



青岛大学
QINGDAO UNIVERSITY

计算机科学技术学院

实 验 报 告

2022 - 2023 学 年 第 一 学 期

专 业： 智能科学与技术
班 级： 智能 3 班
学 号： 2019202216
姓 名： 李坤璘
课程名称： 计算机组成原理
指导教师： 尹卓

实验项目名称： 整机实验（一）单累加器

日期： 2022.11.28

实验环境： 仿真软件

实验目的：

利用仿真软件的资源，自行开发出一台可实际运行的计算机。

实验要求：

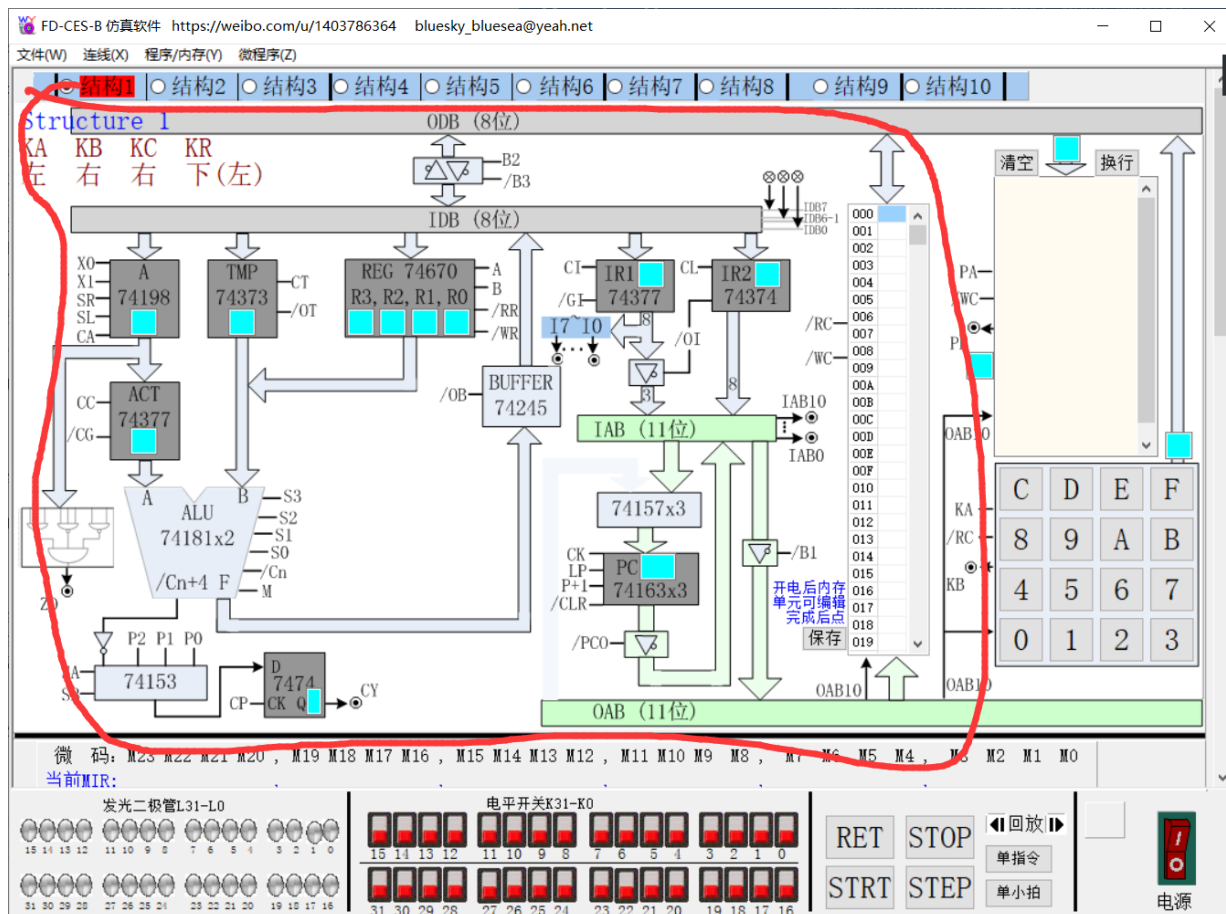
- (1) 运算器:采用单累加器多寄存器结构(开关 KA、KB、KC、KR 分别置左、右、右、下)。
- (2) 指令系统:多于 16 条指令,外设和内存统一编址、统一操作指令。
- (3) 内存寻址方式:立即数寻址、直接寻址、寄存器直接寻址、寄存器间接寻址。
- (4) 微程序设计微操作信号发生器。

实验内容：

一、整机实验的题目要求

- (1) 运算器:采用单累加器多寄存器结构(开关 KA、KB、KC、KR 分别置左、右、右、下)。
- (2) 指令系统:多于 16 条指令,外设和内存统一编址、统一操作指令。
- (3) 内存寻址方式:立即数寻址、直接寻址、寄存器直接寻址、寄存器间接寻址。
- (4) 微程序设计微操作信号发生器。

二、整机的逻辑图



三、指令系统编码和指令执行的过程

| 指令助记符 | 指令功能 | 指令编码 I7I6I5I4I3I2I1 | 节拍 | 微操作 | 控制信号 | 控制信号有效性 |
|---------------|-------------------|------------------------|----|---|--|--|
| 取值微指令 | | | T0 | (PC) → IAB → OAB (M) → ODB → IDB → IR1 (PC) + 1 → PC (MD) → MPC 接数 (A) → ACT | PC0, B1 RC, B2, B3, CI , GI P+1, CK MLD CC, GC | 0, 0 0, 0, 0, ↑, 0 1, ↓ 0 ↑, 0 |
| ADD A, Ri | (A) + (Ri) → A | 000000Ri | T1 | (ACT) + (Ri) → BUF → IDB → A 置 CY | RR, A, B Cn', M, S3, S 2, S1, S0 OB, X0, X1, CA SA, SB, CP | 0, I1, IO 1, 0, 1, 0, 0, 1 0, 1, 1, ↑ 1, 1, ↑ |
| | | | | | | |
| ADD A, Ri | (A) - (Ri) → A | 000001Ri | T1 | (ACT) - (Ri) → BUF → IDB → A 置 CY | RR, A, B Cn', M, S3, S 2, S1, S0 OB, X0, X1, CA SA, SB, CP | 0, I1, IO 0, 0, 0, 1, 1, 0 0, 1, 1, ↑ 1, 1, ↑ |
| MOV A, @Ri | (I2I1I0(Ri)) → A | 001000Ri | T1 | (Ri) → BUF → IDB → IR2 | RR, A, B, OB, C L Cn', M, S3, S 2, S1, S0 | 0, I1, IO, 0, ↑ 1, 1, 1, 0, 1, 0 |
| | | | T2 | (IR1 IR2) → IAB → OAB (M) → ODB → IDB → A | OI, B1 RC, B2, B3, X0, X1 , CA | 0, 0, 0, 0, 0, 1, 1, ↑ |
| MOV A, Ri | (Ri) → A | 010000Ri | T1 | (Ri) → BUF → IDB → A | RR, A, B Cn', M, S3, S 2, S1, S0 OB, X0, X1, CA | 0, I1, IO 1, 1, 1, 0, 1, 0 0, 1, 1, ↑ |
| MOV Ri, A | A → (Ri) | 010001Ri | T1 | (ACT) → BUF → IDB → Ri | Cn', M, S3, S 2, S1, S0 OB, WR, A, B | 1, 0, 0, 0, 0, 0 0, 0, I1, IO |
| MOV A, #data | data → A | 011000001, data | T1 | (PC) → IAB → OAB (M) → ODB → IDB → A (PC) + 1 → PC | PC0, B1 RC, B2, B3, X0 , X1, CA P+1, CK | 0, 0, 0, 0, 0, 1, 1, ↑ 1, ↓ |
| MOV Ri, #data | data → Ri | 011001 Ri, data | T1 | (PC) → IAB → OAB (M) → ODB → | PC0, B1 RC, B2, B3, WR , A, B | 0, 0, 0, 0, 0, I1, IO |

| | | | | | | |
|--|--|---|----|---|--|---------------------------------|
| | | | | IDB→Ri (PC)+1→PC | P+1, CK | 1, ↓ |
| LD A addr | (addr)→A | 1000 a10a9a8, a7~a0 | T1 | (PC)→IAB →OAB (M)→ODB→ IDB→IR2 (PC)+1→PC | PC0, B1 RC, B2, B3, CL P+1, CK | 0, 0, 0, 0, 0, ↑ 1, ↓ |
| | | | T2 | (IR1 IR2) →IAB→OAB (M)→ODB→ IDB→A | OI, B1 RC, B2, B3, X0, X1 , CA | 0, 0, 0, 0, 0, 1, 1, ↑ |
| STA addr | A→(addr) | 10100 a10 a9 a8, a7~a0 | T1 | (PC)→IAB →OAB (M)→ODB→ IDB→IR2 (PC)+1→PC | PC0, B1 RC, B2, B3, CL P+1, CK | 0, 0, 0, 0, 0, ↑ 1, ↓ |
| | | | T2 | (IR1 IR2) →IAB→OAB (M)→ODB→ IDB→IR2 (PC)+1→PC | OI, B1 RC, B2, B3, CL P+1, CK | 0, 0, 0, 0, 0 1, ↓ |
| RLC A | C, A 左环移 一位 | 11000000 | T1 | CY, (A) 左环 移一位 | X0, X1, CA SA, SB, CP | 0, 1, ↑ 0, 1, ↑ |
| RRC A | C, A 右环移 一位 | 11000100 | T1 | CY, (A) 右 环移一位 | X0, X1, CA SA, SB, CP | 1, 0, ↑ 1, 0, ↑ |
| JZ addr JC addr JA0 addr JMP addr | addr $\xrightarrow{\text{条件}}$ PC | 111000 a9 a8 a7~a0 111010 a9 a8 a7~a0 11110 0 a9 a8 a7~a0 111110 a9 a8 a7~a0 | T1 | (PC)→IAB →OAB (M)→ODB→ IDB→IR2 (PC)+1→PC | PC0, B1 RC, B2, B3, CL P+1, CK | 0, 0, 0, 0, 0, ↑ 1, ↓ |
| | | | T2 | (IR1 IR2) →IAB IAB 条件成立 →PC | OI LP | 0 1 |
| HALT | 置 0 RUN | 11111111 | T1 | 置 0 RUN | DR, RCP | 0, ↑ |

四、微指令格式和指令微程序

24 位微码对应关系：M23-M0（24 位微指令码）

| | M23 | M22 | M21 | M20 | M19 | M18 | M17 | M16 | M15 | M14 | M13 | M12 |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 有效电平 | * | * | * | * | * | * | * | * | 0 | 1 | 1 | * |
| | S3 | S2 | S1 | S0 | /Cn | M | X1 | X0 | OI | CL' | CP' | / |
| | M11 | M10 | M9 | M8 | M7 | M6 | M5 | M4 | M3 | M2 | M1 | M0 |
| 有效电平 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| | CG | OT | LP' | OB | GI | P+1 | DR | MLD | WC' | RC' | RR' | WR' |

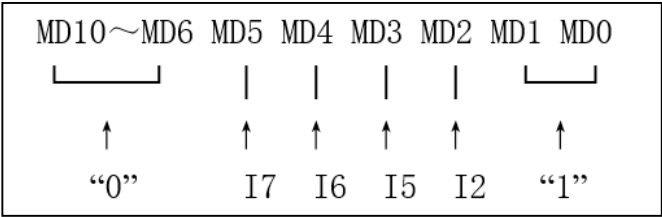
指令微程序：

| 指令助记符 | 位 | 23 22 21 20 | 19 18 17 16 | 15 14 13 12 | 11 10 9 8 | 7 6 5 4 | 3 2 1 0 | 伪指令码 （十六进制） |
|------------|------|----------------|----------------|-----------------|-----------------|------------------|-------------------|----------------|
| | 信号 | S3 S2 S1 S0 | Cn' M X1 X0 | OI CL' CP' / | CG OT LP' OB | GI P+1 DR MLD | WC' RC' RR WR' | |
| | 有效电平 | **** | **** | 011* | 0000 | 0100 | 0001 | |
| | 微地址 | | | | | | | |
| 取值伪指令 | 000H | 0000 | 0000 | 1000 | 0111 | 0110 | 1010 | 00876A |
| | 001 | | | | | | | |
| | 002 | | | | | | | |
| ADD A, Ri | 003 | 1001 | 1011 | 1010 | 1110 | 1011 | 1100 | 9BAEBC |
| | 004 | 0000 | 0000 | 1000 | 0111 | 0110 | 1010 | 00876A |
| | 005 | | | | | | | |
| | 006 | | | | | | | |
| SUB A, Ri | 007 | 0110 | 0011 | 1000 | 1110 | 1011 | 1100 | 638EBC |
| | 008 | 0000 | 0000 | 1000 | 0111 | 0110 | 1010 | 00876A |
| | 009 | | | | | | | |
| | 00A | | | | | | | |
| MOV A, @Ri | 00B | 1010 | 1100 | 1100 | 1110 | 1011 | 1100 | ACCEBC |
| | 00C | 0000 | 0011 | 0000 | 1111 | 1011 | 1010 | 030FBA |
| | 00D | 0000 | 0000 | 1000 | 0111 | 0110 | 1010 | 00876A |
| | 00E | | | | | | | |
| | 00F | | | | | | | |
| | 010 | | | | | | | |
| | 011 | | | | | | | |
| | 012 | | | | | | | |
| MOV A, Ri | 013 | 1010 | 1111 | 1000 | 1110 | 1011 | 1100 | AF8EBC |
| | 014 | 0000 | 0000 | 1000 | 0111 | 0110 | 1010 | 00876A |
| | 015 | | | | | | | |
| | 016 | | | | | | | |
| MOV Ri, A | 017 | 0000 | 1000 | 1000 | 1110 | 1011 | 1111 | 088EBF |
| | 018 | 0000 | 0000 | 1000 | 0111 | 0110 | 1010 | 00876A |

| | | | | | | | | |
|------------------|-----|------|------|------|------|------|------|--------|
| | 019 | | | | | | | |
| | 01A | | | | | | | |
| MOV A, #data | 01B | 0000 | 0011 | 1000 | 1111 | 1111 | 1010 | 038FFA |
| | 01C | 0000 | 0000 | 1000 | 0111 | 0110 | 1010 | 00876A |
| | 01D | | | | | | | |
| | 01E | | | | | | | |
| NOV Ri, #data | 01F | 0000 | 0000 | 1000 | 1111 | 1111 | 1011 | 038FFA |
| | 020 | 0000 | 0000 | 1000 | 0111 | 0110 | 1010 | 00876A |
| | 021 | | | | | | | |
| | 022 | | | | | | | |
| LDA addr | 023 | 0000 | 0000 | 1100 | 1111 | 1111 | 1010 | 00CFEA |
| | 024 | 0000 | 0011 | 0000 | 1111 | 1011 | 1010 | 030FBA |
| | 025 | 0000 | 0000 | 1000 | 0111 | 0110 | 1010 | 00876A |
| | 026 | | | | | | | |
| LDA addr | 027 | 0000 | 0000 | 1100 | 1111 | 1111 | 1010 | 00CFEA |
| | 028 | 0000 | 0011 | 0000 | 1111 | 1011 | 1010 | 030FBA |
| | 029 | 0000 | 0000 | 1000 | 0111 | 0110 | 1010 | 00876A |
| | 02A | | | | | | | |
| STA addr | 02B | 0000 | 0000 | 1100 | 1111 | 1111 | 1010 | 00CFFA |
| | 02C | 0000 | 1000 | 0000 | 1110 | 1011 | 0110 | 080EB6 |
| | 02D | 0000 | 0000 | 1000 | 0111 | 0110 | 1010 | 00876A |
| | 02E | | | | | | | |
| STA addr | 02F | 0000 | 0000 | 1100 | 1111 | 1111 | 1010 | 00CFFA |
| | 030 | 0000 | 1000 | 0000 | 1110 | 1011 | 0110 | 080EB6 |
| | 031 | 0000 | 0000 | 1000 | 0111 | 0110 | 1010 | 00876A |
| | 032 | | | | | | | |
| RLC A | 033 | 0000 | 0010 | 1010 | 1111 | 1011 | 1110 | 02AFBE |
| | 034 | 0000 | 0000 | 1000 | 0111 | 0110 | 1010 | 00876A |
| | 035 | | | | | | | |
| | 036 | | | | | | | |
| RRC A | 037 | 0000 | 0001 | 1010 | 1111 | 1011 | 1110 | 01AFBE |
| | 038 | 0000 | 0000 | 1000 | 0111 | 0110 | 1010 | 00876A |
| | 039 | | | | | | | |
| | 03A | | | | | | | |
| JZ addr | 03B | 0000 | 0000 | 1100 | 1111 | 1111 | 1010 | 00CFFA |
| JC addr | 03C | 0000 | 0000 | 0000 | 1101 | 1011 | 1110 | 000DBE |
| JA0 addr | 03D | 0000 | 0000 | 1000 | 0111 | 0110 | 1010 | 00876A |
| JMP addr | 03E | | | | | | | |
| HALT | 03F | 0000 | 0000 | 1000 | 1111 | 1001 | 1110 | 008F9E |
| | 040 | 0000 | 0000 | 1000 | 0111 | 0110 | 1010 | 00876A |

五、确定微程序入口地址形成方法

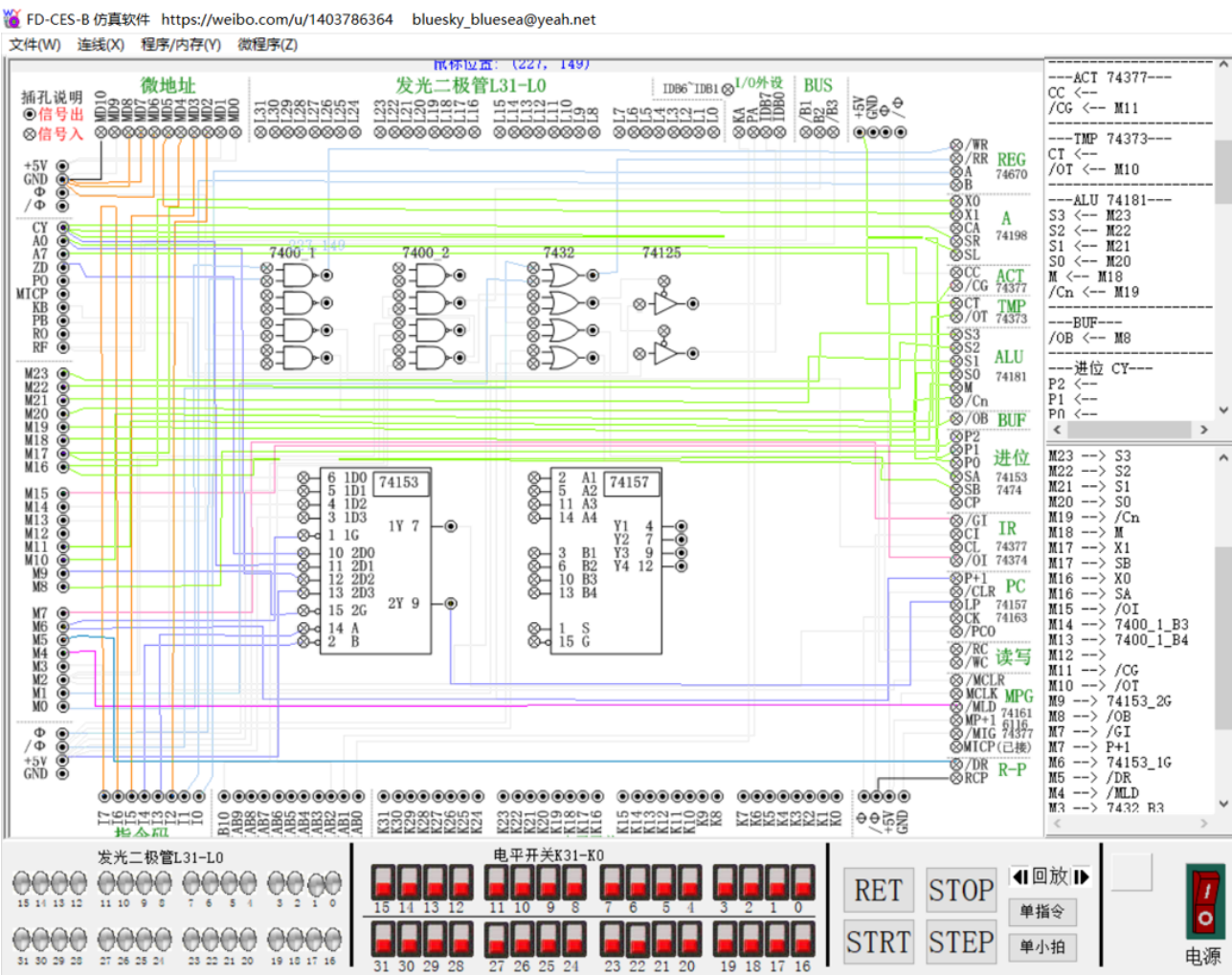
以指令操作码（8 位）为核心，扩展成 11 位的微程序地址即 MD10~MD0. 如果指令系统仅 8 条指令，IR1 的 I7. I6. I5 为指令操作码，每条指令执行不超过 4 拍，即微指令可由多至 4 条微指令构成：



全部微程序首地址如下：

| 指令操作码 I7I6I5 | 微程序首址 MD10 ~ MD0 |
|-----------------|---------------------|
| 0 0 0 | 003H |
| 0 0 1 | 007H |
| 0 1 0 | 00BH |
| 0 1 1 | 00FH |
| 1 0 0 | 013H |
| 1 0 1 | 017H |
| 1 1 0 | 01BH |
| 1 1 1 | 01FH |

六、.ABL 内容



右侧对连线具体情况进行了描述，便于进行快速的引脚定位，所有内容如下：

---I/O 外设---

KA <-- IAB0

PA <-- IAB1

IDB7 <-- 74125_Y2

IDB0 <-- 74125_Y1

IDB6~IDB1 <--

---BUS---

/B1 <-- RF

B2 <-- 7400_2_Y2

/B3 <-- RF

---REG 74670---

/WR <-- 7400_1_Y2

/RR <-- 7432_Y1

A <-- I0

B <-- I1

---A 74198---

X0 <-- M16

X1 <-- M17

CA <-- /φ

SR <-- CY

SL <-- CY

---ACT 74377---

CC <-- /φ

/CG <-- M11

---TMP 74373---

CT <-- +5V

/OT <-- M10

---ALU 74181---

S3 <-- M23

S2 <-- M22

S1 <-- M21

S0 <-- M20

M <-- M18

/Cn <-- M19

---BUF---

/OB <-- M8

---进位 CY---

P2 <-- A7

P1 <-- A0

P0 <-- CY

SA <-- M16

SB <-- M17

CP <-- 7400_1_Y4

---IR1 IR2---

/GI <-- M7

CI <-- /φ

CL <-- 7400_1_Y3

/OI <-- M15

---PC 74163---

P+1 <-- M6

/CLR <-- +5V

LP <-- 74153_2Y

CK <-- φ

/PCO <-- 74153_1Y

---读写---

/RC <-- 7432_Y4

/WC <-- 7432_Y3

---MPG---

/MCLR <-- RO

MCLK <-- PO

/MLD <-- M4

MP+1 <-- +5V

/MIG <-- GND

---R-P 启停---

/DR <-- M5

---微程序地址---

MD9 <-- GND

MD8 <-- GND

MD7 <-- GND

MD6 <-- GND

MD5 <-- I7

MD4 <-- I6

MD3 <-- I5

MD2 <-- I2

MD1 <-- +5V

MD0 <-- +5V

---LED 灯---

---7400_1---

7400_1_A2 <-- φ

7400_1_B2 <-- M0

7400_1_A3 <-- φ

7400_1_B3 <-- M14

7400_1_A4 <-- φ

7400_1_B4 <-- M13

---7400_2---

7400_2_A2 <-- 7400_2_Y3

7400_2_B2 <-- 7400_2_Y4

7400_2_A3 <-- M2

7400_2_B3 <-- M2

7400_2_A4 <-- IAB10

7400_2_B4 <-- IAB2

---7432---

7432_A1 <-- φ

7432_B1 <-- M1

7432_A2 <-- 7400_2_Y4

7432_B2 <-- M2

7432_A3 <-- φ

7432_B3 <-- M3

7432_A4 <-- φ

7432_B4 <-- M2

---74125---

74125_A1 <-- KB

74125_C1 <-- 7432_Y2

74125_A2 <-- PB

74125_C2 <-- 7432_Y2

---74153---

74153_1D0 <-- +5V

74153_1D1 <-- +5V

74153_1D2 <-- +5V

74153_1D3 <-- +5V

74153_1G <-- M6

74153_2D0 <-- ZD

74153_2D1 <-- CY

74153_2D2 <-- A0

74153_2D3 <-- +5V

74153_2G <-- M9

74153_A <-- I3

74153_B <-- I4

---74157---

CY --> SR

CY --> SL

CY --> P0

CY --> 74153_2D1

A0 --> P1

A0 --> 74153_2D2

A7 --> P2

ZD --> 74153_2D0

PO --> MCLK

MICP -->

KB --> 74125_A1

PB --> 74125_A2

RO --> /MCLR

RF --> /B1

RF --> /B3

M23 --> S3

M22 --> S2

M21 --> S1

M20 --> S0

M19 --> /Cn

M18 --> M

M17 --> X1

M17 --> SB

M16 --> X0

M16 --> SA

M15 --> /OI

M14 --> 7400_1_B3

M13 --> 7400_1_B4

M12 -->

M11 --> /CG

M10 --> /OT

M9 --> 74153_2G

M8 --> /OB

M7 --> /GI

M6 --> P+1

M6 --> 74153_1G

M5 --> /DR

M4 --> /MLD

M3 --> 7432_B3

M2 --> 7400_2_A3

M2 --> 7400_2_B3

M2 --> 7432_B2

M2 --> 7432_B4

M1 --> 7432_B1

M0 --> 7400_1_B2

I7 --> MD5

I6 --> MD4

I5 --> MD3

I4 --> 74153_B

I3 --> 74153_A

I2 --> MD2

I1 --> B

I0 --> A

IAB10 --> 7400_2_A4

IAB2 --> 7400_2_B4

IAB1 --> PA

IAB0 --> KA

ϕ --> 7400_1_A2

ϕ --> 7400_1_A3

ϕ --> 7400_1_A4

ϕ --> CK

ϕ --> 7432_A1

ϕ --> 7432_A3

ϕ --> 7432_A4

ϕ --> CI

ϕ --> CA

ϕ --> CC

+5V --> MD1

+5V --> MD0

+5V --> 74153_1D0

+5V --> 74153_1D1

+5V --> 74153_1D2

+5V --> 74153_1D3

+5V --> 74153_2D3

+5V --> /CLR
+5V --> MP+1
+5V --> CT
GND --> MD9
GND --> MD8
GND --> MD7
GND --> MD6
GND --> /MIG
7400_1_Y2 --> /WR
7400_1_Y3 --> CL
7400_1_Y4 --> CP
7400_2_Y2 --> B2
7400_2_Y3 --> 7400_2_A2
7400_2_Y4 --> 7400_2_B2
7400_2_Y4 --> 7432_A2
7432_Y1 --> /RR
7432_Y2 --> 74125_C1
7432_Y2 --> 74125_C2
7432_Y3 --> /WC
7432_Y4 --> /RC
74125_Y1 --> IDB0
74125_Y2 --> IDB7
74153_1Y --> /PCO
74153_2Y --> LP

七、BIT.DEF 内容

S3 S2 S1 S0 /CN M X1 X0 OI CL CP NONE

CG OT LP OB GI P+1 DR MLD WC RC RR WR

八、INS.DEF 内容

?2

-HALT

1

11111111

-ADD

4

A,Ri

000000Ri

-SUB

4

A,Ri

000001Ri

-MOV

5

A,@RI

001000RI

-MOV

4

A,RI

010000RI

-MOV

4

RI,A

010001RI

-MOV

2

A,#DATA

01100000

DATA

-MOV

2

RI,#DATA

011001RI

DATA

-LDA

3

ADDR

10000AAA

A7~A0

-STA

3

ADDR

10100AAA

A7~A0

-RLC

4

A

11000000

-RRC

4

A

11000100

-JZ

3

ADDR

111000AA

A7~A0

-JC

3

ADDR

111010AA

A7~A0

-JAO

3

ADDR

111100AA

A7~A0

-JMP

3

ADDR

111110AA

A7~A0

九、MOP.MID 内容

详细文本如下：

;000H~002H

-00876A

-

-

;ADD A,RI

;003H~006H

-9BAEBC

-00876A

-

-

;SUB A,RI

;007H~00AH

-638EBC

-00876A

-

-

;MOV A,@RI

;00BH~012H

-ACCEBC

-030FBA

-000876A

-

-

-

-

-

;MOV A,RI

;013H~016H

-AF8EBC

-00876A

-

-

;MOV RI,A

;017H~01AH

-088EBF

-00876A

-

-

;MOV A #DATA

;01BH~01EH

-038FFA

-00876A

-

-

;MOV RI,#DATA

;01FH~022H

-008FFB

-00876A

-

-

;LDA ADDR

;023H~026H

-00CFFA

-030FBA

-00876A

-

;LDA ADDR

;027H~02AH

-00CFFA

-030FBA

-00876A

-

;STA ADDR

;02BH~02EH

-00CFFA

-080EB6

-00876A

-

;STA ADDR

;02FH~032H

-00CFFA

-080EB6

-00876A

-

;RLC A

;033H~036H

-02AFBE

-00876A

-

-

;RRC A

;037H~03AH

-01AFBE

-00876A

-

-

;JZ ADDR

-00CFFA

;JC ADDR

-000DBE

;JAO ADDR

-00876A

;JMP ADDR

-

;HALT

;03FH~040H

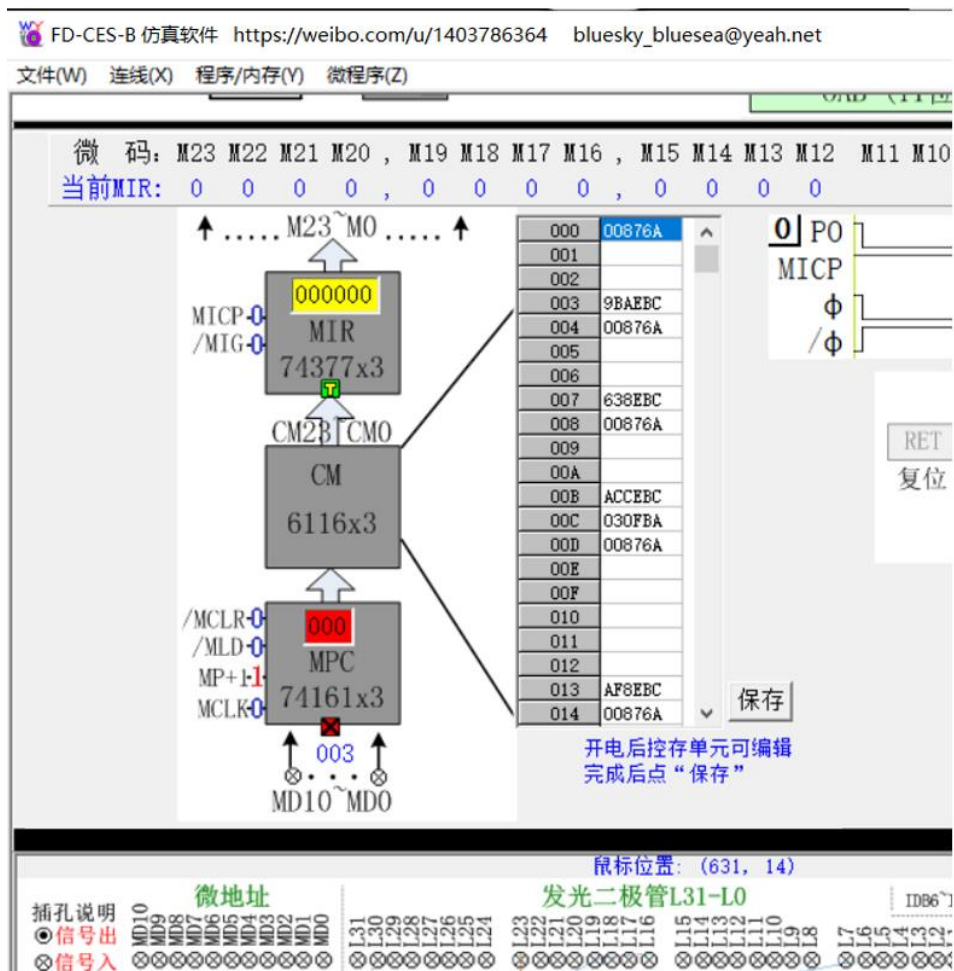
-008F9E

-00876A

十、调试单条指令过程

调试基本步骤如下：

①读取 mop 文件到控存



②编写.asm 文件（以调试存取及停机指令为例）

ORG 005

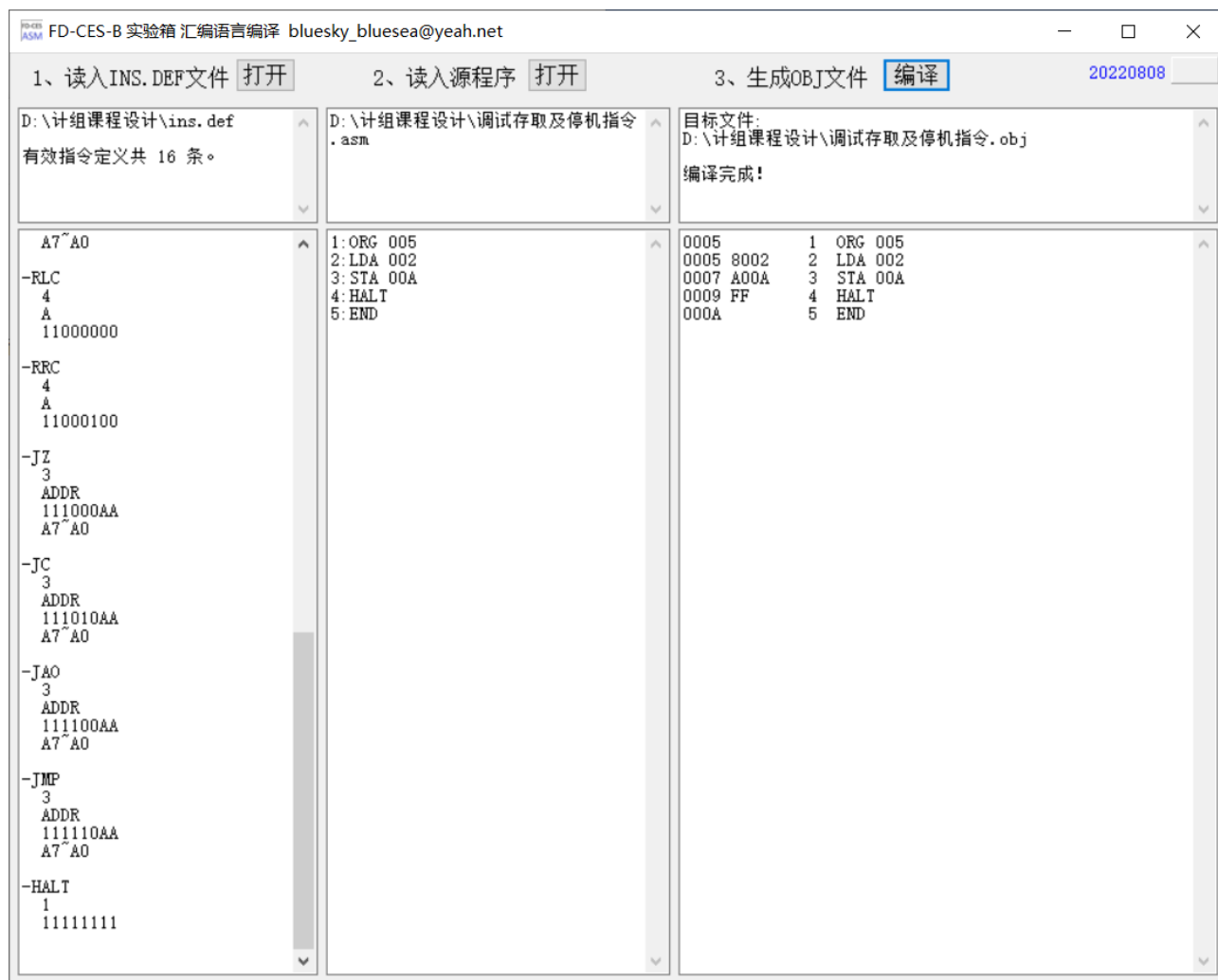
LDA 002

STA 00A

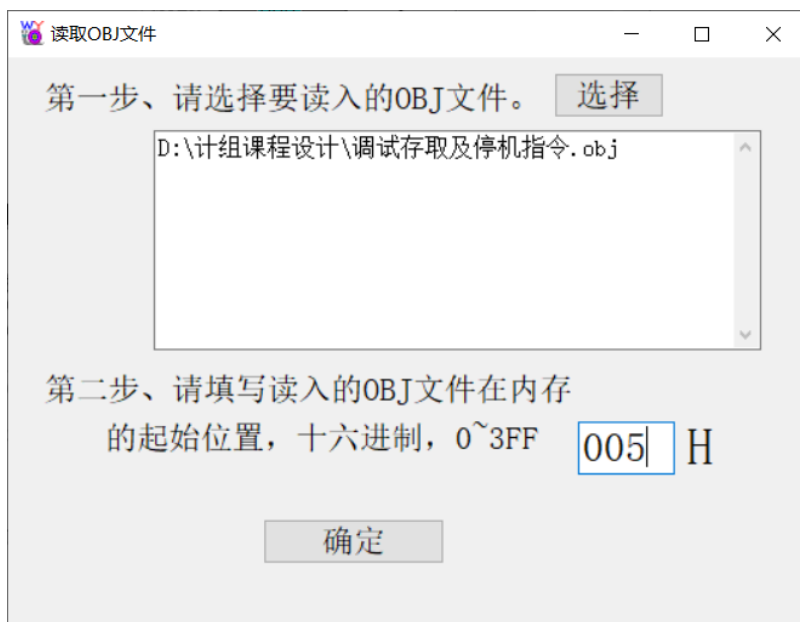
HALT

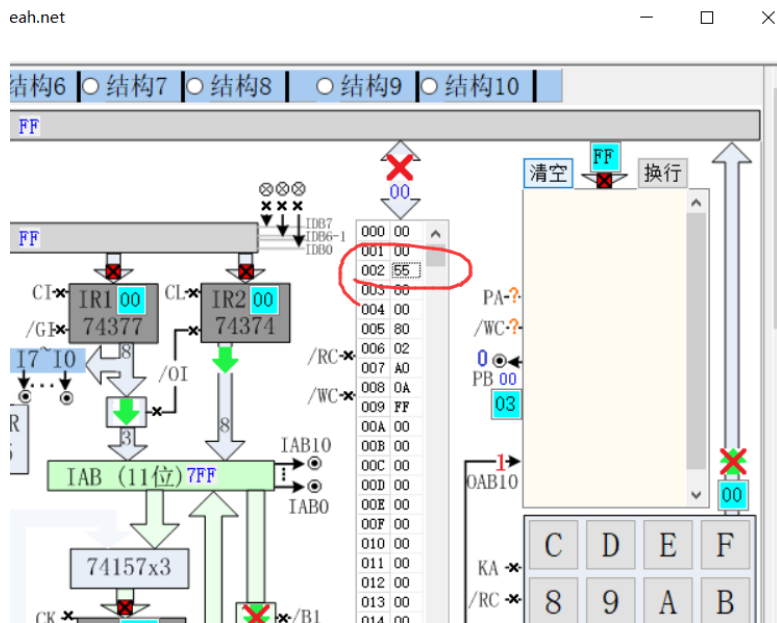
END

③.asm 文件导入编译工具，生成.obj 文件



④读入.obj 文件，随后运行





1.调试存取及停机指令：

先在 002 内存单元中写入 55，然后跳到 005 地址下单步运行程序，过程显示如下：

OAB ODB

005 80

006 02

002 55

007 A0

008 0A

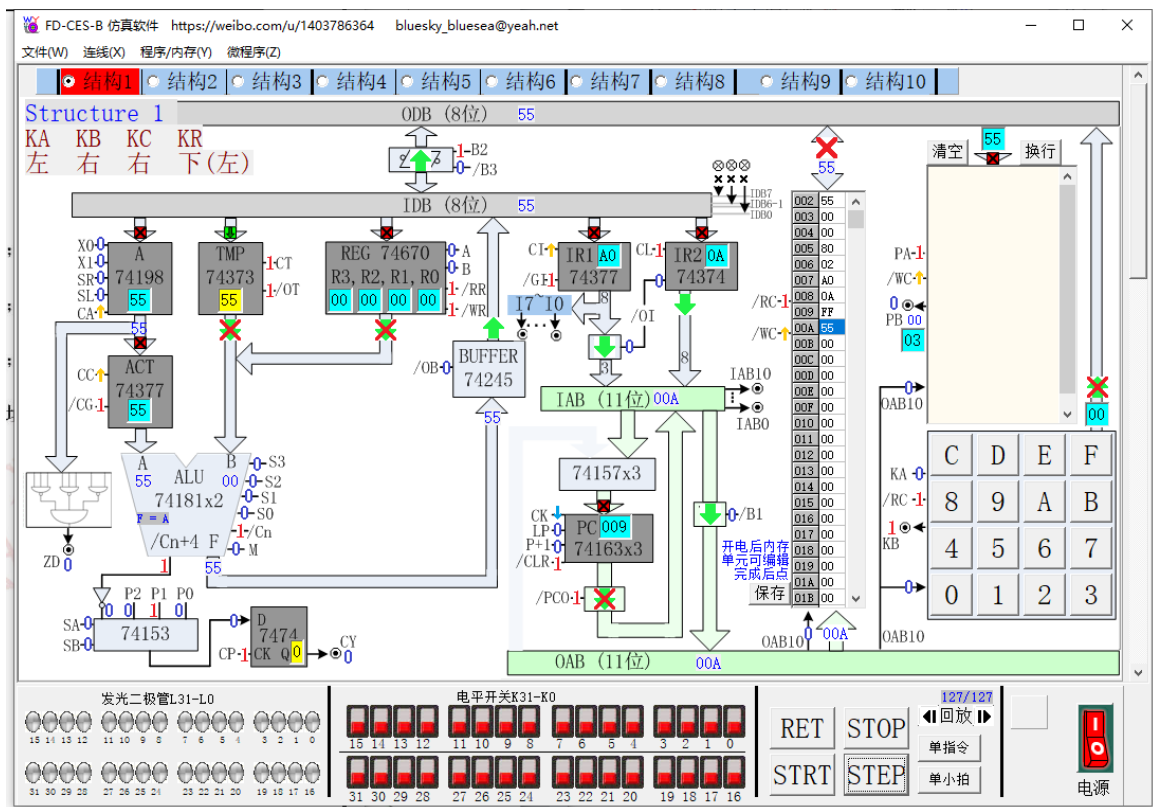
00A 55

009 FF

7FF FF

| | |
|-----|----|
| 000 | 00 |
| 001 | 00 |
| 002 | 55 |
| 003 | 00 |
| 004 | 00 |
| 005 | 80 |
| 006 | 02 |
| 007 | A0 |
| 008 | 0A |
| 009 | FF |
| 00A | 55 |
| 00B | 00 |

结果是(00A)=55



2.调试加法右移指令

先在 002 单元中写入 55，在 003 单元中写入 A9，到 00B 起始地址单拍运行如下：

OAB ODB

00B 80

00C 02

002 55 取出 002 单元中的 55

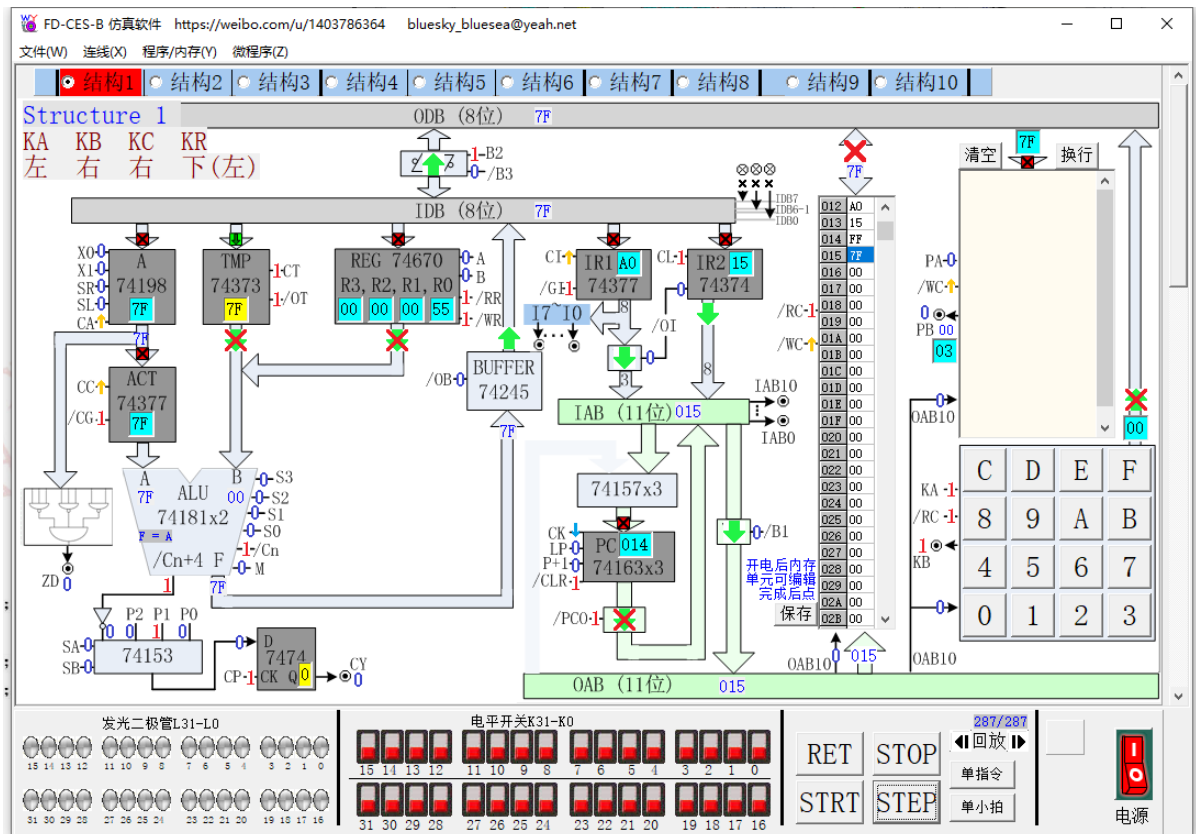
00D 44

7FF 55

00E 80
 00F 03
 003 A9 取出 003 单元中的 A9
 010 00
 7FF FE 加法的和为 FE
 011 C4 右移
 7FF FF
 012 A0
 013 15
 015 7F 015 内存中为 7F
 014 FF
 7FF FF

| | |
|-----|----|
| 000 | 00 |
| 001 | 00 |
| 002 | 55 |
| 003 | A9 |
| 004 | 00 |
| 005 | 80 |
| 006 | 02 |
| 007 | A0 |
| 008 | 0A |
| 009 | FF |
| 00A | 55 |
| 00B | 80 |
| 00C | 02 |
| 00D | 44 |
| 00E | 80 |
| 00F | 03 |
| 010 | 00 |
| 011 | C4 |
| 012 | A0 |
| 013 | 15 |
| 014 | FF |
| 015 | 7F |
| 016 | 00 |

结果是(015)=7F



3.调试减法左移指令

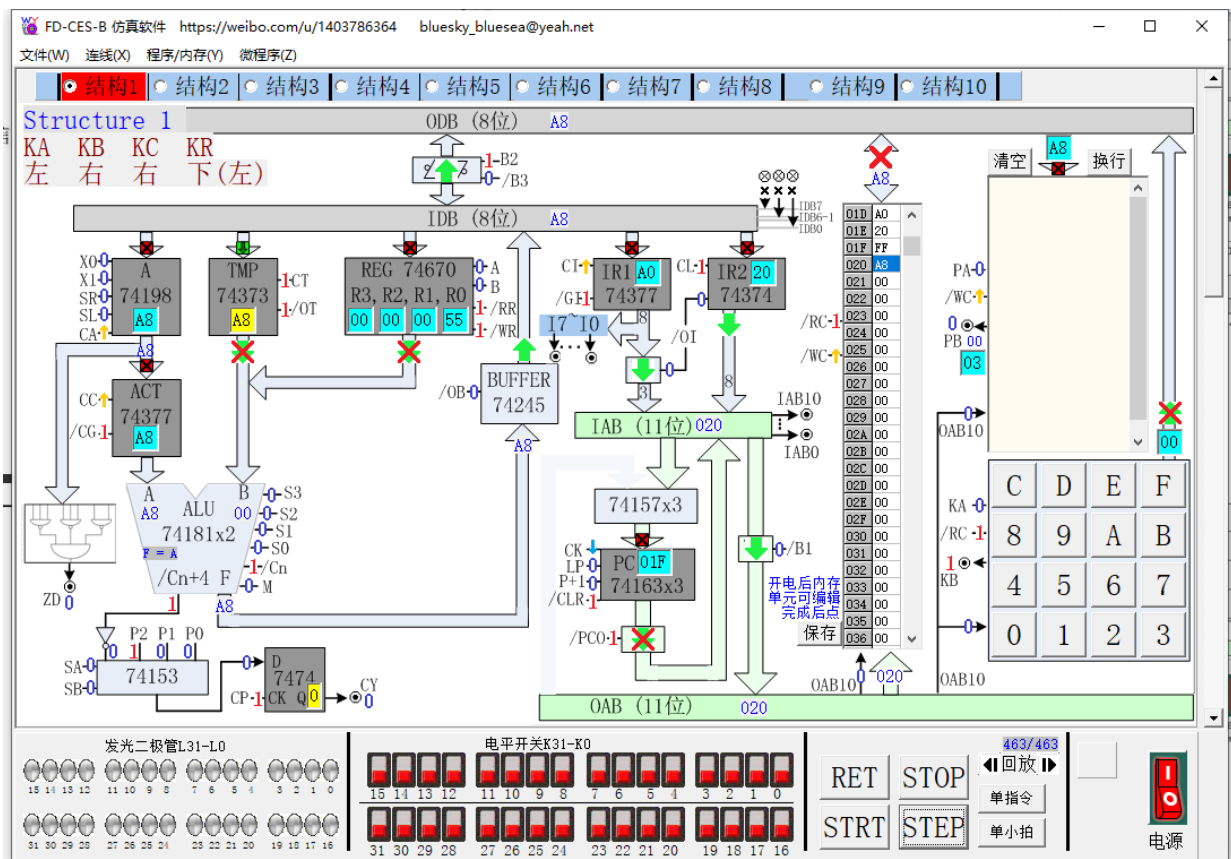
直接调至入口地址 016，单步执行程序如下：

OAB ODB
 016 80

| | | |
|-----|----|-------------|
| 017 | 02 | |
| 002 | 55 | |
| 018 | 44 | |
| 7FF | 55 | |
| 019 | 80 | |
| 01A | 03 | |
| 003 | A9 | |
| 01B | 04 | |
| 7FF | 54 | A9-55=54 |
| 01C | C0 | 左移 |
| 7FF | FF | |
| 01D | A0 | |
| 01E | 20 | |
| 020 | A8 | 存入 020 为 A8 |
| 01F | FF | |
| 7FF | FF | |

结果为(020)=A8

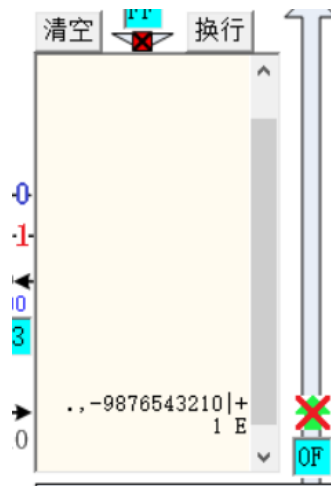
| | |
|-----|----|
| 01D | A0 |
| 01E | 20 |
| 01F | FF |
| 020 | A8 |



4.程序跳转指令

单步调试如下：

| OAB | ODB | |
|-----|-----|-----------------------------------|
| 030 | 84 | |
| 031 | 04 | ;此时先键入 0 来表示 KB 为 0，而后键入一个字符，然后继续 |
| 404 | FF | |
| 032 | F0 | |
| 033 | 30 | |
| 030 | FF | |
| 034 | 84 | |
| 035 | 01 | |
| 401 | 00 | |
| 036 | A4 | |
| 037 | 02 | |
| 402 | 00 | ;打印机会转一下，打印刚才键入的一个字符 |
| 038 | 84 | |
| 039 | 94 | |
| 404 | FF | |
| 03A | 44 | |
| 7FF | 7F | |
| 03B | 00 | |
| 7FF | 7F | |
| 03C | E8 | |
| 03D | 38 | |
| 038 | FF | |
| 03E | F8 | |
| 03F | 30 | |
| 030 | FF | |
| 030 | 84 | ;返回程序首执行 |



| 编 码 | 00H | 01H | 02H | 03H | 04H | 05H | 06H | 07H | 08H | 09H | 0AH | 0BH | 0CH | 0DH | 0EH | 0FH |
|-----------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|-----|-----|-----|--------|---------------|
| 左 轮 字 符 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | - | , | . | # | 0 | 1 加 换 行 |
| 中 间 轮 字 符 | G | K | - | P | D | ? | M | d | + | - | 空 格 | R | | A | 空 格 | 换 行 |
| 右 轮 字 符 | + | × | ÷ | <> | * | S | T | M | C | = | 空 格 | - | % | E | + | x 加 换 行 |

十一、应用程序.ASM 内容

1、调试存取及停机指令

ORG 005

LDA 002

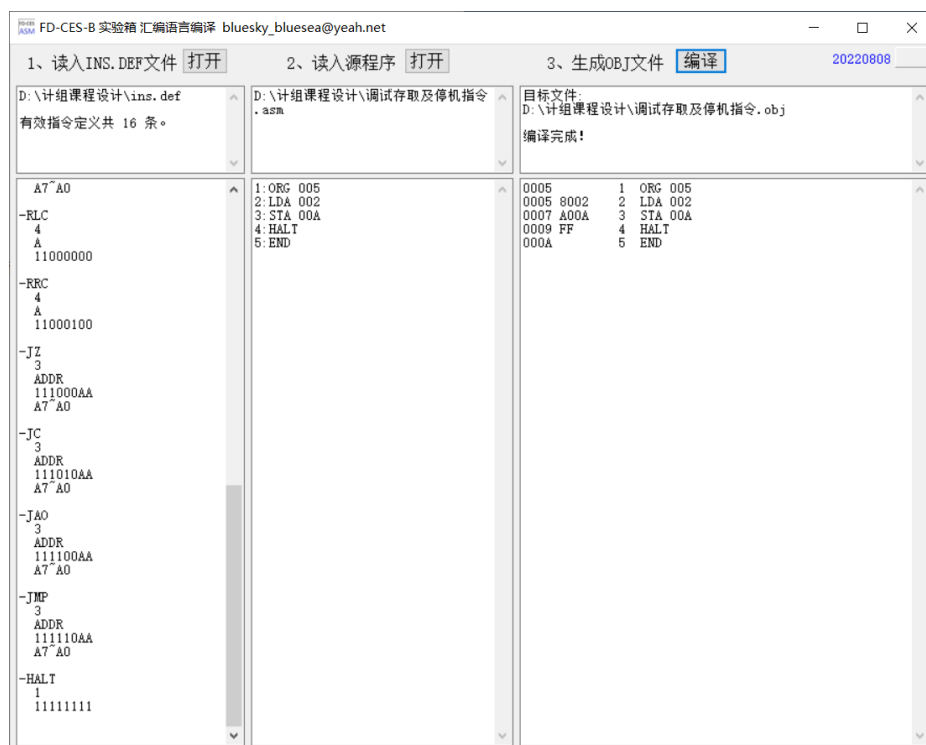
STA 00A

HALT

END

;功能：测试取数与存数及停机指令

;结果应为（00A）=55 (002 单元的内容测试前载入 55H)



2、调试加法右移指令

ORG 00B

LDA 002

MOV R0,A

LDA 003

ADD A,R0

RRC A

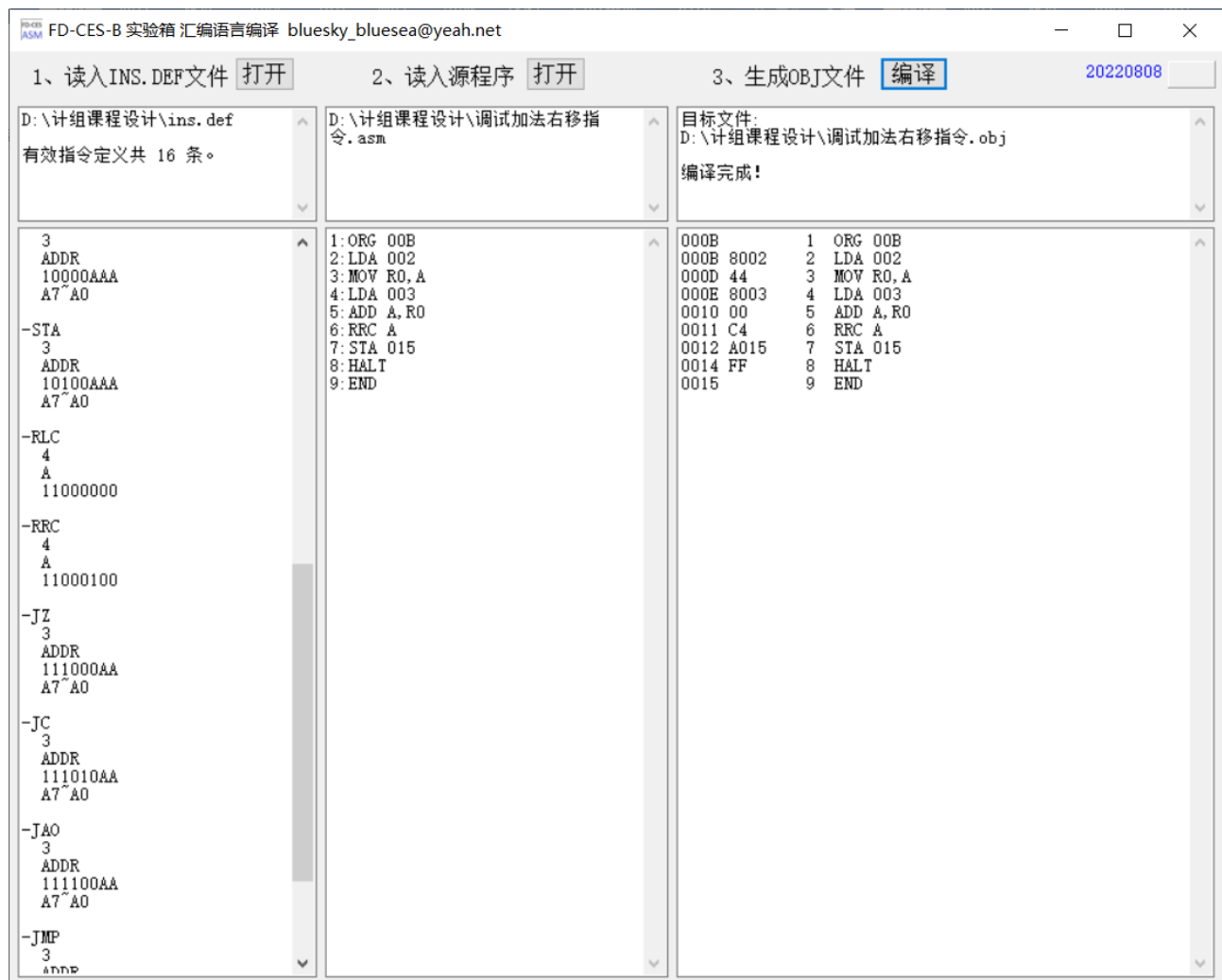
STA 015

HALT

END

;功能：测试 SUB、RLC 指令

;结果应为 (020)=A8 (002 单元的内容测试前载入 55H,003 单元的内容测试前载入 A9H)



3、调试减法左移指令

ORG 016

LDA 002

MOV R0,A

LDA 003

SUB A,R0

RLC A

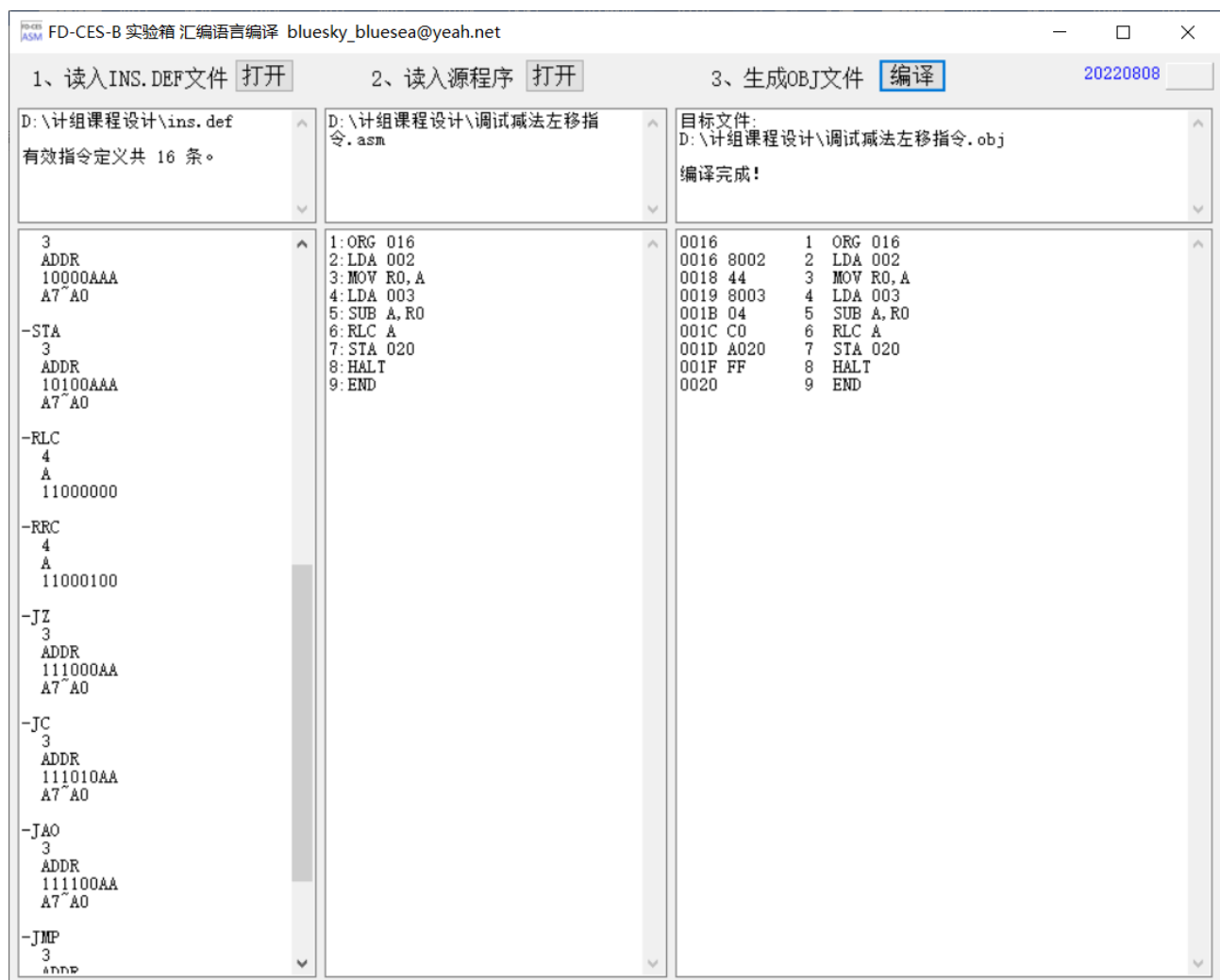
STA 020

HALT

END

;功能：测试 MOV、ADD、RRC 指令

;结果应为 (020)=A8 (002 单元的内容测试前载入 55H,003 单元的内容测试前载入 A9H)



4、程序调转指令

ORG 030

LDA 404 ;读询问口

JAO 030 ;KB=0? 若 KB!=0 则跳到 030 继续读询问口

LDA 401 ;KB=0, 读键值

STA 402 ;402 为打印机端口, 打印该字符

LDA 404 ;读询问口

MOV R0,A ;读询问口后, A 中存储 PB 的状态, 将 A 中的值送到 R0

ADD A,R0 ;(A)+(R0)->A 若 PB=1 则 1+1=0 CY=1 若 PB=0 则 0+0=0 CY=0

JC 038 ;若 CY=1, 则跳到 038 继续读询问口

JMP 030 ;返回程序起始位置继续运行

END

;功能: 测试键盘读入及打印机输出

;结果应为打印机能打印出从键盘读入的字符



十二、实验的步骤与遇到的问题及解决方法

1. 控存文件一开始没写对，导致在做实验时单步调试总会出错，实验结果与预想值有偏差，通过一步步核验最后发现问题及时更改控存文件。
2. 连线时未看清连线文件说明，导致有几条线连接错误造成实验结果出错，及时与同学沟通交流发现问题所在，重新更改连线。
3. 打印机打印字符时一开始对编码 0FH 的左轮右轮中间轮的字符没搞明白，后与同学交流以及自己的不断尝试搞明白。

十三、思考及体会

- ① 完成整机一实验，我对计算机的各个模块部件之间的关系理解的更加清晰，对上学期的组成原理中的指令系统及微程序的相关知识也有了更深的认识
- ② 通过编写不同的文件了解其在整机实验中的功能，也学会了如何使用打印机来进行打印字符，能区分左轮，右轮，中间轮，也学会了如何单拍运行查看指令等操作。
- ③ 使用单步调试的过程还有有些生疏，毕竟涉及到硬件模拟操作是个人的弱项，课下还是要多加练习为实验二做准备。

实验项目名称： 整机实验（题目三）

日期： 2022.12.15

实验环境： 模拟软件

实验目的：

利用仿真软件的资源，自行开发出一台可实际运行的计算机。

实验要求：

给出几个整机实验题，自主选取研制一台具有相关性能的实验计算机

实验内容：

一、整机实验的题目要求

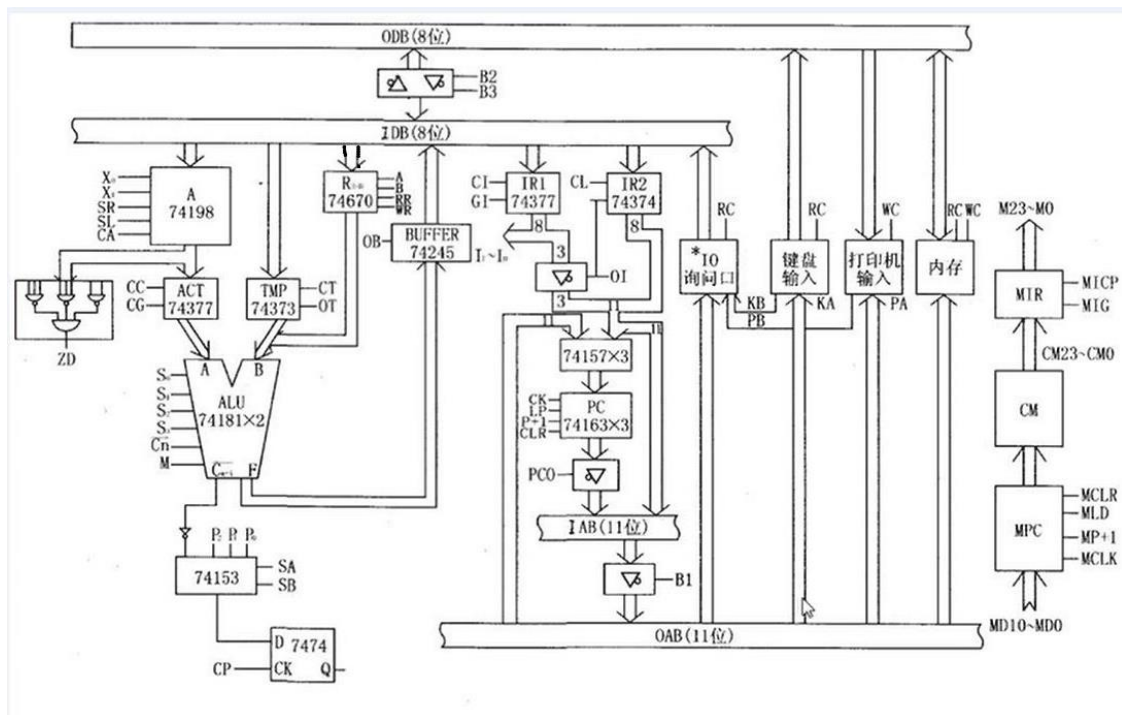
题目三：研制一台性能如下的实验计算机

- (1) 不带外部设备。
- (2) 运算器采用单累加器多通用寄存器结构。
- (3) 操作数寻址方式有：直接地址寻址、立即数寻址、寄存器直接寻址
- (4) 指令系统由如下 8 条指令组成：

| 指令编码 | | 指令助记符 | 指令功能 |
|-------------------------|---------|-------------|---|
| 第一字节 | 第二字节 | | |
| 17 16 15 14 13 12 11 10 | | | |
| 0 00 00 x Ri | | ADD A,Ri | $(A) + (Ri) \rightarrow A$ |
| 0 00 01 x Ri | | MOV Ri, A | $(A) \rightarrow Ri$ |
| 0 00 10 x x x | | SRC A | (A) 带进位 C 右环移一位 |
| 0 00 11 x x x | d7 ~ d0 | MOV A,#data | data \rightarrow A |
| 0 01 000 a9 a8 | a7 ~ a0 | LDA addr | $(addr) \rightarrow A$ |
| 0 01 010 a9 a8 | a7 ~ a0 | STA addr | $(A) \rightarrow addr$ |
| 0 01 100 a9 a8 | a7 ~ a0 | JC addr | 若 CY=1 则 $addr \rightarrow PC$, 否则 PC 加 1 |
| 0 01 110 a9 a8 | a7 ~ a0 | JMP addr | $addr \rightarrow PC$ |

- (5) 能执行双字节加法程序

二、整机的逻辑图



开关位置选择:

| | | | | |
|----|----|----|----|----|
| 开关 | KA | KB | KC | KR |
| 位置 | 左 | 右 | 右 | 下 |

三、指令系统编码和指令执行的过程

| 指令助记符 | 指令编码 I7-I0 | 执行流程 |
|--------------|----------------------|---|
| 取指微指令 | | (PC)→IAB→OAB (M)→ODB→IDB→IR1 (PC)+1→PC (MD)→MPC 接数 (A)→ACT |
| ADD A, RI | 000000Ri | (ACT)+(Ri)→BUF→IDB→A 置 CY |
| MOV RI, A | 000010Ri | (ACT)→BUF→IDB→RI |
| SRC A | 00010000 | CY, (A) 右环移一位 |
| MOV A, #DATA | 00011000 DATA | (PC)→IAB→OAB (M)→ODB→IDB→A (PC)+1→PC |
| LDA ADDR | 001000a9a8, a7-a0 | (PC)→IAB→OAB (M)→ODB→IDB→IR2 (PC)+1→PC (IR1, IR2)→IAB→OAB, (M)→ODB→IDB→A |
| STA ADDR | 001010a9a8 | (PC)→IAB→OAB |

| | | |
|----------|----------------------|---|
| | a7-a0 | (M)→ODB→IDB→IR2 (PC)+1→PC (IR1, IR2)→IAB→OAB, (ACT)→BUF→IDB→ODB |
| JC ADDR | 0011000a9a8 a7-a0 | (PC)→IAB→OAB (M)→ODB→IDB→IR2 (PC)+1→PC (IR1, IR2)→IAB IAB→(PC) (条件成立) |
| JMP ADDR | 001110a9a8 a7-a0 | (PC)→IAB→OAB (M)→ODB→IDB→IR2 (PC)+1→PC (IR1, IR2)→IAB IAB→(PC) (条件成立) |

四、微指令格式和指令微程序

24 位微码对应关系：M23-M0(24 位微指令码)

| | | | | | | | | | | | | |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | M23 | M22 | M21 | M20 | M19 | M18 | M17 | M16 | M15 | M14 | M13 | M12 |
| 有效电平 | * | * | * | * | * | * | * | * | 0 | 1 | 1 | * |
| | S3 | S2 | S1 | S0 | /Cn | M | X1 | X0 | OI | CL' | CP' | / |
| | | | | | | | | | | | | |
| | M11 | M10 | M9 | M8 | M7 | M6 | M5 | M4 | M3 | M2 | M1 | M0 |
| 有效电平 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| | CG | OT | LP' | OB | GI | P+1 | DR | MLD | WC' | RC' | RR' | WR' |

| 指令助记符 | 微地址(十六进制) | 微指令码(十六进制) |
|--------------|-----------|------------|
| 取指微指令 | 000 | 00876A |
| | 001 | |
| | 002 | |
| ADD A, RI | 003 | 9BAEBC |
| | 004 | 00876A |
| | 005 | |
| | 006 | |
| MOV RI, A | 007 | 088EBF |
| | 008 | 00876A |
| | 009 | |
| | 00A | |
| SRC A | 00B | 01AFBE |
| | 00C | 00876A |
| | 00D | |
| | 00E | |
| MOV A, #DATA | 00F | 038FFA |
| | 010 | 00876A |

| | | |
|----------|-----|--------|
| | 011 | |
| | 012 | |
| LDA ADDR | 013 | 00CFFA |
| | 014 | 030FBA |
| | 015 | 00876A |
| | 016 | |
| STA ADDR | 017 | 00CFFA |
| | 018 | 080EB6 |
| | 019 | 00876A |
| | 01A | |
| JC ADDE | 01B | 00CFFA |
| | 01C | 000DBE |
| | 01D | 00876A |
| | 01E | |
| JMP ADDR | 01F | 00CFFA |
| | 020 | 000DBE |
| | 021 | 00876A |
| | 022 | |

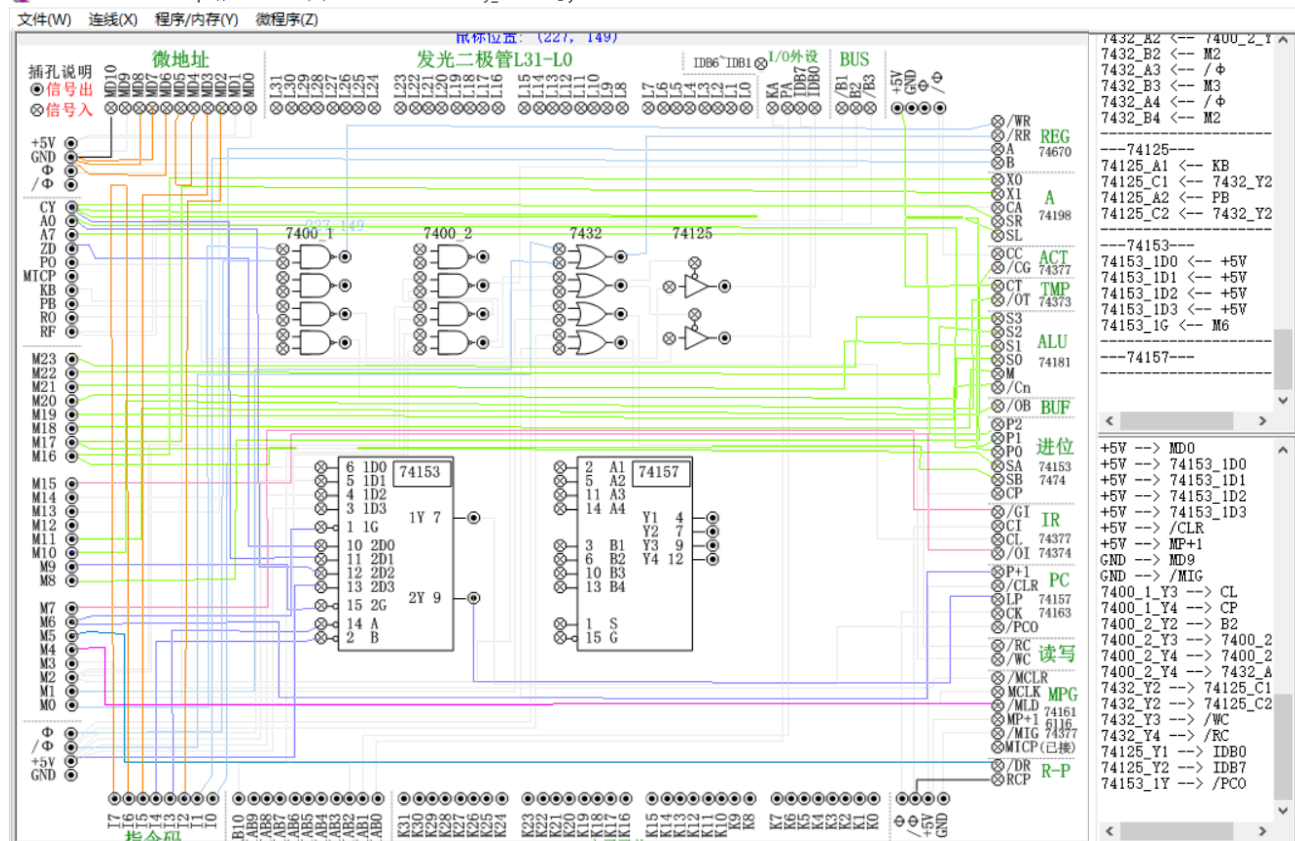
五、确定微程序入口地址形成方法

指令系统有 8 条微指令，选取 3 个指令操作码为核心，根据指令系统编码，可选取 IR1 的 I5,I4,I3 指令操作码为微程序入口地址的形成信号，其中 MD5~MD10 均接 0，MD0、MD1 均接 1，MD4 接 I5，MD3 接 I4，MD2 接 I3，如下图所示

| 指令操作码 I5I4I3 | 微程序首址 MD10~MD0 |
|-----------------|-------------------|
| 000 | 003H |
| 001 | 007H |
| 010 | 00BH |
| 011 | 00FH |
| 100 | 013H |
| 101 | 017H |
| 110 | 01BH |
| 111 | 01FH |

六、.ABL 内容

FD-CES-B 仿真软件 <https://weibo.com/u/1403786364> bluesky_blueseas@yeah.net



由于实验二大部分是在实验一的基础上实现的，因此下面只呈现与其不同的连线：

---微程序地址---

MD9 <-- GND

MD8 <-- GND

MD7 <-- GND

MD6 <-- GND

MD5 <-- GND

MD4 <-- I5

MD3 <-- I4

MD2 <-- I3

MD1 <-- +5V

MD0 <-- +5V

---74153---

74153_1D0 <-- +5V

74153_1D1 <-- +5V

74153_1D2 <-- +5V

74153_1D3 <-- +5V

74153_1G <-- M6

74153_2D0 <-- ZD

74153_2D1 <-- CY

74153_2D2 <-- A0

74153_2D3 <-- +5V

74153_2G <-- M9

74153_A <-- I4

74153_B <-- I3

七、BIT.DEF 内容

S3 S2 S1 S0 /CN M X1 X0 OI CL CP NONE

CG OT LP OB GI P+1 DR MLD WC RC RR WR

八、INS.DEF 内容

?2

-ADD

4

A,Ri

000000Ri

-MOV

4

Ri,A

000010Ri

-SRC

4

A

00010000

-MOV

2

A,#DATA

00011000

D7~D0

-LDA

3

ADDR

001000A9A8

A7~A0

-STA

3

ADDR

001010A9A8

A7~A0

-JC

3

ADDR

001100A9A8

A7~A0

-JMP

3

ADDR

001110A9A8

A7~A0

九、MOP.MID 内容

;取址微指令

-00876A

-

-

;ADD A,Ri

-9BAEBC

-00876A

-

-

;SUB A Ri

-638EBC

-00876A

-

-

;MOV A,@Ri

-ACCEBC

-030FBA

-00876A

-

-

-

-

-

;MOV A Ri

-AF8EBC

-00876A

-

-

;MOV Ri A

-088EBF

-00876A

-

-

;MOV A ,#data

-038FFA

-00876A

-

-

;MOV RI #data

-008FFB

-00876A

-

-

;LDA addr

-00CFFA

-030FBA

-00876A

-

;LDA addr

-00CFFA

-030FBA

-00876A

-

;STA addr

-00CFFA

-080EB6

-00876A

-

;STA addr

-00CFFA

-080EB6

-00876A

-

;RLC A

-02AFBE

-00876A

-

-

;RRC A

-01AFBE

-00876A

-

-

;JZ Addr

-00CFFA

;JC addr

-000DBE

;JA0 addr

-00876A

;JMP addr

-

;HALT

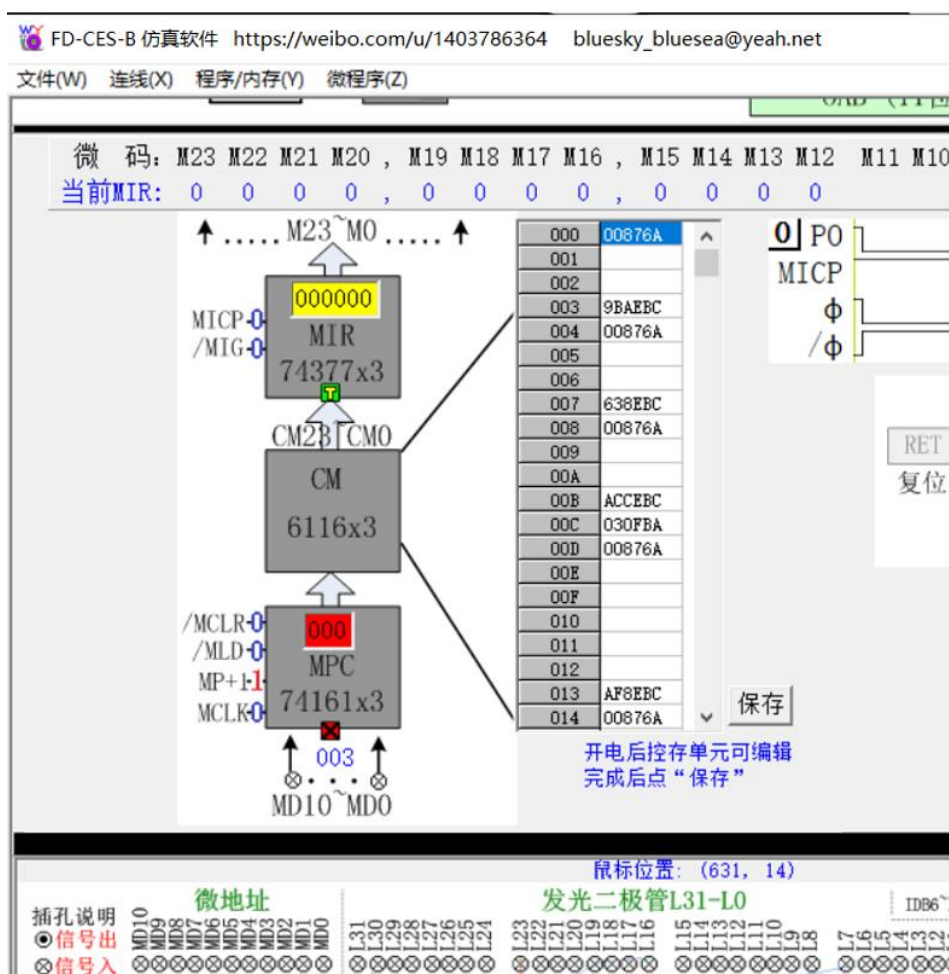
-008F9E

-00876A

十、调试单条指令过程

调试基本步骤如下：

①读取 mop 文件到控存



②编写.asm 文件

```
ORG 050

LDA 001          ;X 的低位->A
MOV R0,A         ;(A)->R0
LDA 002          ;Y 的低位->A
ADD A,R0         ;X 低位+Y 低位->A 有进位置 CY=1, 无进位置 CY=0
JC CARRY         ;有无进位? 有进位则跳到 CARRY,无进位则顺序执行
;无进位
```

```
STA 005          ;(A)->005,X 与 Y 的低位相加和送至 005
LDA 003          ;X 的高位->A
MOV R0,A         ;(A)->R0
LDA 004          ;Y 的高位->A
ADD A,R0         ;X 的高位+Y 的高位->A
STA 006          ;(A)->006,X 与 Y 的高位相加结果送至 006
```

;有进位

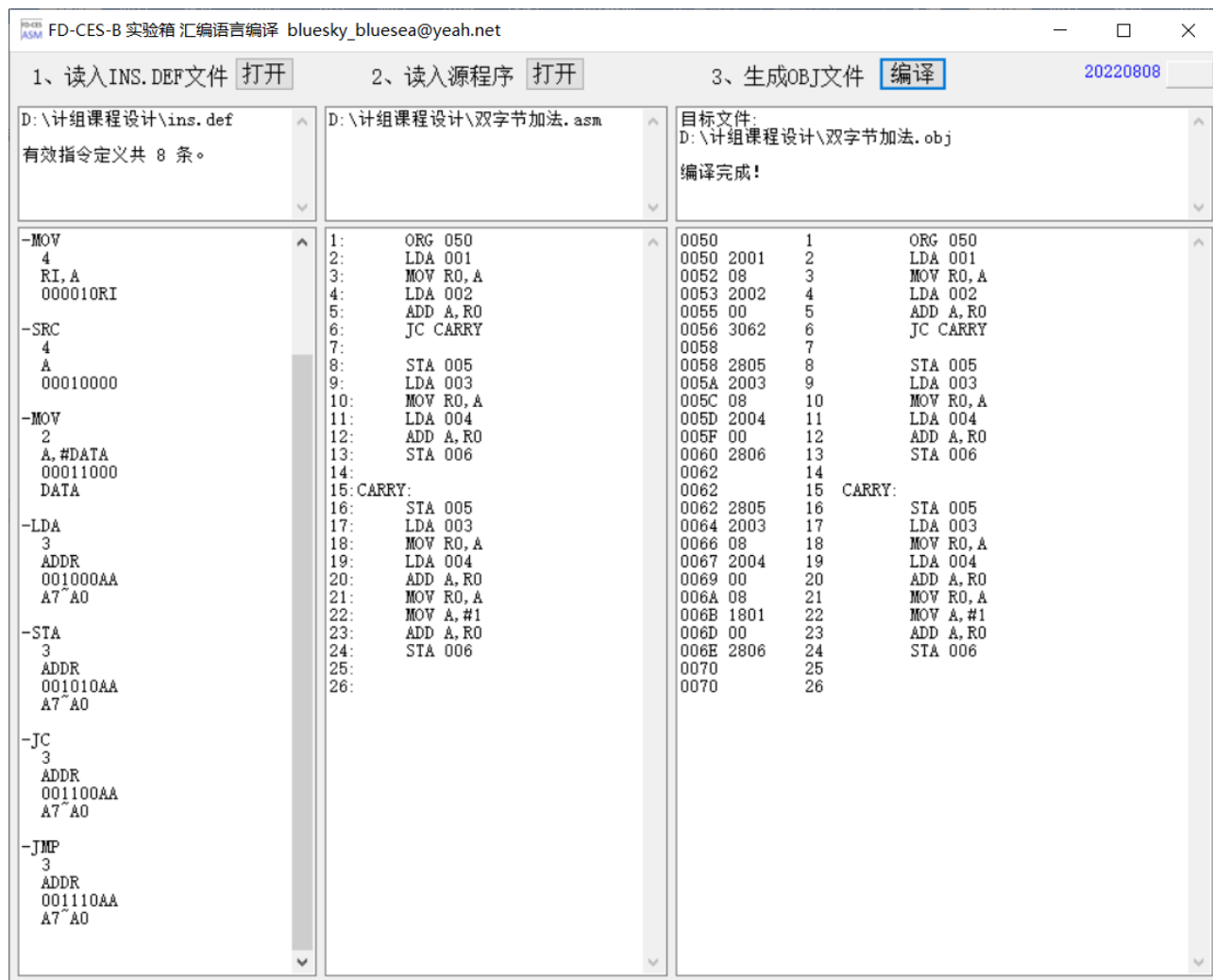
CARRY:

```
STA 005          ;(A)->005,X 与 Y 的低位相加和送至 005
LDA 003          ;X 的高位->A
MOV R0,A         ;(A)->R0
LDA 004          ;Y 的高位->A
ADD A,R0         ;X 的高位+Y 的高位->A
MOV R0,A         ;(A)->R0
MOV A,#1         ;1->A
ADD A,R0         ;X 与 Y 的高位相加和再加上来自低位相加的进位 1
STA 006          ;(A)->006, X 与 Y 的高位相加结果送至 006
```

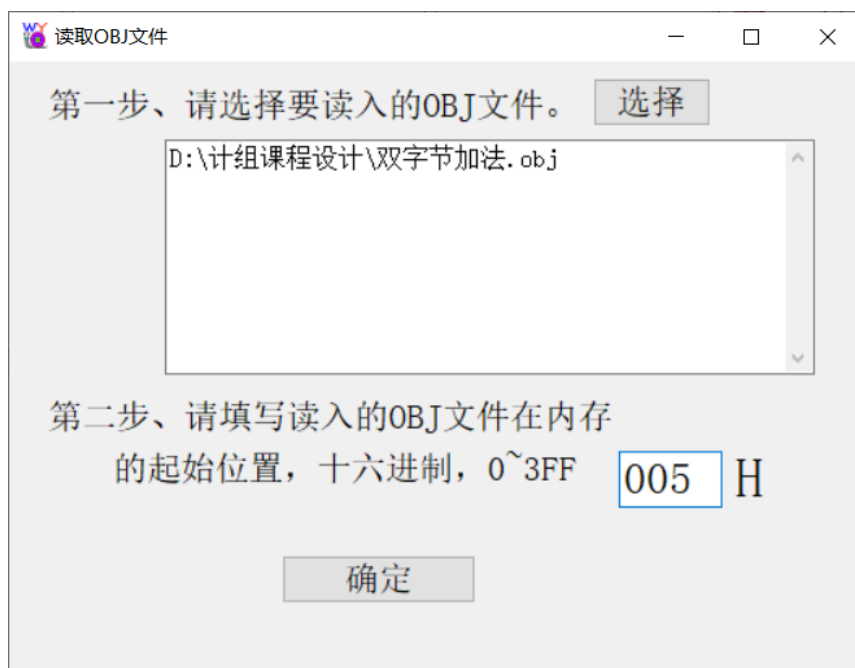
;功能：实现双字节加法

;结果 X=34BE,Y=EE7F,运行程序后 005 中存储低位相加和 3D,006 存储高位相加和加低位进位 23

③.asm 文件导入编译工具，生成.obj 文件



④读入.obj 文件，随后运行



| 地址 | 指令 | 指令编码 | |
|------|----------|------|-------------------------|
| 050. | LDA 001 | 20 | ;(001)=BE->A |
| 051 | | 01 | |
| 052 | MOV R0,A | 08 | ;(A)=BE->R0 |
| 053 | LDA 002 | 20 | ;(002)=7F->A |
| 054 | | 02 | |
| 055 | ADD A,R0 | 00. | ;BE+7F=3D->A 有进位 CY=1 |
| 056 | JC | 30 | ;JC 判断 CY 是否为 1 |
| 057 | | 62 | ;CY=1 跳转到 062 |
| 062 | STA 005 | 28. | ;(A)=3D->005 |
| 063 | | 05 | |
| 064 | LDA 003 | 20. | ;(003)=34->A |
| 065 | | 03 | |
| 066 | MOV R0,A | 08. | ;(A)=34->R0 |
| 067 | LDA 004 | 20 | ;(004)=EE->A |
| 068 | | 04 | |
| 069 | ADD A,R0 | 00 | ;34+EE=22->A |
| 06A | MOV R0,A | 08 | ;(A)=22->R0 |
| 06B. | MOV A,#1 | 18 | ;1->A |
| 06C. | | 01 | |
| 06D. | ADD A,R0 | 00 | ;22+1=23->A 加上来自低位相加的进位 |
| 06E. | STA 006 | 28 | ;(A)=23->006 |
| 06F | | 06 | |

单拍运行按 STEP 键显示如下：

| P9 P8 P7 | P6 P5 | |
|----------|----------|----------------|
| (OAB 信息) | (ODB 信息) | |
| 0.0.1. | B. E. | ;输入 X 的低位->001 |
| 0.0.2. | 7. F. | ;输入 Y 的低位->002 |
| 0.0.3. | 3. 4. | ;输入 X 的高位->003 |
| 0.0.4. | E. E. | ;输入 Y 的高位->004 |

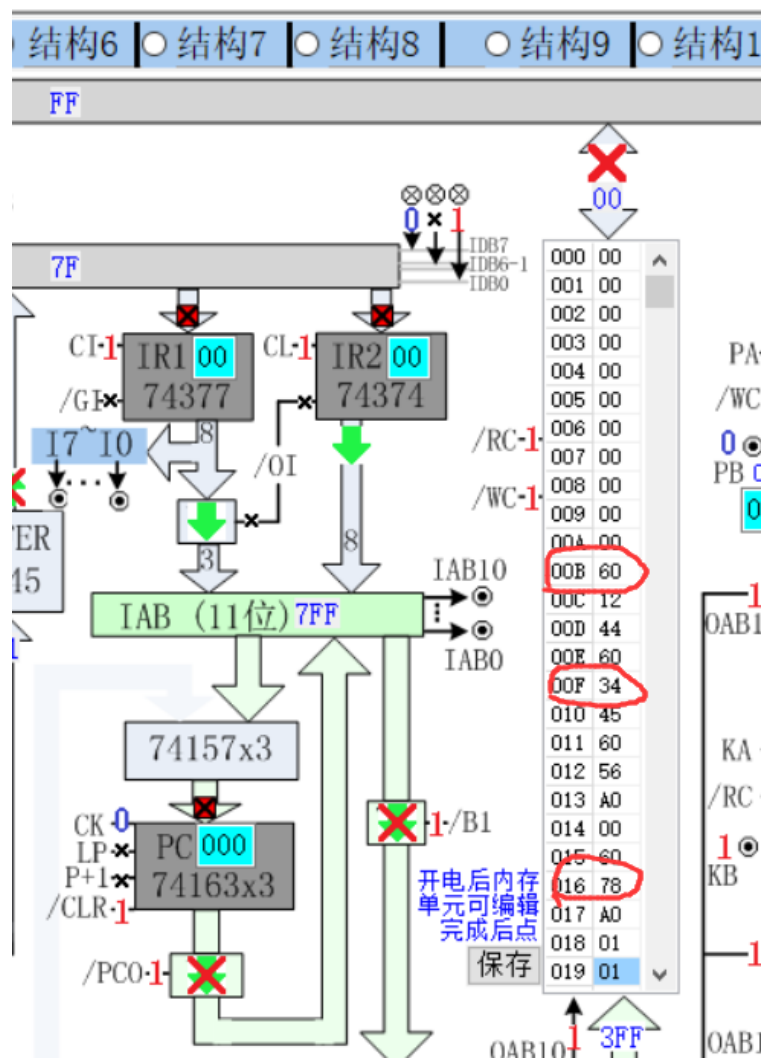
| | | |
|----------|-------|-----------------------|
| 0. 0. 5. | 0. 0. | ;运行前设置(005)=0 |
| 0. 0. 6. | 0. 0. | ;运行前设置(006)=0 |
| | | ;键入首址 050 |
| 0. 5. 0. | 2. 0. | ;(001)=BE->A |
| 0. 5. 1. | 0. 1 | |
| 0. 0. 1. | B. E. | |
| 0. 5. 2. | 0. 8. | ;(A)=BE->R0 |
| 7FF | B. E. | |
| 0. 5. 3. | 2. 0. | ;(002)=7F->A |
| 0. 5. 4. | 0. 2. | |
| 0. 0. 2. | 7. F. | |
| 0. 5. 5. | 0. 0. | ;BE+7F=3D->A 有进位 CY=1 |
| 7FF. | 3. D. | |
| 0. 5. 6. | 3. 0. | ;JC 判断 CY 是否为 1 |
| 0. 5. 7. | 6. 2. | ;CY=1 跳转到 062 |
| 0. 6. 2. | 2. 8. | ; (A)=3D->005 |
| 0. 6. 3. | 0. 5. | |
| 0. 0. 5. | 3. D. | |
| 0. 6. 4. | 2. 0. | ;(003)=34->A |
| 0. 6. 5. | 0. 3. | |
| 0. 0. 3. | 3. 4. | |
| 0. 6. 6. | 0. 8. | ;(A)=34->R0 |
| 7FF | 3. 4. | |
| 0. 6. 7. | 2. 0. | ;(004)=EE->A |
| 0. 6. 8. | 0. 4. | |
| 0. 0. 4. | E. E. | |
| 0. 6. 9. | 0. 0. | ;34+EE=22->A |
| 7FF | 2. 2. | |
| 0. 6. A. | 0. 8. | ;(A)=22->R0 |
| 7FF | 2. 2. | |
| 0. 6. B. | 1. 8. | ;1->A |

0. 6. C. 0. 1.
0. 6. D. 0. 0. ;22+1=23->A 加上来自低位相加的进位
- 7FF 2. 3.
0. 6. E. 2. 8. ; (A)=23->006
0. 6. F. 0. 6.
0. 0. 6. 2. 3.

;程序结束

0. 0. 5. 3. D. ;输入 005 查看该内存单元的内容
0. 0. 6. 2. 3. ;输入 006 查看该内存单元的内容

软件结果如下：



实验结果正确。

十一、应用程序.ASM 内容

```
ORG 050

LDA 001          ;X 的低位->A
MOV R0,A         ;(A)->R0
LDA 002          ;Y 的低位->A
ADD A,R0         ;X 低位+Y 低位->A 有进位置 CY=1, 无进位置 CY=0
JC CARRY        ;有无进位? 有进位则跳到 CARRY,无进位则顺序执行
;无进位
```

```
STA 005          ;(A)->005,X 与 Y 的低位相加和送至 005
LDA 003          ;X 的高位->A
MOV R0,A         ;(A)->R0
LDA 004          ;Y 的高位->A
ADD A,R0         ;X 的高位+Y 的高位->A
STA 006          ;(A)->006,X 与 Y 的高位相加结果送至 006
```

;有进位

CARRY:

```
STA 005          ;(A)->005,X 与 Y 的低位相加和送至 005
LDA 003          ;X 的高位->A
MOV R0,A         ;(A)->R0
LDA 004          ;Y 的高位->A
ADD A,R0         ;X 的高位+Y 的高位->A
MOV R0,A         ;(A)->R0
MOV A,#1         ;1->A
ADD A,R0         ;X 与 Y 的高位相加和再加上来自低位相加的进位 1
STA 006          ;(A)->006, X 与 Y 的高位相加结果送至 006
```

;功能：实现双字节加法

;结果 X=34BE,Y=EE7F,运行程序后 005 中存储低位相加和 3D,006 存储高位相加和加低位进位 23

十二、实验的步骤与遇到的问题及解决方法

1.遇到了实验一相同的问题，即控存文件一开始没写对，导致在做实验时单步调试总会出错，实验结果与预想值有偏差，通过一步步核验最后发现问题及时更改控存文件。

2.连线时未看清连线文件说明，导致有几条线连接错误造成实验结果出错，也是因为自己眼高手低了在实验一基础上改错了，及时与同学沟通交流发现问题所在，重新更改连线。

十三、思考及体会

通过整机二的实验，对整机实验的操作步骤更加熟悉，有了整机一的基础，对文件的编写和程序调试也变得容易，对双字节加法的机器计算过程更加清楚。

实验结束了，但是学习还没有结束，这次实验发现自己对计算机组成原理的知识已经生疏了许多，毕竟是考研科目也要涉及到的，课下借着实验的热乎劲头将微指令重新复习一下。