



# 实验3 基本模型机系统设计实验 (第7、8次课)

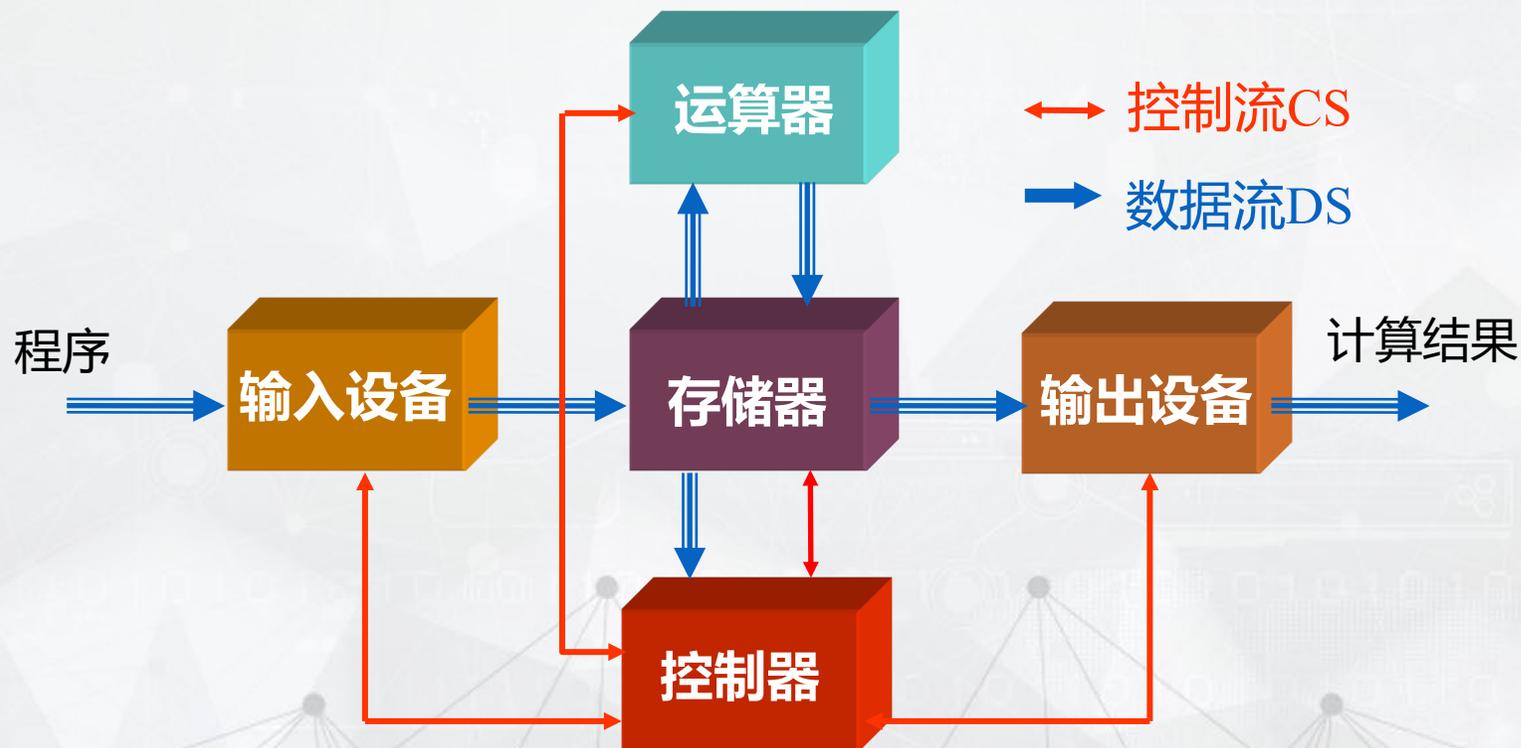
2022.11



哈尔滨工程大学计算机实验教学中心

# 基本模型机体系结构

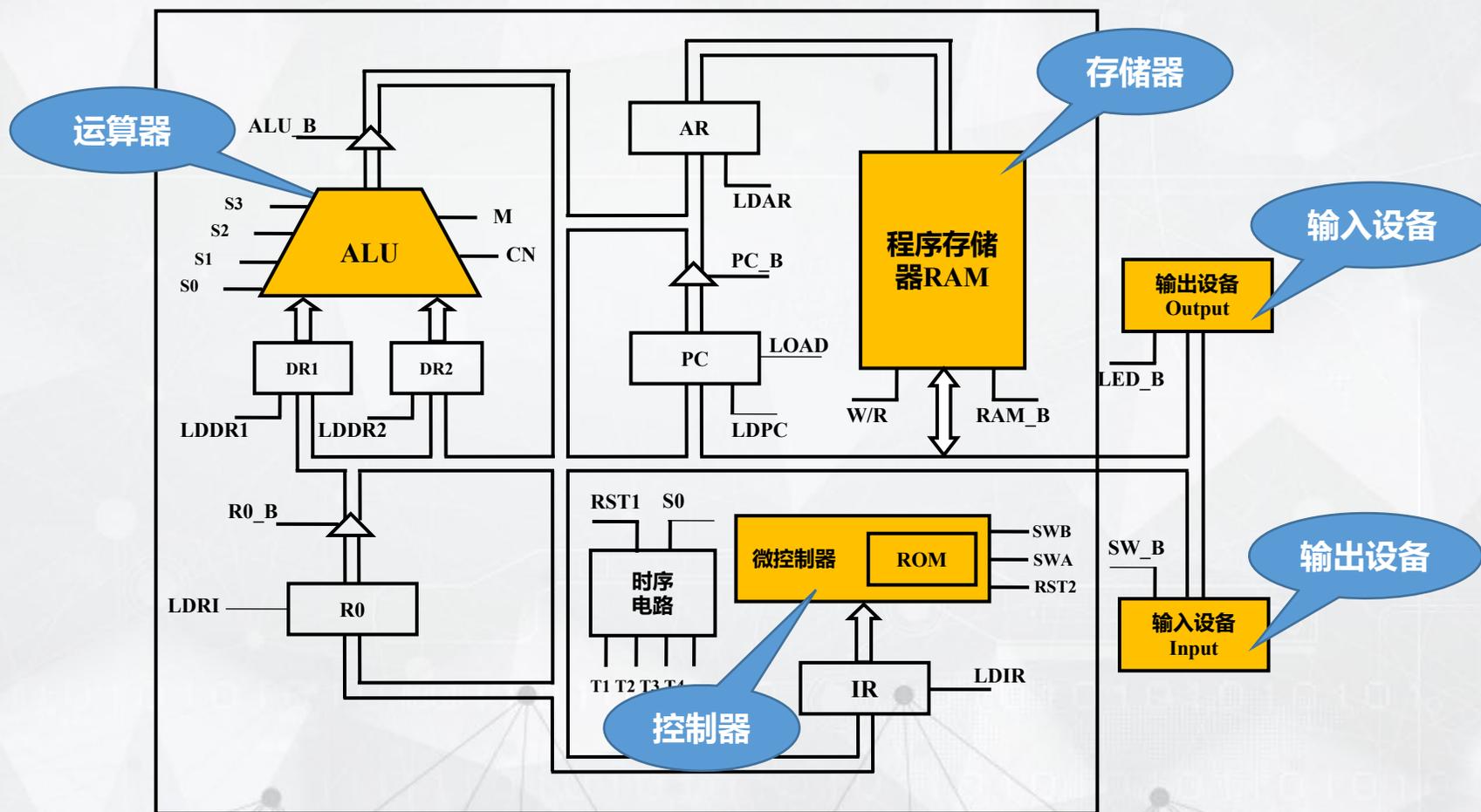
基本模型机采用冯诺依曼体系结构，可划分为5个主要模块：运算器，控制器，存储器，输入设备和输出设备。



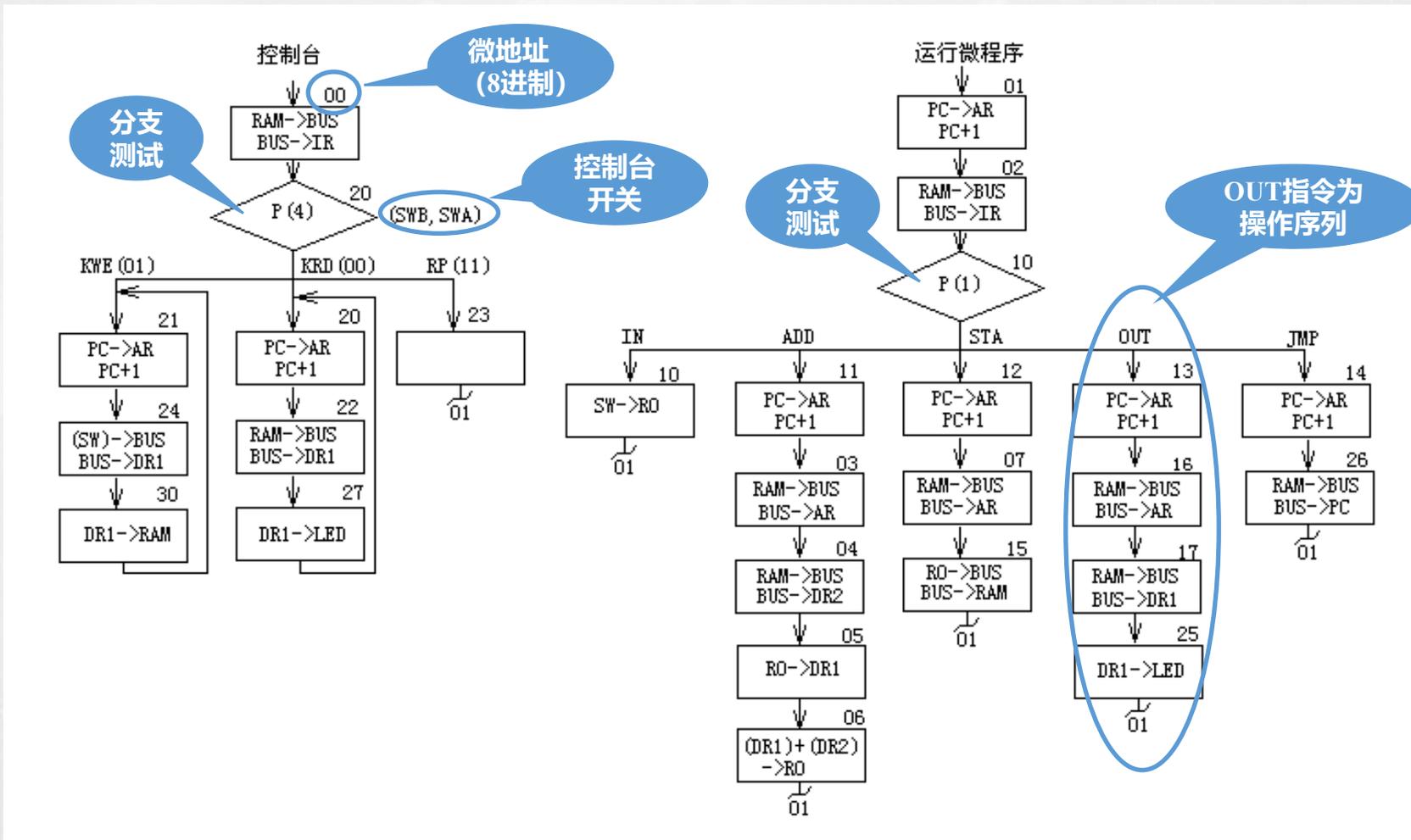
# 基本模型机指令与控制台命令

助记符	机器指令	说明	SWB	SWA	控制台命令
IN	00000000	“Input Device” →R0	0	0	读内存 (KRD)
ADD Addr	00010000 XXXXXXXX	R0+[Addr]→R0	0	1	写内存 (KWE)
STA Addr	00100000 XXXXXXXX	R0→[Addr]	1	1	启动程序 (RP)
OUT Addr	00110000 XXXXXXXX	[Addr]→ “Output Device”			
JMP Addr	01000000 XXXXXXXX	Addr→PC			

# 基本模型机数据通路



# 基本模型微程序流程



# 基本模型机微代码定义

## 微命令编码格式定义

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	UA5	UA4	UA3	UA2	UA1	UA0

## 微命令控制信号的功能

A9、A8字段			A字段				B字段				C字段			
17	16	选择	15	14	13	选择	12	11	10	选择	9	8	7	选择
0	0	SW_B	0	0	0		0	0	0		0	0	0	
0	1	RAM_B	0	0	1	LDRI	0	0	1	R0_B	0	0	1	P (1)
1	0	LED_B	0	1	0	LDDR1	0	1	0	R1_B	0	1	0	P (2)
1	1		0	1	1	LDDR2	0	1	1	R2_B	0	1	1	P (3)
			1	0	0	LDIR	1	0	0		1	0	0	P (4)
			1	0	1	LOAD	1	0	1	ALU_B	1	0	1	
			1	1	0	LDAR	1	1	0	PC_B	1	1	0	LDPC
			1	1	1		1	1	1		1	1	1	

# 基本模型机控制信号的存储

微地址 (8进制)	S3 S2 S1 S0	M Cn	WE A9 A8	A	B	C	UA5—UA0	微指令 (16进制)
0 0	0 0 0 0	0 0	0 1 1	000	000	100	0 1 0 0 0 0	018110
0 1	0 0 0 0	0 0	0 1 1	110	110	110	0 0 0 0 1 0	01ED82
0 2	0 0 0 0	0 0	0 0 1	100	000	001	0 0 1 0 0 0	00C048
0 3	0 0 0 0	0 0	0 0 1	110	000	000	0 0 0 1 0 0	00E004
0 4	0 0 0 0	0 0	0 0 1	011	000	000	0 0 0 1 0 1	00B005
0 5	0 0 0 0	0 0	0 1 1	010	001	000	0 0 0 1 1 0	01A206
0 6	1 0 0 1	0 1	0 1 1	001	101	000	0 0 0 0 0 1	959A01
0 7	0 0 0 0	0 0	0 0 1	110	000	000	0 0 1 1 0 1	00E00D
1 0	0 0 0 0	0 0	0 0 0	001	000	000	0 0 0 0 0 1	001001
1 1	0 0 0 0	0 0	0 1 1	110	110	110	0 0 0 0 1 1	01ED83
1 2	0 0 0 0	0 0	0 1 1	110	110	110	0 0 0 1 1 1	01ED87
1 3	0 0 0 0	0 0	0 1 1	110	110	110	0 0 1 1 1 0	01ED8E
1 4	0 0 0 0	0 0	0 1 1	110	110	110	0 1 0 1 1 0	01ED96
1 5	0 0 0 0	0 0	1 1 1	000	001	000	0 0 0 0 0 1	038201
1 6	0 0 0 0	0 0	0 0 1	110	000	000	0 0 1 1 1 1	00E00F
1 7	0 0 0 0	0 0	0 0 1	010	000	000	0 1 0 1 0 1	00A015
2 0	0 0 0 0	0 0	0 1 1	110	110	110	0 1 0 0 1 0	01ED92
2 1	0 0 0 0	0 0	0 1 1	110	110	110	0 1 0 1 0 0	01ED94
2 2	0 0 0 0	0 0	0 0 1	010	000	000	0 1 0 1 1 1	00A017
2 3	0 0 0 0	0 0	0 1 1	000	000	000	0 0 0 0 0 1	018001
2 4	0 0 0 0	0 0	0 0 0	010	000	000	0 1 1 0 0 0	002018
2 5	0 0 0 0	0 1	0 1 0	000	101	000	0 0 0 0 0 1	050A01
2 6	0 0 0 0	0 0	0 0 1	101	000	110	0 0 0 0 0 1	00D181
2 7	0 0 0 0	0 1	0 1 0	000	101	000	0 1 0 0 0 0	050A10
3 0	0 0 0 0	0 1	1 1 1	000	101	000	0 1 0 0 0 1	068A11

先确定每段微程序的第一条微代码的**微地址**（ROM单元地址）。然后每段微程序中，当前微指令的低六位指向下一个微指令在ROM中存放的地址，即下一个微地址。

Addr	+0	+1	+2
000	000000011000000100010000	000000011110110110000010	00000001100000001001000
003	000000001110000000000100	000000001011000000000101	000000011010001000000110
006	1001010110011010000000001	000000001110000000001101	00000000001000000000001
011	000000011110110110000011	000000011110110110000111	000000011110110110001110
014	000000011110110110010110	000000111000001000000001	00000001110000000001111
017	000000001010000000010101	000000011110110110010010	000000011110110110010100
022	0000000010100000000010111	000000011000000000000001	00000000010000000011000
025	0000010100001010000000001	0000000011010001100000001	000001010000101000010000
030	000001101000101000010001	000000000000000000000000	00000000000000000000000
033	000000000000000000000000	000000000000000000000000	00000000000000000000000



# 实验任务与步骤

- 1、新建工程，新建框图文件 (\*.bdf)，设计基本模型机电路。采用层次化设计方法。自定义元件，包括寄存器、译码器、时序信号发生器等。添加元器件库中的宏模块 lpm\_counter、lpm\_mux、总线型2选1电路BUS\_MUX、编码器74148等。保存文件。

主菜单 “File” → “New Project Wizard”，新建工程Computer（实体名）  
主菜单 “File” → “New” 项，选择Block Diagram/Schematic File，新建框图文件，保存为Computer.bdf。

- 2、设置器件

主菜单 “Assignments” → “Device” 项，选择Cyclone IV E系列EP4CE55F23C8芯片

- 3、编译电路

主菜单 “Processing” → “Start Compliation” 项，启动编译

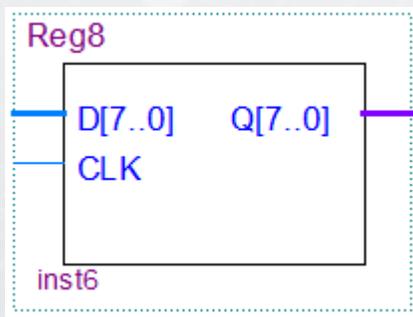
# 自定义8位寄存器元件

1.利用框图设计位寄存器电路，电路设计文件Reg8.bdf，将Reg8.bdf拷贝到总线工程目录

2.主菜单“File”→“Create/Update”项，  
选择“Create Symbol Files for Current File”

由Reg8.bdf生成Reg8.bsf，即生成自定义8位寄存器元件符号

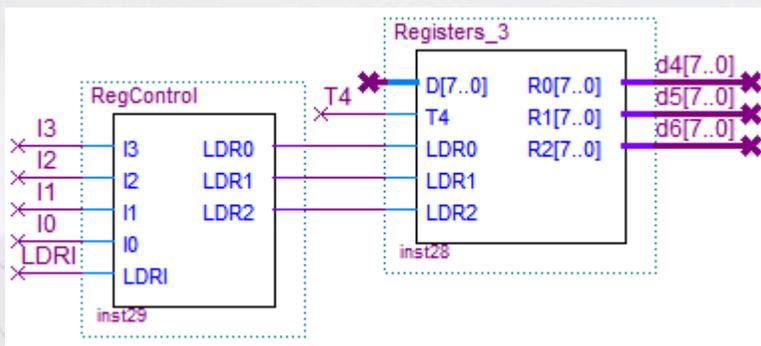
3.在元器件库中，在Project目录下选择自定义元件Reg8，加入到总线电路图中



数据寄存器Reg8可以暂存8位数据。  
当CLK上升沿到来时，输出端Q输出输入端D的值。即 $Q[7..0]=D[7..0]$

# 自定义数据寄存器组元件

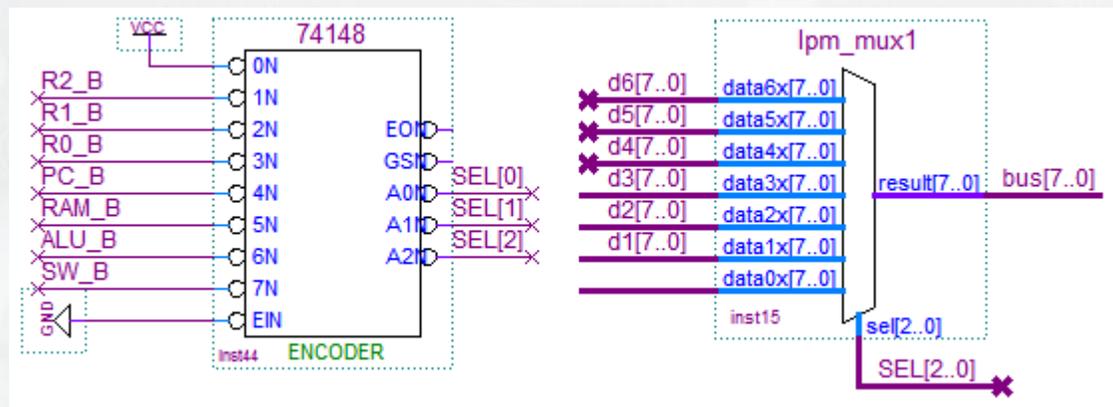
- 1.利用框图设计数据寄存器组电路，采用层次化设计方法，采用自定义寄存器元件Reg8构建寄存器组Registers\_3，并设计控制电路RegControl。
- 2、利用“File”→“Create/Update”命令生成元件符号（\*bsf）
- 3.将RegControl.bdf、RegControl.bsf、Registers3.bdf和Registers3.bsf拷贝到模型机工程目录中。将元件添加到模型机顶层电路图Computer.bdf中。



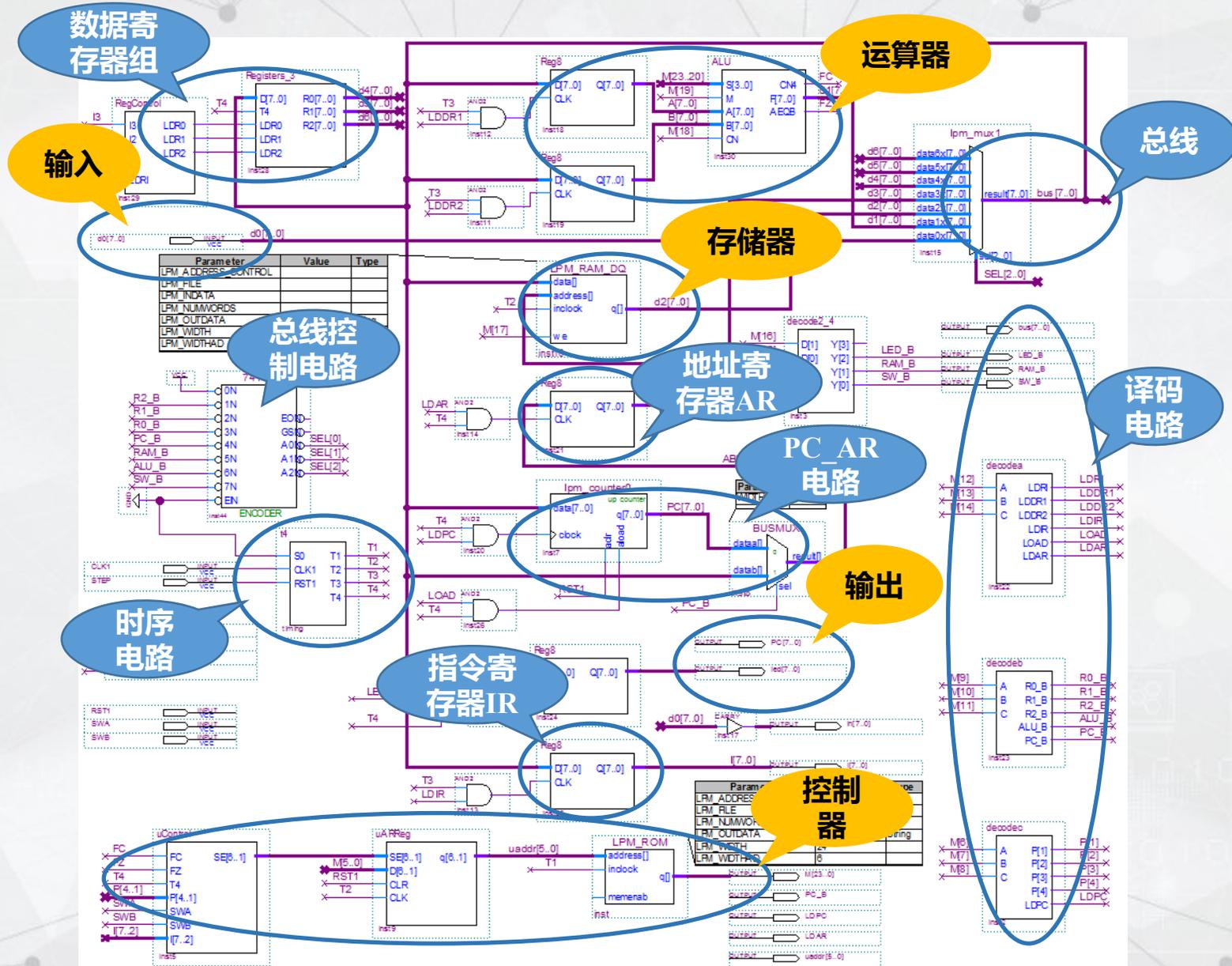
Registers3包含3个寄存器，R0、R1和R2。能够根据指令I的低2位选择指定的寄存器。

# 总线控制信号生成电路

利用框图设计总线控制信号生成电路，采用元器件中的编码器74148，将微程序控制器生成的总线控制信号编码生成多路选择开关的控制信号。控制总线上的设备的写总线操作，让选择哪个设备将数据输出到总线上。



# 基本模型机电电路图



# 实验任务与步骤

4、新建波形图文件 (\*.vwf) ,设置仿真时间, 添加输入输出端口, 设置输入信号值, 保存文件。运行仿真。

**建立仿真波形文件：**主菜单“File”→“New”项, 选择University Program VWF, 新建\*.vwf, 打开波形编辑器。

**设置仿真时间：**主菜单“Edit”→“Set End Time”项。

**添加输入输出端口：**波形编辑器窗口主菜单“Edit”→“Insert”→“Insert Node or Bus”

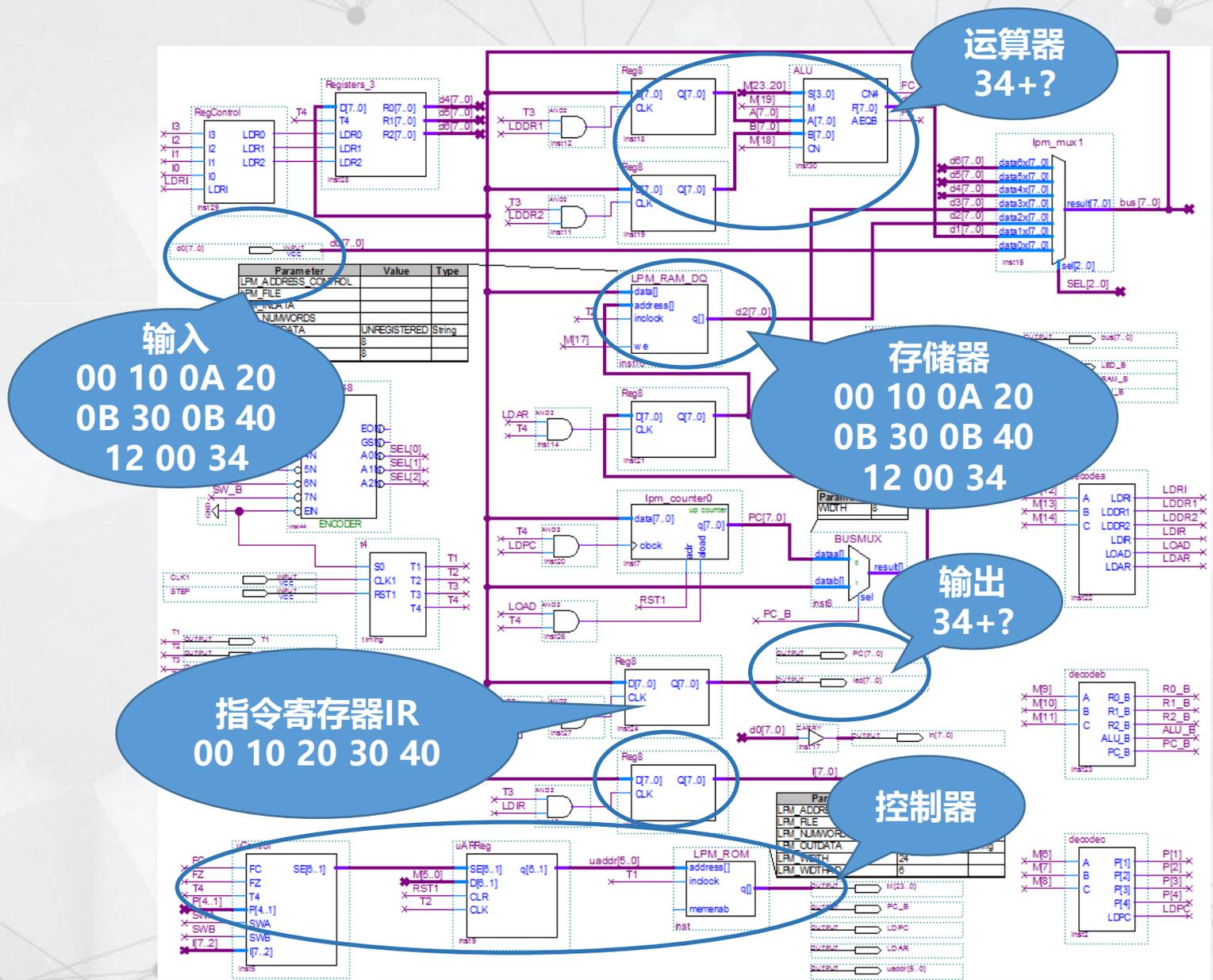
**运行仿真：**波形编辑器窗口主菜单“Simulation”→“Run Functional Simulation”项。

# 基本模型机程序代码1

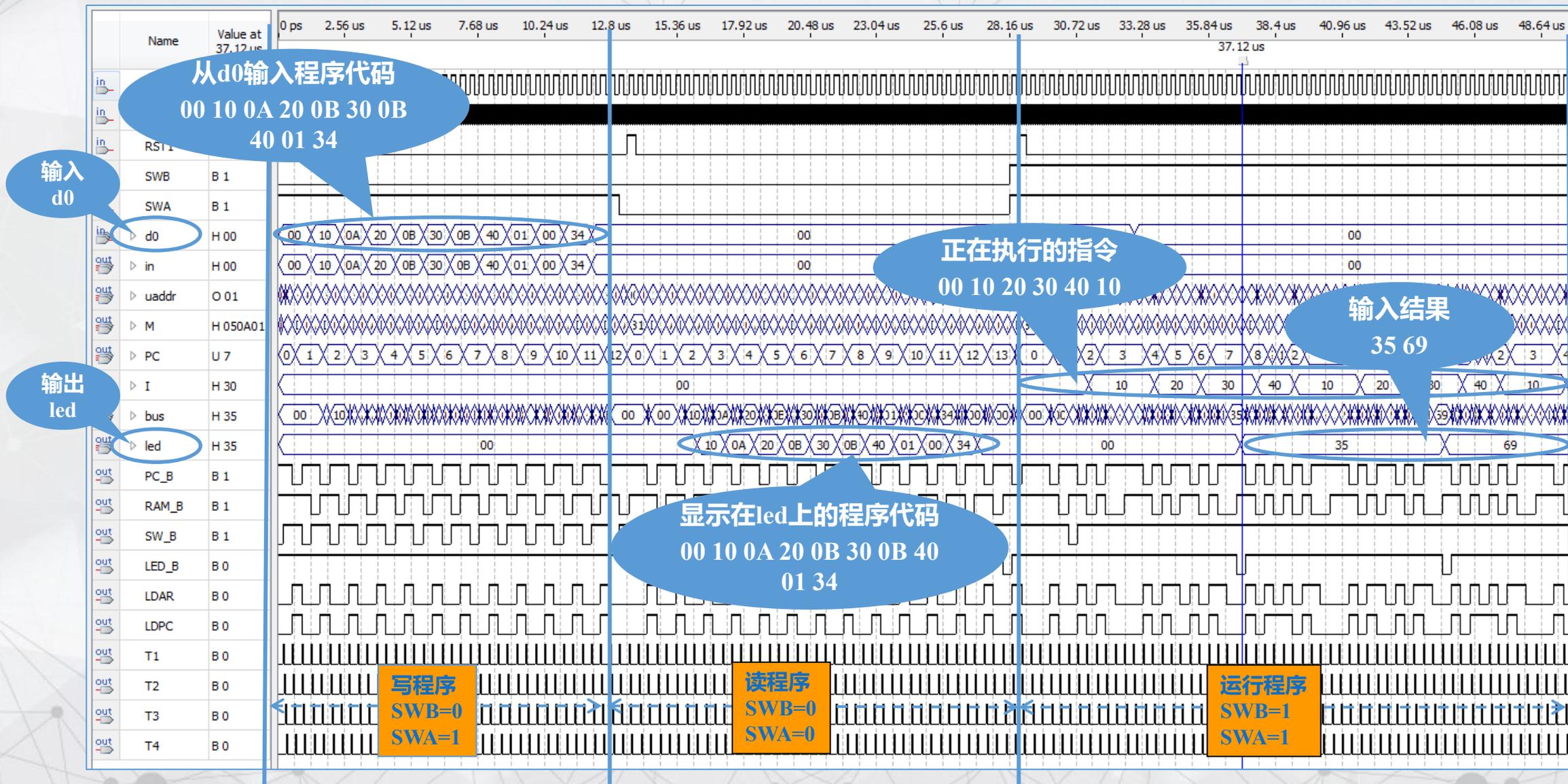
00 10 0A 20 0B 30 0B 40 01 XX 34

RAM地址	RAM数据 (程序代码)	助记符	说明
00	00	IN	“Input Device”→R0
01	10	ADD[0AH]	R0+[0AH]→R0
02	0A		地址0A
03	20	STA[0BH]	R0→[0BH]
04	0B		地址0B
05	30	OUT[0BH]	[0BH]→“Output Device”
06	0B		地址0B
07	40	JMP[01H]	01H→PC
08	01		
09			
0A	34		自定义加数34，所在RAM地址为0A
0B			求和结果，所在RAM地址为0B

# 基本模型机-执行程序



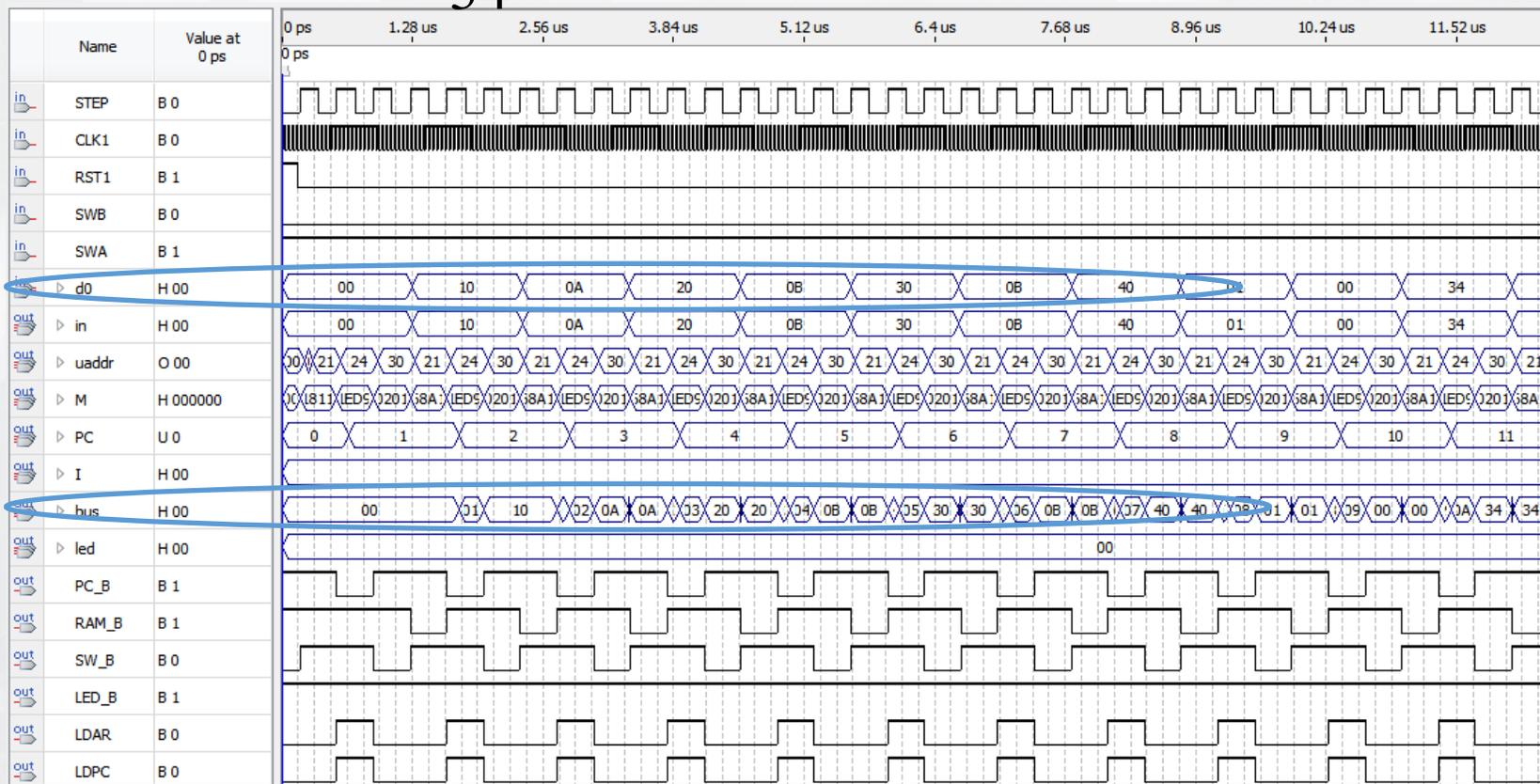
# 基本模型机仿真图 (总体)



# 基本模型机仿真图 (写内存SWB=0 SWA=1)

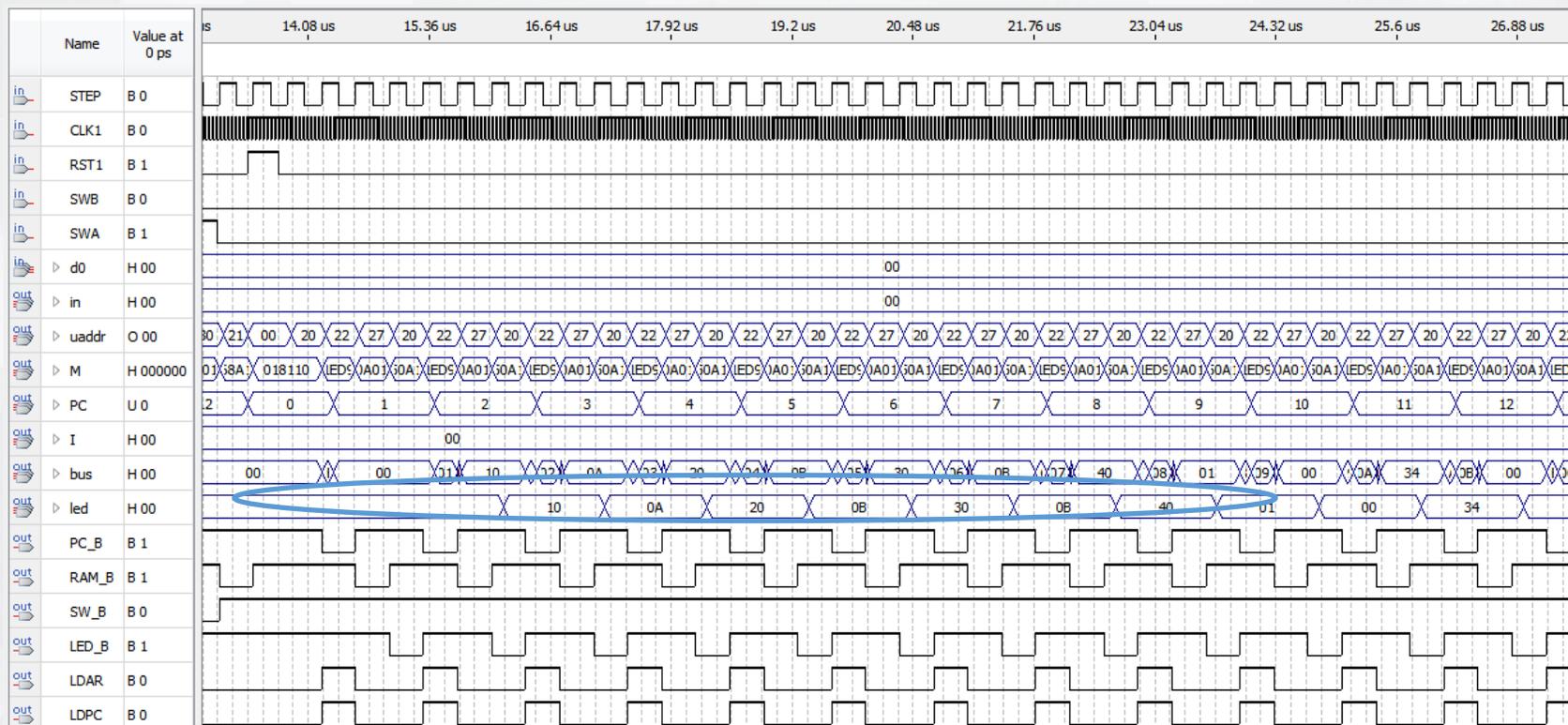
从d0输入程序代码00 10 0A 20 0B 30 0B 40 01 XX

34



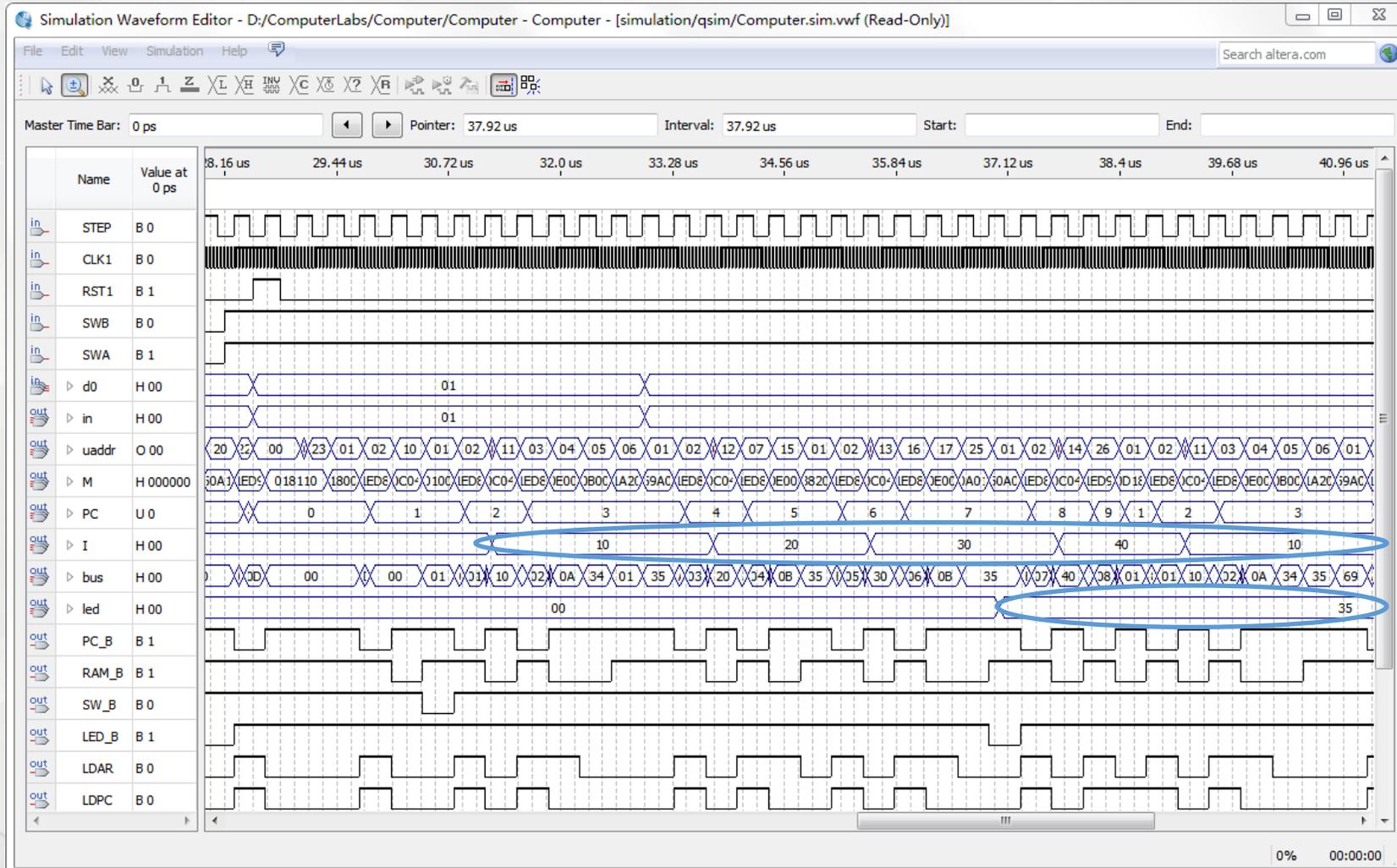
# 基本模型机仿真图 (读内存SWB=0 SWA=0)

从led输出程序代码00 10 0A 20 0B 30 0B 40 01 34



# 基本模型机仿真图 (执行程序SWB=1 SWA=1)

I表示当前正在执行的指令：IN、ADD、STA、OUT、JMP



# 基本模型机程序代码2

00 20 0D 10 0D 10 0D 10 0E 20 0F 30 0F XX 09

RAM 地址	RAM 数据 (程序代码)	助记符	说 明
00	00	IN	“INPUT DEVICE” → R0
01	20	STA[0DH]	R0 → [0DH]
02	0D		
03	10	ADD[0DH]	R0+[0DH] → R0
04	0D		
05	10	ADD[0DH]	R0+[0DH] → R0
06	0D		
07	10	ADD[0EH]	R0+[0EH] → R0
08	0E		
09	20	STA[0FH]	
0A	0F		
0B	30	OUT[0FH]	[0FH] → OUT 输出口
0C	0F		
0D			
0E	09		自定
0F			求和结果

# 实验任务与步骤

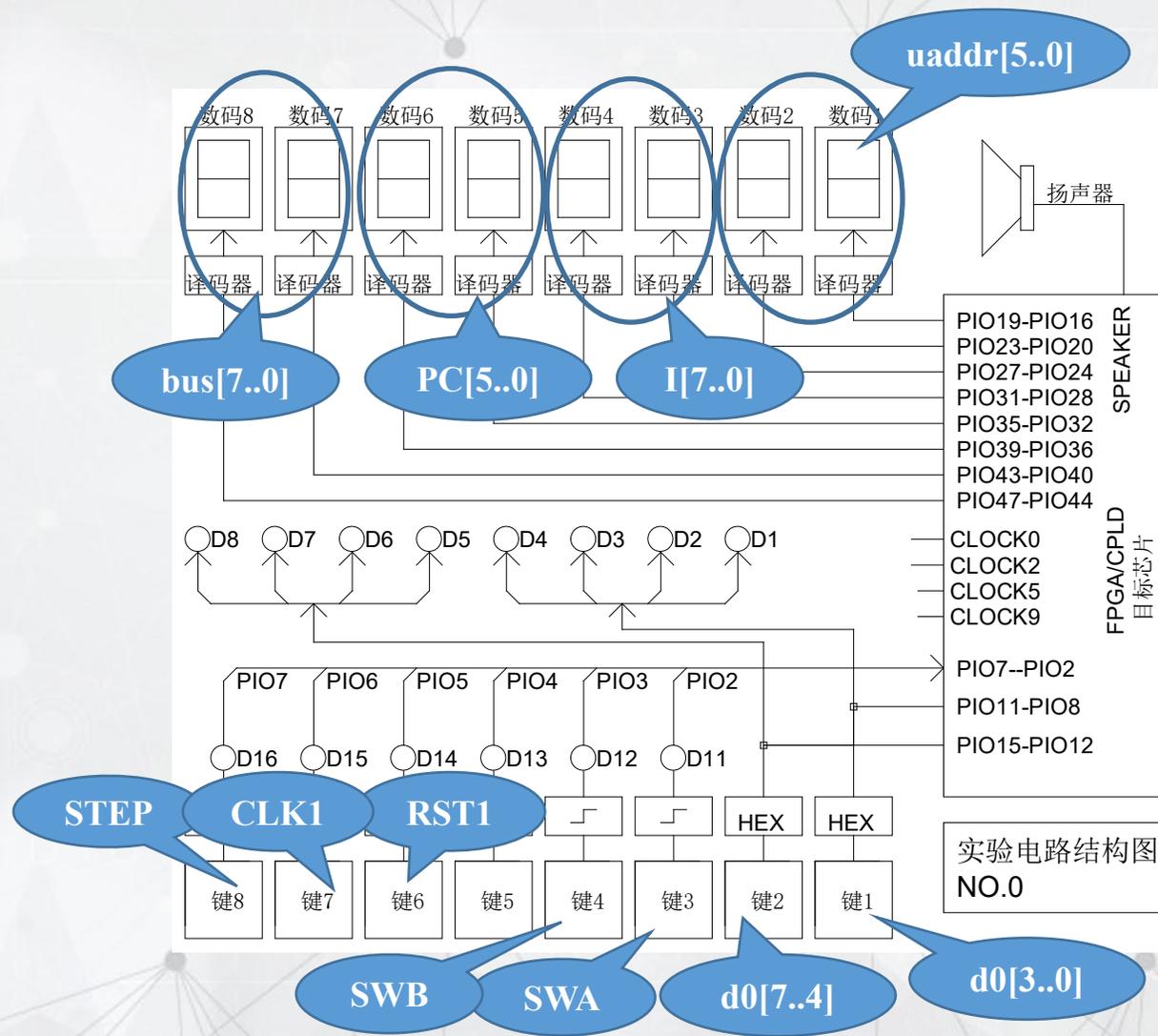
5、选择KX-CDS实验台，选择合适的电路模式结构，例如NO.0，对照电路模式图和引脚表，查找引脚号。打开编程器，输入引脚号，对电路进行引脚锁定，编译工程。

**主菜单“Assignments”→“Pin Planner”项，在Location栏中输入引脚号**

6、下载sof文件到FPGA实验台，演示基本模型机的功能。

**主菜单“Tools”→“Programmer”项，打开编程器，设置硬件，连接实验台。在Programmer窗口，点击Start按钮，Progress为100%时，下载完毕**

# 选择FPGA实验台电路结构No.0



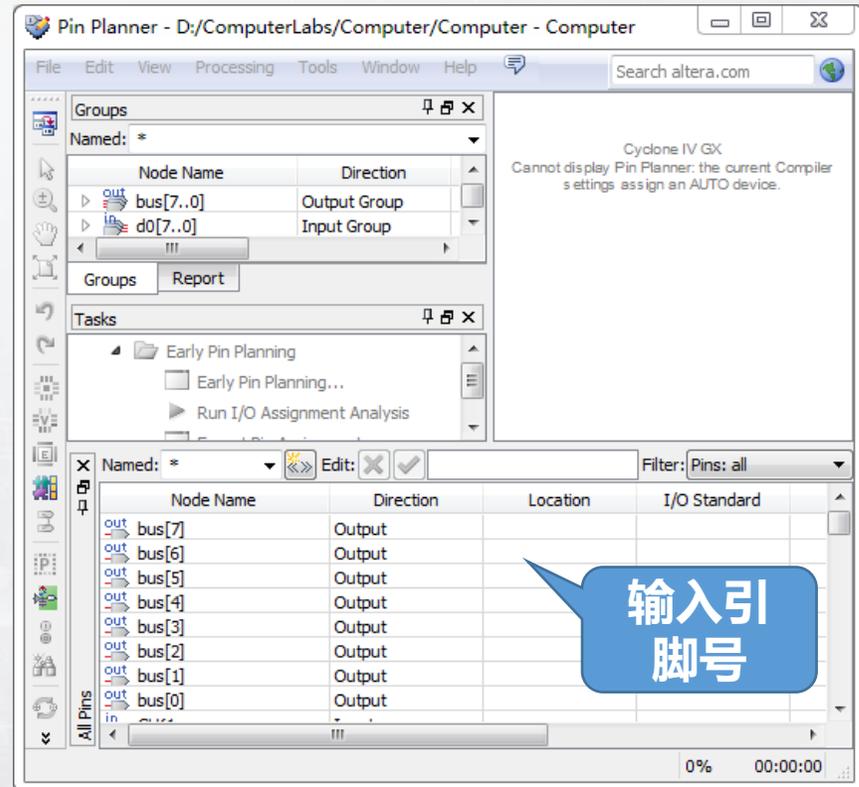
# FPGA实验台引脚锁定

参照电路模式图No.0，确定引脚名称，再查找引脚表，获得引脚号

引脚锁定方案(No.0)

输入输出信号	外设	引脚名称	引脚号
STEP	按键8		
CLK1	按键7		
RST1	按键6		
SWB	按键4		
SWA	按键3		
d0[7..4]	按键2		
d0[3..0]	按键1		
bus[7..4]	数码管8		
bus[3..0]	数码管7		
PC[7..4]	数码管6		
PC[3..0]	数码管5		
I[7..4]	数码管4		
I[3..0]	数码管3		
uaddr[5..4]	数码管2		
uaddr[3..0]	数码管1		

主菜单“Assignments”→“Pin”项，在Location栏中输入引脚号



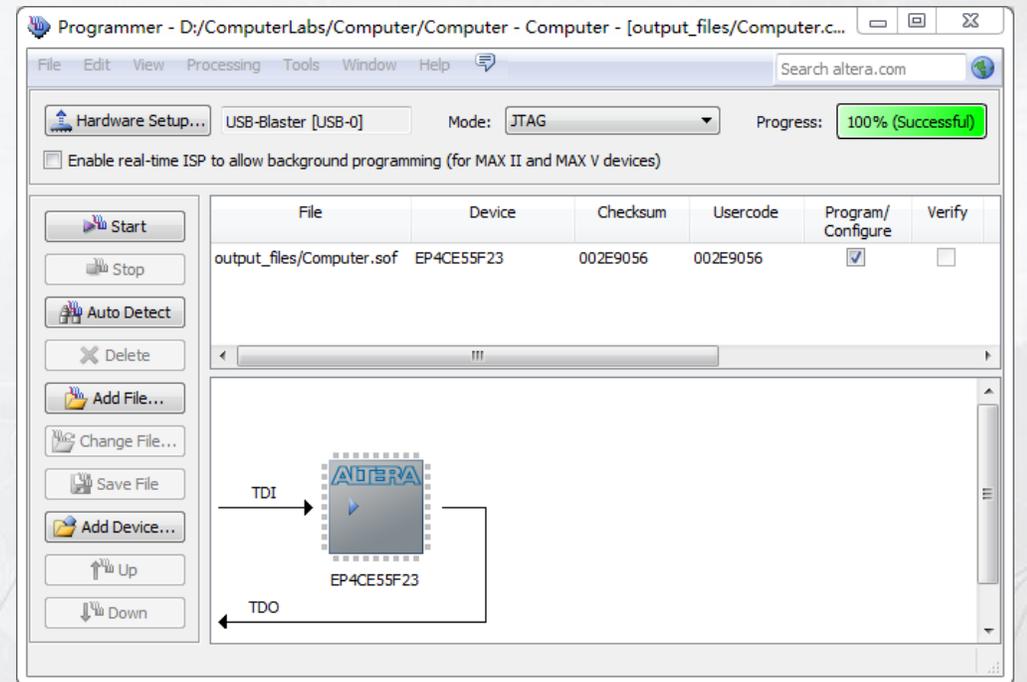
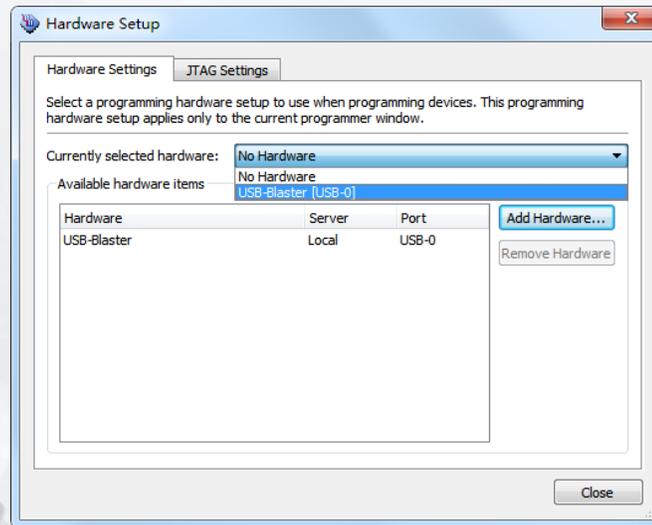
# 连接实验台下载电路

主菜单“Tools”→“Programmer”项，打开编程器，点击“Hardware Setup”按钮，选择USB-Blaster硬件。

在Programmer窗口，点击Start按钮，Progress为100%时，下载完毕。

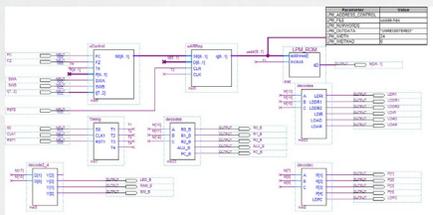
**注意：**实验台需要打开电源，并且将其JTAG接口与计算机通过USB线连接。

KX-CDS实验台

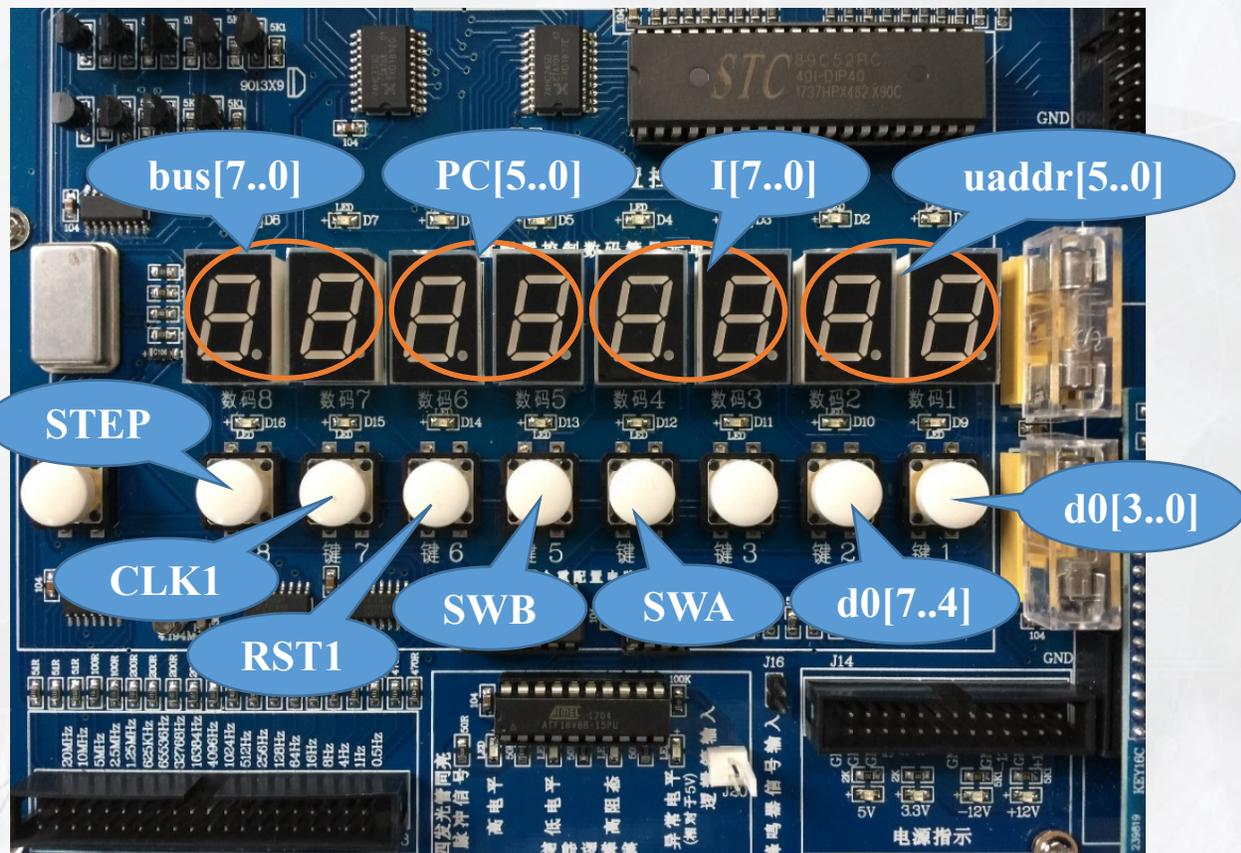
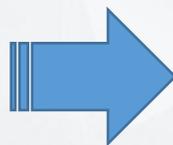


# 基本模型机实验台演示

KX-CDS实验台



下载



# 模型机扩展——带移位功能模型机系统设计

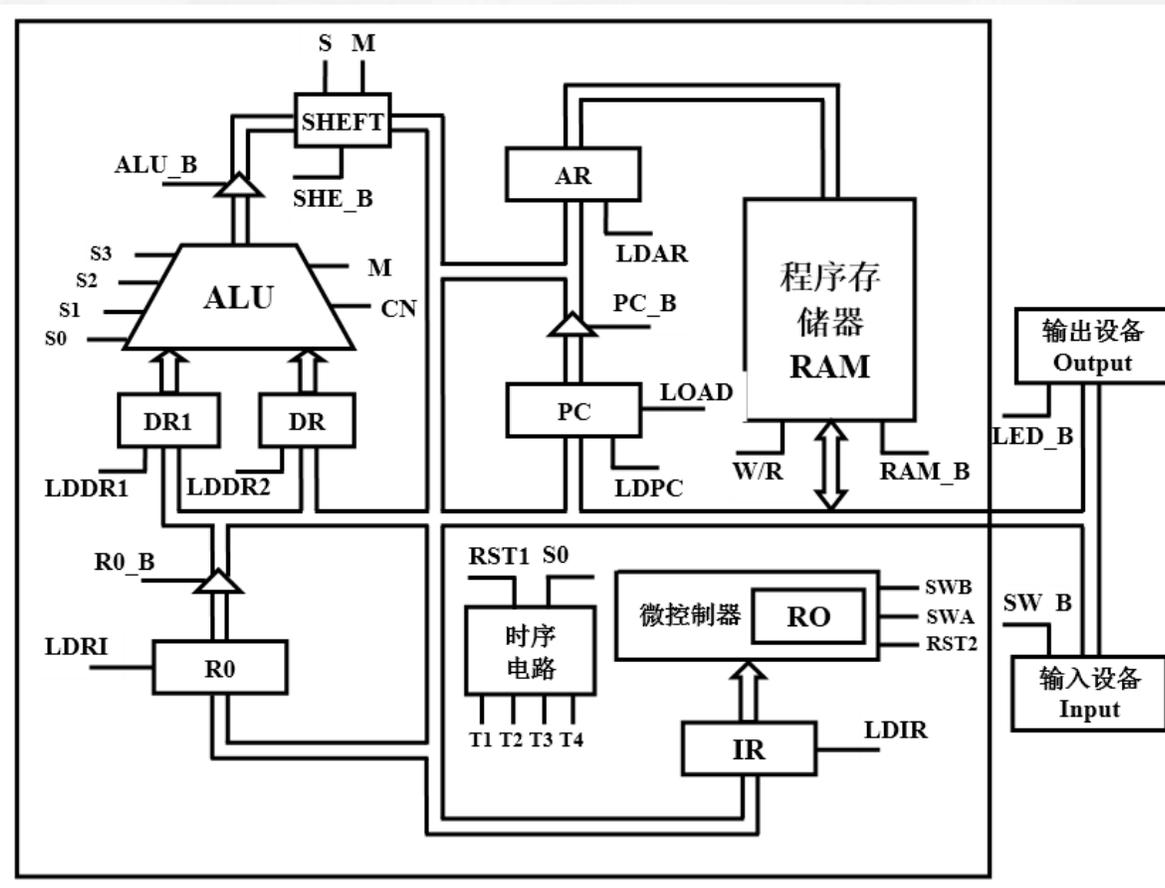
- 第一步：指令设计

在5条基本机器指令基础上，增加4条移位运算指令，并编写相应的微程序。

助记符	机器指令	说明
IN	00000000	“Input Device” → R0
ADD Addr	00010000 XXXXXXXX	$R0 + [Addr] \rightarrow R0$
STA Addr	00100000 XXXXXXXX	$R0 \rightarrow [Addr]$
OUT Addr	00110000 XXXXXXXX	$[Addr] \rightarrow$ “Output Device”
JMP Addr	01000000 XXXXXXXX	$Addr \rightarrow PC$
RR	0101 0000	R0循环右移一位
RRC	0110 0000	R0带进位循环右移一位
RL	0111 0000	R0循环左移一位
RLC	1000 0000	R0带进位循环左移一位

# 模型机扩展——带移位功能模型机系统设计

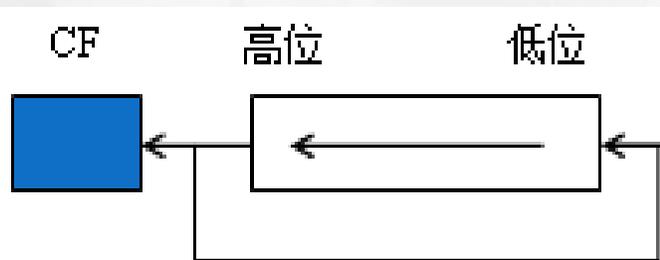
- 第二步：数据通路设计



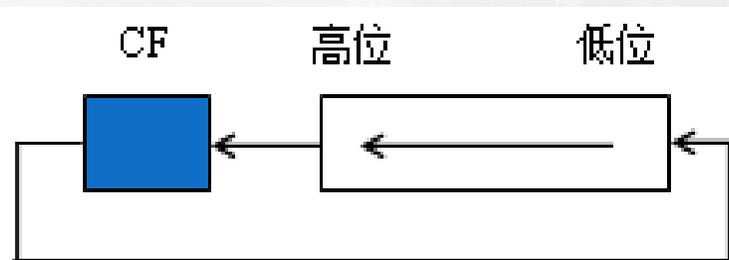
# 双向移位寄存器功能表

输入				输出									功能
CP	S1	S0	M	Q <sub>7</sub>	Q <sub>6</sub>	Q <sub>5</sub>	Q <sub>4</sub>	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	CN	
↑	0	0	任意	0	0	0	0	0	0	0	0	0	保持
↑	1	0	0	D <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	循环右移
↑	1	0	1	C0	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	带进位循环右移
↑	0	1	0	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	D <sub>7</sub>	D <sub>7</sub>	循环左移
↑	0	1	1	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	C0	D <sub>7</sub>	带进位循环左移
↑	1	1	任意	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	0	装数

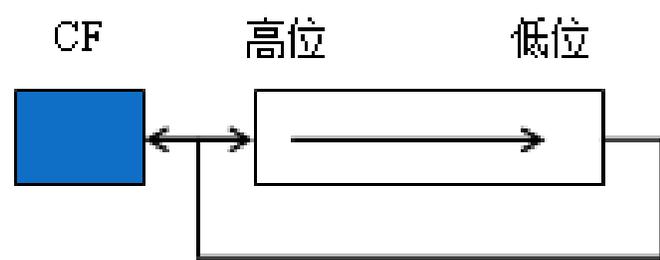
# 移位运算功能



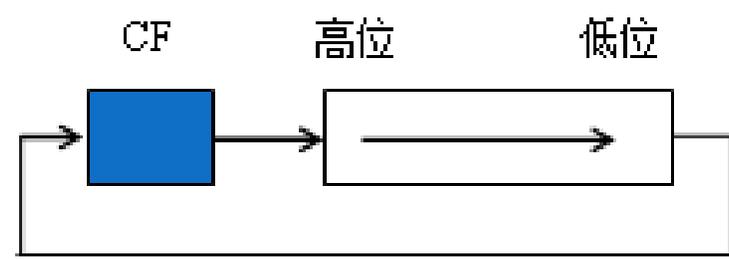
(a) 循环左移



(b) 带进位循环左移



(c) 循环右移



(d) 带进位循环右移

# 模型机扩展——带移位功能模型机系统设计

- 第三步：微程序流程图设计  
参考基本模型机微程序流程图设计方法
- 第四步：计微指令代码，完成微代码表  
参考基本模型机微指令设计方法
- 第五步：完成硬件电路设计  
在基本模型机硬件电路基础上，增加移位寄存器。
- 第六步：编写测试程序，完成软件仿真  
利用9条指令编写程序，完成仿真波形图
- 第七步：下载实验台，完成演示。

现在开始实验！（第7次课）

## 实验3 基本模型机系统设计实验

### 1、基本模型机实验：参考教材7.1.3

基本模型机系统原理：参考教材7.1.1

参考ppt：基本模型机指令系统与控制台命令设计

参考ppt：微指令格式与微程序存储

寄存器组电路文件：基本模型机部分电路文件.rar（QQ群）

### 2、完成电路设计、编译和仿真。

按照ppt中的实验仿真要求完成仿真波形图。仿真基本模型机程序代码1，  
要求：ADD的两个操作数分别是2个学号的后两位。

3、2个人一组。实体名后面加2个学号的后两位，例如computer0709

4、答疑与验收方式：QQ或腾讯会议，按组验收

现在开始实验！（第8次课）

## 实验3 基本模型机系统设计实验

### 1、基本模型机实验：参考教材7.1.3

基本模型机系统原理：参考教材7.1.1

参考ppt：基本模型机指令系统与控制台命令设计

参考ppt：微指令格式与微程序存储

寄存器组电路文件：基本模型机部分电路文件.rar（QQ群）

### 2、完成电路设计、编译和仿真。

利用5条指令编写复杂功能的测试程序并仿真

选做：下载实验台并测试

选做：带移位功能模型机

双向移位寄存器电路文件：sheft.vhd（QQ群）

带移位功能模型机微代码表文件：ucode.hex（QQ群）

### 3、答疑与验收方式：按组验收，现场演示答辩。