



实验2 计算机基本部件与总线控制 实验 (第3次课)

2022.10



哈尔滨工程大学计算机实验教学中心

ALU实验

实验目的

- 1.掌握ALU电路结构和工作原理。
- 2.掌握Quartus Prime软件环境和FPGA实验台的使用方法。
- 3.掌握利用框图输入法设计ALU电路的方法。
- 4.验证ALU功能。

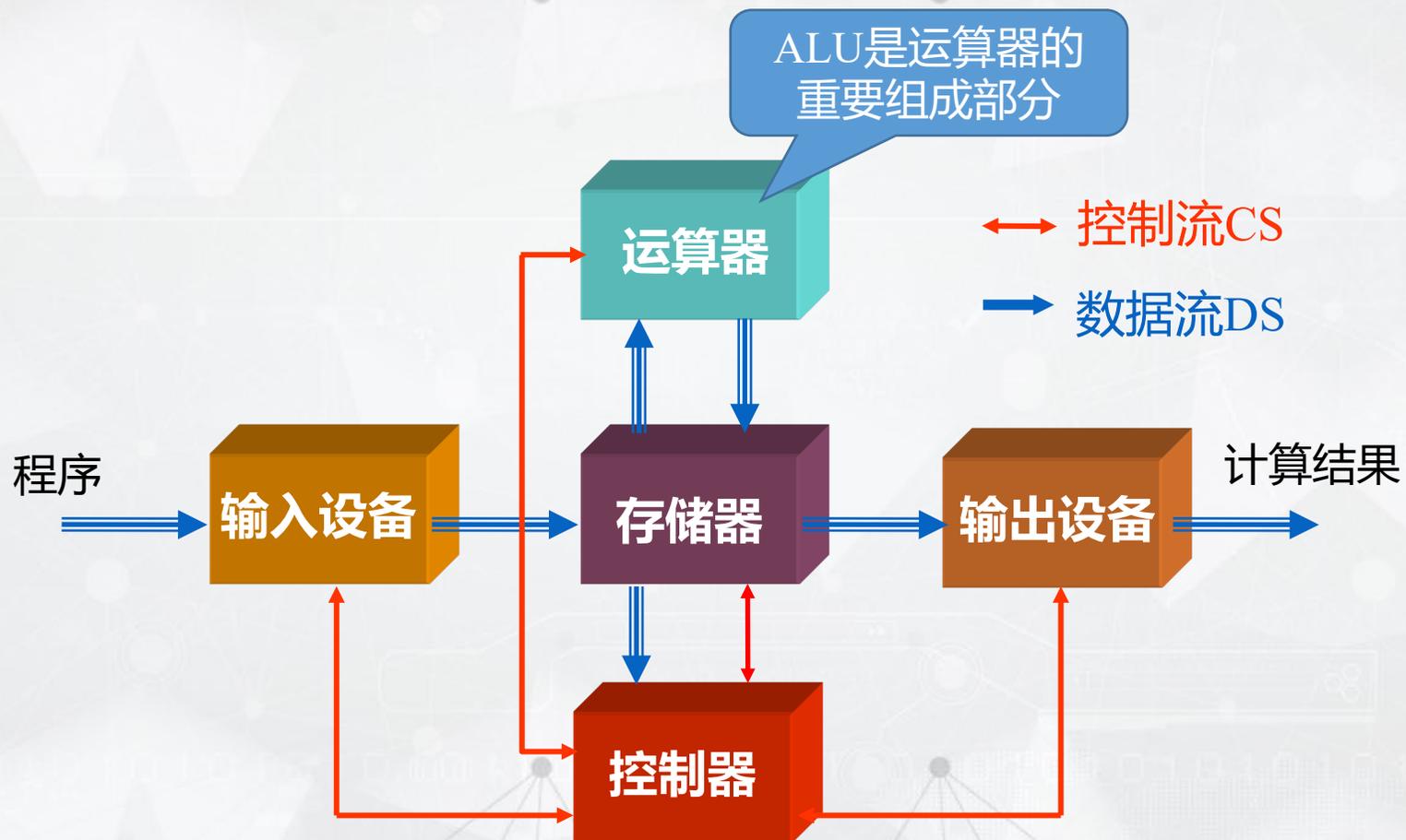
实验内容

实现算术、逻辑运算功能。完成电路的设计、仿真、编程下载和实验台演示。

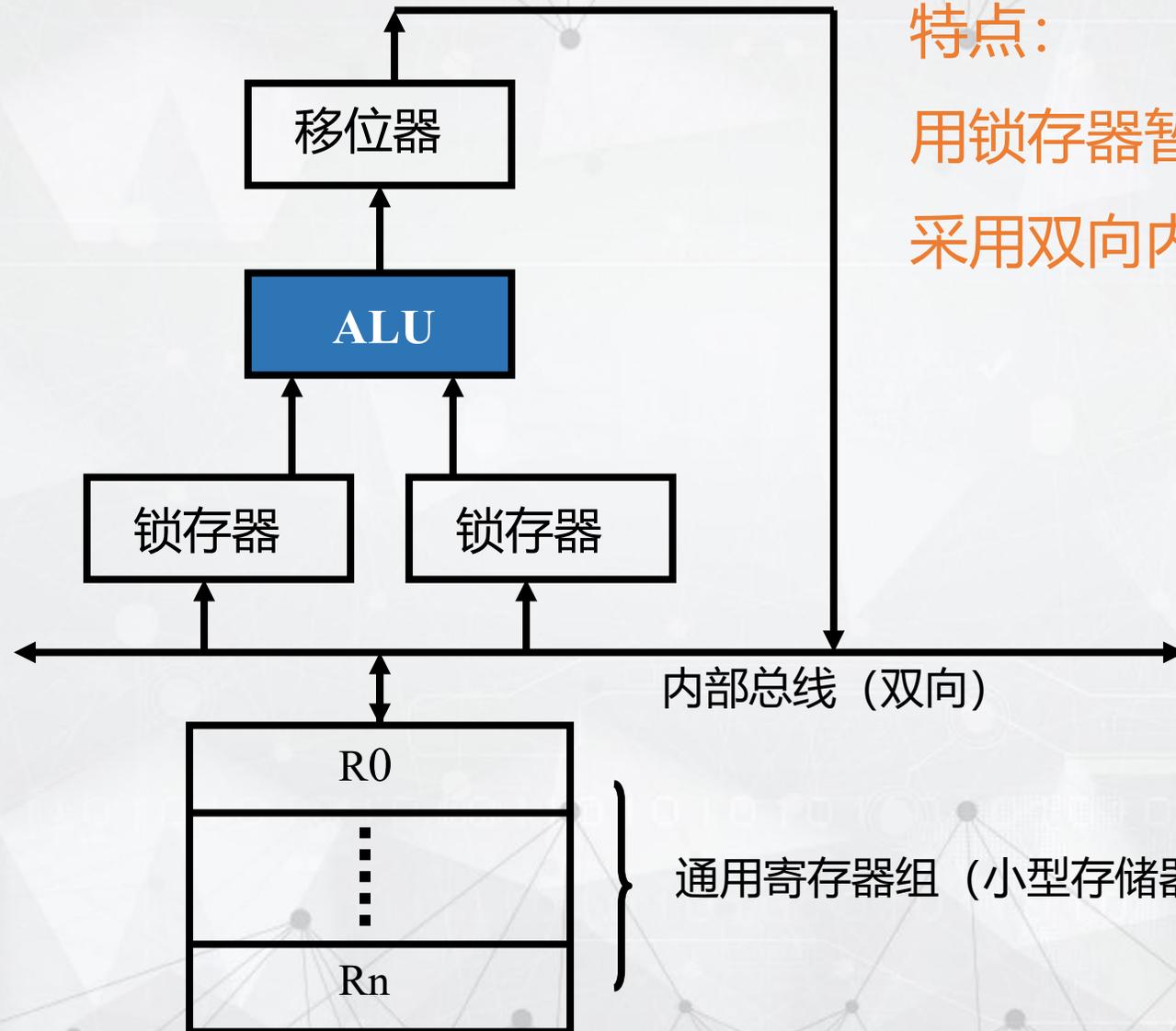
算术逻辑单元(ALU, arithmetic and logic unit)

- ✓ 运算器是计算机中执行各种算术和逻辑运算操作的部件。运算器的基本操作包括加、减、乘、除四则运算，与、或、非、异或等逻辑操作，以及移位、比较和传送等操作。
- ✓ 运算器是由算术逻辑单元(ALU)、累加寄存器、数据缓冲寄存器和状态条件寄存器等部件组成，它是数据加工处理部件。
- ✓ 算术逻辑单元(arithmetic and logic unit) 是实现多组算术运算和逻辑运算的组合逻辑电路，简称ALU。

ALU在基本模型机中的位置



ALU是运算器的重要组成部分



特点:

用锁存器暂存操作数;

采用双向内总线。

ALU实验基本原理

ALU是中央处理器(CPU)的执行单元，是专门执行算术和逻辑运算的数字电路。ALU是计算机中央处理器的核心组成部分。

实验中选用Quartus Prime\Maxplus2元器件库中的ALU，其中ALU可由两个74LS181以串联形式构成8位字长的ALU。也可以用VHDL等硬件描述语言来设计ALU。



ALU基本功能

选择				M=1 逻辑运算	M=0 算术运算	
S ₃	S ₂	S ₁	S ₀		Cn=1 (无进位)	Cn=0 (有进位)
0	0	0	0	$F=A$	$F=A$	$F=A+1$
0	0	0	1	$F=A+B$	$F=A+B$	$F=(A+B)+1$
0	0	1	0	$F=A \cdot B$	$F=A+/B$	$F=(A+/B)+1$
0	0	1	1	$F=0$	$F=-1$	$F=0$
0	1	0	0	$F=A \cdot B$	$F=A+A \cdot /B$	$F=A+A \cdot /B+1$
0	1	0	1	$F=A+/B$	$F=(A+B)+A \cdot /B$	$F=(A+B)+A \cdot /B+1$
0	1	1	0	$F=A \oplus B$	$F=A-B-1$	$F=A-B$
0	1	1	1	$F=A \cdot /B$	$F=A \cdot /B-1$	$F=A \cdot /B$
1	0	0	0	$F=A+B$	$F=A+A \cdot B$	$F=A+A \cdot B+1$
1	0	0	1	$F=A \ominus B$	$F=A+B$	$F=A+B+1$
1	0	1	0	$F=B$	$F=(A+/B)+A \cdot B$	$F=(A+/B)+A \cdot B+1$
1	0	1	1	$F=A \cdot B$	$F=A \cdot B-1$	$F=A \cdot B$
1	1	0	0	$F=1$	$F=A+A$	$F=A+A+1$
1	1	0	1	$F=A+/B$	$F=(A+B)+A$	$F=(A+B)+A+1$
1	1	1	0	$F=A+B$	$F=(A+/B)+A$	$F=(A+/B)+A+1$
1	1	1	1	$F=A$	$F=A-1$	$F=A$

注：+表示逻辑或，+表示算术加

实验内容与要求

✓在Quartus Prime软件中，利用HDL语言或框图设计8位运算器。

完成仿真，仿真要求：

(1)合理设置各个输入信号，仿真运算器的48种功能。算术运算和逻辑运算，操作数A和B任意设置。

(2)验证其中5种运算，人工手动计算得到结果(理论值)。然后从波形图上找到相应的运算结果(仿真值)，比较理论值和仿真值是否一致。

✓将并下载到FPGA实验台进行测试。

选择合适的电路模式结构，对电路进行引脚锁定，编程下载到FPGA实验台，直观的演示电路的功能。

实验任务与步骤

- 1、新建工程，新建框图文件 (*.bdf) ，设计输入ALU电路图。采用元器件库中 2片74ls181芯片串联。保存文件。

主菜单“File”→“New Project Wizard”，新建工程ALU（实体名）

主菜单“File”→“New”项，选择Block Diagram/Schematic File，新建框图文件，保存为 ALU.bdf。

- 2、设置器件

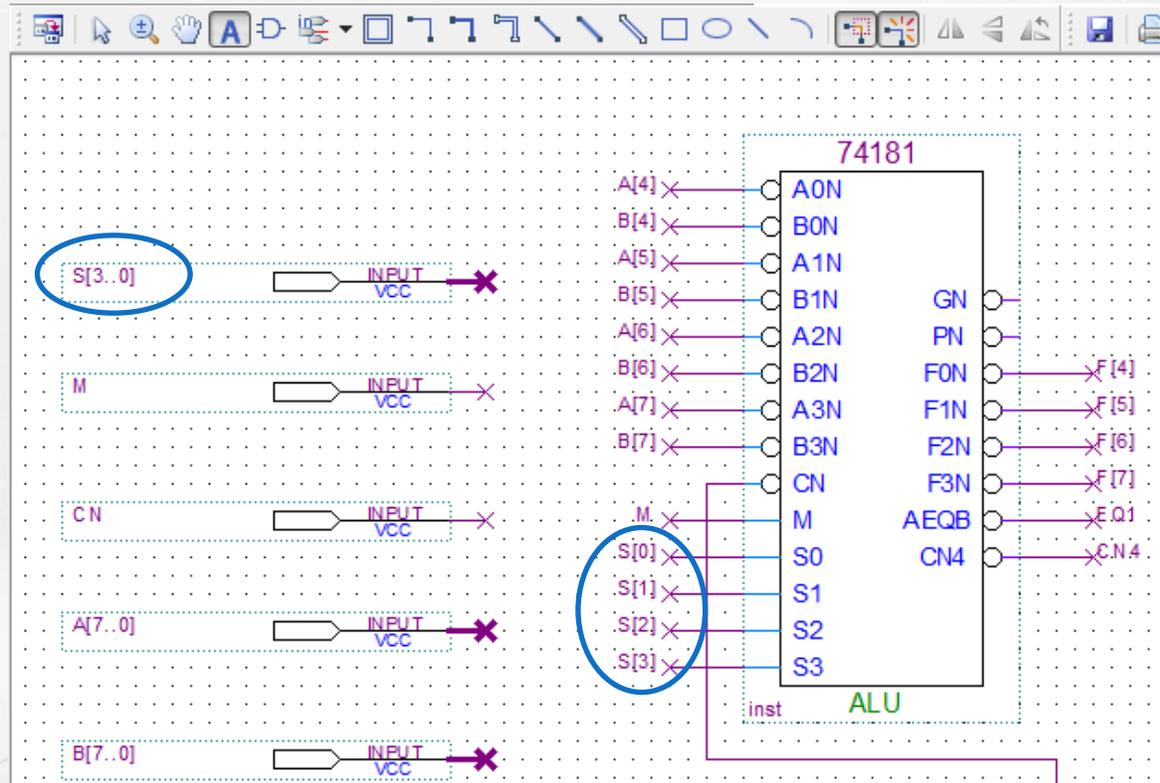
主菜单“Assignments”→“Device”项，选择Cyclone IV E系列 EP4CE55F23C8芯片

- 3、编译电路

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

通过“网络标号”来连接总线和端口

要通过“网络标号”来连接总线和端口,可以在连线上设置网络标号。即利用网络标号,两个具有相同名称的线和端口就连在一起。



利用属性对话框输入引脚名称（网络标号）

The image shows a logic design software interface with a 74181 component. The component has several pins: A[4], B[4], A[5], B[5], A[6], B[6], A[7], B[7], A[0], B[0], A[1], B[1], A[2], B[2], A[3], B[3], CN, M, S[0], S[1], S[2], S[3]. The pins are color-coded: purple for network labels and green for non-network labels. A callout points to a thick line labeled 'Bus型 (粗线)'. Another callout points to the 'Name' field in the 'Node Properties' dialog box, which contains 'S[0]', with the text '在此输入引脚名称 (网络标号)'. A third callout points to the 'Properties' option in the context menu, with the text '注意网络标号颜色为紫色，非标号颜色为绿色。引脚名称建议通过属性菜单来设置。'. The 'Node Properties' dialog box has tabs for 'General', 'Font', and 'Format'. The 'Name' field is highlighted with a blue circle. The 'Properties' option in the context menu is also highlighted with a blue circle.

Bus型 (粗线)

在此输入引脚名称 (网络标号)

注意网络标号颜色为紫色，非标号颜色为绿色。引脚名称建议通过属性菜单来设置。

Node Properties

General Font Format

Name: S[0]

Hide name in block design file.

OK

Cut Copy Paste Delete Toggle Connection Dot Conduit Line Bus Line Node Line Locate Flip Horizontal Flip Vertical Rotate by Degrees Zoom In Zoom Out Zoom... Fit in Window Fit Selection in Window Properties Add Node to SignalTap II Logic Analyzer

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

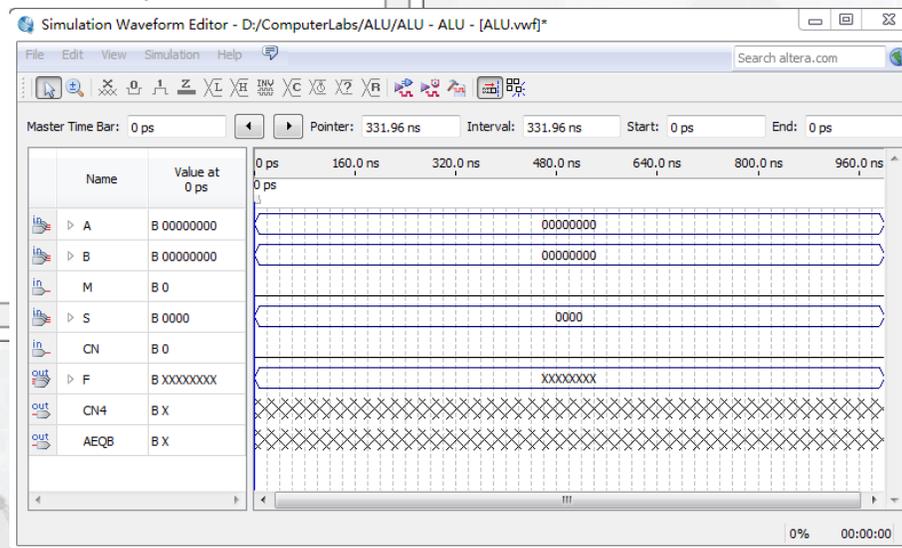
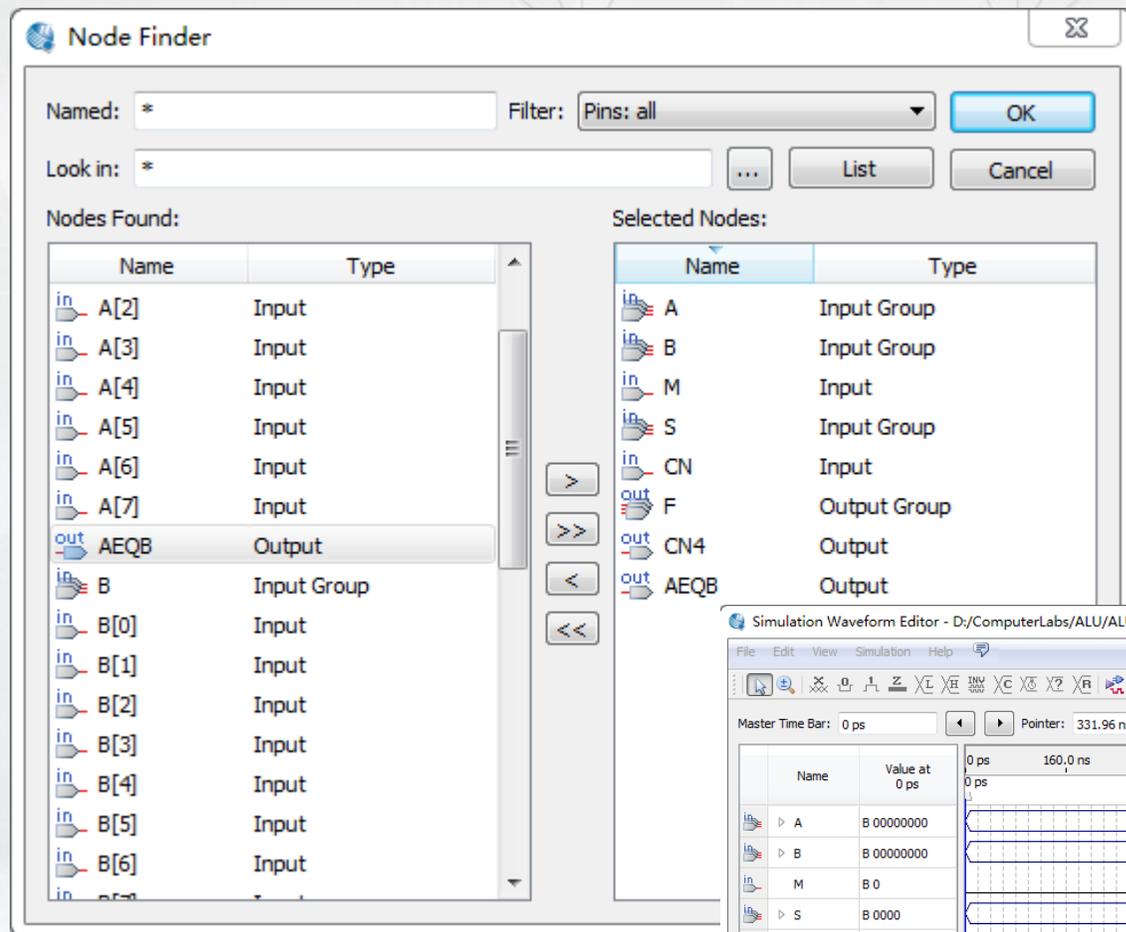
建立仿真波形文件：主菜单“File”→“New”项, 选择University Program VWF, 新建*.vwf, 打开波形编辑器。

设置仿真时间：主菜单“Edit”→“Set End Time”项。

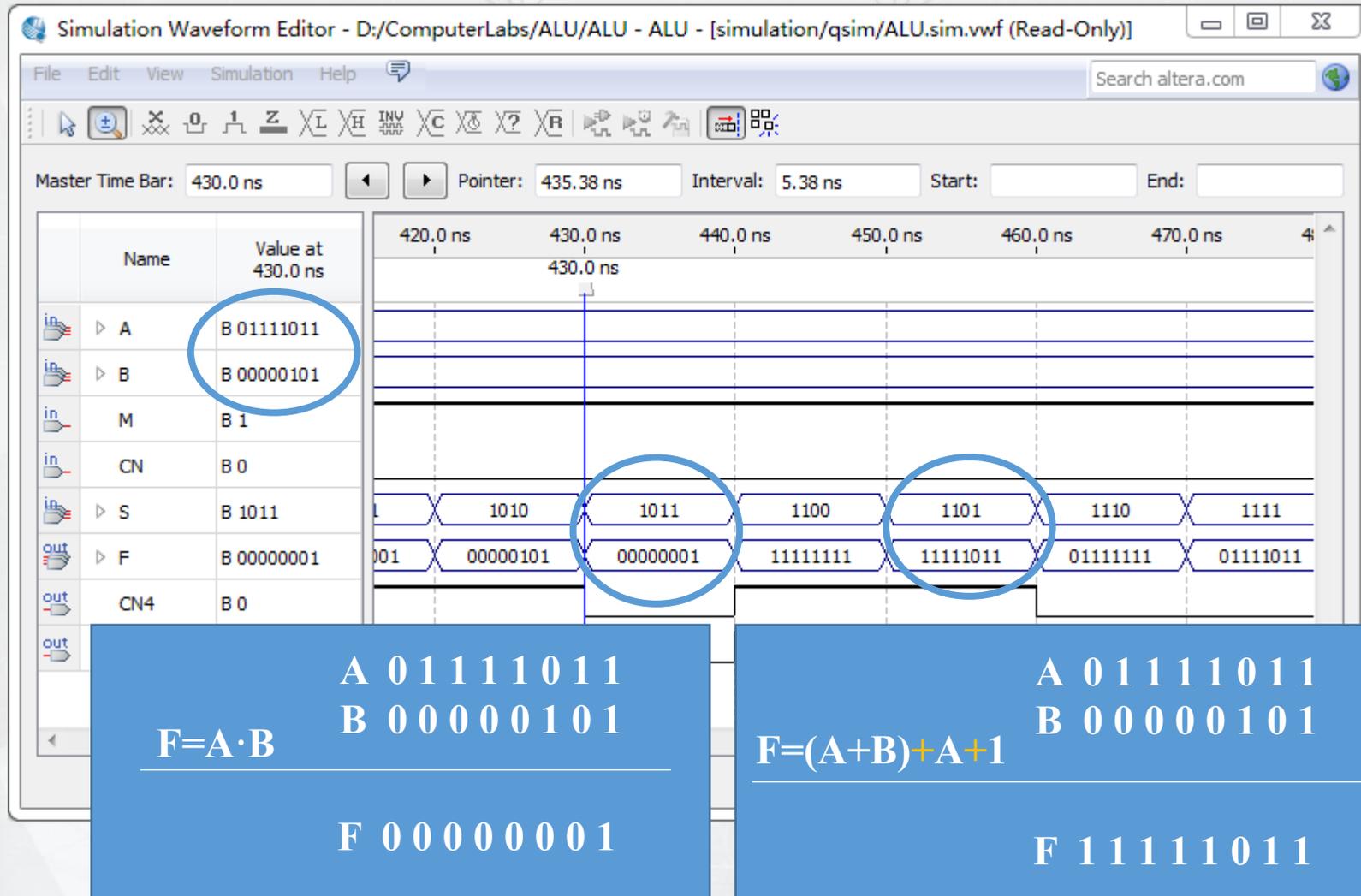
添加输入输出端口：波形编辑器窗口主菜单“Edit” → “Insert”→“Insert Node or Bus”

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

ALU仿真波形图——添加节点与总线



ALU仿真波形图分析示例



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.1，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。

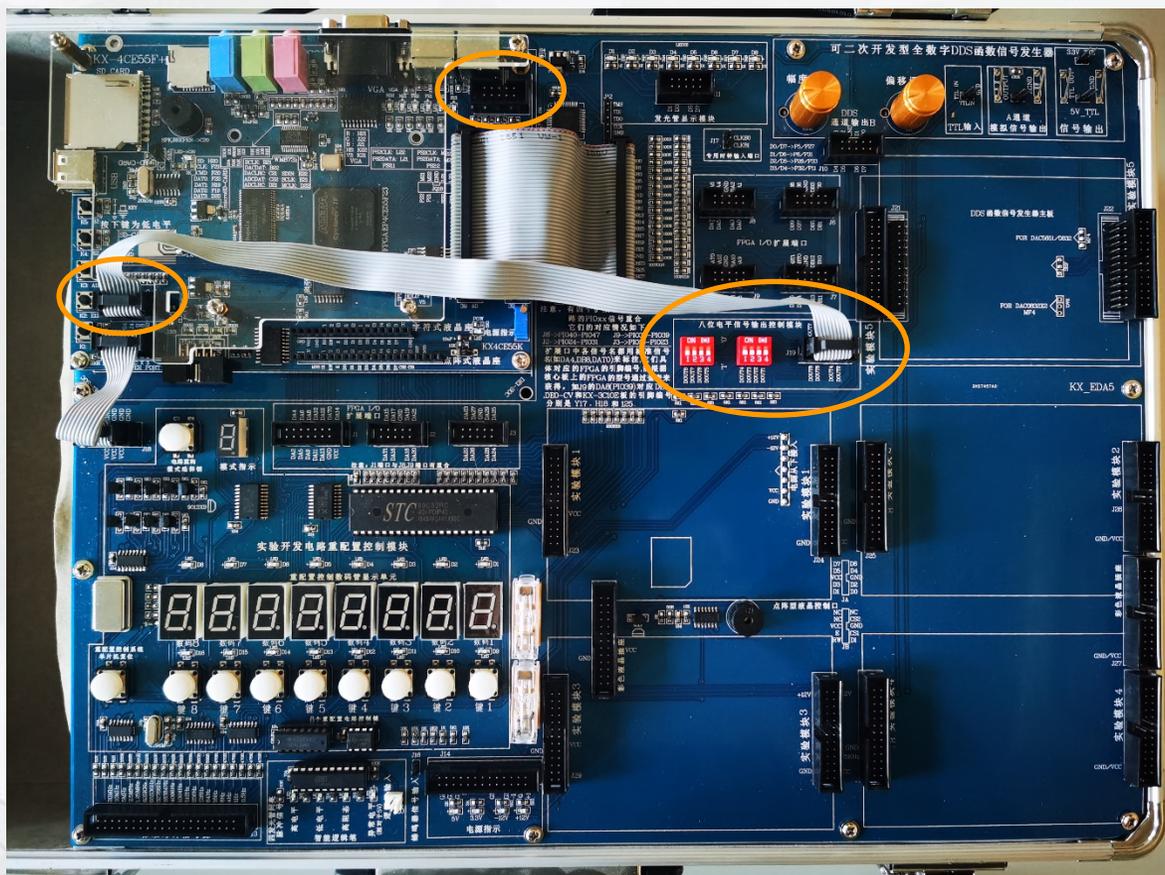
主菜单“Assignments”→“Pin”项，在Location栏中输入引脚号

6、下载sof文件到FPGA实验台，演示ALU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

FPGA实验台电平开关的使用

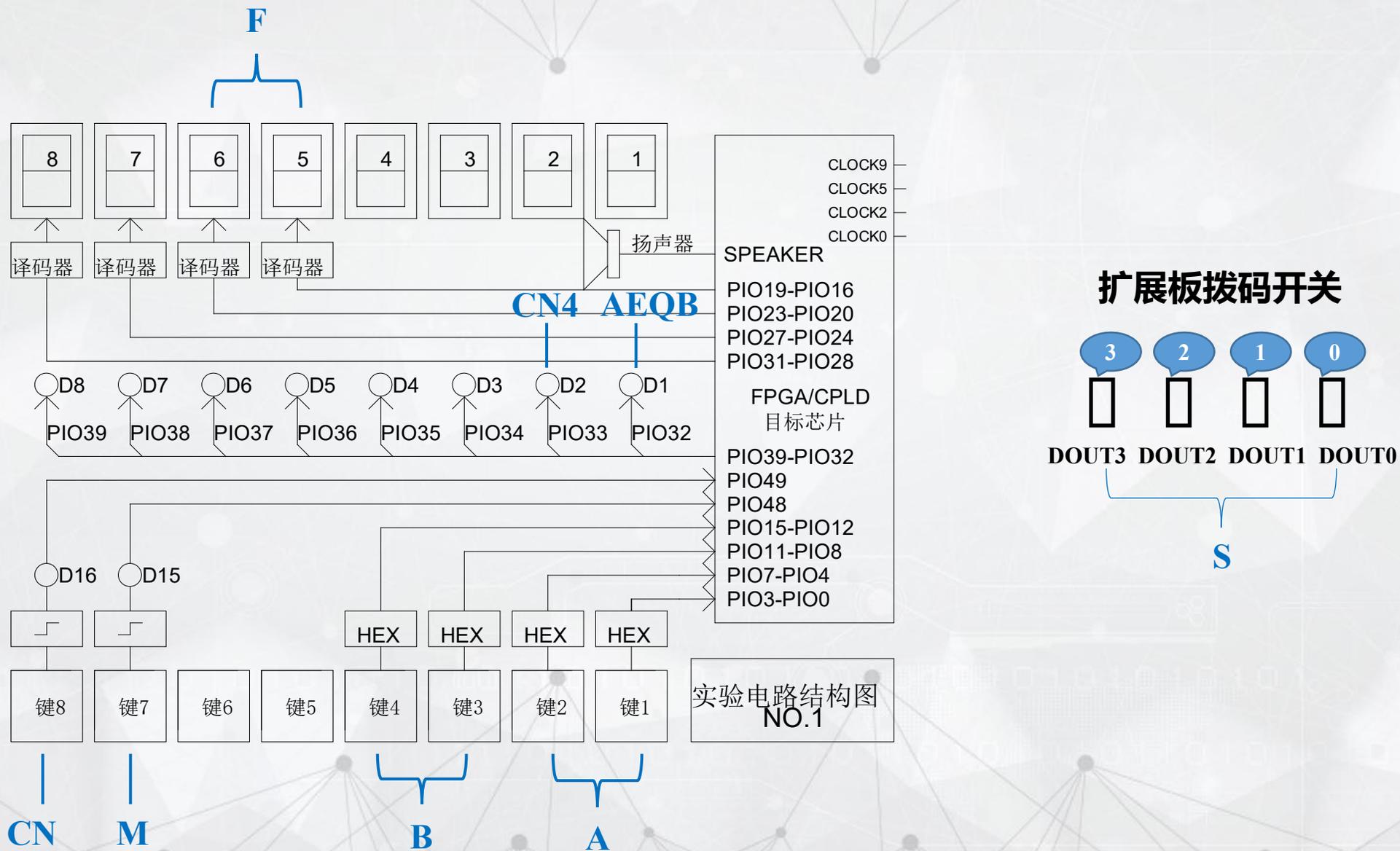
KX-CDS实验台



拨码开关，拨上为0，拨下为1
两个十芯口任选，通过排线连接拨码开关

核心板左下十芯口			
DOUT1	input	G4	外接J19拨码开关1
DOUT2	input	H4	外接J19拨码开关2
DOUT3	input	J4	外接J19拨码开关3
DOUT4	input	J3	外接J19拨码开关4
DOUT5	input	B2	外接J19拨码开关5
DOUT6	input	J5	外接J19拨码开关6
DOUT7	input	C2	外接J19拨码开关7
DOUT8	input	B1	外接J19拨码开关8
核心板右上JQ18十芯口			
DOUT1	input	L22	外接J19拨码开关1
DOUT2	input	R21	外接J19拨码开关2
DOUT3	input	L21	外接J19拨码开关3
DOUT4	input	R22	外接J19拨码开关4
DOUT5	input	M22	外接J19拨码开关5
DOUT6	input	P21	外接J19拨码开关6
DOUT7	input	M21	外接J19拨码开关7
DOUT8	input	P22	外接J19拨码开关8

选择FPGA实验台电路结构No.1

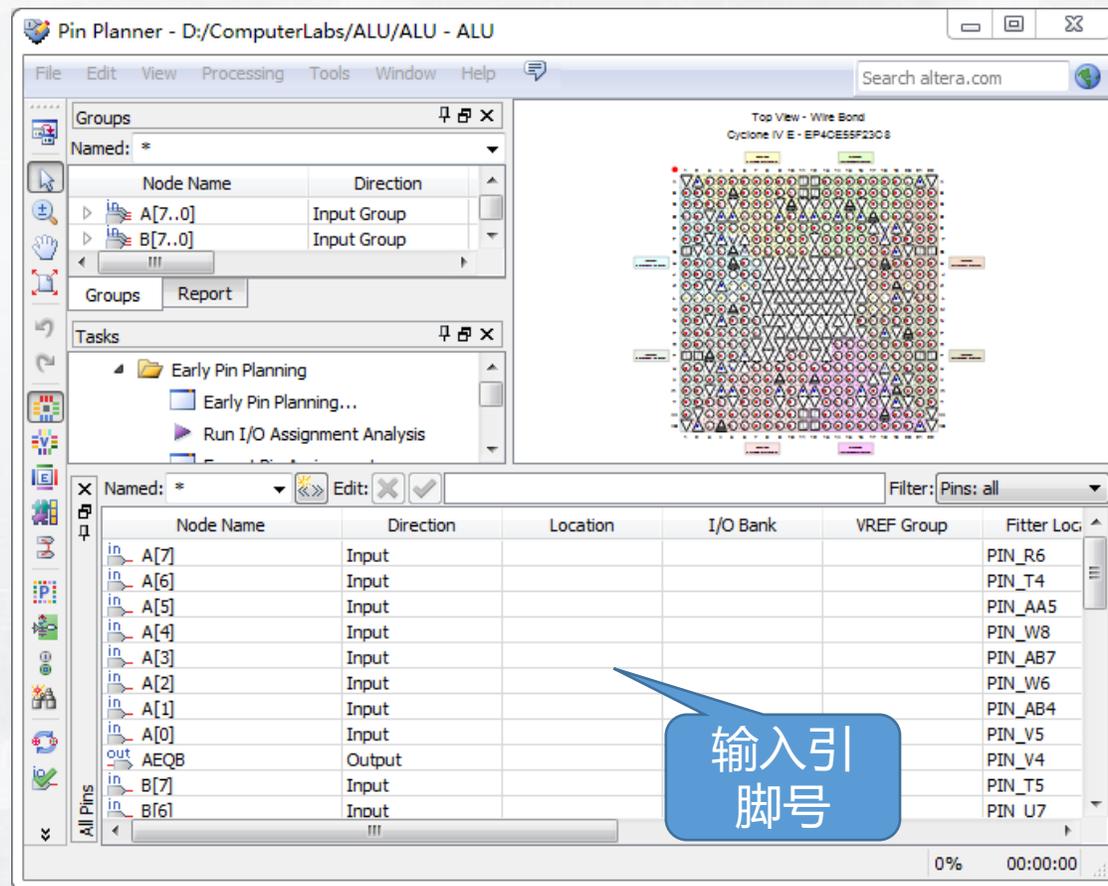


ALU引脚锁定方案(No.1)

输入/输出端口	外设	引脚名称	引脚号
CN	按键8	PIO49	
M	按键7	PIO48	
S[3..0]	拨码开关3、2、1、0	通过十芯口扩展DOUT3~DOUT0	
B[7..4]	按键4	PIO15、PIO14、PIO13、PIO12	
B[3..0]	按键3	PIO11、PIO10、PIO9、PIO8	
A[7..4]	按键2	PIO7、PIO6、PIO5、PIO4	
A[3..0]	按键1	PIO3、PIO2、PIO1、PIO0	
F[7..4]	数码管6	PIO23、PIO22、PIO21、PIO20	
F[3..0]	数码管5	PIO19、PIO18、PIO17、PIO16	
CN4	LED灯1	PIO32	
AEQB	LED灯2	PIO33	

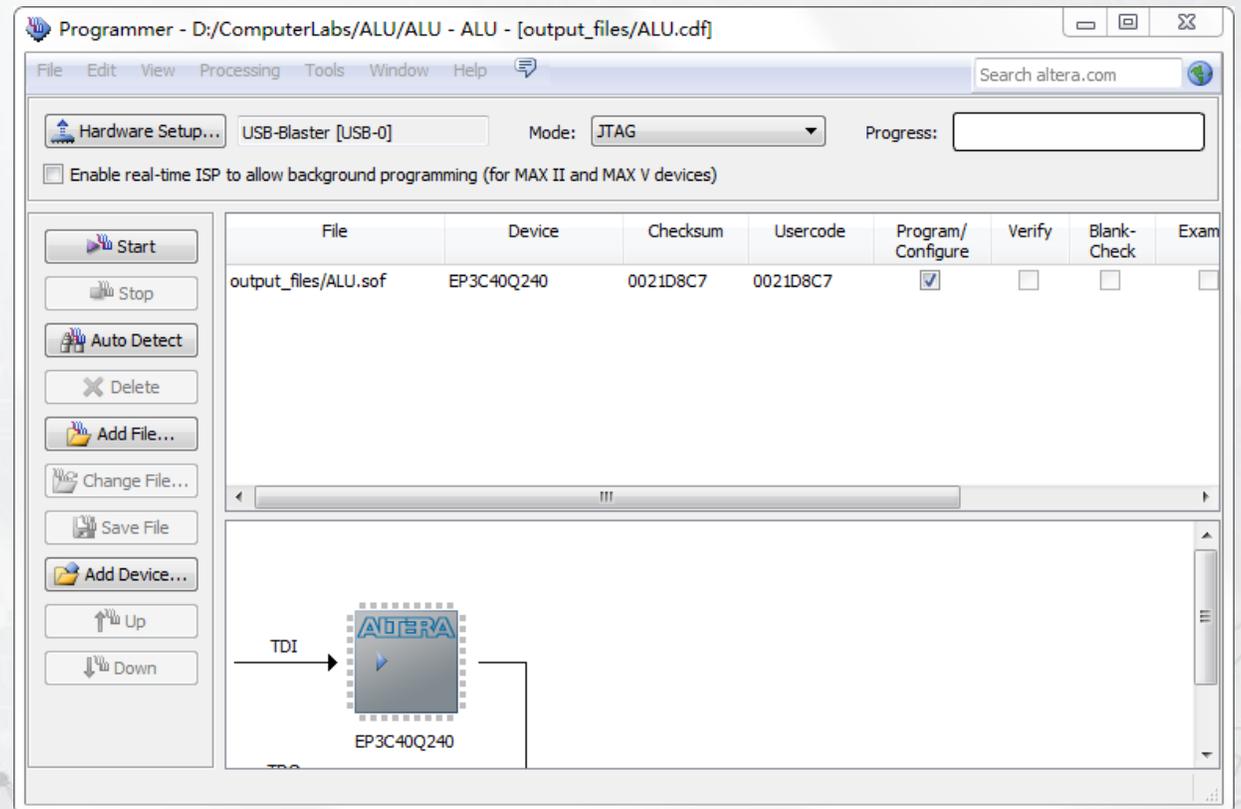
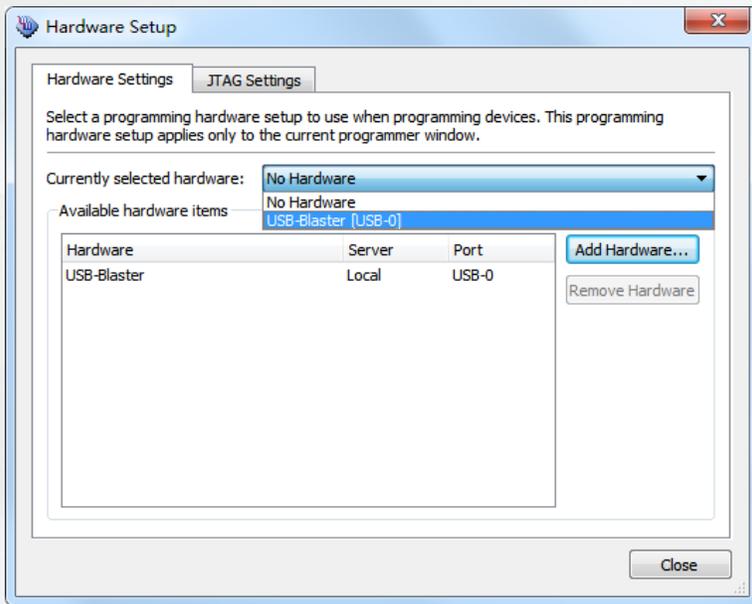
引脚锁定——(KX-CDS实验台EP4CE55F23C8芯片)

- ✓ 主菜单“Assignment”→“Device”项，选择芯片EP4CE55F23C8
- ✓ 主菜单“Assignment”→“Pin Planner”项，在Location栏中输入引脚号，编译。



连接实验台、编程器配置 (KX-CDS实验台, EP4CE55F23C8芯片)

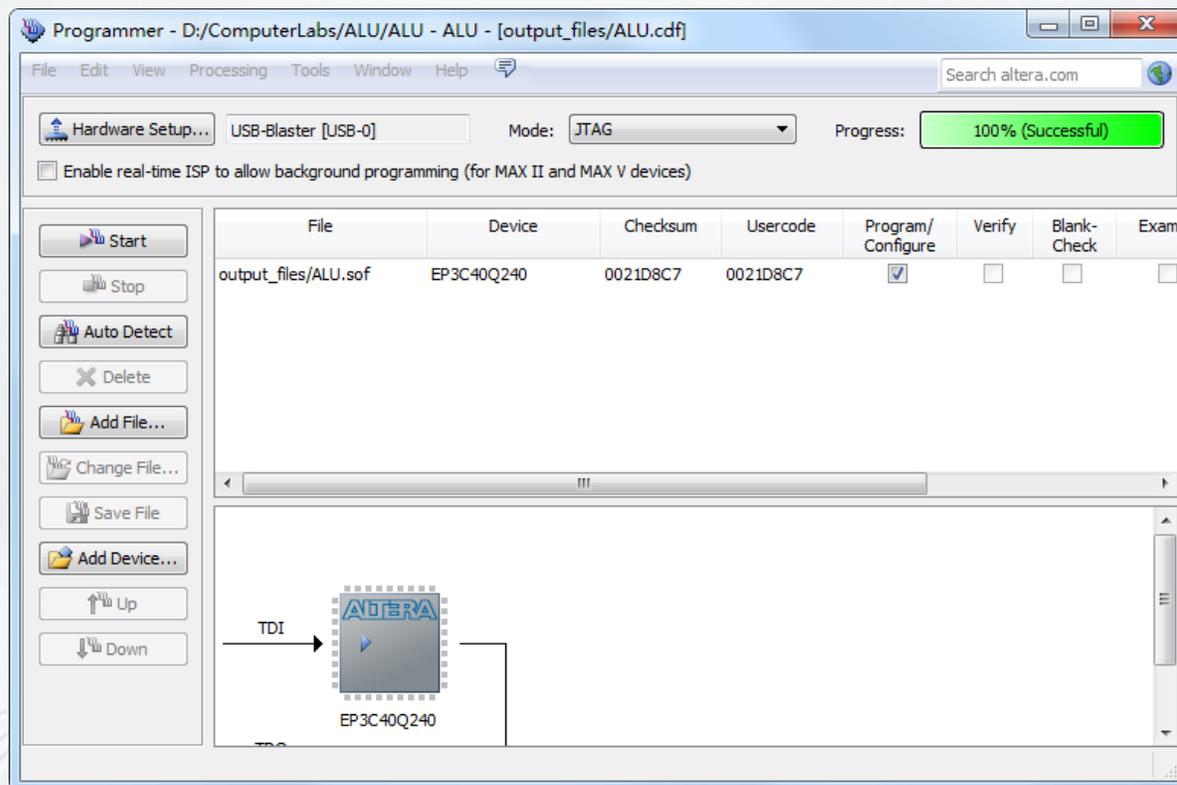
主菜单“Tools”→“Programmer”项, 打开编程器, 点击“Hardware Setup”按钮, 在“Hardware Setup”对话框中点击“Add Hardware”按钮。选择USB-Blaster[USB-0],最后点击Close按钮。



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。并且计算机已经安装了实验台的驱动。

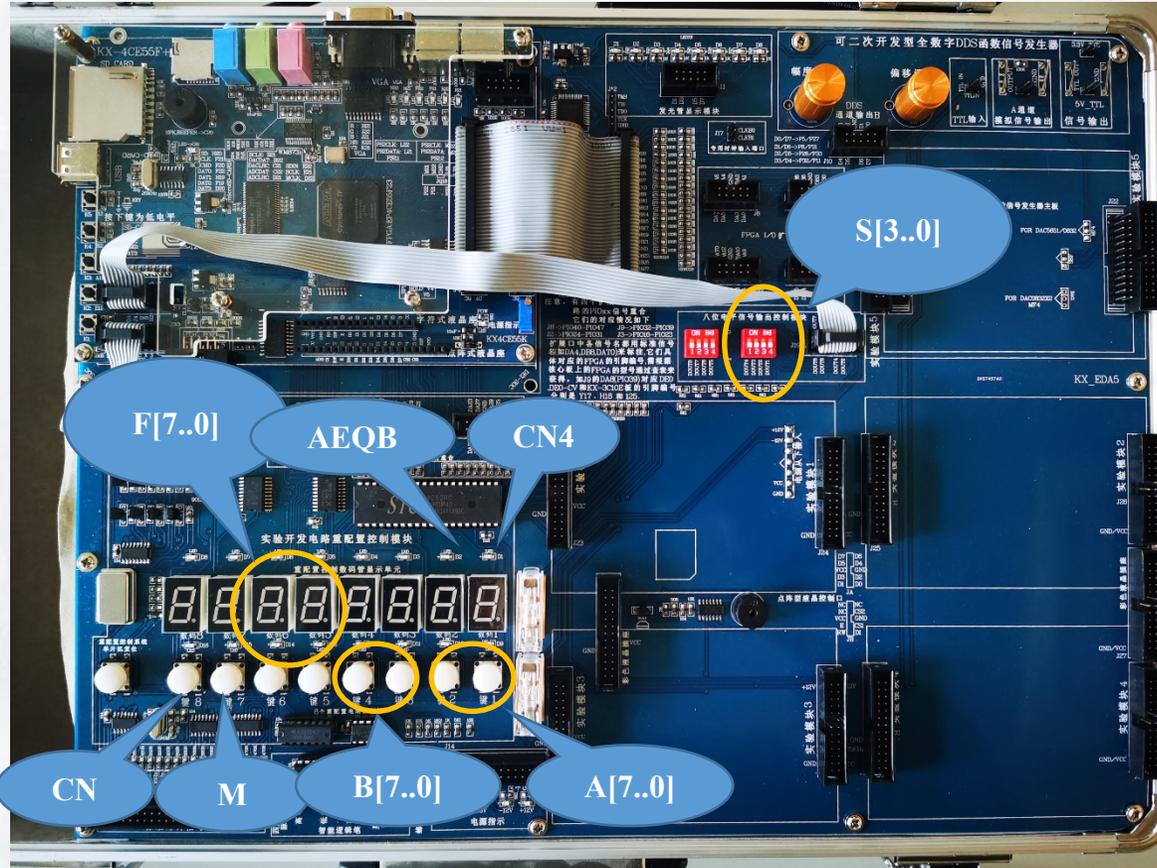
下载和测试 (KX-CDS实验台, EP4CE55F23C8芯片)

在Programmer窗口, 勾选ProgramConfigure复选框, 点击Start按钮, Progress为100%时, 下载完毕



ALU实验台演示

KX-CDS实验台





只读存储器(ROM)实验



存储器种类

计算机存储器分为两种基本类型：

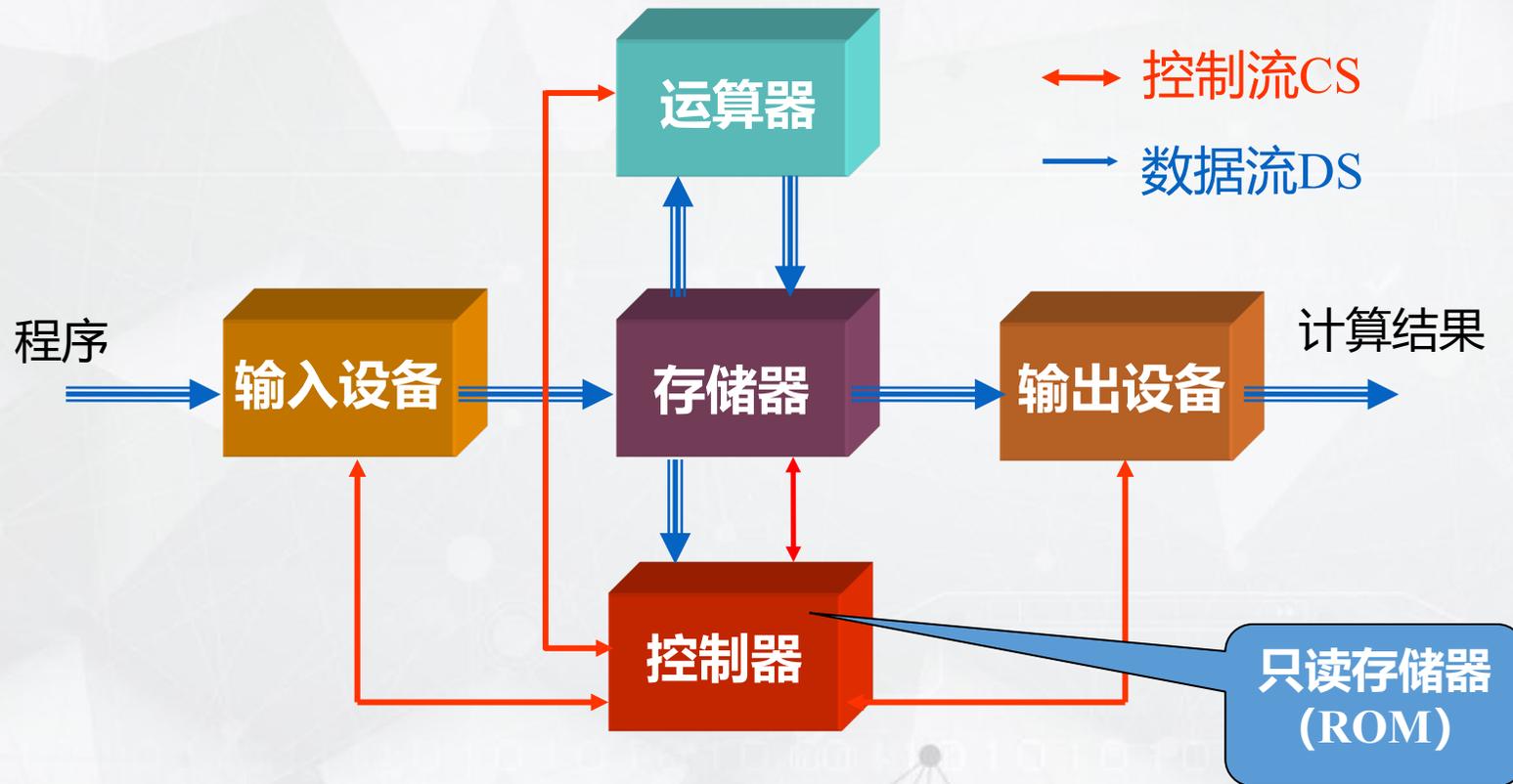
只读存储器（Read-Only Memory, ROM）

- 事先写好，工作过程中只能读出不能改写、数据稳定、断电后数据不会改变。
- 在计算机组成实验中的模型机系统中，ROM在控制器中，用于存储模型机的控制信号。

随机存取存储器（Random Access Memory, RAM）

- 随机读写，速度快，断电数据丢失。
- 是与CPU直接交换数据的内部存储器，又称主存。一般用来存储正在运行的程序的临时数据。

ROM在计算机中的位置



ROM实验

实验目的

- 1.掌握ROM存储器电路结构和工作原理。
- 2.掌握Quartus Prime软件环境和FPGA实验台的使用方法。
- 3.掌握利用框图输入法设计ROM存储器电路的方法。
- 4.验证ROM存储器功能。

实验内容

完成ROM存储器电路的设计、仿真、编程下载和实验台演示。

ROM实验内容与要求

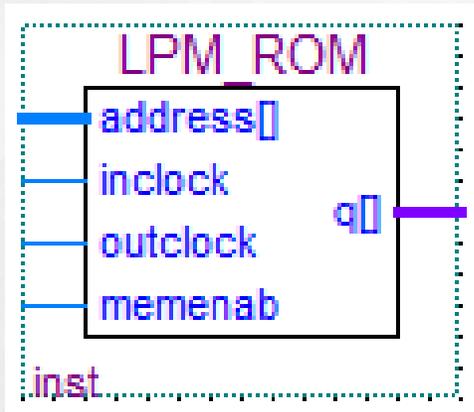
在Quartus Prime软件中，利用框图以及IP核（lpm_rom）设计ROM电路，建立初始化数据文件*.hex，将微代码表中的数据输入到表格中。完成ROM的数据初始化。

完成仿真，仿真要求：

合理设置各个输入信号，将ROM中存储的数据依次读出，观察输出结果是否和*.hex中的数据一致。

ROM实验原理

在Quartus Prime中，IP库中有许多可调用的LPM (Library Parameterized Modules)参数化的模块库，可构成ROM——lpm_rom。



输入/输出端口	功能
clock	输入时钟脉冲
q[23..0]	lpm_rom的24位数据输出端
address[5..0]	lpm_rom的6位读出地址

ROM初始化数据设置

ROM中的数据是在对FPGA现场配置时，通过配置文件一起写入存储单元的。

在Quartus Prime中，初始化数据文件格式有2种：

- (1) Memory Initialization File (.mif)
- (2) Hexadecimal (Intel-Format) File (.hex)

建立mif/hex文件，通过表格形式输入ROM初始化数据，保存文件。在设置lpm_rom时指定ROM初始化数据文件的路径即可完成ROM初始化数据设置。

实验任务与步骤

- 1、新建工程，新建框图文件 (*.bdf) ，设计输入ROM电路，采用元器件库中 宏模块lpm_rom 。保存文件。

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

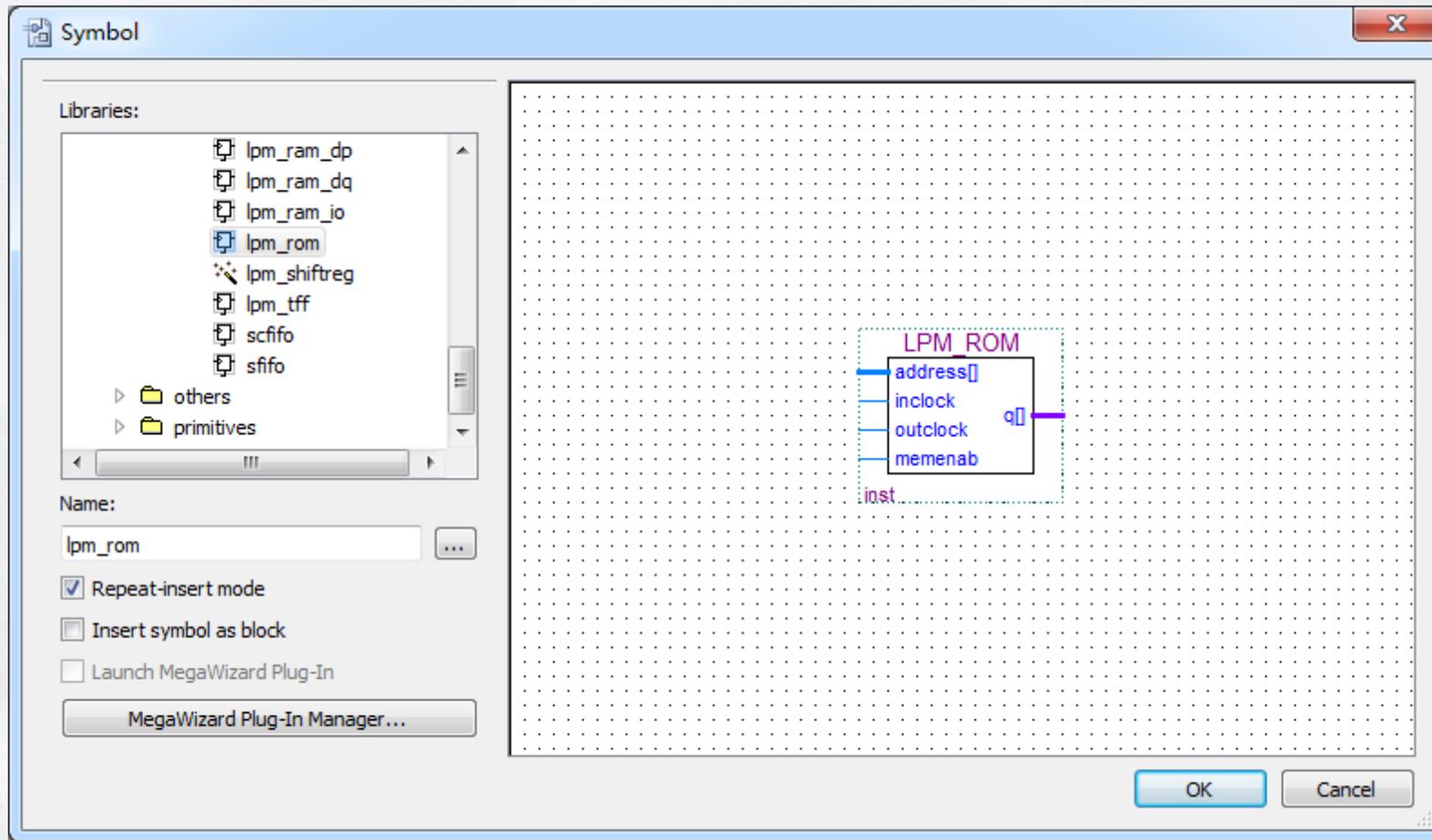
- 2、设置lpm_rom参数

选中LPM_ROM，右键菜单→“Properties”

- 3、设置器件

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

从元器件库中直接添加ROM——lpm_rom



设置lpm_rom参数

在框图编辑器中，选中LPM_ROM，右键菜单→“Properties”

The image shows two overlapping 'Symbol Properties' dialog boxes. The background dialog is on the 'Ports' tab, showing a table of ports. A callout points to the 'Status' column, indicating that 'Used' ports should be selected and 'Unused' ports should be disabled. The foreground dialog is on the 'Parameter' tab, showing a table of parameters. Callouts highlight the 'LPM_FILE' parameter as the 'initialization data file', the 'LPM_WIDTH' parameter (value 24) as the 'number of data items', and the 'LPM_WIDTHAD' parameter (value 6) as the 'data width'.

Name	Alias	Inversion	Status	Direction
1 address[LPM_WIDTHAD-1..0]	address[]	None	Used	INPUT No
2 indock	indock	None	Used	INPUT No
3 memenab	memenab	None	Unused	INPUT
4 outdock	outdock	None	Unused	INPUT
5 q[LPM_WIDTH-1..0]	q[]	None	Used	OUTPUT

Name	Value	Type	Description
1 LPM_ADDRESS_CONTROL			Should the address and control ports
2 LPM_FILE			File containing initial contents of mem
3 LPM_NUMWORDS			Number of memory words, default is :
4 LPM_OUTDATA	"UNREGISTERED"		Should the output data be registered
5 LPM_WIDTH	24		Data width in bits, any integer > 0
6 LPM_WIDTHAD	6		Number of address lines, any integer
7 <NEW>			

添加输入输出端口

The screenshot shows the Quartus II 64-bit IDE interface. The main workspace displays a schematic with an LPM ROM component. The component's inputs are labeled 'address[5..0]' and 'clock', and its output is labeled 'q[23..0]'. A parameter table is visible on the right side of the schematic, listing the following parameters and values:

Parameter	Value
LPM_ADDRESS_CONTROL	
LPM_FILE	
LPM_NUMWORDS	
LPM_OUTDATA	"UNREGISTERED"
LPM_WIDTH	24
LPM_WIDTHAD	6

Two blue callout bubbles are present: one pointing to the 'LPM_WIDTH' parameter with the text '数据宽度24', and another pointing to the 'LPM_WIDTHAD' parameter with the text '地址宽度6'.

实验任务与步骤

4、新建ROM数据文件 (*.hex) ， 设置参数， 数据个数为64（地址宽度为6） ， 数据宽度为24。保存文件。

主菜单“File”→“New”项， 选择Hexadecimal(Intel-Format) File， 新建存储器数据文件， 保存为 ucode.hex。

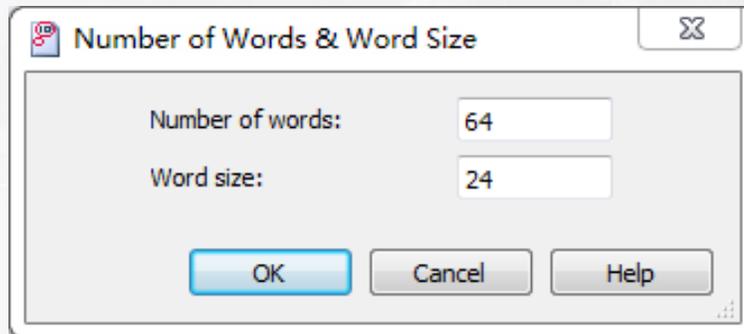
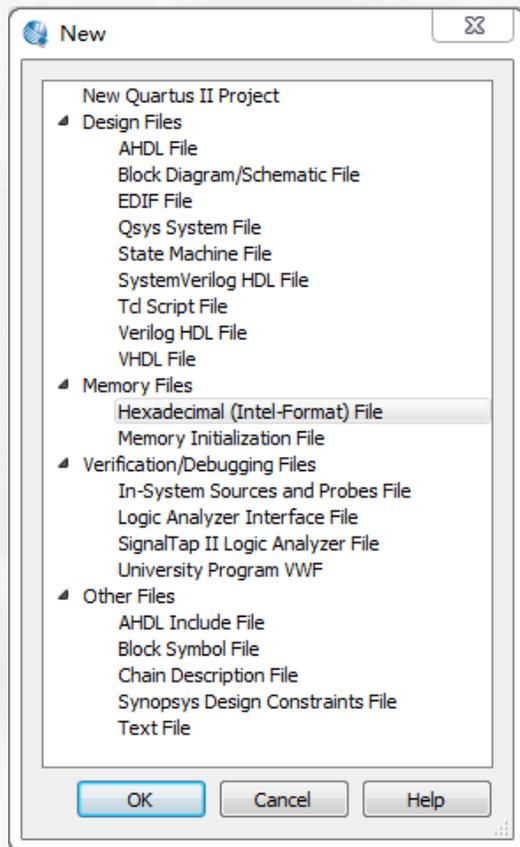
5、 在ucode.hex文件中输入数据。为ROM绑定初始化数据文件ucode.hex

选中LPM_ROM,右键菜单→“Properties” → “Parameter”， 设置LPM_FILE为ucode.hex

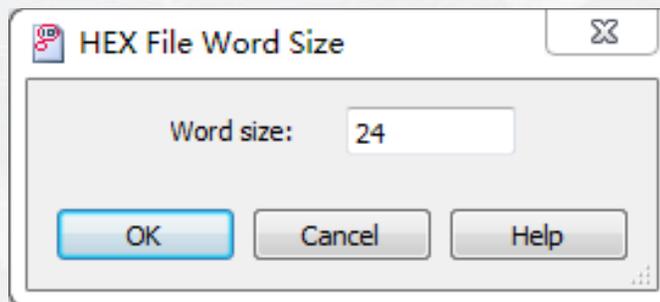
6、 编译电路

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

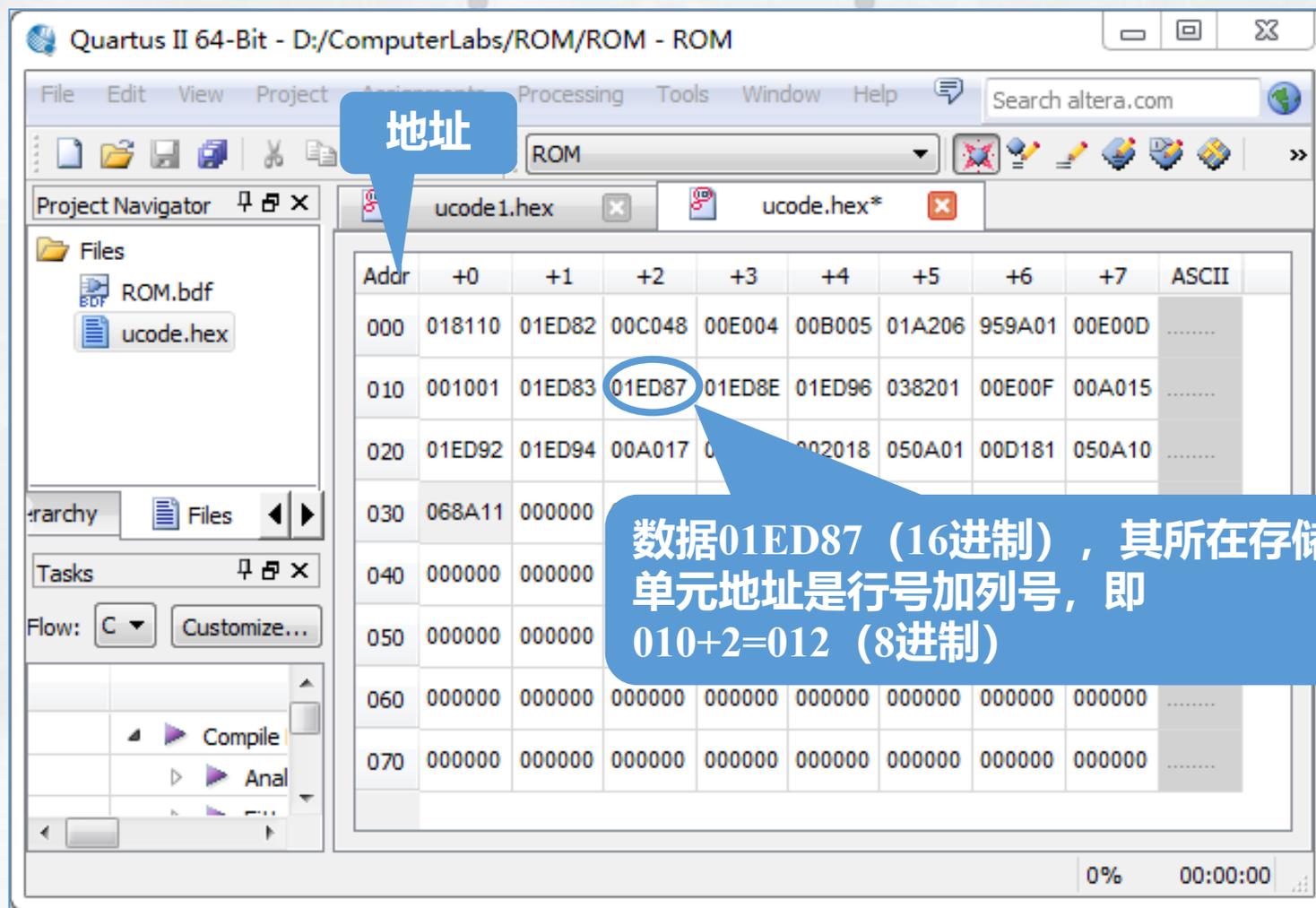
新建ROM数据文件 (*.hex) , 设置参数



注意：以后每次打开*.hex文件会弹出对话框，需要输入数据宽度24



初始化数据文件ucode.hex



地址

Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
000	018110	01ED82	00C048	00E004	00B005	01A206	959A01	00E00D
010	001001	01ED83	01ED87	01ED8E	01ED96	038201	00E00F	00A015
020	01ED92	01ED94	00A017	002018	050A01	00D181	050A10
030	068A11	000000							
040	000000	000000							
050	000000	000000							
060	000000	000000	000000	000000	000000	000000	000000	000000
070	000000	000000	000000	000000	000000	000000	000000	000000

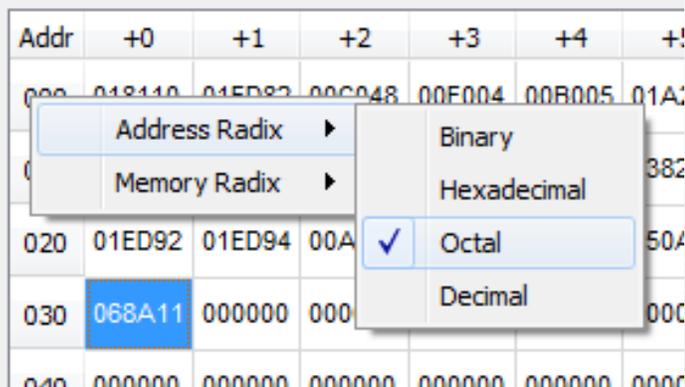
数据01ED87 (16进制), 其所在存储单元地址是行号加列号, 即 $010+2=012$ (8进制)

设置初始化数据文件表格地址和数据的数制

Hex文件的表格中，地址和数据的数制可以通过右键菜单来设置，可以设置二进制、十六进制等数制。

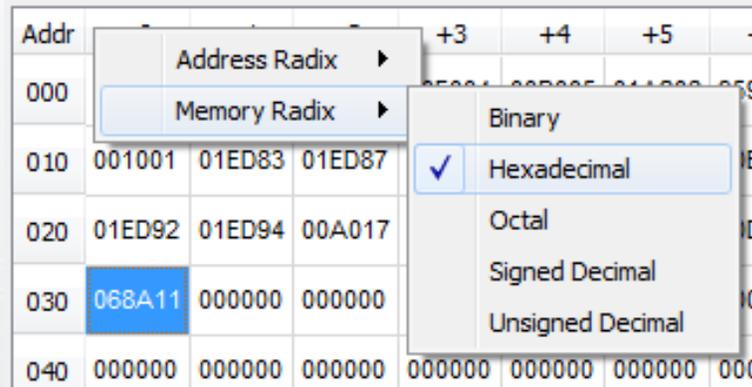
设置地址为8进制

Addr	+0	+1	+2	+3	+4	+5
000	018110	01ED83	00C048	00E004	00B005	01A...
010						382
020	01ED92	01ED94	00A...			50A
030	068A11	000000	000			000
040	000000	000000	000000	000000	000000	0000

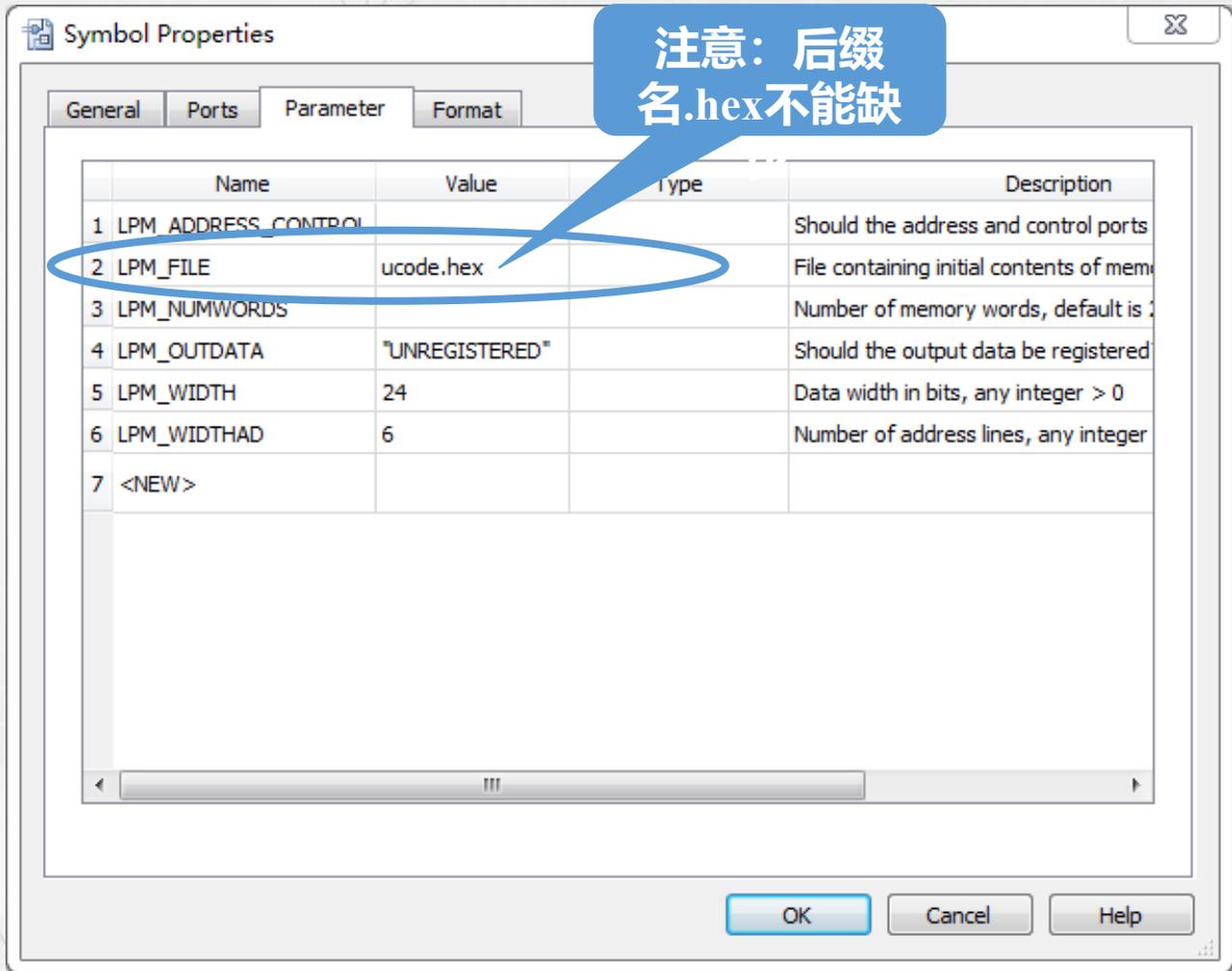
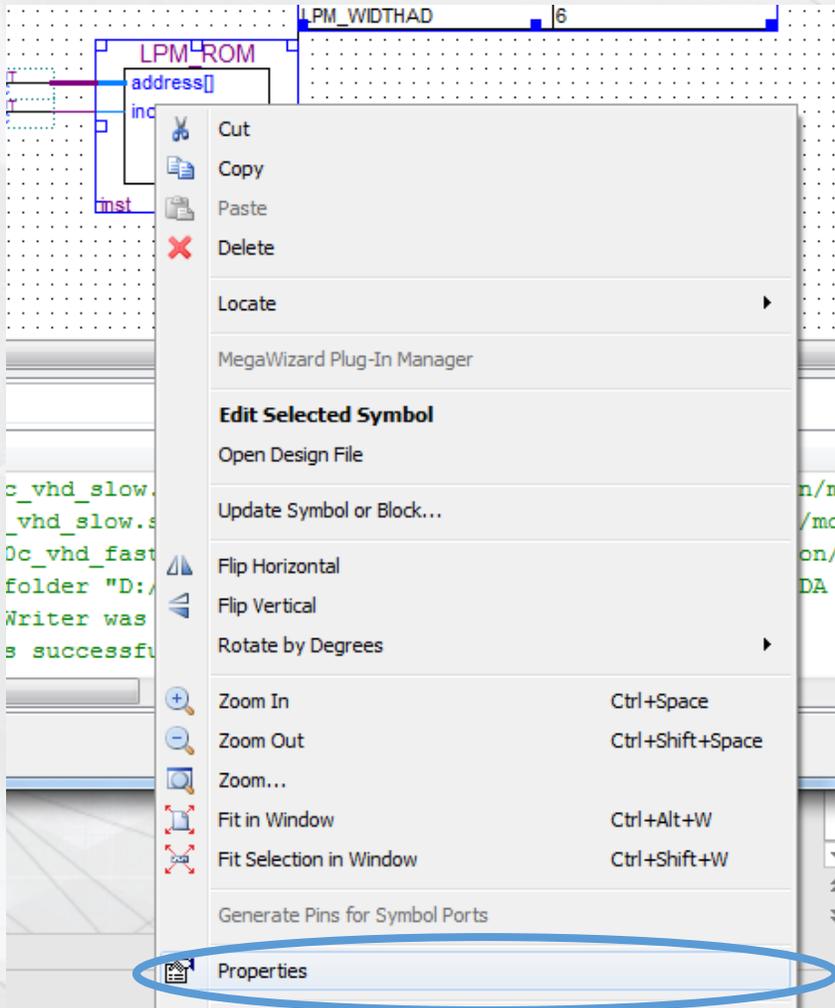


设置存储数据为16进制

Addr	+3	+4	+5	+6
000				
010	001001	01ED83	01ED87	
020	01ED92	01ED94	00A017	
030	068A11	000000	000000	
040	000000	000000	000000	000000



LPM_ROM与ucode.hex绑定



实验任务与步骤

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

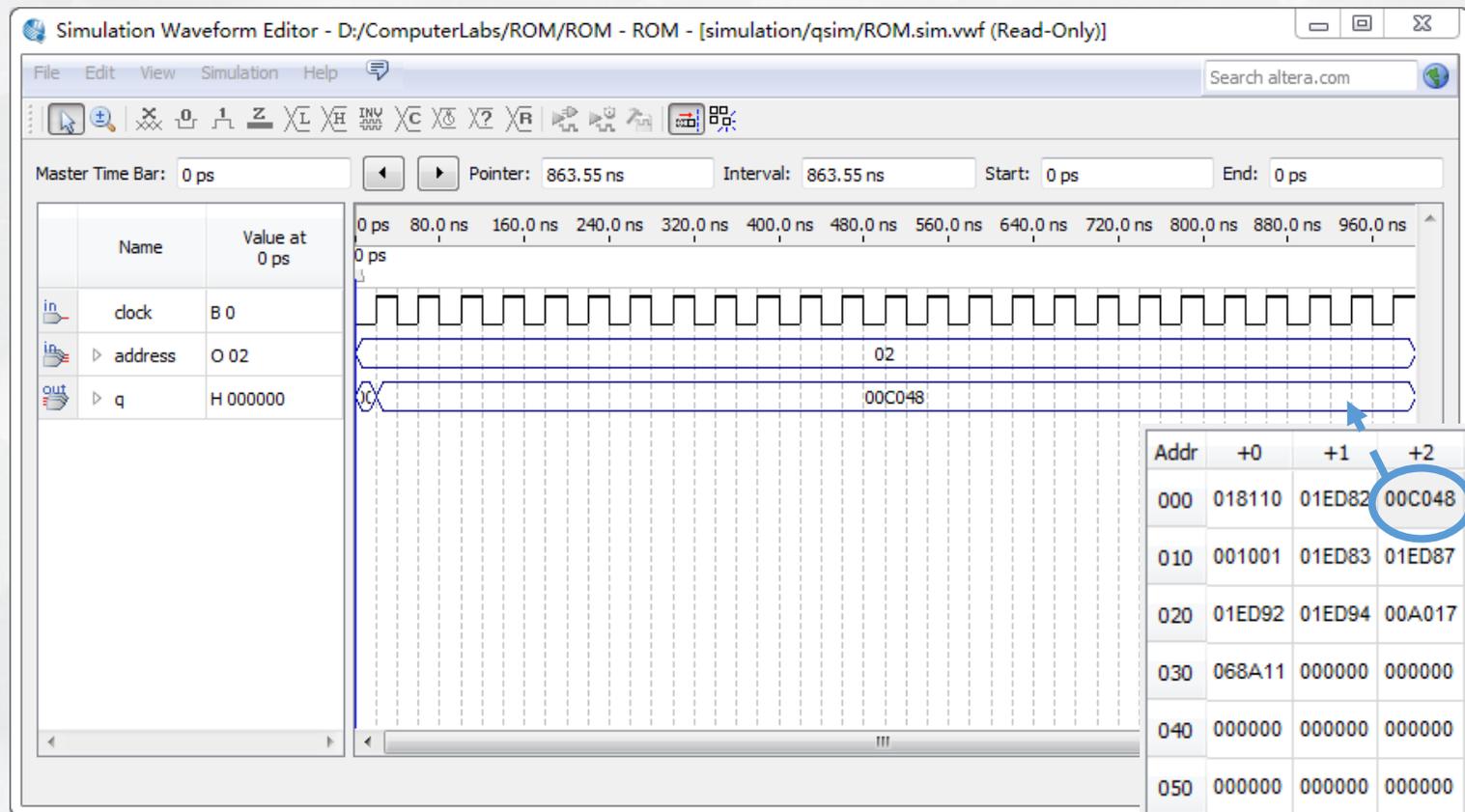
建立仿真波形文件：主菜单“File”→“New”项，选择University Program VWF，新建*.vwf，打开波形编辑器。

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。

ROM存储器仿真波形图示例



Addr	+0	+1	+2	+3	+4	+5	+6	+7
000	018110	01ED82	00C048	00E004	00B005	01A206	959A01	00E00D
010	001001	01ED83	01ED87	01ED8E	01ED96	038201	00E00F	00A015
020	01ED92	01ED94	00A017	018001	002018	050A01	00D181	050A10
030	068A11	000000	000000	000000	000000	000000	000000	000000
040	000000	000000	000000	000000	000000	000000	000000	000000
050	000000	000000	000000	000000	000000	000000	000000	000000
060	000000	000000	000000	000000	000000	000000	000000	000000
070	000000	000000	000000	000000	000000	000000	000000	000000

仿真波形图与ucode.hex中的数据一致

实验任务与步骤

8、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。

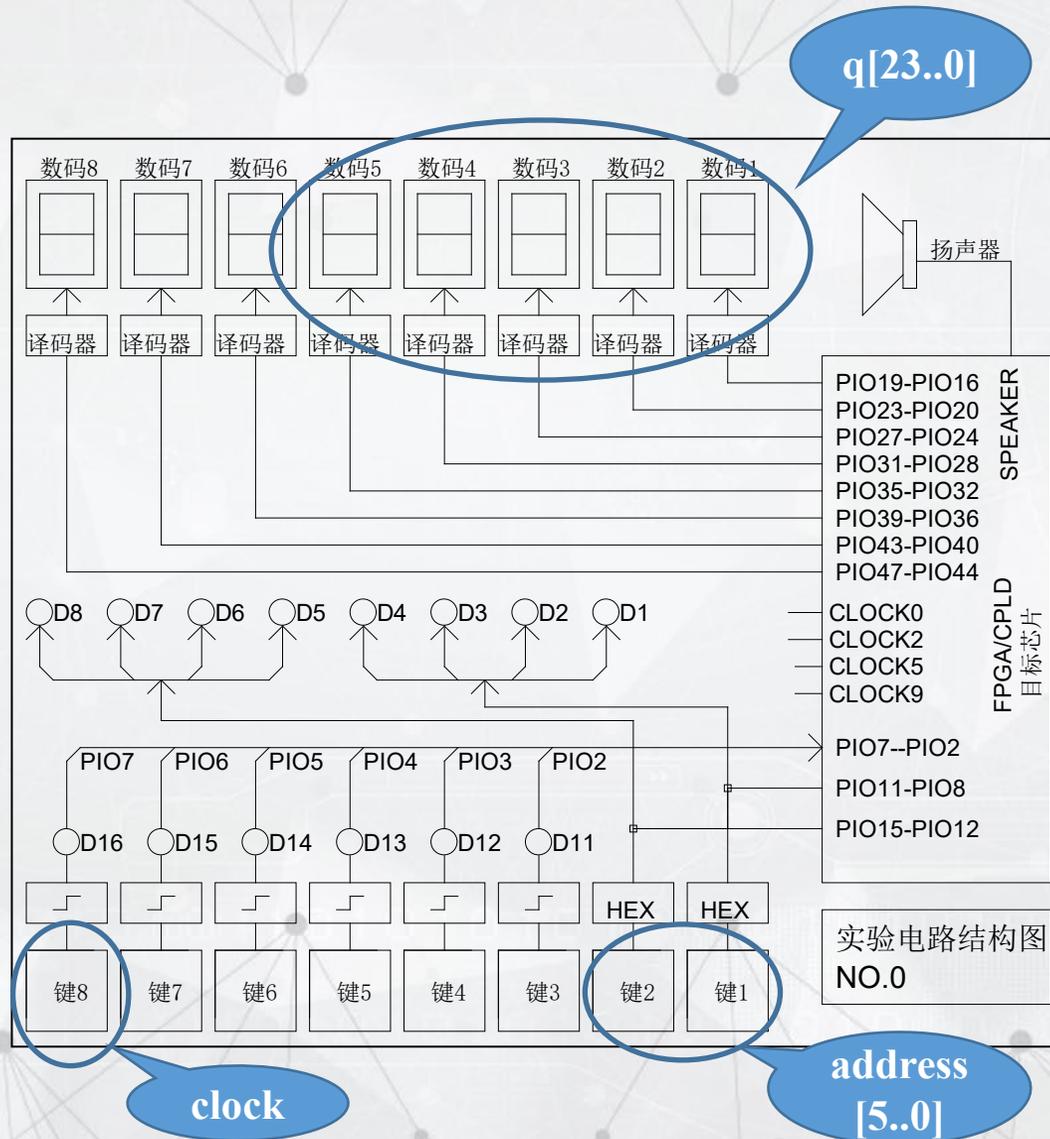
主菜单“Assignments”→“Pin”项

9、下载sof文件到FPGA实验台，演示ROM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



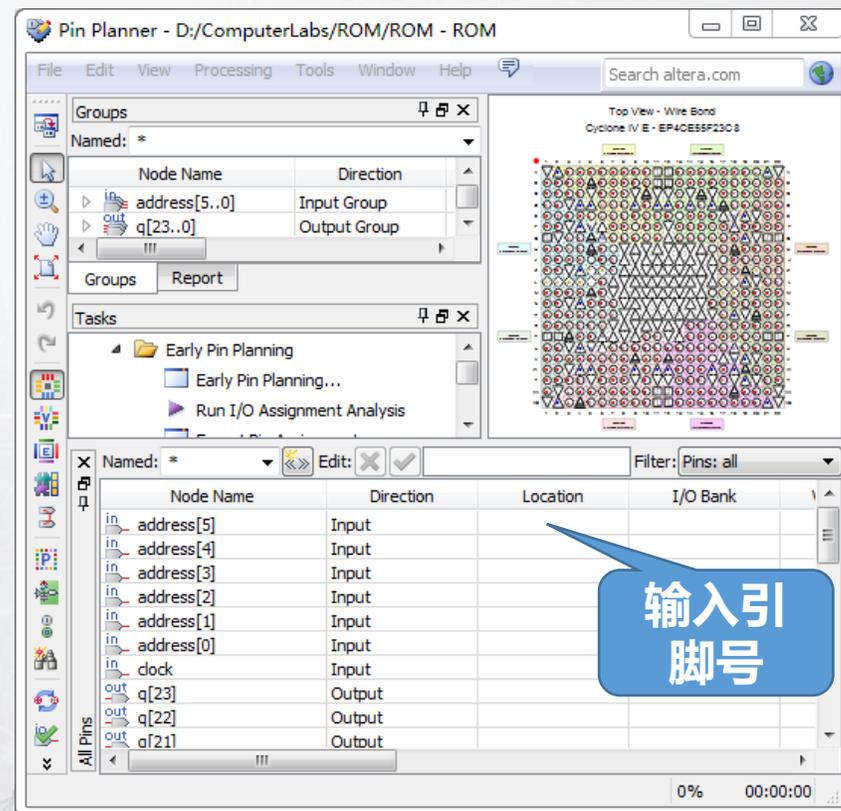
FPGA实验台引脚锁定

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/输出端口	外设	引脚名称	引脚号
clock	按键8		
address[5..4]	按键2		
address[3..0]	按键1		
q[23..20]	数码管6		
q[19..16]	数码管5		
q[15..12]	数码管4		
q[11..8]	数码管3		
q[7..4]	数码管2		
q[3..0]	数码管1		

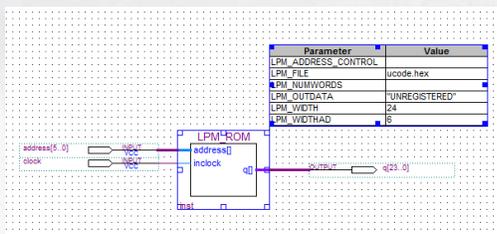
主菜单“Assignments”→“Pin”项，在 Location 栏中输入引脚号



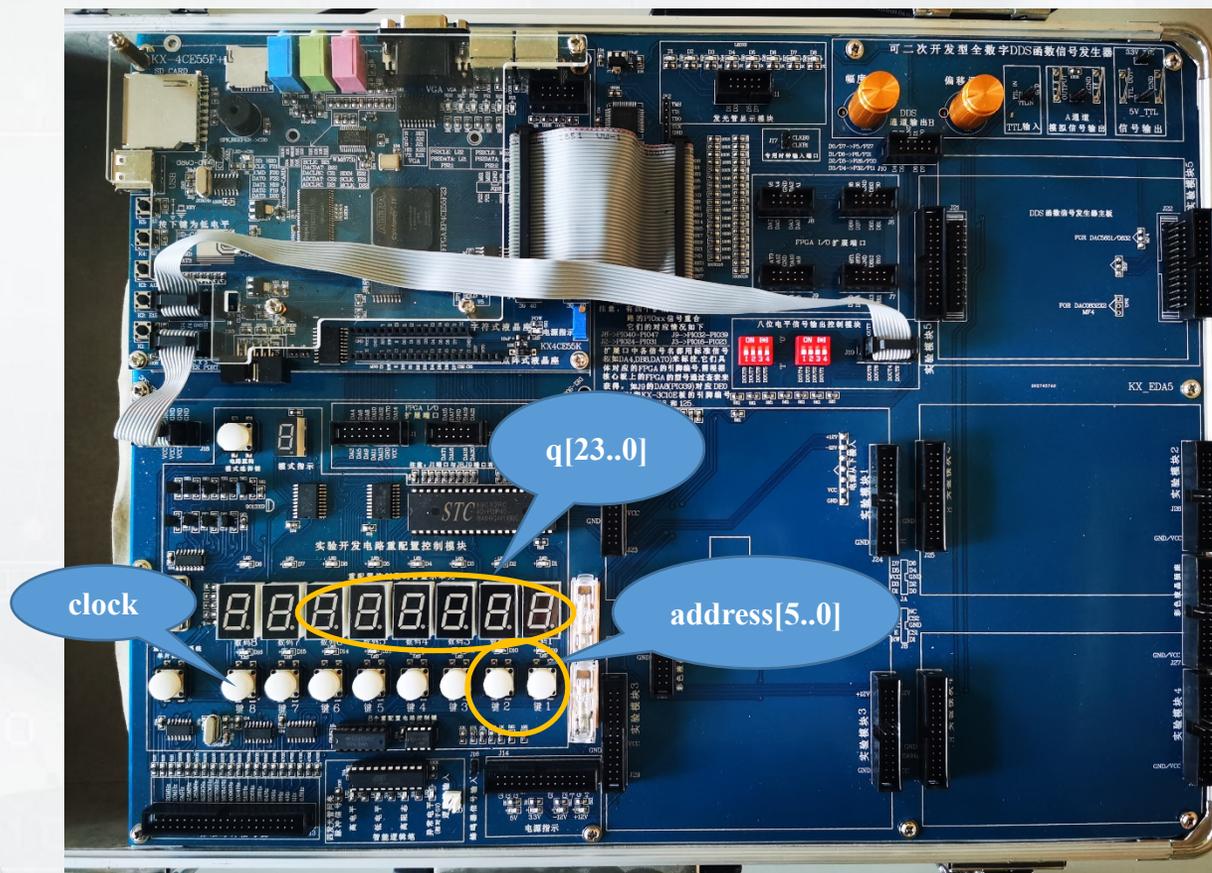
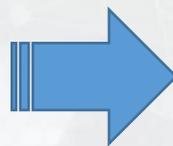
ROM实验台演示

KX-CDS实验台

- ✓ 主菜单“Tools”→“Programmer”项，打开编程器，点击“Hardware Setup”按钮，选择ByteBlasterMV，注意：KX-CDS实验台需要打开电源，并且与计算机通过数据线连接
- ✓ 勾选ProgramConfigure复选框
- ✓ 在Programmer窗口，点击Start按钮，Progress为100%时，下载完毕。



下载





随机存取存储器 (RAM)实验



存储器种类

计算机存储器分为两种基本类型：

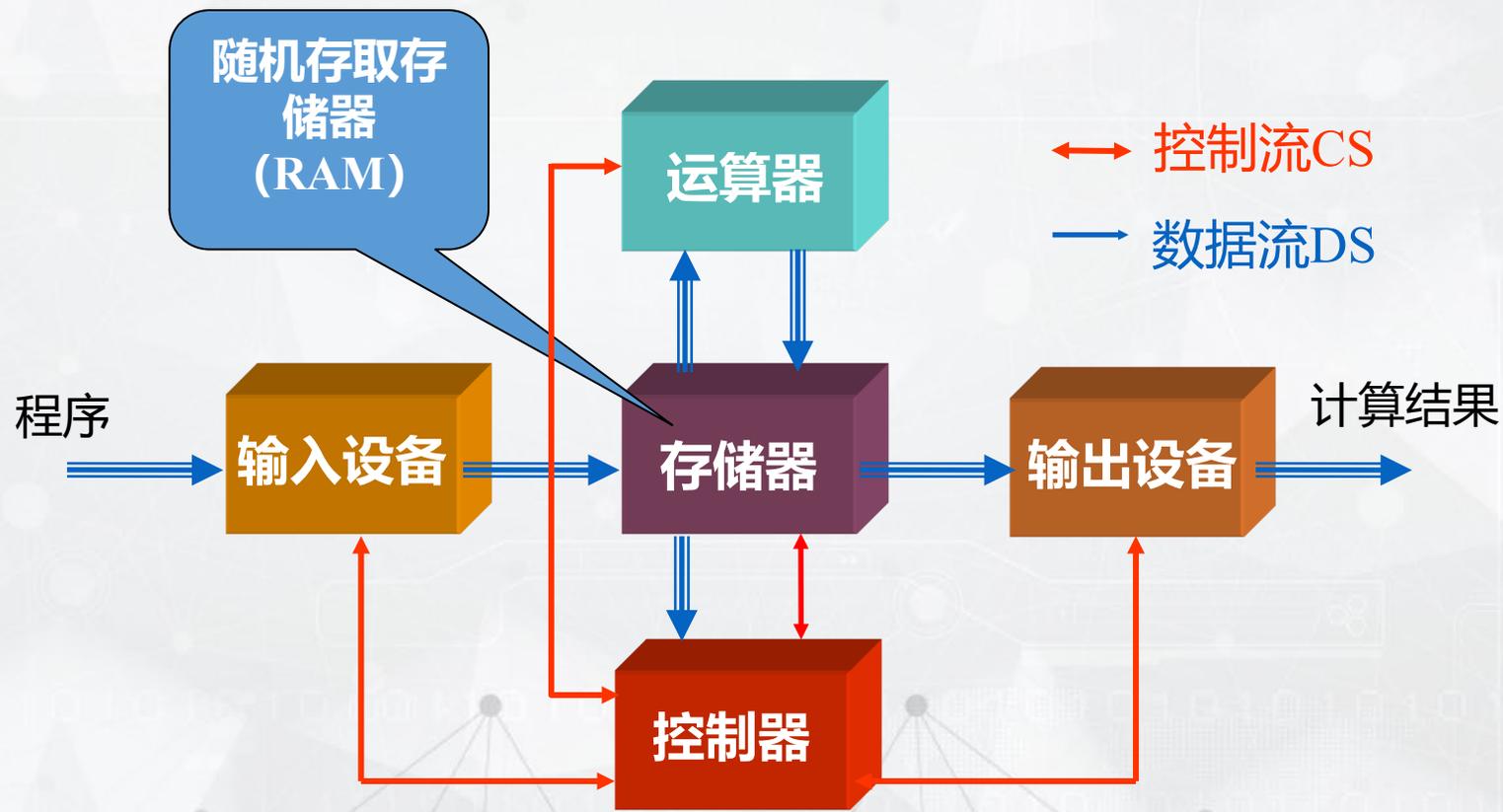
只读存储器（Read-Only Memory, ROM）

- 事先写好，工作过程中只能读出不能改写、数据稳定、断电后数据不会改变。
- 在计算机组成实验中的模型机系统中，ROM在控制器中，用于存储模型机的控制信号。

随机存取存储器（Random Access Memory, RAM）

- 随机读写，速度快，断电数据丢失。
- 是与CPU直接交换数据的内部存储器，又称主存。一般用来存储正在运行的程序的临时数据。

存储器RAM在计算机中的位置



RAM实验

实验目的

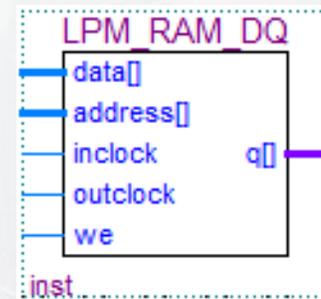
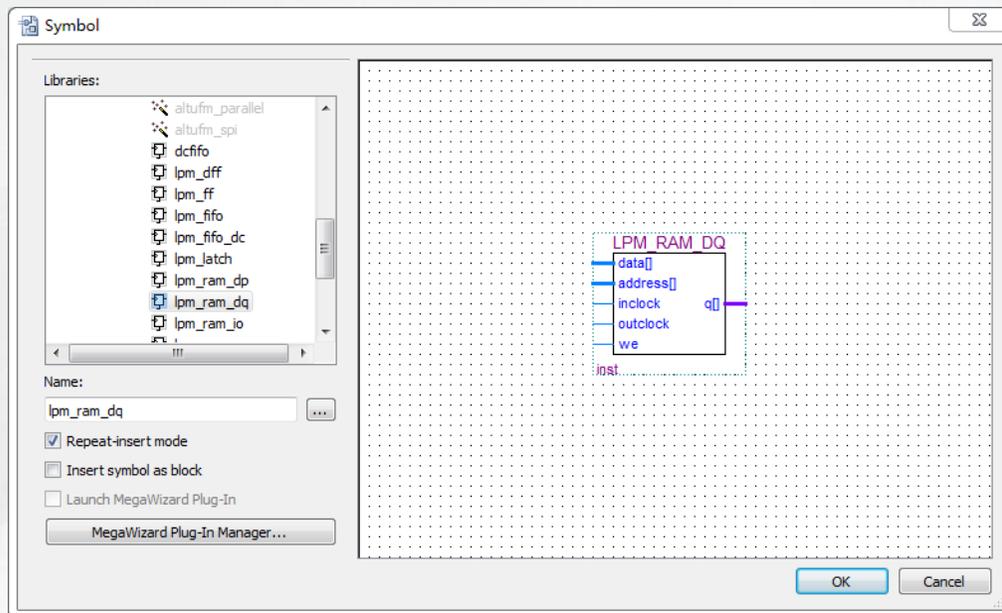
- 1.掌握RAM存储器电路结构和工作原理。
- 2.掌握Quartus Prime软件环境和FPGA实验台的使用方法。
- 3.掌握利用框图输入法设计RAM存储器电路的方法。
- 4.验证RAM存储器功能。

实验内容

完成RAM存储器电路的设计、仿真、编程下载和实验台演示。

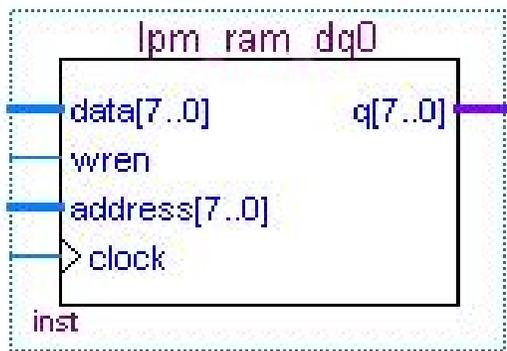
RAM实验原理

在Quartus Prime中，IP库中有许多可调用的LPM (Library Parameterized Modules)参数化的模块库，可构成RAM——lpm_ram_dq。



RAM实验原理

lpm_ram_dq有5组信号:



lpm_ram_dq的结构图

端口名称	功能
data[7..0]	8位数据输入端
wren	读/写控制端, 高电平进行写操作, 低电平进行读操作
address[7..0]	读出和写入地址
clock	读/写时钟脉冲
q[7..0]	lpm_ram的8位数据输出端

RAM实验内容与要求

在Quartus Prime软件中，利用框图及IP核（lpm_ram_dq）设计RAM电路。

完成仿真，仿真要求：

- 1.模拟将一段程序的指令代码依次顺序写进RAM。
- 2.模拟程序运行时，从RAM读出指令和数据的流程。主要体现RAM地址自动加1程序顺序执行程序 and 从总线上获取RAM地址访问RAM两种情况。

实验任务与步骤

- 1、新建工程，新建框图文件 (*.bdf) ，设计输入RAM电路，采用元器件库中宏模块lpm_ram_dq。保存文件。

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

- 2、设置器件

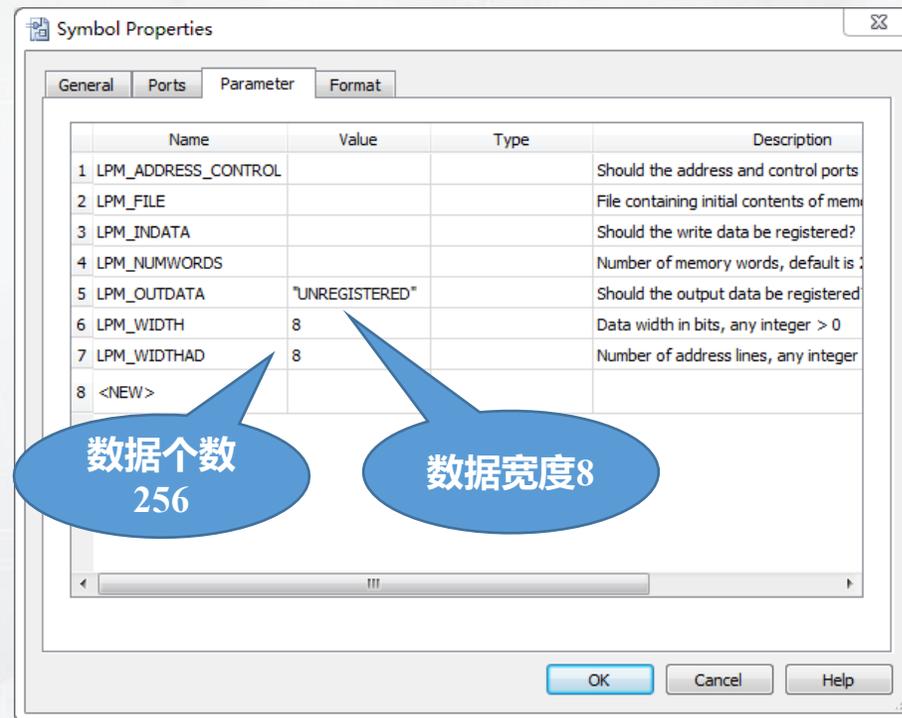
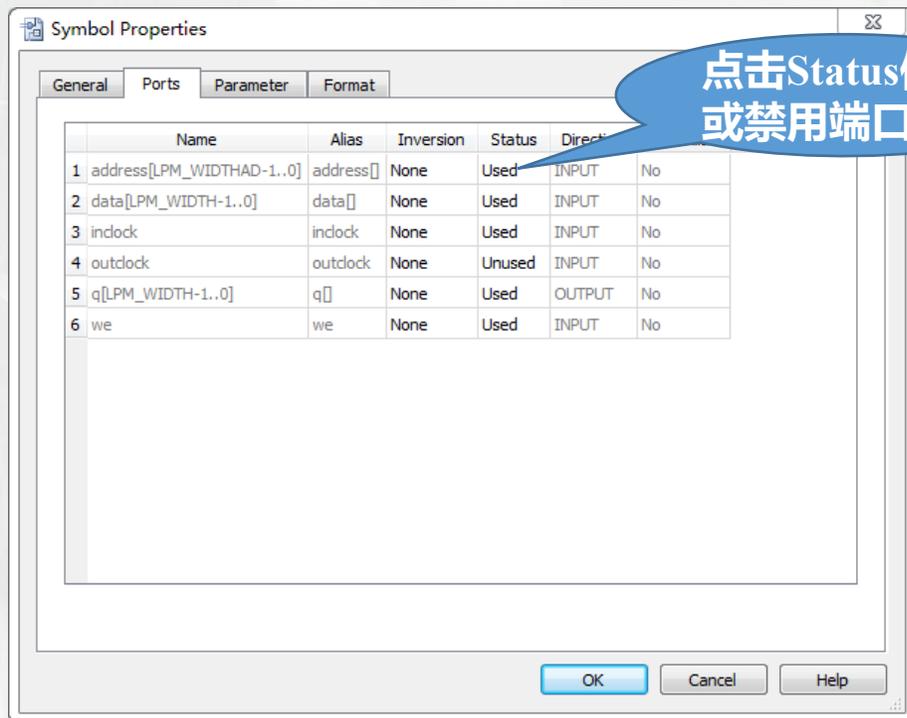
主菜单“Assignments”→“Device”项，选择Cyclone IV E系列 EP4CE55F23C8芯片

- 3、编译电路

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

设置lpm_ram_dq参数

在框图编辑器中，选中LPM_RAM_DQ，右键菜单→“Properties”



添加输入输出端口

Parameter Table:

Parameter	Value
LPM_ADDRESS_CONTROL	
LPM_FILE	
LPM_INDATA	
LPM_NUMWORDS	
LPM_OUTDATA	"UNREGISTERED"
LPM_WIDTH	8
LPM_WIDTHAD	8

Callout 1: 数据宽度8

Callout 2: 地址宽度8

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。

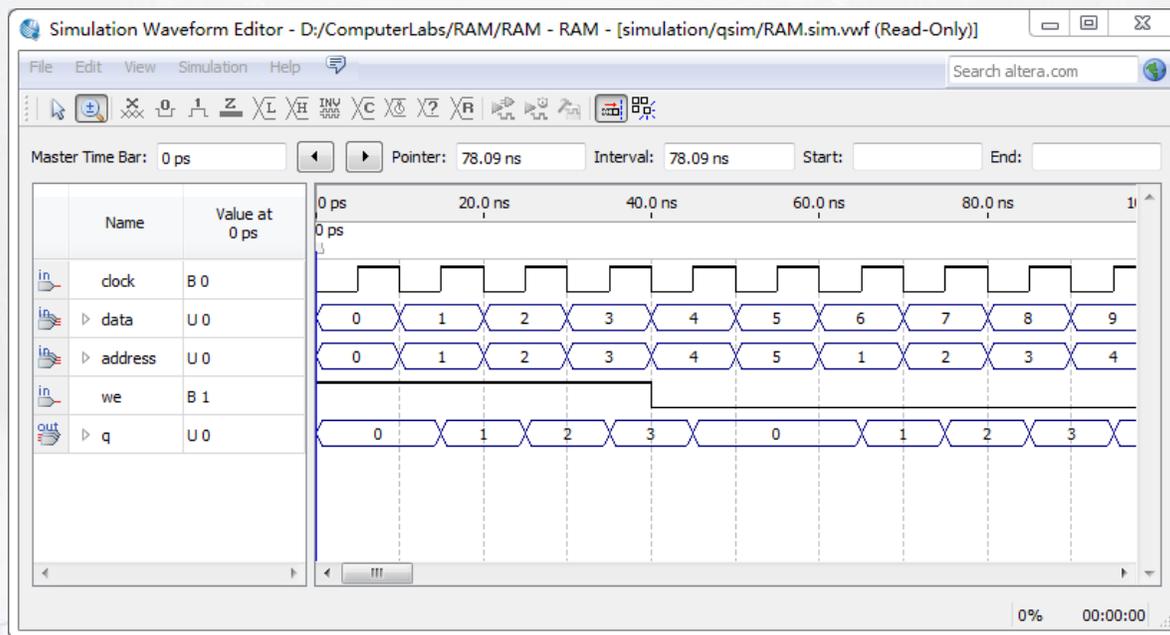
设置仿真时间: 主菜单“Edit”→“Set End Time”项。

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。

RAM仿真波形图示例

从仿真波形上可以看出，we为高电平时，将数据0、1、2、3存储到地址为0、1、2、3的RAM存储单元中。当we为低电平时，q依次显示了从RAM读出的4个数据0、1、2、3。



RAM实验仿真要求

RAM地址	RAM中存储的程序代码
00	00
01	10
02	0A
03	20
04	0B
05	30
06	0B
07	40
08	01
09	
0A	34
0B	

在波形图中设置合理的RAM地址，模拟模型机执行程序时RAM地址的变化。

程序运行时，RAM地址的变化是:00、01、02、0A、03、04、0B、05、06、0B....。

实验任务与步骤

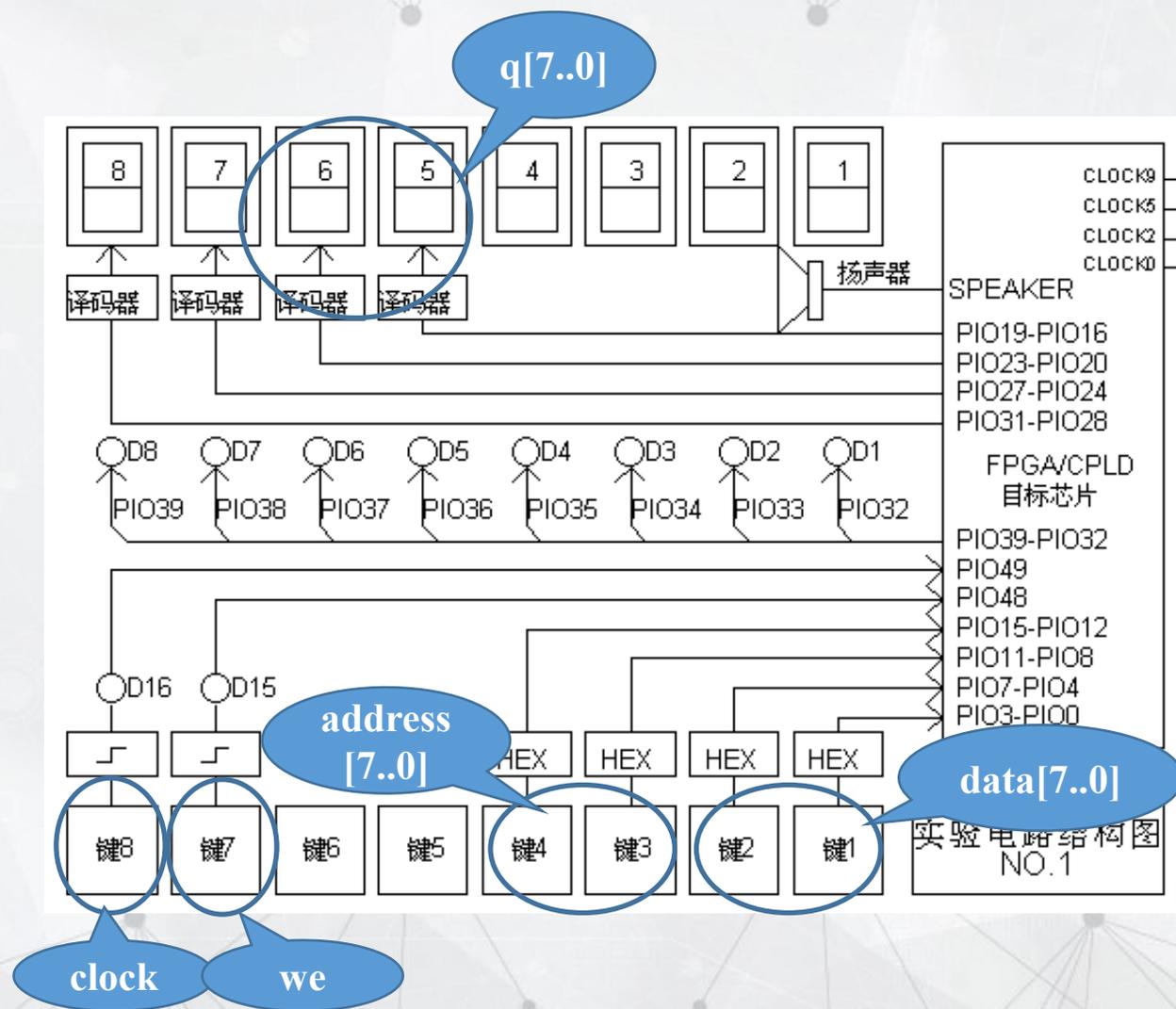
5、选择KX-CDS实验台，选择合适的电路模式结构，例如NO.1，对照电路模式图和引脚表，查找引脚号。打开编程器，输入引脚号，对电路进行引脚锁定，编译工程。

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

6、下载sof文件到FPGA实验台，演示RAM的读写功能。

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.1



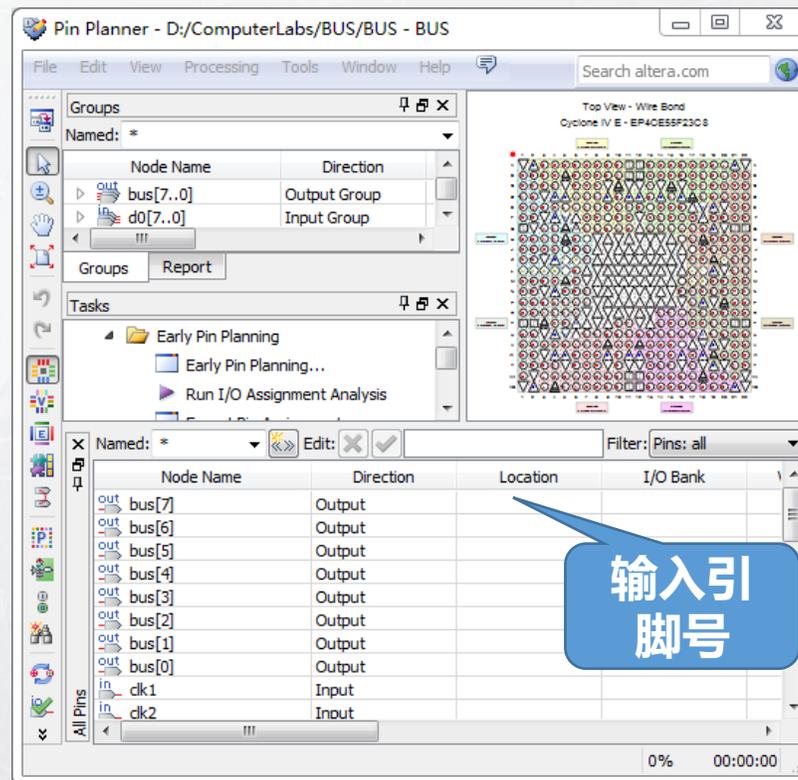
FPGA实验台引脚锁定

参照电路模式图No.1，确定引脚名称，再查找引脚表，获得引脚号

引脚锁定方案(No.1)

输入/输出端口	外设	引脚名称	引脚号
clock	按键8		
address[7..4]	按键4		
address[3..0]	按键3		
data[7..4]	按键2		
data[3..0]	按键1		
we	按键7		
q[7..4]	数码管6		
q[3..0]	数码管5		

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

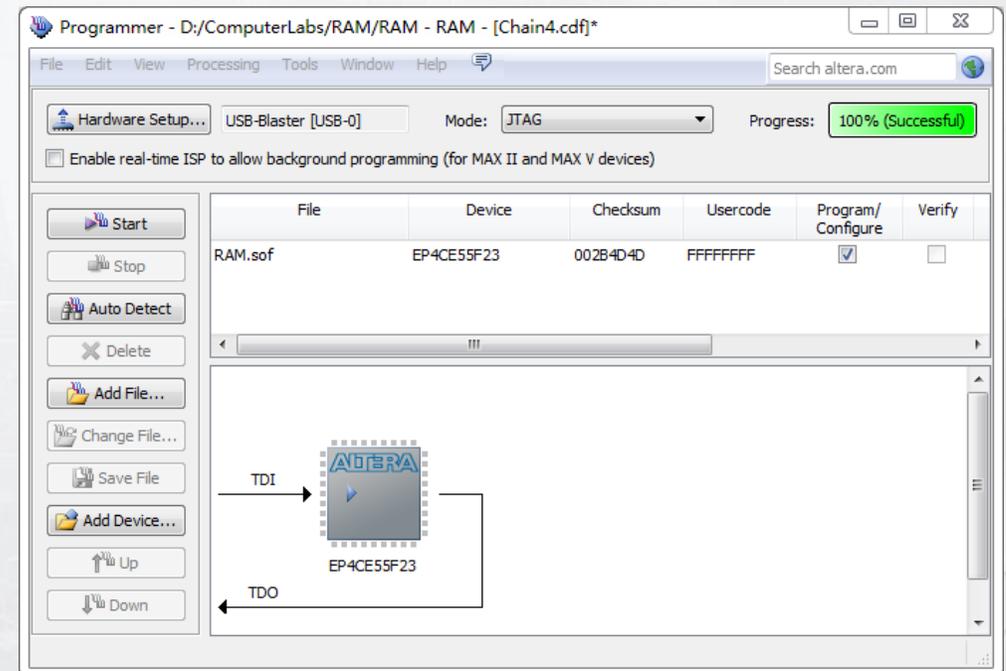
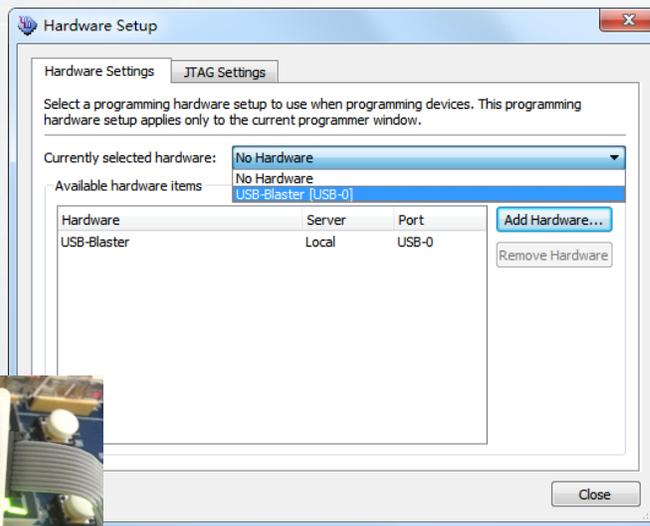


连接实验台下载电路

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。

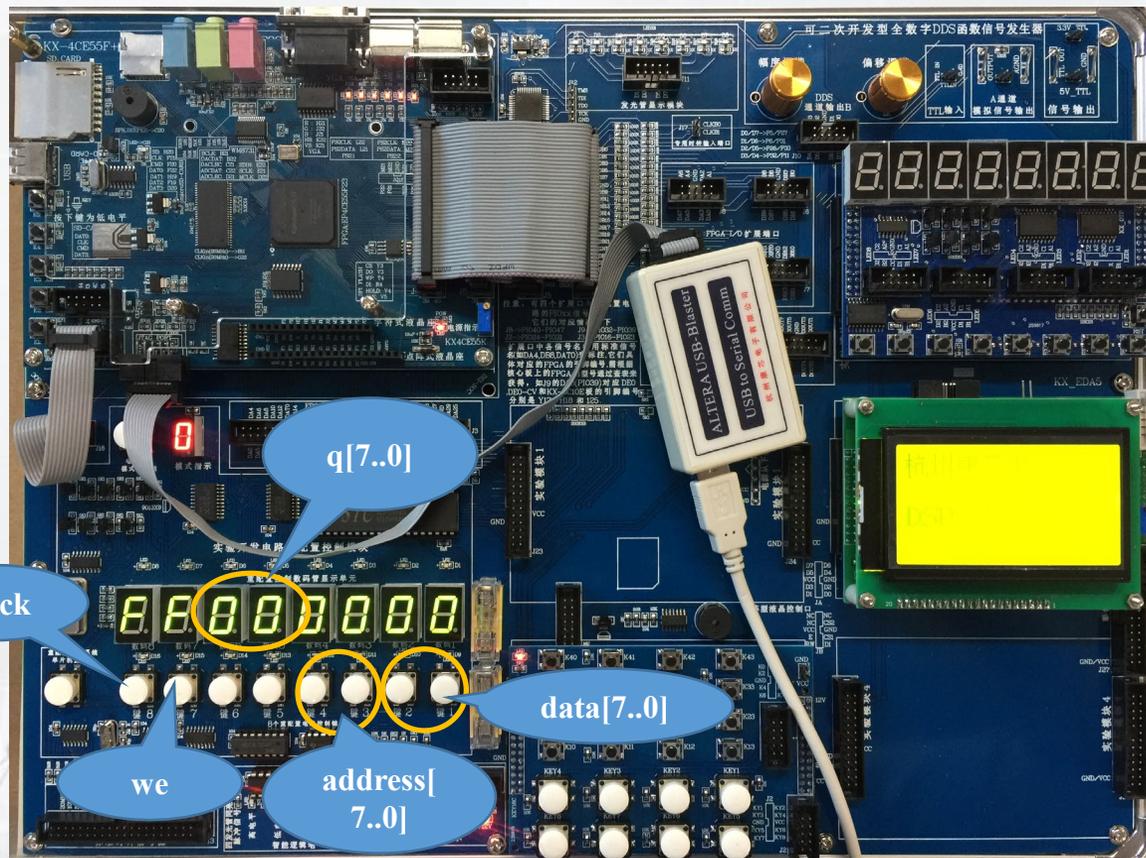
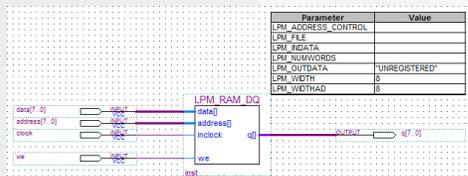
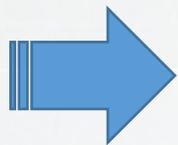
KX-CDS实验台



RAM实验台演示

KX-CDS实验台

下载



现在开始实验！（第3次课）

实验2 计算机基本部件与总线控制实验

1、运算器实验：参考教材250页-254页， 221页

存储器实验：ROM参考教材254页-260页

RAM参考教材260页-264页

2、分别建立三个工程，完成电路设计、编译和仿真。按照每个实验的要求，全面仿真各个电路的功能。

3、2个人一组。实体名后面加2个学号的后两位，例如mux21a0709

4、下次课预习：

总线传输实验：参考教材6.5

FPGA实验台操作方法：参考教材2.3.5