

带移位功能（9条指令） 模型机系统设计

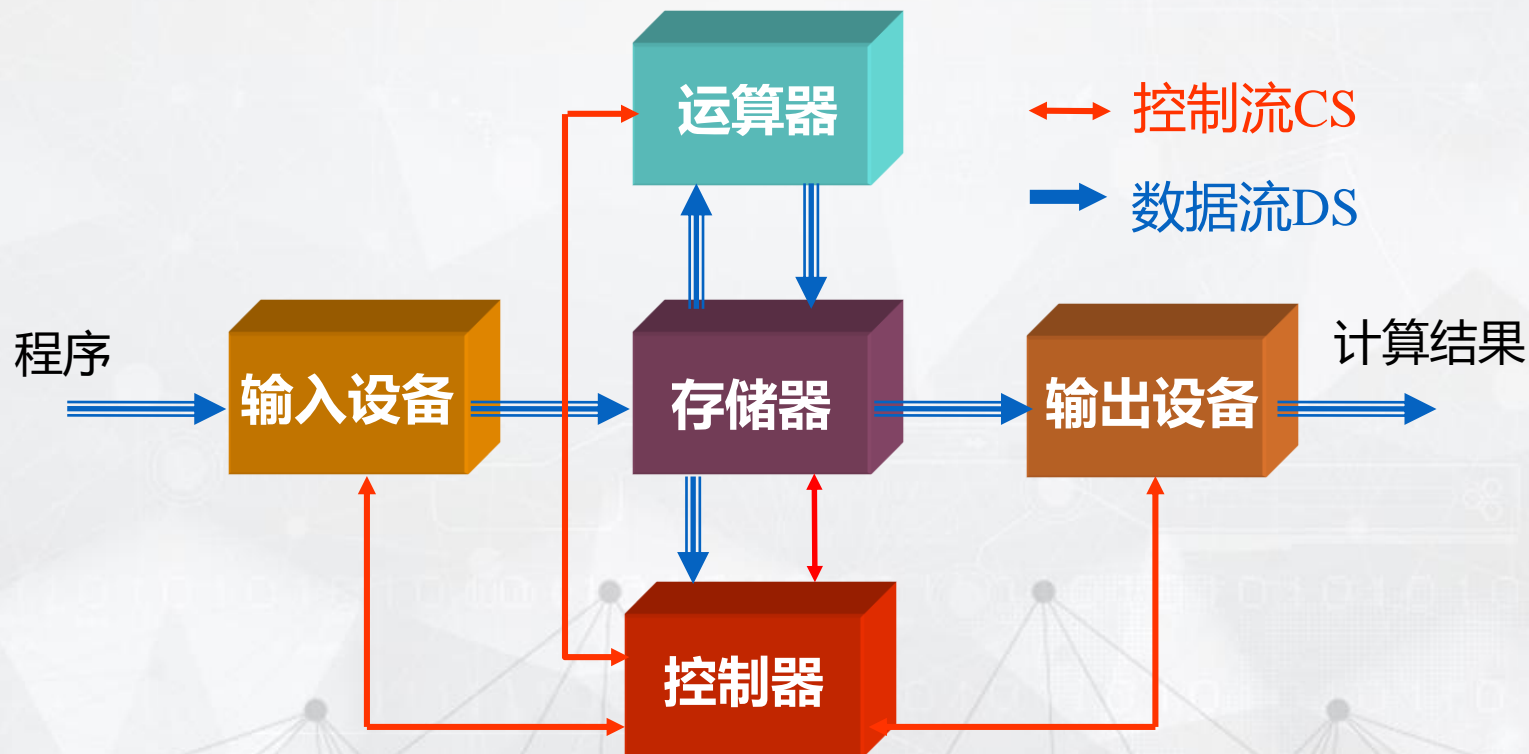
2024. 6



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。



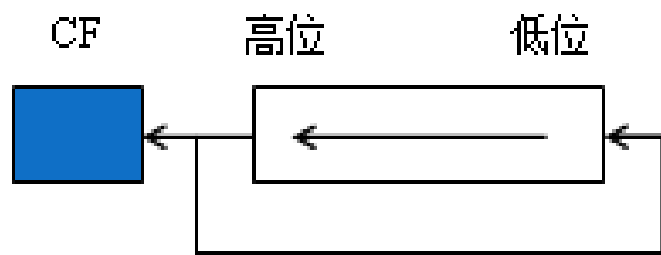
带移位模型机指令与控制台命令

——9条机器指令，3条控制台命令

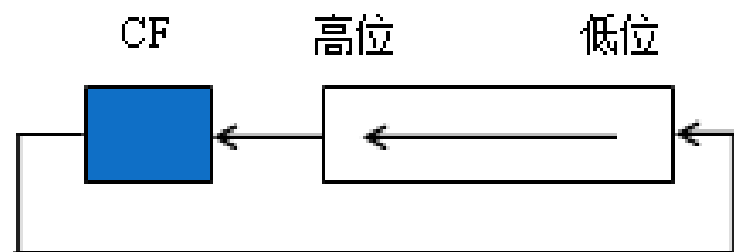
助记符	机器指令	说明
IN	00000000	“Input Device” →R0
ADD Addr	00010000 XXXXXXXX	$R0+[Addr] \rightarrow R0$
STA Addr	00100000 XXXXXXXX	$R0 \rightarrow [Addr]$
OUT Addr	00110000 XXXXXXXX	$[Addr] \rightarrow$ “Output Device”
JMP Addr	01000000 XXXXXXXX	$Addr \rightarrow PC$
RR	0101 0000	R0循环右移一位
RRC	0110 0000	R0带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

SWB	SWA	控制台命令
0	0	读内存 (KRD)
0	1	写内存 (KWE)
1	1	启动程序 (RP)

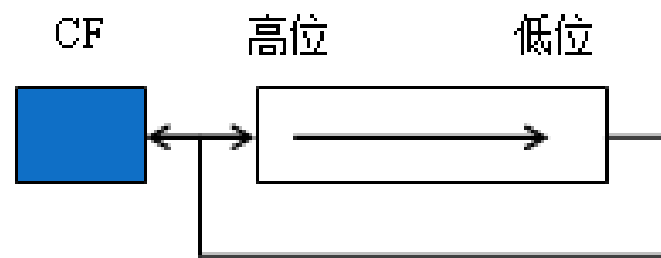
移位运算功能



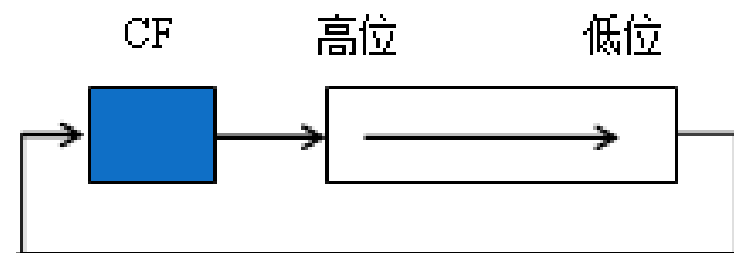
(a) 循环左移



(b) 带进位循环左移

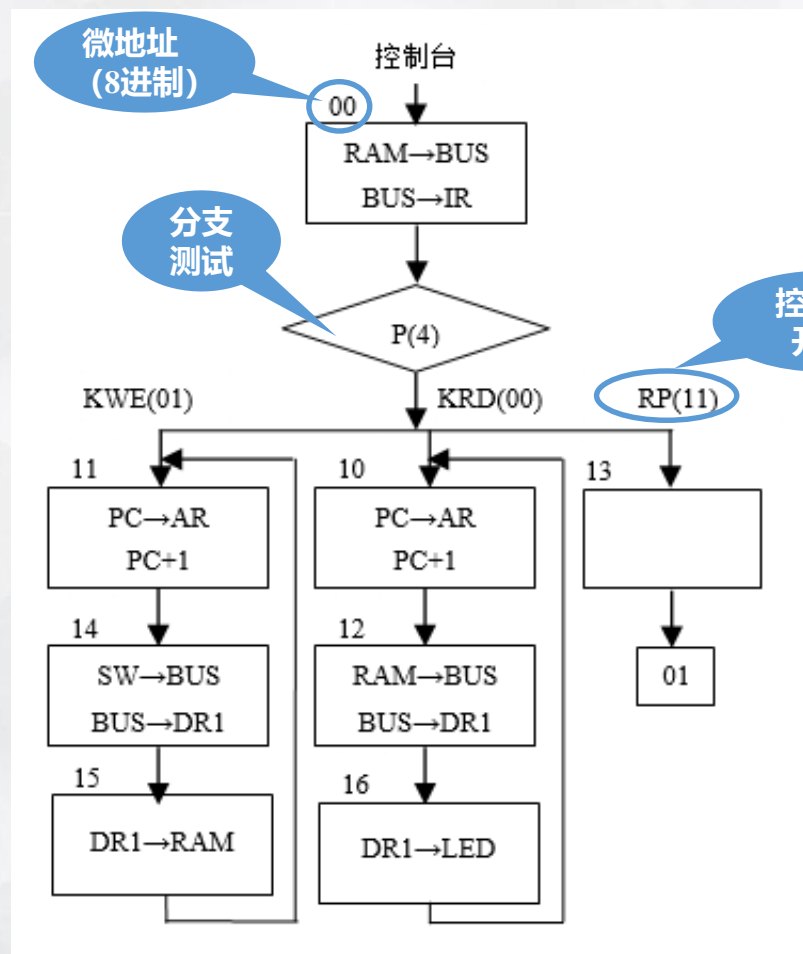


(c) 循环右移

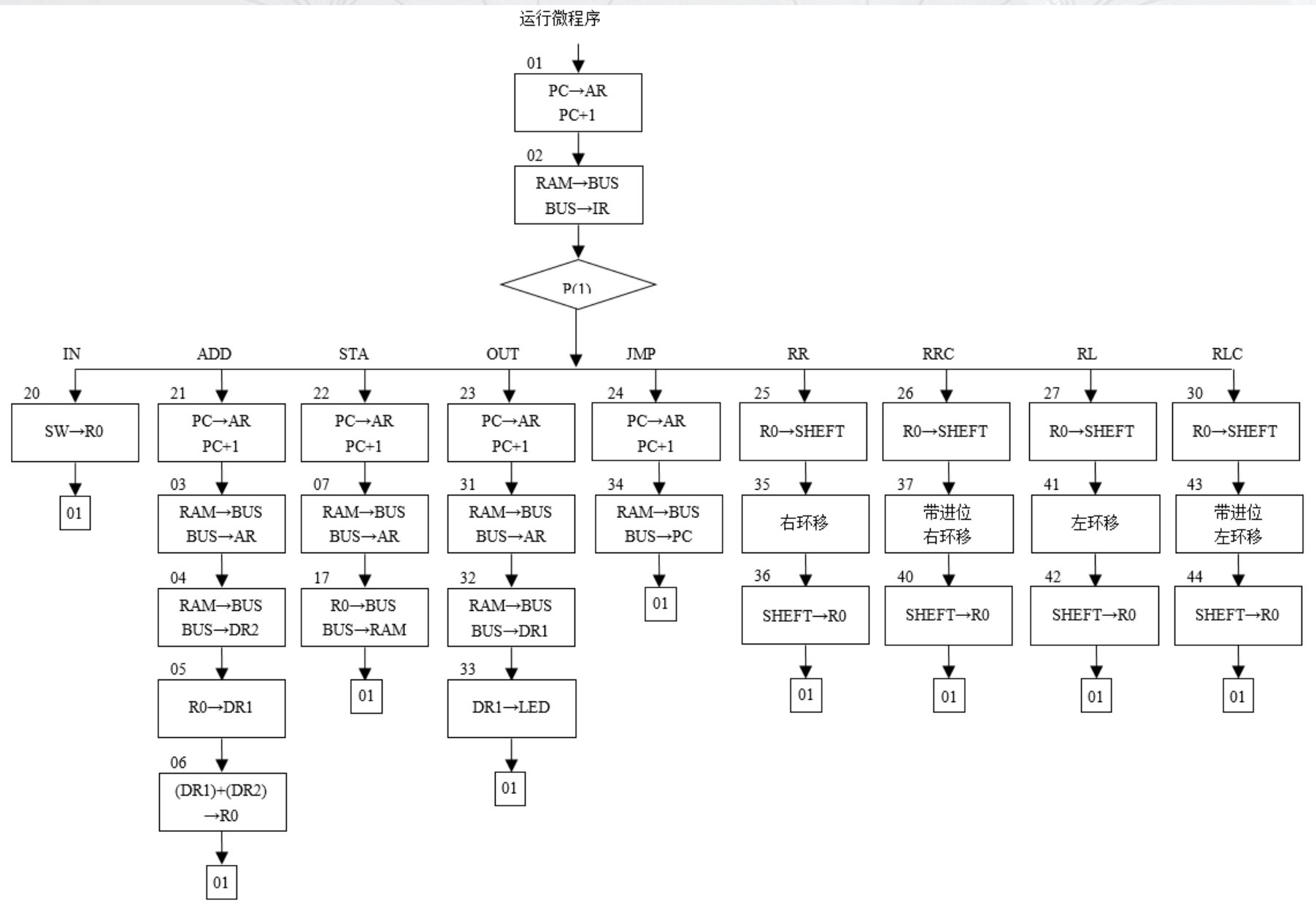


(d) 带进位循环右移

带移位功能模型机微程序流程



带移位功能模型机微程序流程



带移位功能模型机微代码定义

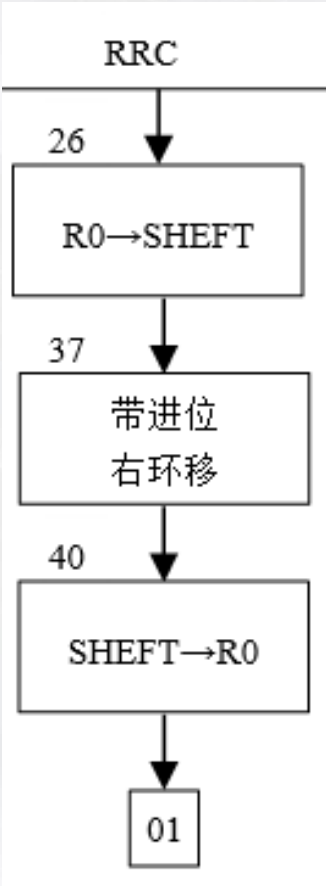
微命令编码格式定义

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	RS_B	0	0	1	P (1)
1	0	LED_B	0	1	0	LDDR1	0	1	0	RD_B	0	1	0	P (2)
1	1		0	1	1	LDDR2	0	1	1	RI_B	0	1	1	P (3)
			1	0	0	LDIR	1	0	0	SHE_B	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	

带进位循环右移微程序流程与微代码



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 0 1 0 0 0	018108
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 1 0 0 0 0	00C050
...
2 5	0	0	1	1	0	0	0	1	1	000	001	000	0 1 1 1 0 1	31821D
2 6	0	0	1	1	0	0	0	1	1	000	001	000	0 1 1 1 1 1	31821F
2 7	0	0	1	1	0	0	0	1	1	000	001	000	1 0 0 0 0 1	318221
3 0	0	0	1	1	0	0	0	1	1	000	001	000	1 0 0 0 1 1	318223
...
3 7	0	0	1	0	1	0	0	1	1	000	100	000	1 0 0 0 0 0	298820
4 0	0	0	0	0	0	0	0	1	1	001	100	000	0 0 0 0 0 1	019801
...

带移位模型机控制信号的存储

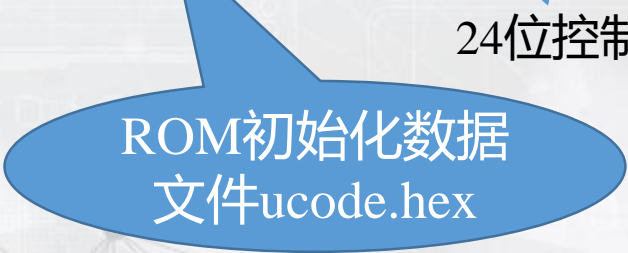
微地址 (8 进制)	S3 S2 S1 S0	M Cn	WE A9 A8	A	B	C	UA5——UA0	微指令 (16 进制)
0 0	0 0 0 0	0 0	0 1 1	000	000	100	0 1 0 0 0 0	018110
0 1	0 0 0 0	0 0	0 1 1	110	110	110	0 0 0 0 1 0	01ED82
0 2	0 0 0 0	0 0	0 0 1	100	000	001	0 0 1 0 0 0	00C048
0 3	0 0 0 0	0 0	0 0 1	110	000	000	0 0 0 1 0 0	00E004
0 4	0 0 0 0	0 0	0 0 1	000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

需要更换成9条指令模型机的微代码表

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	100101011001101000000001	000000001110000000001101	000000000010000000000001
011	000000011110110110000011	000000011110110110000111	000000011110110110001110
014	000000011110110110010110	000000011100000100000001	00000001110000000001111
017	000000001010000000010101	000000011110110110010010	000000011110110110010100
022	000000001010000000010111	000000011000000000000001	000000000010000000011000
025	000001010000101000000001	000000001101000110000001	000001010000101000010000
030	000001101000101000010001	000000000000000000000000	000000000000000000000000
033	000000000000000000000000	000000000000000000000000	000000000000000000000000

24位控制信号



带移位功能模型机系统设计步骤

- 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。参考9条指令功能表

- 第二步：数据通路设计

参考带移位功能模型机数据通路图

- 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。

实验任务与步骤

- 1、新建工程，新建框图文件 (*.bdf)，设计带移位模型机电路。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和自定义元件（双向移位寄存器）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

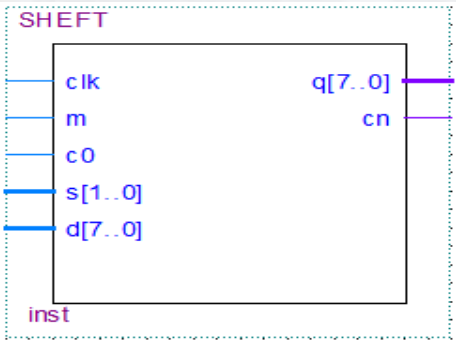
主菜单 “Assignmemts” → “Device” 项，选择Cyclone IV E系列
EP4CE55F23C8芯片

- 3、编译电路

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

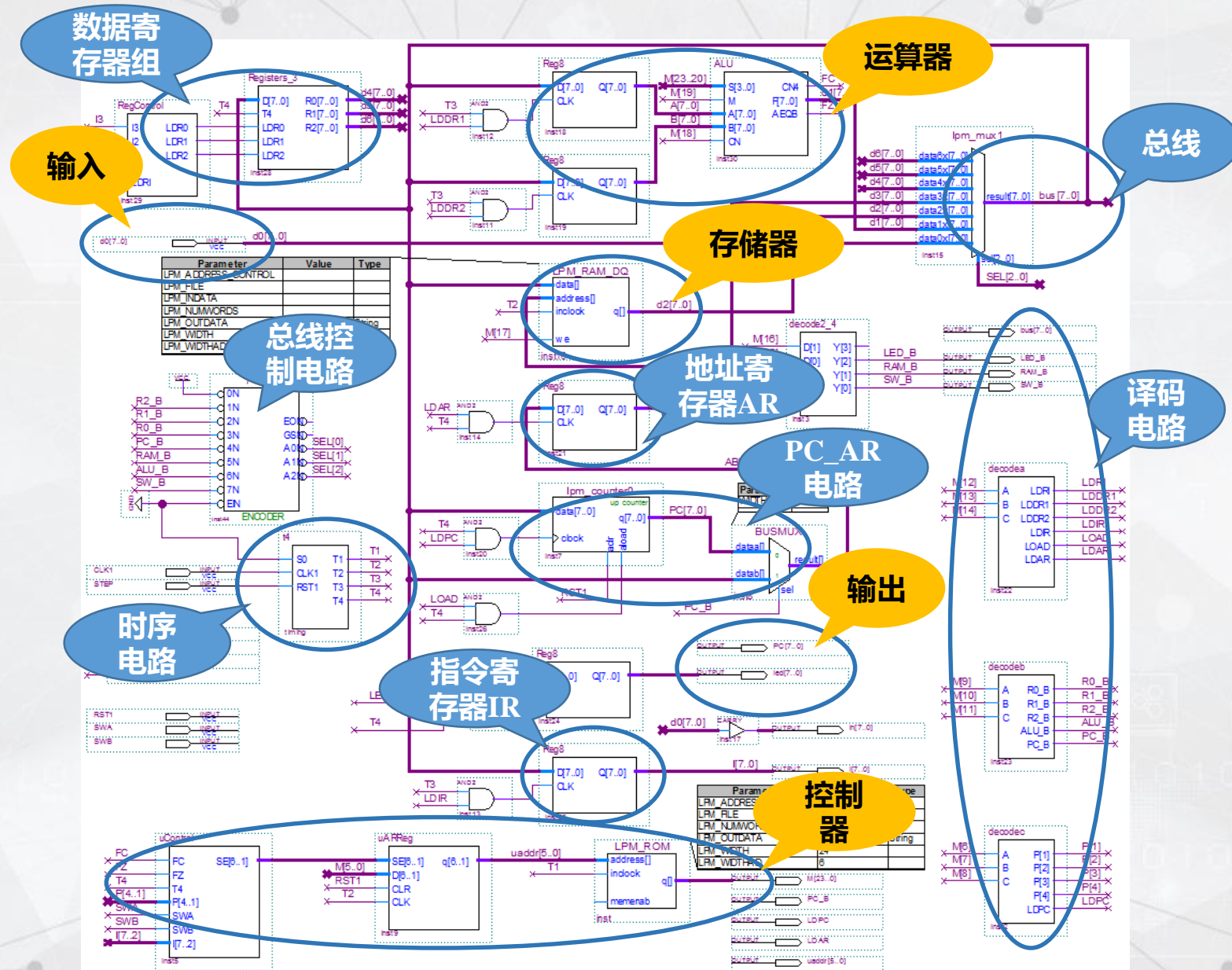
自定义双向移位寄存器元件

- 1.利用框图设计双向移位寄存器电路，电路设计文件sheft.bdf，将sheft.bdf拷贝到工程目录
- 2.主菜单“File”→“Create/Update”项，选择“Create Symbol Files for Current File”
由sheft.bdf生成sheft.bsf，即生成自定义双向移位寄存器元件符号
- 3.在元器件库中，在Project目录下选择自定义元件sheft，加入到电路图中



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

基本模型机电路图



带移位模型机电路设计与波形图仿真要求

1、设计移位指令，编写微程序代码，实现带移位运算功能的模型机系统。

2、在基本模型机基础上，增加双向移位寄存器SHEFT。

方式一：将SHEFT部件挂到总线上，总线多加一路分支。（必做）

方式二：ALU输出接到SHEFT数据输入端，SHEFT输出端接总线。（选做）

方式二，需要考虑ALU和SHEFT公用的控制信号S和M如何分配？是否可行？

3、扩展微控制器编码/译码电路（`decoderb`和74148），增加SHE_B控制信号，实现双向移位寄存器读总线和写总线控制。完成电路设计和编译。

3、利用9条机器指令编写具有运算或者控制等功能的应用程序。完成波形图仿真。

如何生成自定义元件？

选择主菜单“File”中的“Create/Update”→“Create Symbol Files For Current File”项，即可将当前电路（*.bdf或者*.vhd）变成一个元件符号存盘，以待在高层次设计中调用。

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”

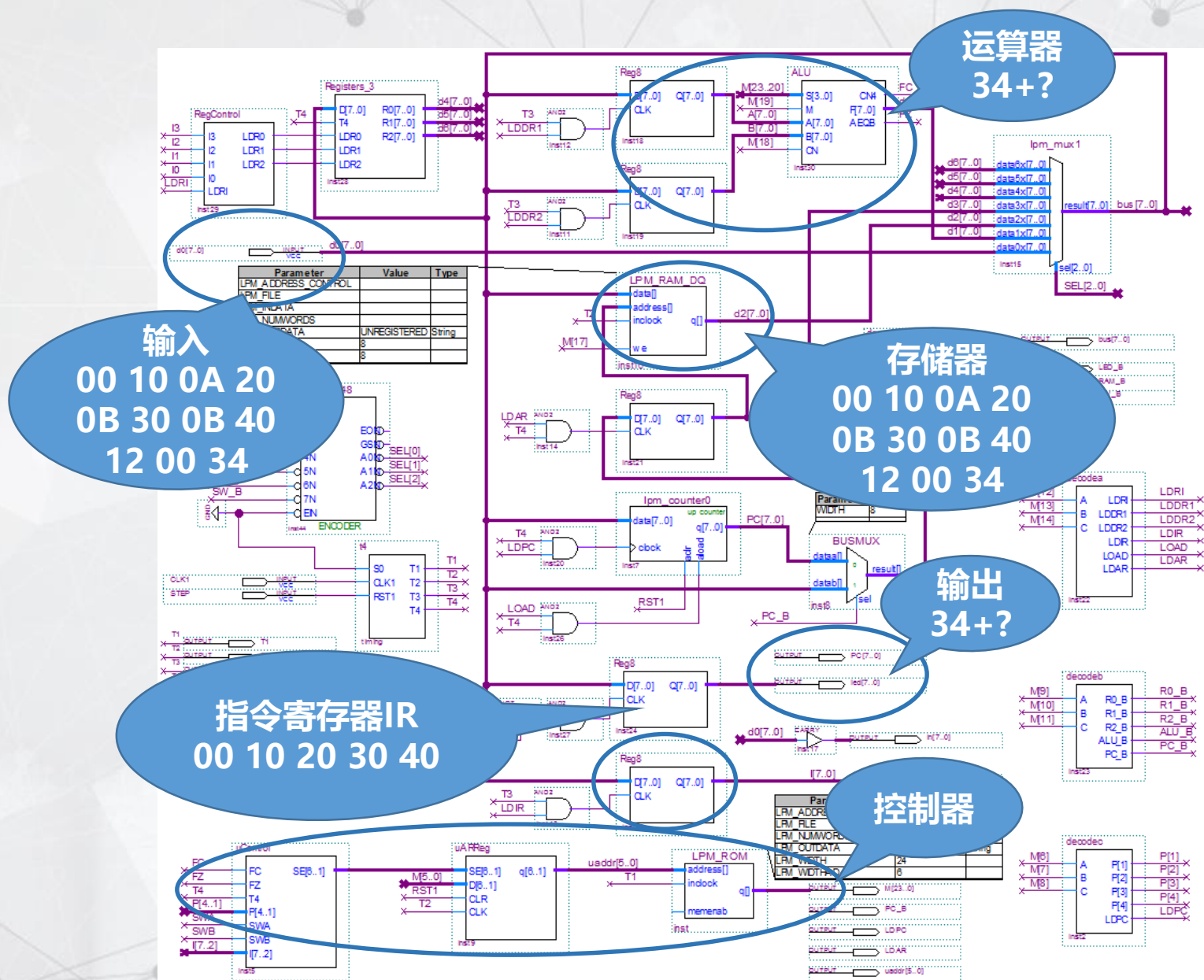
运行仿真：波形编辑器窗口主菜单“Simulation”→“Run Functional Simulation”项。

示例：基本模型机程序代码

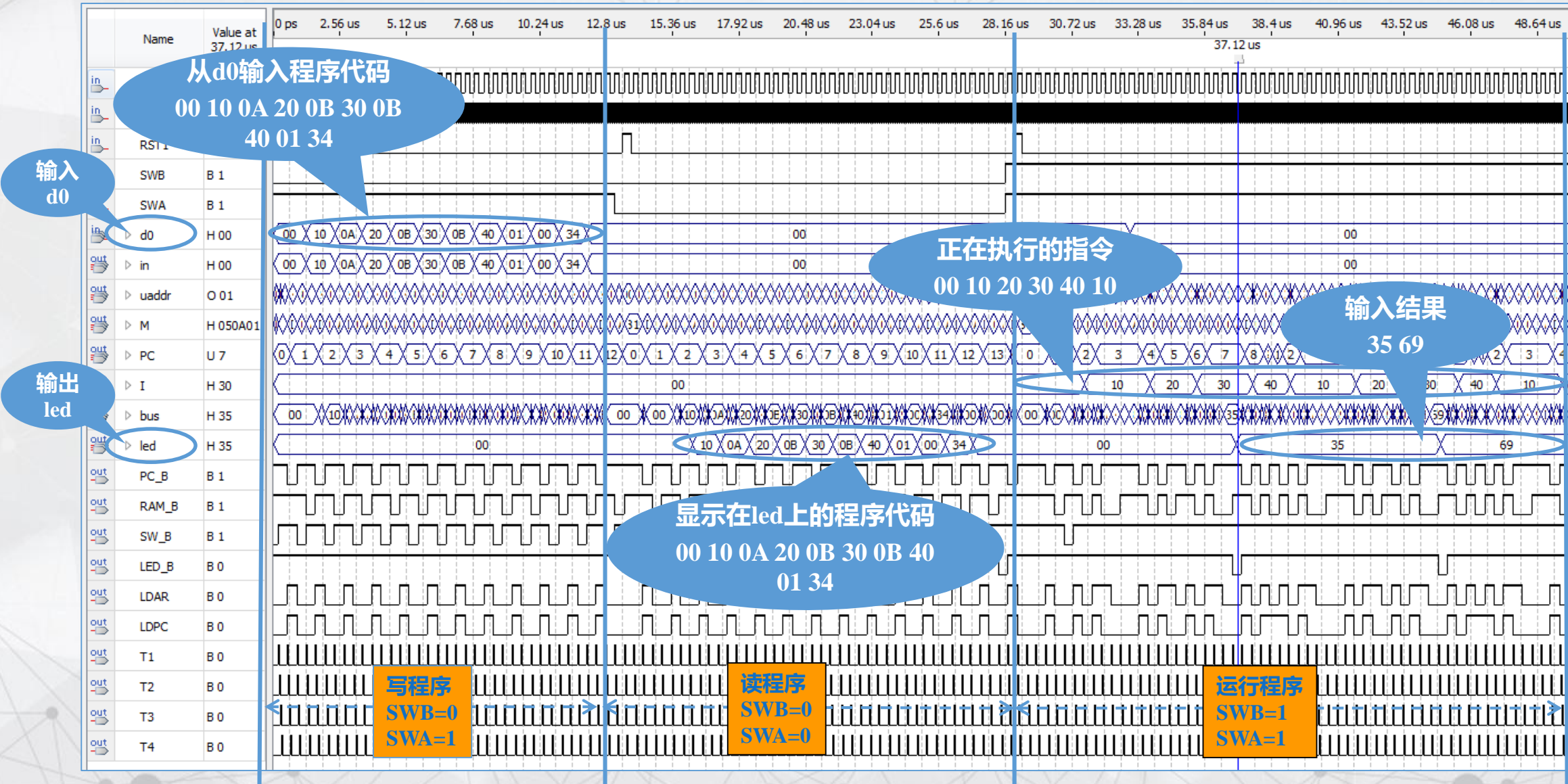
00 10 0A 20 0B 30 0B 40 01 XX 34

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

示例：基本模型机-执行程序



示例：基本模型机仿真图（总体）



实验任务与步骤

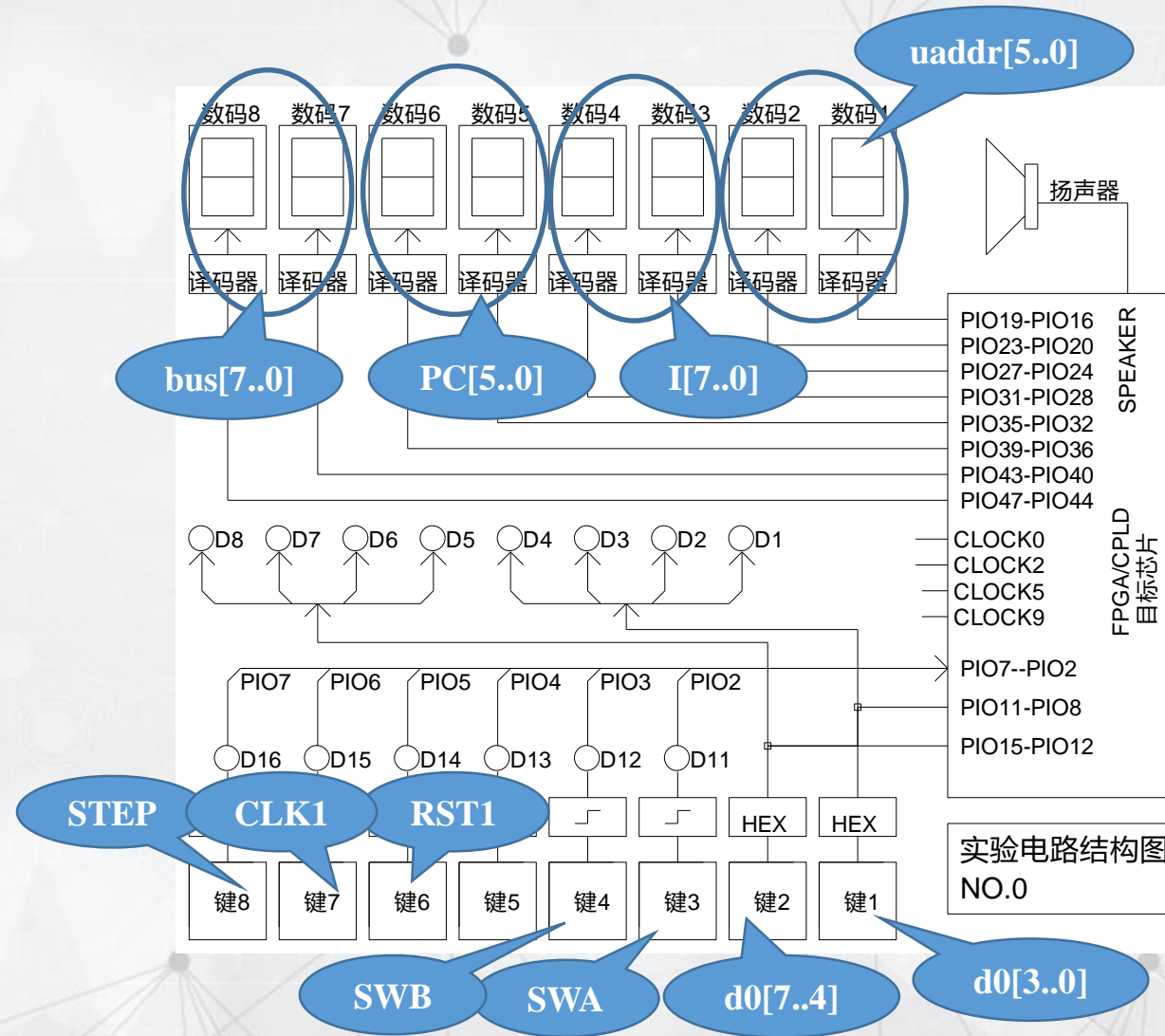
5、选择KX-CDS实验台，选择合适的电路模式结构，例如NO.0，对照电路模式图和引脚表，查找引脚号。打开编程器，输入引脚号，对电路进行引脚锁定，编译工程。

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



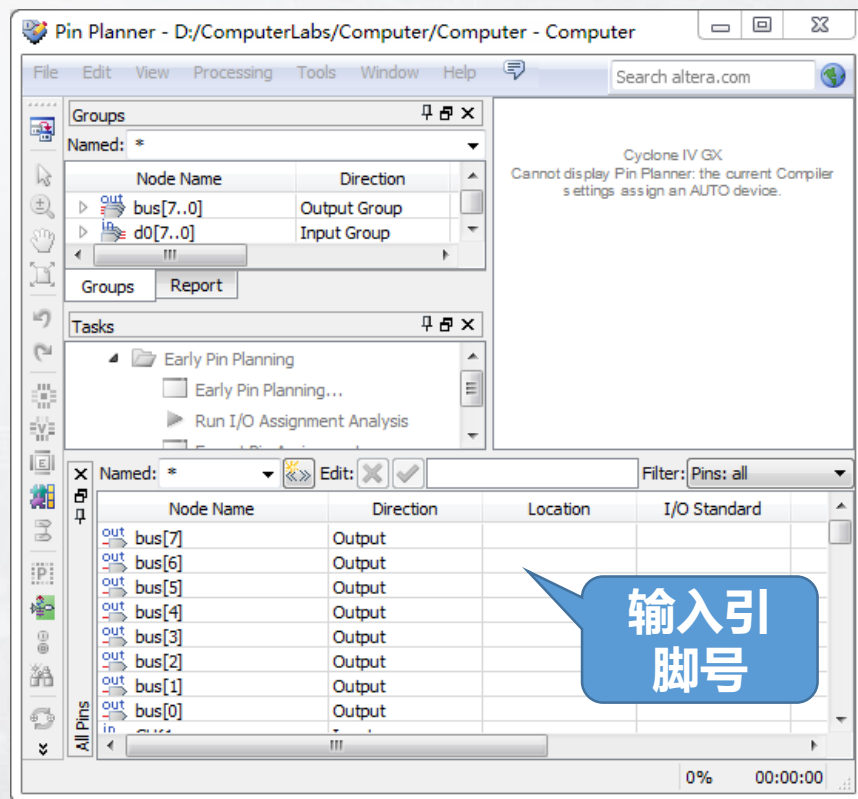
FPGA实验台引脚锁定

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



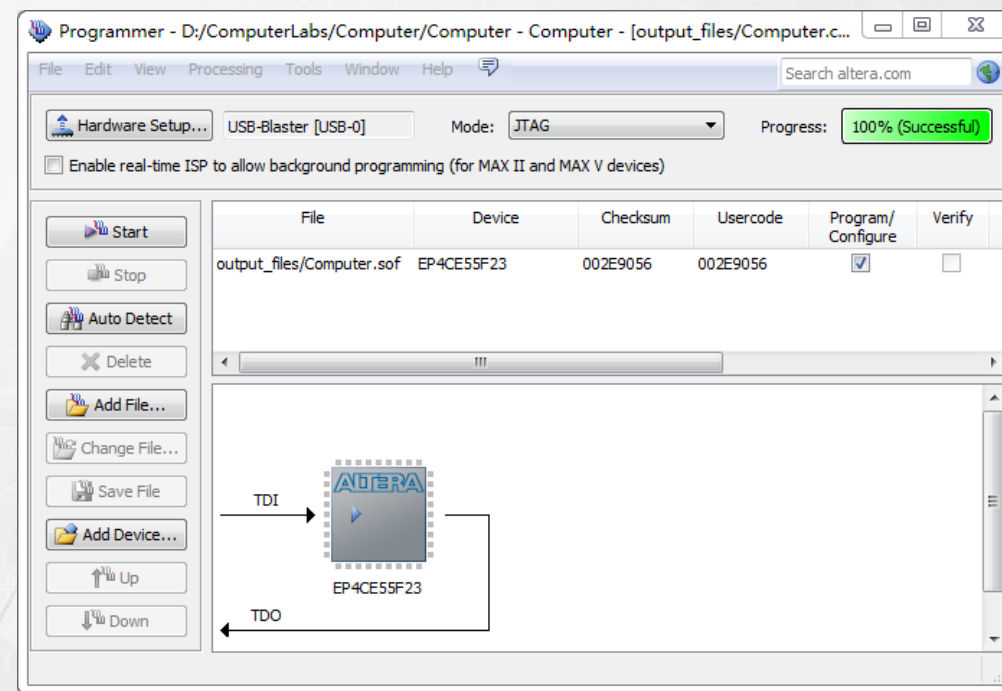
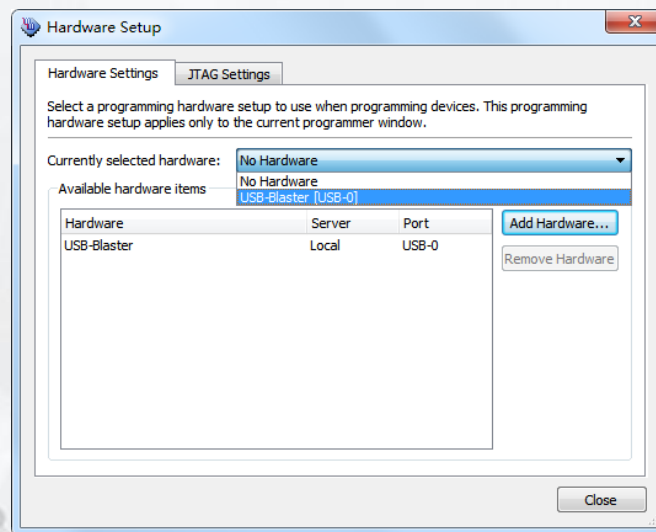
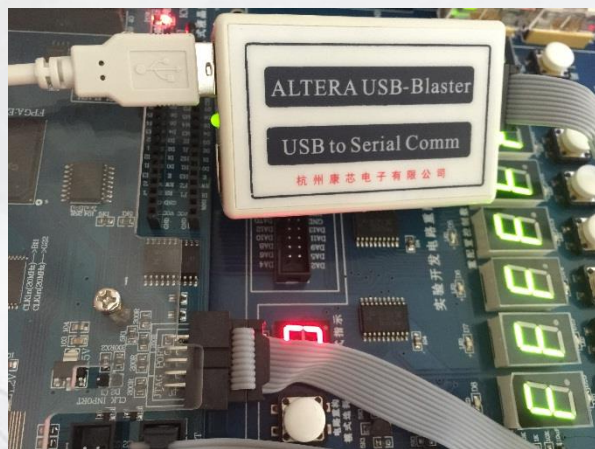
连接实验台下载电路

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。

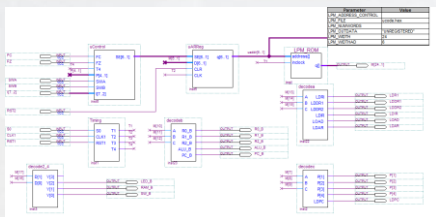
注意：实验台需要打开电源，并且将其JTAG接口与计算机通过USB线连接。

KX-CDS实验台

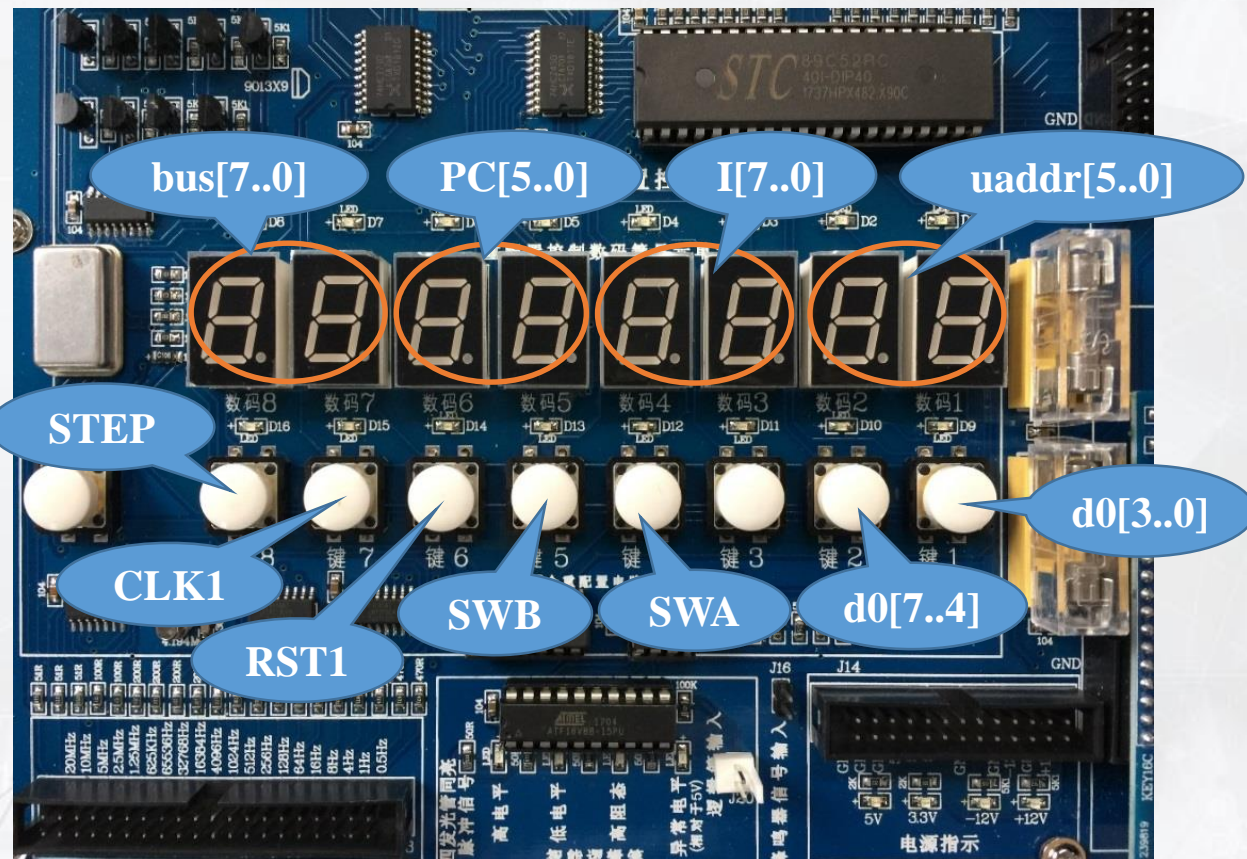
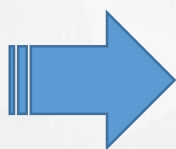


带移位功能模型机实验台演示

KX-CDS实验台



下载



现在开始实验（第2、3次课）！

带移位功能模型机电路设计、编译和仿真

1、基本模型机系统原理：参考教材7.1.1

参考ppt：双向移位寄存器实验.ppt

参考ppt：带移位功能(9条指令)模型机系统设计.ppt

参考电路：基本模型机电路、双向移位寄存器电路

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。编写带移位模型机程序代码，完成仿真波形图，仿真要求：测试9条指令。

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