0	0	0	0	0	0	1	110	000	000	001111
1 7	00A015	0	0	0	0	0	0	0	0	1	010	000	000	010101
2 0	01ED92	0	0	0	0	0	0	0	1	1	110	110	110	010010
2 1	01ED94	0	0	0	0	0	0	0	1	1	110	110	110	010100
2 2	00A018	0	0	0	0	0	0	0	0	1	010	000	000	011000
2 3	018001	0	0	0	0	0	0	0	1	1	000	000	000	000001
2 4	002017	0	0	0	0	0	0	0	0	0	010	000	000	010111
2 5	010A01	0	0	0	0	0	0	0	1	0	000	101	000	000001
2 6	00D181	0	0	0	0	0	0	0	0	1	101	000	110	000001
2 7	038A11	0	0	0	0	0	0	1	1	1	000	101	000	010001
3 0	010A10	0	0	0	0	0	0	0	1	0	000	101	000	010000

A 字段				B 字段				C 字段			
15	14	13	选择	12	11	10	选择	9	8	7	选择
0	0	0		0	0	0		0	0	0	
0	0	1	LDRi	0	0	1	RS-B	0	0	1	P (1)
0	1	0	LDDR1	0	1	0	RD_B	0	1	0	P (2)
0	1	1	LDDR2	0	1	1	RJ_B	0	1	1	P (3)
1	0	0	LDIR	1	0	0	SFT_B	1	0	0	P (4)
1	0	1	LOAD	1	0	1	ALU-B	1	0	1	AR
1	1	0	LDAR	1	1	0	PC-B	1	1	0	LDPC

拟定指令流程和微命令序列



建立数据通路



微程序控制

<<new>> <input checked="" type="checkbox"/> Filter on node names: *							
	status	From	To	Assignment Name	Value	Enabled	Entity
1	✓ ...		IN[0]	Location	PIN_60	Yes	
2	✓ ...		IN[7]	Location	PIN_39	Yes	
3	✓ ...		IN[6]	Location	PIN_42	Yes	
4	✓ ...		IN[5]	Location	PIN_83	Yes	
5	✓ ...		IN[4]	Location	PIN_77	Yes	
6	✓ ...		IN[3]	Location	PIN_74	Yes	
7	✓ ...		IN[2]	Location	PIN_70	Yes	
8	✓ ...		IN[1]	Location	PIN_65	Yes	
9	✓ ...		SWA	Location	PIN_64	Yes	
10	✓ ...		SWB	Location	PIN_66	Yes	
11	✓ ...		STEP	Location	PIN_34	Yes	
12	✓ ...		RST0	Location	PIN_84	Yes	
13	✓ ...		P10_1	Location	PIN_51	Yes	
14	✓ ...		P11_2	Location	PIN_50	Yes	
15	✓ ...		P12_3	Location	PIN_44	Yes	
16	✓ ...		P36_9	Location	PIN_59	Yes	
17	✓ ...		P37_10	Location	PIN_54	Yes	
18	✓ ...		CLK	Location	PIN_23	Yes	
19		<<new>>	<<new>>	<<new>>			

绑定引脚

Project Navigator

Entity

Cyclone IV E: EP4CE6E22C8

Lab6

Hierarchy

Files

Design Units

IP Compor

Tasks

Flow: Compilation

Customize...

Task	Time
✓ Compile Design	00:00:21
✓ > Analysis & Synthesis	00:00:07
✓ > Fitter (Place & Route)	00:00:06
✓ > Assembler (Generate programming files)	00:00:03
✓ > TimeQuest Timing Analysis	00:00:03
✓ > EDA Netlist Writer	00:00:02
Program Device (Open Programmer)	

Home

Lab6.bdf

Table of Contents

Flow Summary

Flow Settings

Flow Non-Default Global Settings

Flow Elapsed Time

Flow OS Summary

Flow Log

Analysis & Synthesis

Fitter

Assembler

TimeQuest Timing Analyzer

EDA Netlist Writer

Flow Messages

Flow Suppressed Messages

Flow Summary

Flow Status

Quartus II 64-Bit Version

Revision Name

Top-level Entity Name

Family

Device

Timing Models

Total logic elements

Total combinational functions

Dedicated logic registers

Total registers

Total pins

Total virtual pins

Total memory bits

Embedded Multiplier 9-bit elements

Total PLLs

Successful - Fri Apr 04 11:06:02 2025

13.1.0 Build 162 10/23/2013 SJ Full Vers

Lab6

Cyclone IV E

EP4CE6E22C8

Final

1,003 / 6,272 (16 %)

984 / 6,272 (16 %)

208 / 6,272 (3 %)

208

29 / 92 (32 %)

0

3,584 / 276,480 (1 %)

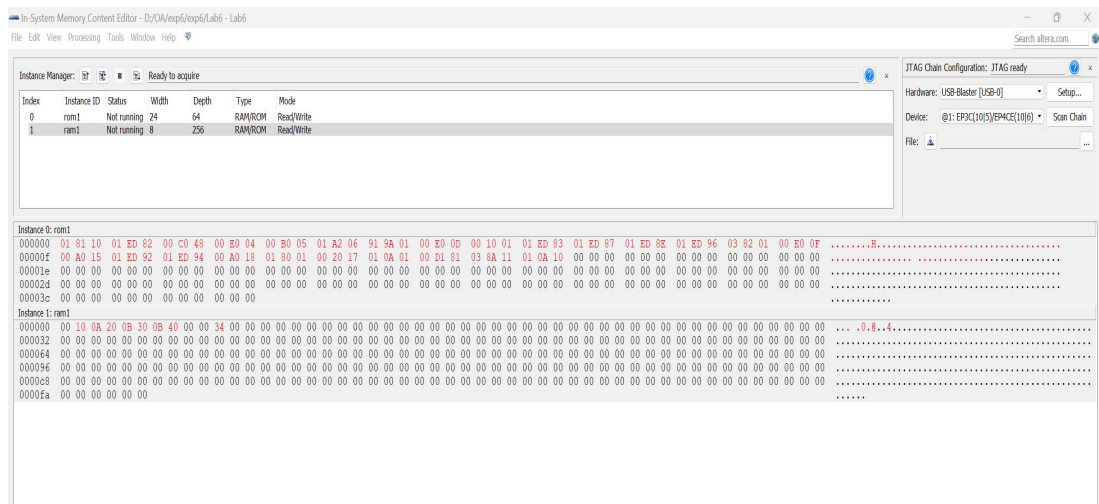
0 / 30 (0 %)

0 / 2 (0 %)

编译并下载至 FPGA

STEP	后续 uA 微地址	MC 微指令	PC	IR 指令	完成功能	执行结果
1	00	018110	00	00	控制台（读/写/运行）功能判断	控制台操作：转入程序运行方式
2	23	018001			SWB、WSA=（11）转 RP，分支转移	转程序执行方式
3	02	01ED82	01	80	执行第 1 条指令。（输入 IN）	PC→AR=00H, PC+1=01H, AR 指向指令地址
4	10	00C048			取指令, 将 RAM 中的指令送指令寄存器	RAM(00H)=00→BUS→IR=00H
5	01	001001		02	接收来自 IN 输入的数据, 送 R0 寄存器	R0=56H, 键 1、键 2 输入数据 56H
6	02	01ED82			执行第 2 条指令,（加法 ADD）	PC→AR=01H, PC+1=02H, AR 指向指令地址
7	11	00C048		90	取指令	RAM(01H)=90H→BUS→IR=90H
8	03	01ED83			间接寻址, AR 指向取数的间接地址	PC→AR=02H, PC+1=03H, RAM=90H
9	04	00E004			取数地址送 AR	RAM(02)=0AH→BUS→AR=0AH
10	05	00B005			从 RAM 中取数送 DR2	RAM(0AH)=D5H→BUS→DR2=D5H
11	06	01A206			将 R0 的数据送 DR1	(R0)=56H→BUS→DR1=56H
12	01	919A01		A0	加法运算: (DR1)+(DR2)→R0	56H+D5H=12BH→R0=2BH, 进位 FC=1
13	02	01ED82	04		执行第 3 条指令（存储 STA）	PC→AR=03H, PC+1=04H
14	12	00C048			取指令	RAM(03H)=20H→BUS→IR=A0H
15	07	01ED87			间接寻址, AR 指向存数的间接地址	PC→AR=04H, PC+1=05H
16	15	00E00D			存数的地址送 AR	RAM(04)=0BH→BUS→AR=0BH
17	01	038201		B0	R0 的内容存入 RAM(0BH)单元	(R0)=2BH→BUS→RAM(0BH)=2BH, 此时 RAM 输出 RAM 0B 单元的原数据 ACH
18	02	01ED82	06		执行第 4 条指令（输出 OUT）	PC→AR=05H, PC+1=06H
19	13	00C048			取指令	RAM(05H)=B0H→BUS→IR=B0H
20	16	01ED8E			间接寻址, AR 指向取数的间接地址	PC→AR=06H, PC+1=07H
21	17	00E00F			取数地址送 AR	RAM(06)=0BH→BUS→AR=0BH
22	25	00A015		C0	从 RAM 中取数送 DR1	RAM(0BH)=2BH→BUS→DR1=2BH
23	01	010A01			DR1 的内容送输出单元 DOUT	DR1=2BH→BUS→DOUT=2BH
24	02	01ED82			执行第 5 条指令（转移 JMP）	PC→AR=07H, PC+1=08H
25	14	00C048			取指令	RAM(07H)=C0H→BUS→IR=C0H
26	26	01ED96			间接寻址, AR 指向转移的间接地址	PC→AR=08H, PC+1=09H
27	01	00D181	00		转移地址送 PC, 转到 00H。	RAM(08H)=00→BUS→PC=00H
28	02	01ED82	01		执行第 1 条指令——程序循环	PC→AR=00H, PC+1=01H
29	10	00C048	01	00	取指令	
...						

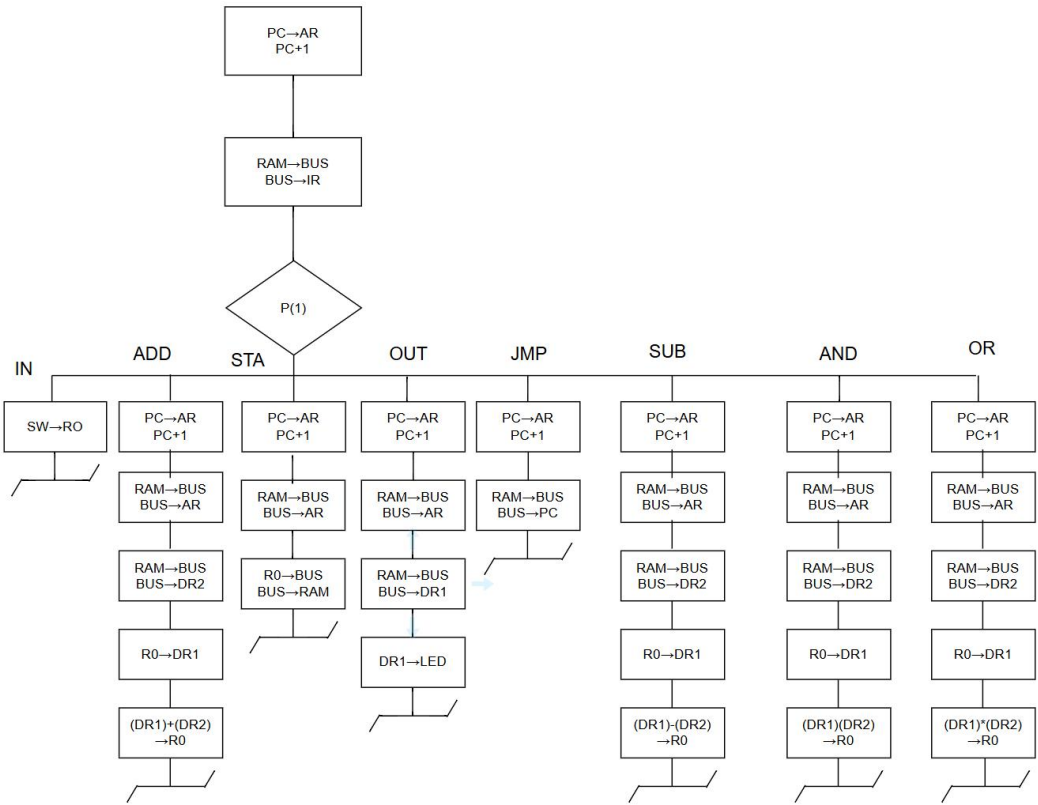
微指令执行情况



In-System Memory Content Editor 从 FPGA 中的 ROM 和 RAM 中读取的波形数据

五、 实验总结及问题分析

思考题，除了已有的五条指令外，再设计减法 SUB，AND，OR 3 条指令，总共 8 条指令，编写响应的微程序流程图，写出微程序代码表。



SUB, AND, OR 基本与 ADD 指令类似，只不过在最后一步微指令为

$(DR1) - (DR2) \rightarrow R0$ ， $(DR1) (DR2) \rightarrow R0$ ， $(DR1) * (DR2) \rightarrow R0$ 。

微程序代码表如下：

微地址	微指令	S3 -S0	M	Cn	WE	A9	A8	A	B	C	UA5 UA0
27	01ED9A	0000	0	0	0	1	1	110	110	110	011010
28	01ED9E	0000	0	0	0	1	1	110	110	110	011110
29	01ED20	0000	0	0	0	1	1	110	110	100	100000
30	00E01B	0000	0	0	0	0	1	110	000	000	011011
31	00B01C	0000	0	0	0	0	1	011	000	000	011100

32	01A01D	0000	0	0	0	1	1	010	000	000	011110
33	616A01	0110	0	0	0	1	0	110	101	000	000001
34	00E01F	0000	0	0	0	0	1	110	000	000	011111
35	00B020	0000	0	0	0	0	1	011	000	000	100000
36	01A01F	0000	0	0	0	1	1	010	000	000	011111
37	B99A01	1011	1	0	0	1	1	001	101	000	000001
38	00E021	0000	0	0	0	0	1	110	000	000	100001
39	00B022	0000	0	0	0	0	1	011	000	000	100010
40	01A023	0000	0	0	0	1	1	010	000	000	100011
41	E99A01	1110	1	0	0	1	1	001	101	000	000001