

第 1 章 TEC-8 计算机硬件综合实验系统	2
1.1 TEC-8 实验系统的用途	2
1.2 TEC-8 实验系统技术特点	2
1.3 TEC-8 实验系统组成	2
1.4 逻辑测试笔.....	3
1.5 TEC-8 实验系统结构和操作	4
1.6 模型计算机指令系统	7
1.7 开关、按钮、指示灯.....	8
1.8 数字逻辑和数字系统实验部分.....	10
1.9 E2PROM 中微代码的修改	11
第 2 章 计算机组织与体系结构基本实验	17
2.1 运算器组成实验	17
2.2 双端口存储器实验	26
2.3 数据通路实验.....	34
2.4 微程序控制器实验.....	42
2.5 CPU 组成与机器指令的执行	49
2.6 中断原理实验.....	54
第 3 章 课程综合设计.....	59
3.1 模型机硬连线控制器设计.....	59
3.2 模型机流水微程序控制器设计.....	67
3.3 模型机流水硬连线控制器设计.....	71
3.4 含有阵列乘法器的 ALU 设计.....	74
第 4 章 数字逻辑与数字系统综合设计实验	78
4.1 简易电子琴实验	78
4.2 简易频率计实验	83
4.3 简易交通灯实验	90
4.4 VGA 接口设计	95
第 5 章: Quartus II 软件简易使用	99
5.1 软件安装:	99
5.2 实验步骤	101

第 1 章 TEC-8 计算机硬件综合实验系统

1.1 TEC-8 实验系统的用途

TEC-8 计算机硬件综合实验系统，以下简称 TEC-8 实验系统，是清华大学科教仪器厂生产的一个**专利产品**，专利号 ZL 2007 2 0149391.9。它用于**数字逻辑与数字系统、计算机组成原理、计算机体系结构**三门课程的实验教学，也可用于数字系统的研究开发，为提高学生的动手能力、培养学生的创新精神提供了一个良好的舞台。

1.2 TEC-8 实验系统技术特点

(1) 模型计算机采用 8 位字长、简单而实用，有利于学生掌握模型计算机整机的工作原理。通过 8 位数据开关用手动方式输入二进制测试程序，有利于学生从最底层开始了解计算机工作原理。

(2) 指令系统采用 4 位操作码，可容纳 16 条指令。已实现加、减、与、加 1、存数、取数、条件转移、无条件转移、输出、中断返回、开中断、关中断和停机等 14 条指令，指令功能非常典型。

(3) 采用双端口存储器作为主存，实现数据总线和指令总线双总线体制，实现指令流水功能，体现出现代 CPU 设计思想。

(4) 控制器采用微程序控制器、硬连线控制器和独立 3 种类型，体现了当代计算机控制器技术的完备性。

(5) 微程序控制器、硬连线控制器和独立之间的转换采用独创的一次全切换方式，切换不用关掉电源，切换简单、安全可靠。

(6) 控制存储器中的微代码可用 PC 计算机下载，省去了 E²PROM 器件的专用编程器和对器件的插、拔。

(7) 运算器中 ALU 采用 2 片 74LS181 实现，4 个 8 位寄存器组用 1 片 EPM7064 实现，设计新颖。

(8) 每一条机器指令的时序采用不定长机器周期方式，符合现代计算机设计思想。

(9) 扩展小板的通用区提供了 15 个双列直插的器件插座，用于《数字逻辑和数字系统》课程的基本实验。

(10) 1 片在系统可编程器件 CPLD 可用于作为硬连线控制器使用，又可用于《数字逻辑与数字系统》课程的大型设计实验。为了安排大型设计实验，提供了用发光二极管代表的按东、西、南、北方向的安排的 12 个交通灯，6 个数码管，1 个喇叭和 1 个 VGA 接口。

(11) 设计《计算机组织与体系结构》课程实验考虑了与前导课程《数字逻辑与数字系统》实验的衔接。由于在《数字逻辑与数字系统》实验中已经进行了大量的接、插线实践，因此在 TEC-8 上进行《计算机组成与体系结构》课程实验接线较少，让学生把精力集中在实验现象的观察、思考和实验原理的理解上。

1.3 TEC-8 实验系统组成

整个 TEC-8 计算机硬件综合实验系统由下列部分构成：电源、试验台、下载电缆、通讯线。下面分别对各个组成部分予以介绍：

(1) 电源

安装在实验箱的下部，输出+5V，最大电流为 3A。220V 交流电源开关安装在实验箱的右

侧。220V 交流电源插座安装在实验箱的背面。实验台上有一个+5V 电源指示灯。

(2) 实验台

实验台安装在实验箱的上部，由一块印制电路板构成。TEC-8 模型计算机安装在这块印制电路板上。学生在实验台上进行实验。

(3) 下载电缆

用于将新设计的硬连线控制器或者其他电路下载到 CPLD 器件中。下载前必须将下载电缆的一端和 PC 机的并行口连接，另一端和实验台上的下载插座连接。

(4) 通讯线

通讯线分为两种一种是 COM 转 COM 的，另外一种 USB 转 COM 的。两种通讯线 COM 转 COM 的和 USB 转 COM 通讯线都是用于在 PC 机上在线修改控制存储器中的微代码。COM 转 COM 两端，一端直接连 PC 机器的 COM 口，另一端连试验台上的 COM 口。USB 通讯线一端接 PC 机的 USB 口，另一端接实验台上的 COM 口。(两种转换线在配件中二选一。原则上标配 COM 转 COM)。

1.4 逻辑测试笔

在数字电路实验中，对信号的测量是一个重要问题。常用的测试工具有示波器、万用表和逻辑测试笔。示波器的好处是直观、准确，用波形显示信号的状态，常用于对连续的周期波形进行测量，数字示波器对非周期信号的测量也很有效，缺点是造价较高。万用表价格便宜，使用方便，对信号电压能进行精确测量，缺点是不能测量脉冲信号。逻辑测试笔常用于测量信号的电平，判断一个较窄的脉冲是否发生以及发生了几个脉冲，缺点是无法对信号的电压作精确测量。数字电路实验中，关心的不是信号的具体电压而是信号的电平，逻辑测试笔作为一种方便、直观的测试工具，得到了广泛应用。TEC-8 实验台上许多信号都连接发光二极管作为指示灯，指示信号的电平，同时配备了逻辑测试笔。TEC-8 实验系统上配置的逻辑测试笔在测试信号的电平时，红灯亮表示高电平，绿灯亮表示低电平，红灯和绿灯都不亮表示高阻态。在测试脉冲个数时，首先按一次 Reset 按钮，使 2 个黄灯 D1、D0 灭，处于测试初始状态。TEC-8 实验台上的逻辑测试笔最多能够测试 3 个连续脉冲。测试信号的状态显示如表 1.1 所示。

表 1.1 指示灯对应得信号的状态

红绿	测试结果	D	测试结果
00	高阻态	00	没有脉冲
10	高电平	01	1 个脉冲
01	低脉冲	10	2 个脉冲
		11	3 个脉冲

数字电路的测试大体上分为静态测试和动态测试两部分。静态测试指的是给定数字电路若干静态输入值，测量输出是否正确。在静态测试基础上，给数字电路输入端加脉冲信号，用示波器或者逻辑测试笔测试数字电路输出是否正确。一般地说，时序电路应当进行动态测试。

1.5 TEC-8 实验系统结构和操作

1.5.1 模型计算机时序信号

TEC-8 模型计算机主时钟 MF 的频率为 1MHz，执行一条微指令需要 3 个节拍脉冲 T1、T2、T3。TEC-8 模型计算机时序采用不定长机器周期，绝大多数指令采用 2 个机器周期 W1、W2，少数指令采用一个机器周期 W1 或者 3 个机器周期 W1、W2、W3。

图 1.1 是 3 个机器周期的时序图。

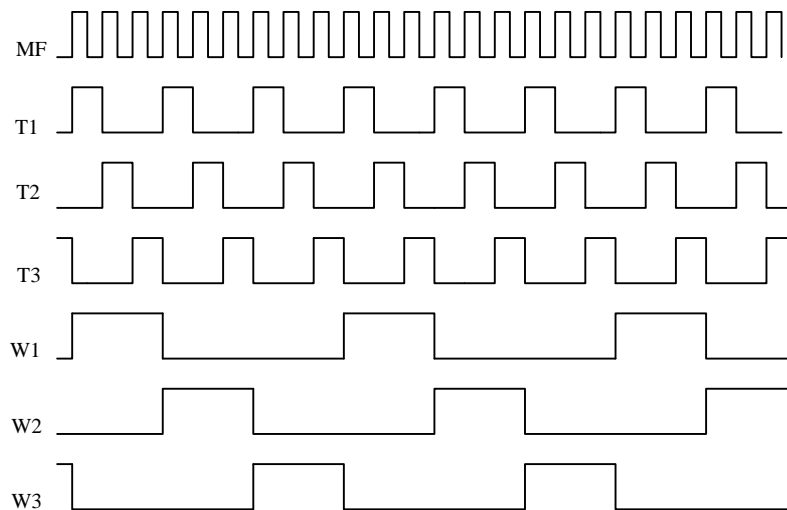


图 1.1 TEC-8 模型计算机时序图

1.5.2 模型计算机组成

图 1.2 是 TEC-8 模型计算机电路框图。下面介绍主要组成模块：时序发生器、算逻运算单元 ALU、双端口寄存器组 7064、数据开关 SD、双端口存储器 7132、组合逻辑控制器、微程序控制器、若干寄存器和若干选择器等；下面分别予以详细介绍：

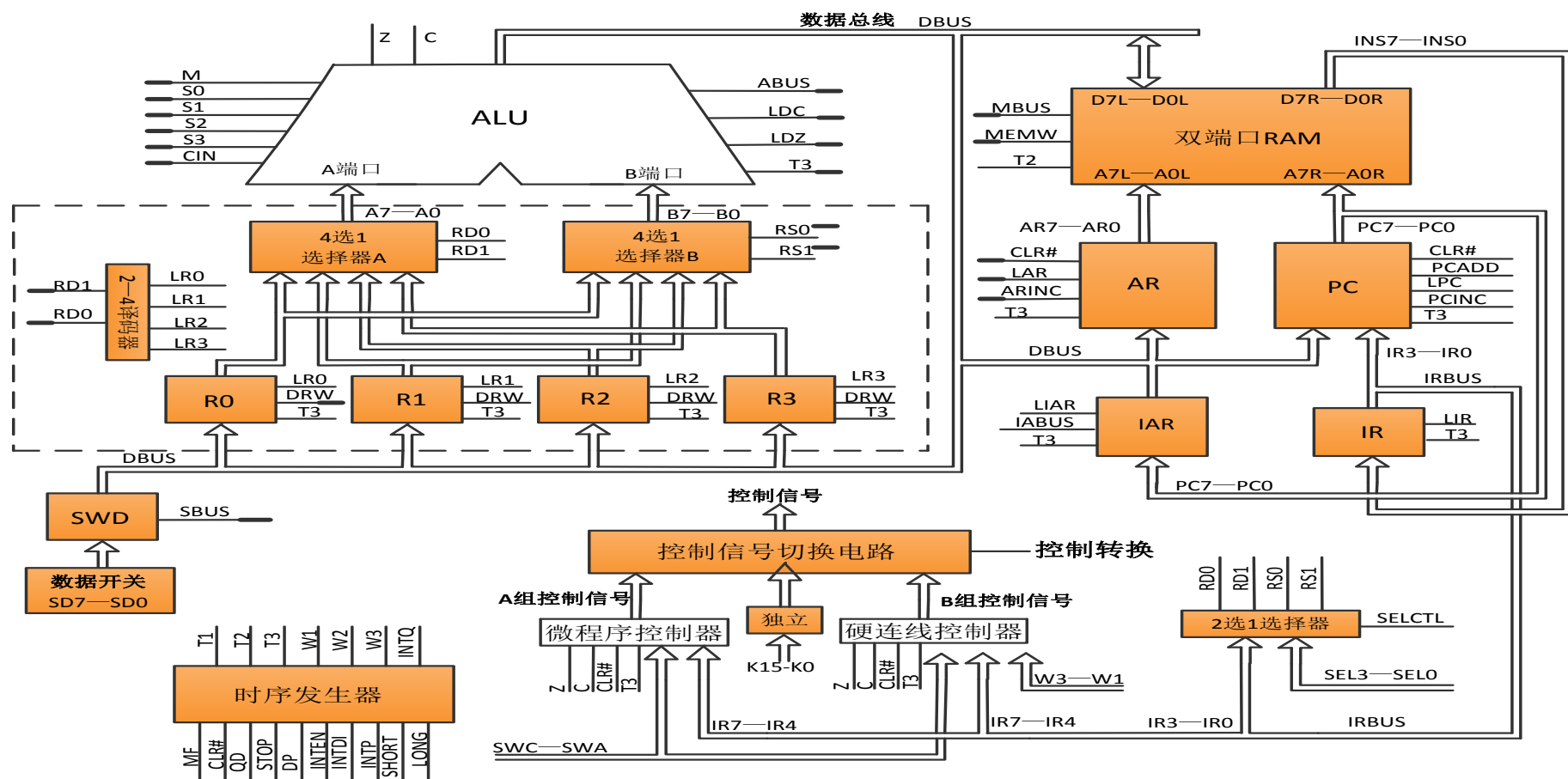


图 1.2 TEC-8 模型计算机框图

1. 时序发生器

它由 2 片 GAL22V10 组成，产生节拍脉冲 T1、T2、T3，节拍电位 W1、W2、W3，以及中断请求信号 ITNQ。主时钟 MF 采用石英晶体振荡器产生的 1MHz 时钟信号。T1、T2、T3 的脉宽为 1 微妙。一个机器周期包含一组 T1、T2、T3。

2. 算术逻辑单元 ALU

算术逻辑单元由 2 片 74LS181 加 1 片 74LS74、1 片 74LS244、1 片 74LS245、1 片 74LS30 组成，进行算术逻辑运算。74LS181 是一个 4 位的算术逻辑器件，2 个 74LS181 级联构成一个 8 位的算术逻辑单元。在 TEC-8 模型计算机中，算术逻辑单元 ALU 对 A 端口的 8 位数和 B 端口的 8 位数进行加、减、与、或和数据传送 5 种运算，产生 8 位数据结果、进位标志 C 和结果为 0 标志 Z。当信号 SBUS 为 1 时，将运算的数据结果送数据总线 DBUS。

3. 双端口寄存器组

双端口寄存器组由 ALTERA 公司的 1 片可编程器件 EPM7064 组成，向 ALU 提供两个运算操作数 A 和 B，保存运算结果。EPM7064 里面包含 4 个 8 位寄存器 R0、R1、R2、R3，4 选 1 选择器 A，4 选 1 选择器 B，2-4 译码器。在图 1.2 中，用虚线围起来的部分全部放在一个 EPM7064 中。4 个寄存器通过 4 选 1 选择器向 ALU 的 A 端口提供 A 操作数，通过 4 选 1 选择器 B 向 ALU 的 B 端口提供 B 操作数，2-4 译码器产生信号 LR0、LR1、LR2 和 LR3，选择保存运算数据结果的寄存器。

4. 数据开关 SD7~SD0

8 位数据开关 SD7~SD0 是双位开关，拨到朝上位置时表示“1”，拨到朝下位置时表示“0”。用于编制程序并把程序放入存储器，设置寄存器 R3~R0 的值。通过拨动数据开关 SD7~SD0 得到的程序或者数据通过 SWD 送往数据总线 DBUS。SWD 是 1 片 74LS244。

5. 双端口 RAM

双端口 RAM 由 1 片 IDT7132 及少许附加电路组成，存放程序和数据。双端口 RAM 是一种 2 个端口可同时进行读、写的存储器，2 个端口各有独立的存储器地址、数据总线和读、写控制信号。在 TEC-8 中，双端口存储器的左端口是个真正的读、写端口，用于程序的初始装入操作，从存储器中取数到数据总线 DBUS，将数据总线 DBUS 上的数写入存储器；右端口设置成只读方式，从右端口读出的指令 INS7~INS0 被送往指令寄存器 IR。

6. 程序计数器 PC、地址寄存器 AR 和中断地址寄存器 IAR

程序计数器 PC 由 2 片 GAL22V10 和 1 片 74LS244 组成向双端口 RAM 的左端口提供存储器地址 PC7~PC0，程序计数器 PC 具有 PC 复位功能，从数据总线 DBUS 上装入初始 PC 功能，PC 加 1 功能，PC 和转移偏量相加功能。

地址寄存器 AR 由 1 片 GAL22V10 组成，向双端口 RAM 的左端口提供存储器地址 AR7~AR0。它具有从数据总线 DBUS 上装入初始 AR 功能和 AR 加 1 功能。

中断地址寄存器 IAR 是 1 片 74LS374，它保存中断时的程序地址 PC。

7. 指令寄存器 IR

指令寄存器是 1 片 74LS273，用于保存从双端口 RAM 中读出的指令。它的输出 IR7~IR4 送往硬连线控制器、微程序控制器，IR3~IR0 送往 2 选 1 选择器。

8. 微程序控制器

微程序控制器产生 TEC-8 模型计算机所需的各种控制信号。它由 5 片 HN58C65、1 片 74LS174、3 片 74LS32 和 3 片 74LS06 组成。5 片 HN58C65 组成控制存储器，存放微程序代码；1 片 74LS174 是微地址寄存器。3 片 74LS32 和 3 片 74LS08 组成微地址转移逻辑。

9. 硬连线控制器

硬连线控制器由 1 片可编程器件 CPLD 组成，产生 TEC-8 模型计算机所需的各种控制信号。

10. 控制信号切换电路

控制信号切换器由 7 片 74LS244 和 1 个三刀转换开关组成。拨动一次转换开关,就能够实现一次控制信号的切换。当转换开关拨到朝上位置时,硬连线控制器灯亮,TEC-8 模型计算机使用硬连线控制器产生的控制信号;当转换开关拨到中间位置时,TEC-8 模型计算机各个部件独立,控制信号需要通过开关来控制;当转换开关拨到朝下位置时,TEC-8 模型计算机使用微程序控制器产生的控制信号。

11. 2 选 1 选择器

2 选 1 选择器由 1 片 74LS244 组成,用于在指令中的操作数 IR3~IR0 和控制信号 SEL3~SEL0 之间进行选择,产生目的寄存器编码 RD1、RD0,产生源寄存器编码 RS1、RS0。

1.6 模型计算机指令系统

TEC-8 模型计算机是个 8 位机,字长是 8 位。多数指令是单字指令,少数指令是双字指令。指令使用 4 位操作码,最多容纳 16 条指令。

已实现加法、减法、逻辑与、加 1、存数、取数、Z 条件转移、C 条件转移、无条件转移、输出、中断返回、开中断、关中断和停机 14 条指令。指令系统如表 12.2 所示。

表 1.2 TEC-8 模型计算机指令系统

名称	助记符	功 能	指令格式		
			IR(7-4)	IR(3-2)	IR(1-0)
加法	ADD Rd, Rs	$Rd \leftarrow Rd + Rs$	0001	Rd	Rs
减法	SUB Rd, Rs	$Rd \leftarrow Rd - Rs$	0010	Rd	Rs
逻辑与	AND Rd, Rs	$Rd \leftarrow Rd \text{ and } Rs$	0011	Rd	Rs
加 1	INC Rd	$Rd \leftarrow Rd + 1$	0100	Rd	XX
取数	LD Rd, [Rs]	$Rd \leftarrow [Rs]$	0101	Rd	Rs
存数	ST Rs, [Rd]	$Rs \rightarrow [Rd]$	0110	Rd	Rs
C 条件转移	JC addr	C=1, 则 $PC \leftarrow @ + \text{offset}$	0111	offset	
Z 条件转移	JZ addr	Z=1, 则 $PC \leftarrow @ + \text{offset}$	1000	offset	
无条件转移	JMP [Rd]	$PC \leftarrow Rd$	1001	Rd	XX
输出	OUT Rs	$DBUS \leftarrow Rs$	1010	XX	Rs
中断返回	IRET	返回断点	1011	XX	XX
关中断	DI	禁止中断	1100	XX	XX
开中断	EI	允许中断	1101	XX	XX
停机	STP	暂停运行	1110	XX	XX

1.2 中,XX 代表随意值。Rs 代表源寄存器号,Rd 代表目的寄存器号。在条件转移指令中,@代表当前 PC 的值,offset 是一个 4 位的有符号数,第 3 位是符号位,0 代表正数,1 代表负数。**注意: @不是当前指令的 PC 值,是当前指令的 PC 值加 1。**

指令系统中,指令操作码 0000B 没有对应的指令,实际上指令操作码 0000B 对应着一条 nop 指令,即什么也不做的指令。当复位信号为 0 时,对指令寄存器 IR 复位,使 IR 的值为 00000000B,对应一条 nop 指令。这样设计的目的是适应指令流水的初始状态要求。

1.7 开关、按钮、指示灯

为了在实验过程中观察各种数据，TEC-8 实验系统设置了大量的指示灯。

1、与运算器有关的指示灯

信号名称	信号灯符号	备 注
数据总线指示灯	D7~D0	
运算器 A 端口指示灯	A7~A0	
运算器 B 端口指示灯	B7~B0	
进位信号指示灯	C	
结果为 0 信号指示灯	Z	

2、与存储器有关的指示灯

信号名称	信号灯符号	备 注
程序计数器指示灯	PC7~PC0	
地址指示灯	AR7~AR0	
存储器右端口数据指示灯	INS7~INS0	
指令寄存器指示灯	IR7~IR0	
双端口存储器右端口数据指示灯	INS7~INS0	

3、与微程序控制器有关的信号指示灯

在使用微程序控制器时，控制信号指示灯指示微程序控制器产生的控制信号以及后继微地址 $N\mu A5 \sim N\mu A0$ 和判别位 $P4 \sim P0$ ，微地址指示灯指示当前的微地址 $\mu A5 \sim \mu A0$ ；微地址指示灯 $\mu A5 \sim \mu A0$ 、后继微地址 $N\mu A4 \sim N\mu A0$ 和判别位指示灯 $P4 \sim P0$ 只在微程序控制器方式下有实际意义，在硬布线控制器和独立方式下没有实际意义。

信号名称	信号灯符号	备 注
微地址指示灯	$\mu A5 \sim \mu A0$	
后继微地址指示灯	$N\mu A5 \sim N\mu A0$	
判别位指示灯	$P4 \sim P0$	

4、节拍脉冲信号和节拍电位信号指示灯

按下启动按钮 QD 后，至少产生一组节拍脉冲 T1、T2、T3，无法用指示灯显示 T1、T2、T3 的状态，因此设置了 T1、T2、T3 观测插孔，使用 TEC-8 实验台上提供的逻辑测试笔能够观测 T1、T2、T3 是否产生。

硬连线控制器产生的节拍电位信号 W1、W2 和 W3 有对应的指示灯。

5、其它指示灯

信号名称	信号灯符号
控制台操作指示灯	=1 时，表明进行控制台操作；
	=0 时，表明运行测试程序；
硬连线控制器指示灯	=1 时，表明使用硬连线控制器；
	=0 时，表明使用微程序控制器或者独立状态；
+5V 指示灯	指示+5V 电源的状态

TEC-8 实验平台上有下列按钮：

1、启动按钮 QD

按一次启动按钮 QD，则产生 2 个脉冲 QD 和 QD#。QD 为正脉冲，QD#为负脉冲，脉冲的宽度与按下 QD 按钮的时间相同。正脉冲 QD 启动节拍脉冲信号 T1、T2 和 T3。

2、复位按钮 CLR

按一次复位按钮 CLR，则产生 2 个脉冲 CLR 和 CLR#。CLR 为正脉冲，CLR#为负脉冲，脉冲的宽度与按下 CLR 按钮的时间相同。负脉冲 CLR#使 TEC-6 模型计算机复位，处于初始状态。

3. 中断按钮 PULSE

按一次中断按钮 PULSE，则产生 2 个脉冲 PULSE 和 PULSE#。PULSE 为正脉冲，PULSE#为负脉冲，脉冲的宽度与按下 PULSE 按钮的时间相同。正脉冲 PULSE 向 TEC-8 模型计算机发出中断请求。

TEC-8 实验台上有下列开关：

1、数据开关 SD7~SD0

这 8 个双位开关用于向寄存器中写入数据、向存储器中写入程序或者用于设置存储器初始地址。当开关拨到朝上位置时为 1，拨到向下位置时为 0。

2、电平开关 S15~S0

这 16 个双位开关用于在实验时设置信号的电平。每个开关上方都有对应的接插孔，供接线使用。开关拨到朝上位置时为 1，拨到向下位置时为 0。

3、单微指令开关 DP

单微指令开关控制节拍脉冲信号 T1、T2、T3 的数目。当单微指令开关 DP 朝上时，处于单微指令运行方式，每按一次 QD 按钮，只产生一组 T1、T2、T3；当单微指令开关 DP 朝下时，处于连续运行方式，每按一次 QD 按钮，开始连续产生 T1、T2、T3，直到按一次 CLR 按钮或者控制器产生 STOP 信号为止。

4. 控制器转换开关

当控制器转换开关朝上时，使用硬连线控制器；当控制器转换开关朝下时，使用微程序控制器。

5. 编程开关

当编程开关朝下时，TEC-8 模型计算机处于正常工作状态；当编程开关朝上时，处于编程状态。在编程状态下，修改控制存储器中的微代码状态，。

6、操作模式开关 SWC、SWB、SWA

操作模式 SWC、SWB、SWA 确定的 TEC-8 模型计算机操作模式如下：

操作模式	实验功能	备注
000	启动程序运行	
001	写存储器	
010	读存储器	
011	读寄存器	
100	写寄存器	
101	运算器组成实验	
110	双端口存储器实验	
111	数据通路实验	

1.8 数字逻辑和数字系统实验部分

TEC-8 实验系统能够满足《数字逻辑和数字系统》课程的实验要求，既可以进行基本实验，也可以进行大型综合性设计实验。

1.8.1 基本实验通用区

基本实验通用区位于 TEC-8 实验台的左上部，里面安排了 2 个 14 芯、2 个 16 芯、2 个 20 芯、1 个 24 芯、1 个 28 芯双列直插插座，供使用中、小规模数字逻辑器件做基本实验用。另外在实验台的中下部还有 1 个 500 欧姆的电位器，当电位器的一端接+5V、另一端接地后，旋转电位器可以改变电位器中间抽头的电压。它可以作为数字器件的输入电压，供测试器件的输入、输出特性使用。

1.8.2 大型综合设计实验装置

为了进行大型综合设计实验，TEC-8 上安排了如下实验装置。

1. 6 个数码管及驱动电路
2. 1 个喇叭及驱动电路
3. 1 个 VGA 接口及驱动电路
4. 12 个发光二极管及驱动电路。12 个发光二极管按东、西、南、北方向设置，每个方向安排红、黄、绿三种颜色的发光二极管，模仿交通灯。
5. 1 个同时可产生 7 路时钟的信号发生器。这 7 路时钟的频率分别是 1MHz、100KHz、10KHz、1KHz、100Hz、10Hz、1Hz，占空比为 50%。其中 1MHz 信号就是 TEC-8 的主时钟 MF；100KHz、10KHz 信号可以通过短路子 DZ3 和 DZ4 进行二选一选择，产生信号 CP1；1KHz、100Hz 信号可以通过短路子 DZ5 和 DZ6 进行二选一选择，产生信号 CP2；10Hz、1Hz 信号可以通过短路子 DZ7 和 DZ8 进行二选一选择，产生信号 CP3。**注意：短路子 DZ3 和 DZ4 不能同时短接；短路子 DZ5 和 DZ6 不能同时短接；短路子 DZ7 和 DZ8 不能同时短接。**时钟信号 MF、CP1、CP2 和 CP3 通过插孔输出，或者通过扁平电缆连接到 EPM7128 的引脚。
6. 一条扁平电缆。当进行大型综合设计实验时，有些实验需要通过扁平电缆将需要的信号和器件 CPLD 的引脚连接。扁平电缆的一端接 34 芯插座 J6 (J6 和 CPLD 的引脚相连)；另一端分为三部分，第一部分接 16 芯插座 J8 (J8 和开关 S₁₅~S₀相连)；第二部分接 12 芯插座 J4 (J4 和 12 个发光二极管 L₁₁~L₀相连)或者接 12 芯插座 J1 (J1 和数码管 LG2、LG1 的驱动相连)；第三部分接 6 芯插座 J5 (J5 和 5 中的时钟信号以及正脉冲 QD、PULSE 相连)。

1.9 E2PROM 中微代码的修改

1. E²PROM 的两种工作方式

TEC-8 模型计算机中的 5 片 E²PROM(CM4~CM0)有两种工作方式，一种叫“正常”工作方式，作为控制存储器使用；一种叫“编程”工作方式并且 DZ11 的短路子放置在“编程”位置，可用于修改 E²PROM 的微代码。当编程开关拨到“正常”位置并且 DZ11 放置在“正常”位置，则 TEC-8 可以正常做实验，CM4~CM0 作为控制存储器使用，里面的微代码正常读出，供数据通路使用。当编程开关拨到“编程”位置时，CM4~CM0 只受 TEC-8 实验系统中的单片机和 DZ11 的控制，用来对 5 片 E²PROM 编程。在编程状态下，不进行正常实验。**特别提示：正常实验时编程开关的位置必须拨到“正常”位置、DZ11 短路子必须放置在左边【“正常”位置】，否则可能破坏 E²PROM 原先的内容。**

2. 安装 Prolific usb-to-serical comm prot 驱动程序

如果通讯线不是 COM 对 COM，而是 COM 对 USB，则需要安装驱动。PC 计算机通过 RS232 串行通讯方式和 TEC-8 实验系统中的单片机 89S52 通讯，从而达到修改控制存储器 E²PROM 的目的。如果 TEC-8 实验系统上的编程线采用 USB 对 COM 通讯线，因此需要一个驱动程序，将 USB 通讯方式转换为 RS232 通讯方式，这个驱动程序就是 Prolific usb-to-serical comm prot。出厂时提供的光盘上有这个驱动程序，在出厂资料盘/7 驱动文件夹内。

当第一次用出厂时提供的编程电缆将 PC 机的一个 USB 口和 TEC-8 实验系统上的 USB 口连接时，PC 机自动检测出安装了新硬件，并自动启动“安装新硬件驱动程序”服务，在 PC 机屏幕上弹出“找到新的硬件向导”第一个对话框，如图 1.3 所示。

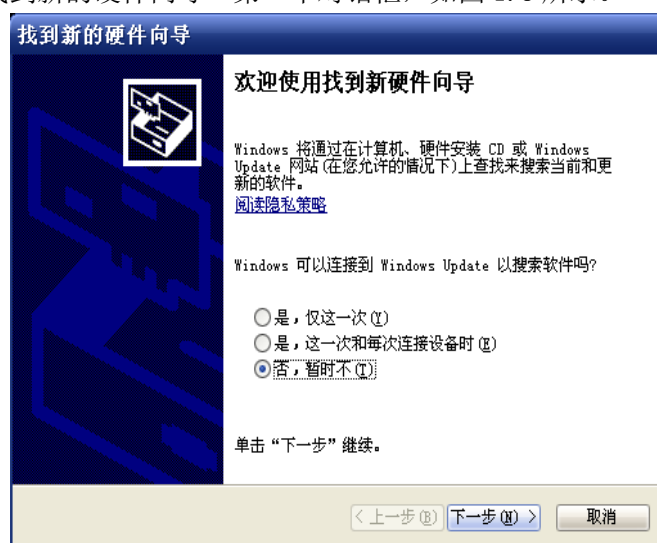


图 1.3 找到新硬件向导对话框 1

直接点击找到新的硬件向导右下角的取消框，采用其他方式安装。打开出厂资料盘/7 驱动目录，直接安装驱动即可。安装完毕后如下所示：



图 1.4 初安装完后串口状态

现在串口的端口号为 COM5，如果需要调整，则如下操作所示：

1. 在 COM5 上右击属性如下所示：



图 1.5 端口属性

2. 点击端口属性如下所示：



1.6 端口设置

3. 更改端口号如下所示：



图 1.7 更改端口号

4. 点击确定后在计算机管理中刷新即可得到需要的端口号，如下所示：



图 1.8 设置完成的串口

3. 串口调试助手 2.2 介绍

顾名思义，串口调试助手是一个调试 PC 机串口的程序，在 TEC-8 实验系统中，首先在 PC 机上通过串口调试程序将新的 E²PROM 数据下载到单片机中，由单片机完成对 E²PROM 的编程。

串口调试助手使用极其简单。通过双击出厂时提供的该软件的图标，PC 机屏幕上出现如图 1.8 所示的该软件对话框。



图 1.9 串口调试助手对话框

(1) 选择串口号

选择和 TEC-8 通讯使用的串口号,在 COM1~COM4 中选择一个。串口的设置要与 CP2102 USB to UART Bridge Controller 驱动程序将 USB 转换的 R232 串口号一致。该串口号可用下列方式得到。

在用编程电缆将 PC 机一个 USB 口和 TEC-8 实验系统连接的情况下,用鼠标右键点击 PC 机桌面上的“我的电脑”图标,弹出一个菜单,如图 1.9 所示。

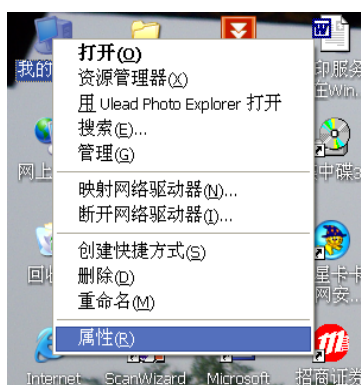


图 1.10 “我的电脑”操作菜单

如图 1.9 那样,点击“属性(R)”菜单项,弹出系统属性对话框,如图 1.10。



图 1.11 系统属性对话框

选中“硬件”菜单项后，点击“设备管理器”按钮，弹出设备管理器窗口，如图 1.12 所示。



图 1.12 设备管理器窗口

在设备管理器窗口中可以找到该 USB 口代替的串口号。图 1.11 中是 COM1。具体的串口号根据 PC 机和当时 PC 机的具体环境而定。

(2) 设置波特率等参数

由于串口调试助手需要和 TEC-8 实验系统上的单片机通讯，因此它设置的串口参数需要和单片机内设置的参数一致，即波特率为 2400 波特，数据位 8 位，无校验位，停止位 1 位。这些参数设置不正确将无法通讯。

(3) 窗口下部空白区为 PC 数据发送窗口，其上面较大的空白区为 PC 数据接收窗口。

4. 修改 CM4~CM0 的步骤(备注：原则上不允许修改以免设备不能正常工作，如果需要修改请联系技术支持部门即可，当然 也可以自行修改，有其他疑问可再联系技术支持部门。如果想批量的修改 ROM 中的内容，可以查看随机光盘，内有 TEC8DOWNTO 软件，不用安装下在即可。跳线方式可参考 E²PROM 的两种工作方式中的特别提示。)

(1) 编写二进制格式的微代码文件

微代码文件的格式是二进制。TEC-8 实验系统上使用的 E²PROM 的器件型号是 HN58C65。虽然 1 片 HN58C65 的容量是 2048 个字节,但是在 TEC-8 实验系统中作为控制存储器使用时,每片 HN58C65 都只使用了 64 个字节。因此在改写控制存储器内容时,首先需要生成 5 个二进制文件,每个文件包含 64 个字节。

(2) 连接编程电缆

在 TEC-8 关闭电源的情况下,用出厂时提供的编程电缆将 PC 机的一个 USB 口和 TEC-8 实验系统上的 USB 口相连。

(3) 将编程开关拨到“编程”位置,DZ11 短路子放置“编程”位置。

(4) 将串口调试助手程序打开,设置好串口号和参数。

(5) 打开电源,按一下单片机复位键。

(6) 发送微代码。

串口调试助手的接收区此时会显示信息‘WAITING FOR COMMAND ...’,提示等待命令。这个等待命令的提示信息是 TEC-8 实验系统发送给串口调试助手的,表示 TEC-8 实验系统已准备好接受命令。

一共有 5 个命令,分别是 0、1、2、3 和 4,分别对应被编程的 CM0、CM1、CM2、CM3 和 CM4。

如果准备修改 CM0,则在数据发送区写入‘0’,按‘手动发送’按钮,将命令‘0’发送给 TEC-8 实验系统,通知它要写 CM0 文件了。

数据接收区会出现‘PLEASE CHOOSE A CM FILE’。通过点击“选择发送文件”按钮选择要写入 CM0 的二进制文件。然后点击“发送文件”按钮将文件发往 TEC-8 实验系统。

TEC-8 实验系统接收数据并对 CM0 编程,然后它读出 CM0 的数据和从 PC 机接收到数据比较,不管正确与否,TEC-8 实验系统都向串口调试助手发回结果信息,在数据接收窗口显示出来。

对一个 E²PROM 编程完成后,根据需要可再对其它 E²PROM 编程,全部完成后,按一次 TEC-8 实验系统上的“单片机复位”按钮结束编程。最后将编程开关拨到“正常”位置,并且将 DZ11 放置在左端。

注意:对 CM0、CM1、CM2、CM3 和 CM4 的编程顺序无规定,只要在发出器件号后紧跟着发送该器件的编程数据(文件)即可。编程也可以只对一个或者几个 E²PROM 编程,不一定对 5 个 E²PROM 全部编程。

第 2 章 计算机组织与体系结构基本实验

2.1 运算器组成实验

2.1.1 运算器组成实验：为程序控制器方式

一、实验类型：原理性+分析性

二、实验目的

- (1)熟悉逻辑测试笔的使用方法。
- (2)熟悉 TEC-8 模型计算机的节拍脉冲 T1、T2、T3；
- (3)熟悉双端口通用寄存器组的读写操作；
- (4)熟悉运算器的数据传送通路；
- (5)验证 74LS181 的加、减、与、或功能；
- (6)按给定的数据，完成几种指定的算术、逻辑运算运算。

三、实验设备

序列号	名 称	数 量	备 注
1	TEC-8 实验系统	1 台	
2	双踪示波器	1 台	
3	直流万用表	1 块	
4	逻辑测试笔	1 支	在实验箱上方

四、实验电路

为了进行本实验，首先需要了解 TEC-8 模型计算机的基本时序。在 TEC-8 中，执行一条微指令(或者在硬连线控制器中完成 1 个机器周期)需要连续的 3 个节拍脉冲 T1、T2 和 T3。它们的时序关系如图 2.1 所示：

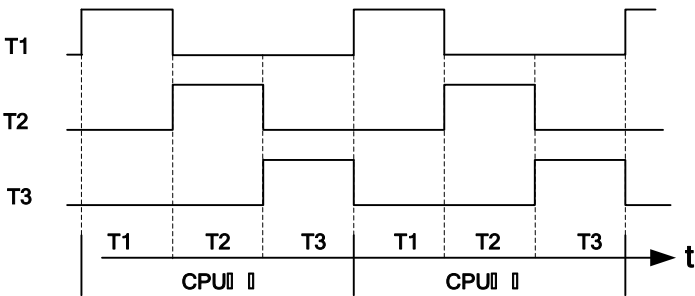


图 2.1 机器周期与 T1、T2、T3 时序关系图

对于运算器操作来说，在 T1 期间，产生 2 个 8 位参与运算的数 A 和 B，A 是被加数，B 是加数；产生控制运算类型的信号 M、S3、S2、S1、S0 和 CIN；产生控制写入 Z 标志寄存器的信号 LDZ 和控制写入 C 标志寄存器的信号 LDC，产生将运算的数据结果送往数据总线 DBUS 的控制信号 ABUS。这些控制信号保持到 T3 结束；在 T2 期间，根据控制信号，完成某种运算功能；在 T3 的上升沿，保存运算的数据结果到一个 8 位寄存器中，同时保存进位标志 C 和结果为 0 标志 Z。

图 2.2 是运算器组成实验的电路图。

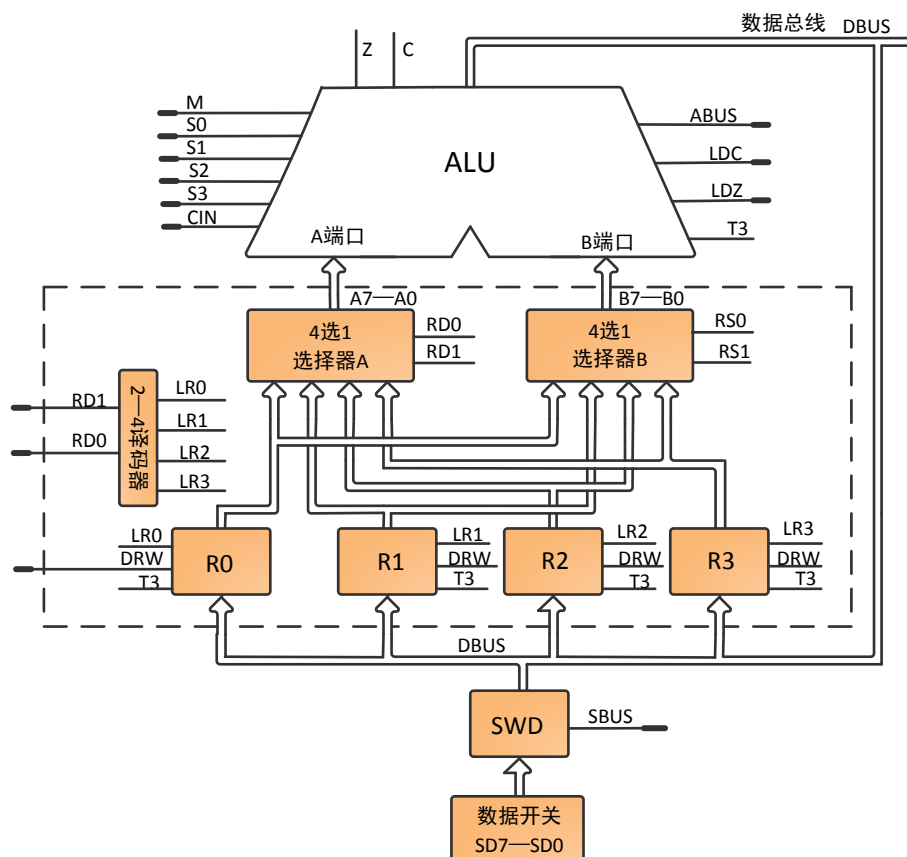


图 2.2 运算器组成实验电路图

双端口寄存器组由 1 片 EPM7064 (U40) (图 2.2 中用虚线围起来的部分) 组成, 内部包含 4 个 8 位寄存器 R0、R1、R2、R3, 4 选 1 选择器 A, 4 选 1 选择器 B 和 1 个 2-4 译码器。根据信号 RD1、RD0 的值, 4 选 1 选择器 A 从 4 个寄存器中选择 1 个寄存器送往 ALU 的 A 端口。根据信号 RS1、RS0 的值, 4 选 1 选择器 B 从 4 个寄存器中选择 1 个寄存器送往 ALU 的 B 端口。2-4 译码器对信号 RD1、RD0 进行译码, 产生信号 LR0、LR2、LR3、LR4, 任何时刻这 4 个信号中只有一个为 1, 其它信号为 0。LR3~LR0 指示出被写的寄存器。当 DRW 信号为 1 时, 如果 LR0 为 1, 则在 T3 的上升沿, 将数据总线 DBUS 上的数写入 R0 寄存器, 余类推。

数据开关 SD7~SD0 是 8 个双位开关。用手拨动这些开关, 能够生成需要的 SD7~SD0 的值。数据开关驱动器 SWD 是 1 片 74 LS 244 (U50)。在信号 SBUS 为 1 时, SD7~SD0 通过 SWD 送往数据总线 DBUS。在本实验中, 使用数据开关 SD7~SD0 设置寄存器 R0、R1、R2 和 R3 的值。

ALU 由 2 片 74LS181 (U41 和 U42)、1 片 74LS74、1 片 74 LS 244、1 片 74 LS 245 和 1 片 74LS30 构成。74LS181 完成算术逻辑运算, 74 LS 245 和 74 LS 30 产生 Z 标志, 74 LS 74 保存标志 C 和标志 Z。ALU 对 A7~A0 和 B7~B0 上的 2 个 8 位数据进行算术逻辑运算, 运算后的数据结果在信号 ABUS 为 1 时送数据总线 DBUS (D7~D0), 运算后的标志结果在 T3 的上升沿保存进位标志位 C 和结果为 0 标志位 Z。加法和减法同时影响 C 标志和 Z 标志, 与操作和或操作只影响 Z 标志。

应当指出, 74LS181 只是许多种能做算术逻辑运算器件中的一种器件, 这里它仅作为一个例子使用。

74LS181 能够进行 4 位的算术逻辑运算, 2 片 74 LS 181 级连在一起能够 8 位运算, 3 片 74LS181 级连在一起能够进行 12 位运算, 余类推。所谓级联方式, 就是将低 4 位 74LS181

的进位输出引脚 $\overline{C_{n+4}}$ 与高 4 位 74LS181 的进位输入引脚 $\overline{C_n}$ 连接。在 TEC-8 模型计算机中，U42 完成低 4 位运算，U41 完成高 4 位运算，二者级连在一起，完成 8 位运算。在 ABUS 为 1 时，运算得到的数据结果送往数据总线 DBUS。数据总线 DBUS 有 4 个信号来源：运算器、存储器、数据开关和中断地址寄存器，在每一时刻只允许其中一个信号源送数据总线。本实验中用到的信号归纳如下：

序列号	信号名	功能说明
1	S3、S2、S1、S0	控制 74LS181 的运算类型
2	CIN	低位 74LS181 的进位输入
3	SEL3、SEL2(RD1、RD0)	选择送 ALU 的 A 端口的寄存器
4	SEL1、SEL0(RS1、RS0)	选择送 ALU 的 B 端口的寄存器
5	DRW	=1 时，在 T3 上升沿对 RD1、RD0 选中的寄存器进行写操作，将数据总线 DBUS 上的数 D7~D0 写入选定的寄存器
6	SETCTL	=1 时，实验系统处于实验台状态。 =0 时，实验系统处于运行程序状态
7	SBUS	=1 时，将运算结果送数据总线 DBUS =0 时，禁止运算结果送数据总线 DBUS
8	ABUS	=1 时，将运算结果送数据总线 DBUS
9		=0 时，禁止运算结果送数据总线 DBUS
10	M	运算模式：M=0 为算术运算；M=1 逻辑运算；
11	LDZ	=1 时，如果运算结果为 0，在 T3 的上升沿，将 1 写入到 Z 标志寄存器；如果运算结果不为 0，将 0 保存到 Z 标志寄存器。
12	LDC	=1 时，在 T3 的上升沿将运算得到的进位保存到 C 标志寄存器。
13	A7~A0	送往 ALU 的 A 端口的数
14	B7~B0	送往 ALU 的 B 端口的数
15	D7~D0	数据总线 DBUS 上的 8 位数
16	C	进位标志
17	Z	结果为 0 标志

上述信号都有对应的指示灯。当指示灯灯亮时，表示对应的信号为 1；当指示灯不亮时，对应的信号为 0。实验过程中，对每一个实验步骤，都要记录上述信号(可以不纪录 SETCTL)的值。另外 $\mu A5 \sim \mu A0$ 指示灯指示当前微地址。

应当指出，74LS181 对减法运算采用的是补码运算方式，即先求得 $[-\text{减数}]$ 的补码，然后和被减数的补码相加的方式完成。因此一个较大的数减去一个较小的数，或者 2 个相等的数相减时产生进位。

五、实验任务

1. 用双踪示波器和逻辑测试笔测试节拍脉冲信号 T1、T2、T3。

2. 对下述 7 组数据进行加、减、与、或运算。

- | | | | |
|-----------------|-----------------|-----------------|-----------------|
| (1)A=F0H, B=10H | (2)A=FFH, B=AAH | (3)A=10H, B=F0H | (4)A=55H, B=AAH |
| (5)A=03H, B=05H | (6)A=C5H, B=61H | (7)A=0AH, B=0AH | |

六、实验步骤

1. 实验准备

将控制器转换开关拨到微程序位置，将编程开关设置为正常位置，将开关 DP 拨到向上位置。打开电源。

2. 用逻辑测试笔测试节拍脉冲信号 T1、T2、T3

(1)将逻辑测试笔的一端插入 TEC-8 实验台上的“逻辑测试笔”上面的插孔中，另一端插入“T1”上方的插孔中。

(2)按复位按钮 CLR，使时序信号发生器复位。

(3)按一次逻辑测试笔框内的 Reset 按钮，使逻辑测试笔上的脉冲计数器复位，2 个黄灯 D1、D0 均灭。

(4)按一次启动按钮 QD，这时指示灯 D1、D0 的状态应为 01B，指示产生了一个 T1 脉冲；如果再按一次 QD 按钮，则指示灯 D1、D0 的状态应当为 10B，表示又产生了一个 T1 脉冲；继续按 QD 按钮，可以看到在单周期运行方式下，每按一次 QD 按钮，就产生一个 T1 脉冲。

(5)用同样的方法测试 T2、T3。

3. 进行加、减、与、或实验

(1)设置加、减、与、或实验模式

按复位按钮 CLR，使 TEC-8 实验系统复位。指示灯 $\mu A5 \sim \mu A0$ 显示 00H。将操作模式开关设置为 SWC=1、SWB=0、SWA=1，准备进入加、减、与、或实验。

按一次 QD 按钮，产生一组节拍脉冲信号 T1、T2、T3，进入加、减、与、或实验。

(2)设置数 A

指示灯 $\mu A5 \sim \mu A0$ 显示 0BH。在数据开关 SD7~SD0 上设置数 A。在数据总线 DBUS 指示灯 D7~D0 上可以看到数据设置的正确不正确，发现错误需及时改正。设置数据正确后，按一次 QD 按钮，将 SD7~SD0 上的数据写入 R0，进入下一步。

(3)设置数 B

指示灯 $\mu A5 \sim \mu A0$ 显示 15H。这时 R0 已经写入，在指示灯 B7~B0 上可以观察到 R0 的值。在数据开关 SD7~SD0 上设置数 B。设置数据正确后，按一次 QD 按钮，将 SD7~SD0 上的数据写入 R1，进入下一步。

(4)进行加法运算

指示灯 $\mu A5 \sim \mu A0$ 显示 16H。指示灯 A7~A0 显示被加数 A(R0)，指示灯 B7~B0 显示加数 B(R1)，D7~D0 指示灯显示运算结果 A+B。按一次 QD 按钮，进入下一步。

(5)进行减法运算

指示灯 $\mu A5 \sim \mu A0$ 显示 17H。这时指示灯 C(红色)显示加法运算得到的进位 C，指示灯 Z(绿色)显示加法运算得到的结果为 0 信号。指示灯 A7~A0 显示被减数 A(R0)，指示灯 B7~B0 显示减数 B(R1)，指示灯 D7~D0 显示运算结果 A-B。按一次 QD 按钮，进入下一步。

(6)进行与运算

指示灯 $\mu A5 \sim \mu A0$ 显示 18H。这时指示灯 C(红色)显示减法运算得到的进位 C，指示灯 Z(绿色)显示减法运算得到的结果为 0 信号。

指示灯 A7~A0 显示数 A(R0)，指示灯 B7~B0 显示数 B(R1)，指示灯 D7~D0 显示运算结果 A and B。按一次 QD 按钮，进入下一步。

(7)进行或运算

指示灯 $\mu A5 \sim \mu A0$ 显示 19H。这时指示灯 Z(绿色)显示与运算得到的结果为 0 信号。指示灯 C 保持不变。指示灯 A7~A0 显示数 A(R0)，指示灯 B7~B0 显示数 B(R1)，指示灯 D7~D0 显示运算结果 A or B。按一次 QD 按钮，进入下一步。

(8)结束运算

指示灯 $\mu A5 \sim \mu A0$ 显示 00H。这时指示灯 Z(绿色)显示或运算得到的结果为 0 信号。指示灯 C 保持不变。

按照上述步骤，对要求的 7 组数据进行运算。

七、实验要求

1. 做好实验预习，掌握运算器的数据传输通路及其功能特性。
2. 写出实验报告，内容是：

(1)实验目的。

(2)根据实验结果填写表 2.1。

表 2.1 运算器组成实验结果数据表

实验数据		实验结果									
数 A	数 B	加			减			与		或	
		数据结果	C	Z	数据结果	C	Z	数据结果	Z	数据结果	Z

(3)结合实验现象，每一实验步骤中，对下述信号在所起的作用进行解释：M、S0、S1、S2、S3、CIN、ABUS、LDC、LDZ、SEL3、SEL2、SEL1、SEL0、DRW、SBUS。并说明在该步骤中，哪些信号是必需的，哪些信号不是必需的，哪些信号必需采用实验中使用的值，哪些信号可以不采用实验中使用的值。

八、可探索和研究的问题

1. ALU 具有记忆功能吗？如果有，如何设计？
2. 为什么在 ALU 的 A 端口和 B 端口的数据确定后，在数据总线 DBUS 上能够直接观测运算的数据结果，而标志结果却在下一步才能观测到？

2.1.2 运算器组成实验：独立方式

一、实验类型：原理性+分析性+操作性

二、实验目的

- (1)熟悉逻辑测试笔的使用方法。
- (2)熟悉 TEC-8 模型计算机的节拍脉冲 T1、T2、T3；
- (3)熟悉双端口通用寄存器组的读写操作；
- (4)熟悉运算器的数据传送通路；
- (5)验证 74LS181 的加、减等各种功能；
- (6)按照表中提供的功能自行验证其中几种即可。

三、实验设备

序列号	名称	数量	备注
1	TEC-8 实验系统	1 台	
2	双踪示波器	1 台	
3	直流万用表	1 块	
4	逻辑测试笔	1 支	在实验箱上方

四、实验电路和实验任务

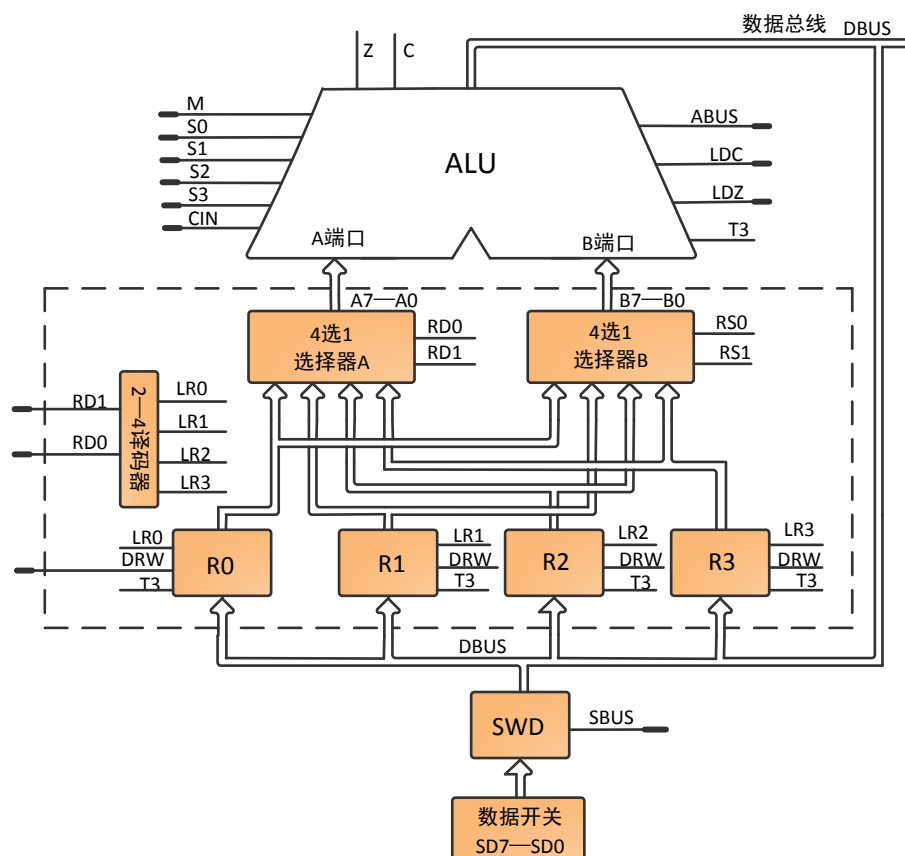


图 2.3 运算器组成实验电路图

(1) 实验电路

上图 2.3 标识出了本实验所用的运算器数据通路图。参与运算的数据首先通过试验台操作板上的 8 个二进制数据开关 SD7-SD0 来设置，然后输入到双端口通用寄存器堆 RF 中。

双端口寄存器堆 RF 由 1 个 ALTERA EPM 7064 实现，功能相当于 4 个八位通用寄存器，用于保存参与运算的数据，运算后的结果也要送到双端口通用寄存器堆 RF 中保存。双端口寄存器堆模块 RF 的控制信号中 RD1、RD0 用于选择送 ALU 的 A 端口(左端口)的通用寄存器。RS1、RS0 用于选择送 ALU 的 B 端口(右端口)的通用寄存器。

(2) 实验任务

1. 按图所示，将运算器模块与实验台操作板上的线路进行连接。由于运算器模块内部的连线已经由印制电路板连接好，故接线任务仅仅是完成数据开关、控制信号模拟开关、与运算模块的外部连线。特别注意：为了建立清楚的整机概念，培养严谨的科研能力，手工连线时绝对有必要的。
2. 用开关 K15-K0 向通用寄存器堆 RF 内的 R3-R0 寄存器置数据。然后读出 R3-R0 的数据，在数据总线 DBUS 上显示出来。
3. 验证 ALU 的正逻辑算术、逻辑运算功能。

注意：进位信号 C 是运算器 ALU 最高进 C_{n+4} 的反，既有进位为 1，无进位为 0。

选择方式				M=1 逻辑运算	M=0 算术运算
S3	S2	S1	S0	逻辑运算	CN=1 【有进位】
0	0	0	0	$F = \neg A$	$F = A$
0	0	0	1	$F = \neg (A+B)$	$F = (A+B)$
0	0	1	0	$F = (\neg A)B$	$F = A + \neg B$
0	0	1	1	$F = 0$	$F = -1$ (补码形式)
0	1	0	0	$F = \neg (AB)$	$F = A$ 加 $\neg B$
0	1	0	1	$F = \neg B$	$F = (A+B)$ 加 $\neg A$
0	1	1	0		$F = A$ 减 B 减 1
0	1	1	1	$F = A/B$	$F = (AB/\neg A)$ 减 1
1	0	0	0	$F = \neg A + B$	$F = A$ 加 AB
1	0	0	1		$F = A$ 加 B
1	0	1	0	$F = B$	$F = (A + \neg B)$ 加 AB
1	0	1	1	$F = AB$	$F = AB$ 减 1
1	1	0	0	$F = 1$	$F = A$ 加 A
1	1	0	1	$F = A + \neg B$	$F = (A+B)$ 加 A
1	1	1	0	$F = A+B$	$F = (A + \neg B)$ 加 A
1	1	1	1	$F = A$	$F = A$ 减 1

4. 具体实验步骤：将“控制转换”开关拨到最中间位置既“独立”灯亮。

第一步：测试寄存器写入和读出；【操作模式：1100】

接线表和置开关如下表：

名称	电平控制信号开关	K6	K5	K4	K3	K2	K1	K0
功能	信号名称	RD1	RD0	DRW	SBUS	RS1	RS0	MBUS
置 R0	操作模式			1	1			
置 R1	写 REG 操作模式：1100		1	1	1			
置 R2		1		1	1			
置 R3		1	1	1	1			

备注：写寄存器完成后可以直接在写寄存器操作模式下，通过 K6、K5 拨动开关查看写入寄存器中的数据，对应的数据灯：A7~A0。通过 K2、K1 拨动开关也可以查看写入寄存器中的数据，对应的数据灯 B7~B0。

第二步：运算器实验【操作模式：1101】

接线表和置开关如下表：

名称	K15	K14	K13	K12	K11	K10	K9	K8
序号	M	S0	S1	S2	S3	CIN	ABUS	LDC
	运算器组成操作模式：1101							
	送两个数到 REG，K6K5、K2K1 分别选择加与被加							
		1			1	1	1	1

名称	K7	K6	K5	K4	K3	K2	K1	K0
序号	LDZ	RD1	RD0	DRW	SBUS	RS1	RS0	MBUS
				1	1			
			1	1	1			
	1							
备注：运算器实验答案只提供了加法运算的控制信号，其他运算功能请参考上页中 ALU 表的运算逻辑功能即可。								

2. 1. 3 实验报告记录:

2.2 双端口存储器实验

2.2.1 双端口存储器实验：微程序控制器方式

一、实验类别 原理性+分析性

二、实验目的

- (1)了解双端口静态存储器 IDT7132 的工作特性及其使用方法；
- (2)了解半导体存储器怎样存储和读取数据；
- (3)了解双端口存储器怎样并行读写；
- (4)熟悉 TEC-8 模型计算机中存储器部分的数据通路。

三、实验设备

序号	名称	数量	备注
1	实验系统	1 台	
2	双踪示波器	1 台	
3	直流万用表	1 块	
4	逻辑测试笔	1 支	在试验箱上方

四、实验电路

图 2.3 是双端口存储器实验的电路图。

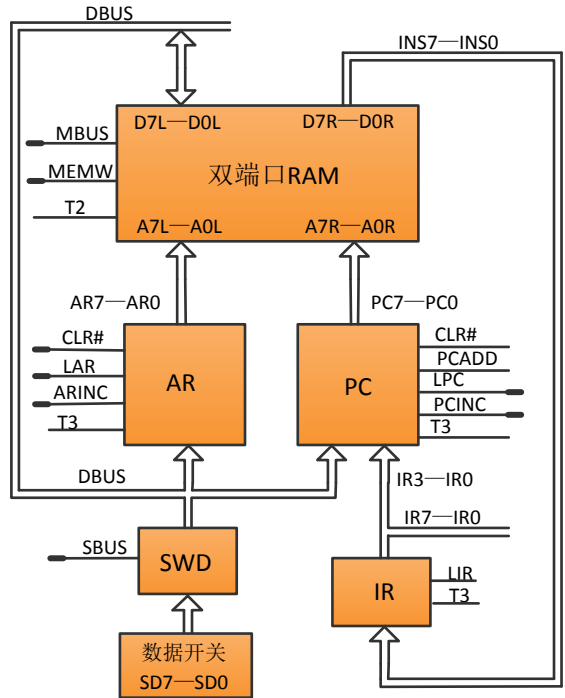


图 2.3 双端口存储器实验电路图

双端口 RAM 电路 由 1 片 IDT7132 及少许附加电路组成，存放程序和数据。IDT7132 有 2 个端口，一个称为左端口，一个称为右端口。2 个端口各有独立的存储器地址线、数据线和 3 个读、写控制信号：CE#、R/W#和 OE#，可以同时对器件内部的同一存储体同时进行读、写。IDT7132 容量为 2048 字节，TEC-8 实验系统只使用 64 字节。

在 TEC-8 实验系统中，左端口配置成读、写端口，用于程序的初始装入操作，从存储器中取数到数据总线 DBUS，将数据总线 DBUS 上的数写入存储器。当信号 MEMW 为 1 时，在 T2

为 1 时，将数据总线 DBUS 上的数 D7~D0 写入 AR7~AR0 指定的存储单元；当 MBUS 信号为 1 时，AR7~AR0 指定的存储单元的数送数据总线 DBUS。右端口设置成只读方式，从 PC7~PC0 指定的存储单元读出指令 INS7~INS0，送往指令寄存器 IR。

程序计数器 PC 由 2 片 GAL22V10(U53 和 U54)组成。向双端口 RAM 的右端口提供存储器地址。当复位信号 CLR#为 0 时，程序计数器复位，PC7~PC0 为 00H。当信号 LPC 为 1 时，在 T3 的上升沿，将数据总线 DBUS 上的数 D7~D0 写入 PC。当信号 PCINC 为 1 时，在 T3 的上升沿，完成 PC 加 1。当 PCADD 信号为 1 时，PC 和 IR 中的转移偏量(IR3~IR0)相加，在 T3 的上升沿，将相加得到的和写入 PC 程序计数器。

地址寄存器 AR 由 1 片 GAL22V10(U58)组成，向双端口 RAM 的左端口提供存储器地址 AR7~AR0。当复位信号 CLR#为 0 时，地址寄存器复位，AR7~AR0 为 00H。当信号 LAR 为 1 时，在 T3 的上升沿，将数据总线 DBUS 上的数 D7~D0 写入 AR。当信号 ARINC 为 1 时，在 T3 的上升沿，完成 AR 加 1。

指令寄存器 IR 是 1 片 74LS273(U47)，用于保存指令。当信号 LIR 为 1 时，在 T3 的上升沿，将从双端口 RAM 右端口读出的指令 INS7~INS0 写入指令寄存器 IR。

数据开关 SD7~SD0 用于设置双端口 RAM 的地址和数据。当信号 SBUS 为 1 时，数 SD7~SD0 送往数据总线 DBUS。本实验中用到的信号归纳如下：

名 称	功 能 说 明
D7~D0	数据总线 DBUS 上的数。
PCINC	=1 时，在 T3 的上升沿 PC 加 1。
AR7~AR0	双端口 RAM 左端口存储器地址。
ARINC	=1 时，在 T3 的上升沿，AR 加 1。
PC7~PC0	双端口 RAM 右端口存储器地址。
SBUS	=1 时，数据开关 SD7~SD0 的数送数据总线 DBUS。
MBUS	=1 时，将双端口 RAM 的左端口数据送到数据总线 DBUS。
INS7~INS0	从双端口 RAM 右端口读出的指令，本实验中作为数据使用。
LAR	=1 时，在 T3 的上升沿，将数据总线 DBUS 上的 D7~D0 写入地址寄存器 AR。
LPC	当它为 1 时，在 T3 的上升沿，将数据总线 DBUS 上的 D7~D0 写入程序计数器 PC。
MEMW	=1 时，在 T2 为 1 期间将数据总线 DBUS 上的 D7~D0 写入双端口 RAM 写入的存储器单元由 AR7~AR0 指定。
LIR	=1 时，在 T3 的上升沿将从双端口 RAM 的右端口读出的指令 INS7~INS0 写入指令寄存器 IR。读出的存储器单元由 PC7~PC0 指定。

上述信号都有对应的指示灯。当指示灯灯亮时，表示对应的信号为 1；当指示灯不亮时，对应的信号为 0。实验过程中，对每一个实验步骤，都要记录上述信号(可以不纪录 SETCTL)的值。另外 $\mu A5 \sim \mu A0$ 指示灯指示当前微地址。

五、实验任务

1. 从存储器地址 10H 开始，通过左端口连续向双端口 RAM 中写入 3 个数：85H，60H，38H。在写的过程中，在右端口检测写的数据是否正确。
2. 从存储器地址 10H 开始，连续从双端口 RAM 的左端口和右端口同时读出存储器的内容。

六、实验步骤

1. 实验准备

将控制器转换开关拨到微程序位置，将编程开关设置为正常位置。打开电源。

2. 进行存储器读、写实验

(1)设置存储器读、写实验模式

按复位按钮 CLR，使 TEC-8 实验系统复位。指示灯 $\mu A5 \sim \mu A0$ 显示 00H。将操作模式开关设置为 SWC=1、SWB=1、SWA=0，准备进入双端口存储器实验。

按一次 QD 按钮，进入存储器读、写实验。

(2)设置存储器地址

指示灯 $\mu A5 \sim \mu A0$ 显示 0DH。在数据开关 SD7~SD0 上设置地址 10H。在数据总线 DBUS 指示灯 D7~D0 上可以看到地址设置的正确不正确，发现错误需及时改正。设置地址正确后，按一次 QD 按钮，将 SD7~SD0 上的地址写入地址寄存器 AR(左端口存储器地址)和程序计数器 PC(右端口存储器地址)，进入下一步。

(3)写入第 1 个数

指示灯 $\mu A5 \sim \mu A0$ 显示 1AH。指示灯 AR7~AR0(左端口地址)显示 10H，指示灯 PC7~PC0(右端口地址)显示 10H。在数据开关 SD7~SD0 上设置写入存储器的第 1 个数 85H。按一次 QD 按钮，将数 85H 通过左端口写入由 AR7~AR0 指定的存储器单元 10H。

(4)写入第 2 个数

指示灯 $\mu A5 \sim \mu A0$ 显示 1BH。指示灯 AR7~AR0(左端口地址)显示 11H，指示灯 PC7~PC0(右端口地址)显示 10H。观测指示灯 INS7~INS0 的值，它是通过右端口读出的由右地址 PC7~PC0 指定的存储器单元 10H 的值。比较和通过左端口写入的数是否相同。在数据开关 SD7~SD0 上设置写入存储器的第 2 个数 60H。按一次 QD 按钮，将第 2 个数通过左端口写入由 AR7~AR0 指定的存储器单元 11H。

(5)写入第 3 个数

指示灯 $\mu A5 \sim \mu A0$ 显示 1CH。指示灯 AR7~AR0(左端口地址)显示 12H，指示灯 PC7~PC0(右端口地址)显示 11H。观测指示灯 INS7~INS0 的值，它是通过右端口读出的由右地址 PC7~PC0 指定的存储器单元 11H 的值。比较和通过左端口写入的数是否相同。在数据开关 SD7~SD0 上设置写入存储器的第 3 个数 38H。按一次 QD 按钮，将第 3 个数通过左端口写入由 AR7~AR0 指定的存储器单元 12H。

(6)重新设置存储器地址

指示灯 $\mu A5 \sim \mu A0$ 显示 1DH。指示灯 AR7~AR0(左端口地址)显示 13H，指示灯 PC7~PC0(右端口地址)显示 12H。观测指示灯 INS7~INS0 的值，它是通过右端口读出的由右地址 PC7~PC0 指定的存储器单元 12H 的值。比较和通过左端口写入的数是否相同。在数据开关 SD7~SD0 重新设置存储器地址 10H。按一次 QD 按钮，将 SD7~SD0 上的地址写入地址寄存器 AR(左端口存储器地址)和程序计数器 PC(右端口存储器地址)，进入下一步。

(7)左、右两 2 个端口同时显示同一个存储器单元的内容。

指示灯 $\mu A5 \sim \mu A0$ 显示 1FH。指示灯 AR7~AR0(左端口地址)显示 10H，指示灯 PC7~PC0(右端口地址)显示 10H。观测指示灯 INS7~INS0 的值，它是通过右端口读出的由右地址 PC7~PC0 指定的存储器单元 10H 的值。观测指示灯 D7~D0 的值，它是从左端口读出的由 AR7~AR0 指定的存储器单元 10H 的值。

按一次 QD 按钮，地址寄存器 AR 加 1，程序计数器 PC 加 1，在指示灯 D7~D0 和指示灯 INS7~INS0 上观测存储器的内容。继续按 QD 按钮，直到存储器地址 AR7~AR0 为 12H 为止。

七、实验要求

1. 做好实验预习，掌握双端口存储器的使用方法和 TEC-8 模型计算机存储器部分的数据通路。

2. 写出实验报告，内容是：

(1)实验目的。

(2)根据实验结果填写表 2.1。

表 2.1 双端口存储器实验结果表

实验数据		实验结果					
左端口存储器地址	通过左端口写入的数据	第一次从右端口读出的数		同时读出时的读出结果			
		右端口存储器地址	读出的数	左端口存储器地址	读出的数	右端口存储器地址	读出的数

(3)结合实验现象，在每一实验步骤中，对下述信号所起的作用进行解释：SBUS、MBUS、LPC、PCINC、LAR、ARINC、MEMW。并说明在该步骤中，哪些信号是必需的，哪些信号不是必需的，哪些信号必需采用实验中使用的值，哪些信号可以不采用实验中使用的值。

八、可研究和探索的问题

1. 在通过左端口向双端口 RAM 写数时，在右端口可以同时观测到左端口写入的数吗？为什么？

2.2.2 双端口存储器实验：独立方式

一、实验类别 原理性+分析性+操作性

二、实验目的

- (1)了解双端口静态存储器 IDT7132 的工作特性及其使用方法；
- (2)了解半导体存储器怎样存储和读取数据；
- (3)了解双端口存储器怎样并行读写；
- (4)熟悉 TEC-8 模型计算机中存储器部分的数据通路。

三、实验设备

序号	名称	数量	备注
1	实验系统	1 台	
2	双踪示波器	1 台	
3	直流万用表	1 块	
4	逻辑测试笔	1 支	在试验箱上方

四、实验电路

图 2.4 是双端口存储器实验的电路图。

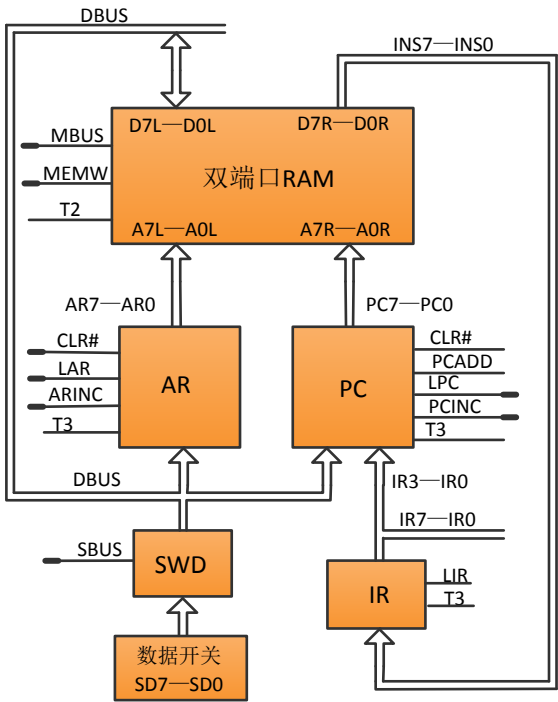


图 2.4 双端口存储器实验电路图

双端口 RAM 电路 由 1 片 IDT7132 及少许附加电路组成，存放程序和数据。IDT7132 有 2 个端口，一个称为左端口，一个称为右端口。2 个端口各有独立的存储器地址线、数据线和 3 个读、写控制信号：CE#、R/W#和 OE#，可以同时为器件内部的同一存储体同时进行读、写。IDT7132 容量为 2048 字节，TEC-8 实验系统只使用 64 字节。

在 TEC-8 实验系统中，左端口配置成读、写端口，用于程序的初始装入操作，从存储器中取数到数据总线 DBUS，将数据总线 DBUS 上的数写入存储器。当信号 MEMW 为 1 时，在 T2 为 1 时，将数据总线 DBUS 上的数 D7~D0 写入 AR7~AR0 指定的存储单元；当 MBUS 信号为 1 时，AR7~AR0 指定的存储单元的数送数据总线 DBUS。右端口设置成只读方式，从 PC7~PC0 指定的存储单元读出指令 INS7~INS0，送往指令寄存器 IR。

程序计数器 PC 由 2 片 GAL22V10 (U53 和 U54) 组成。向双端口 RAM 的右端口提供存储器地址。当复位信号 CLR# 为 0 时，程序计数器复位，PC7~PC0 为 00H。当信号 LPC 为 1 时，在 T3 的上升沿，将数据总线 DBUS 上的数 D7~D0 写入 PC。当信号 PCINC 为 1 时，在 T3 的上升沿，完成 PC 加 1。当 PCADD 信号为 1 时，PC 和 IR 中的转移偏量 (IR3~IR0) 相加，在 T3 的上升沿，将相加得到的和写入 PC 程序计数器。

地址寄存器 AR 由 1 片 GAL22V10 (U58) 组成，向双端口 RAM 的左端口提供存储器地址 AR7~AR0。当复位信号 CLR# 为 0 时，地址寄存器复位，AR7~AR0 为 00H。当信号 LAR 为 1 时，在 T3 的上升沿，将数据总线 DBUS 上的数 D7~D0 写入 AR。当信号 ARINC 为 1 时，在 T3 的上升沿，完成 AR 加 1。

指令寄存器 IR 是 1 片 74LS273 (U47)，用于保存指令。当信号 LIR 为 1 时，在 T3 的上升沿，将从双端口 RAM 右端口读出的指令 INS7~INS0 写入指令寄存器 IR。

数据开关 SD7~SD0 用于设置双端口 RAM 的地址和数据。当信号 SBUS 为 1 时，数 SD7~SD0 送往数据总线 DBUS。本实验中用到的信号归纳如下：

名 称	功 能 说 明
D7~D0	数据总线 DBUS 上的数。
PCINC	=1 时，在 T3 的上升沿 PC 加 1。
AR7~AR0	双端口 RAM 左端口存储器地址。
ARINC	=1 时，在 T3 的上升沿，AR 加 1。
PC7~PC0	双端口 RAM 右端口存储器地址。
SBUS	=1 时，数据开关 SD7~SD0 的数送数据总线 DBUS。
MBUS	=1 时，将双端口 RAM 的左端口数据送到数据总线 DBUS。
INS7~INS0	从双端口 RAM 右端口读出的指令，本实验中作为数据使用。
LAR	=1 时，在 T3 的上升沿，将数据总线 DBUS 上的 D7~D0 写入地址寄存器 AR。
LPC	当它为 1 时，在 T3 的上升沿，将数据总线 DBUS 上的 D7~D0 写入程序计数器 PC。
MEMW	=1 时，在 T2 为 1 期间将数据总线 DBUS 上的 D7~D0 写入双端口 RAM 写入的存储器单元由 AR7~AR0 指定。
LIR	=1 时，在 T3 的上升沿将从双端口 RAM 的右端口读出的指令 INS7~INS0 写入指令寄存器 IR。读出的存储器单元由 PC7~PC0 指定。

上述信号都有对应的指示灯。当指示灯灯亮时，表示对应的信号为 1；当指示灯不亮时，对应的信号为 0。实验过程中，对每一个实验步骤，都要记录上述信号 (可以不纪录 SETCTL) 的值。另外 $\mu A5 \sim \mu A0$ 指示灯指示当前微地址。

五、实验任务

1. 从存储器地址 10H 开始，通过左端口连续向双端口 RAM 中写入 3 个数：85H，60H，38H。在写的过程中，在右端口检测写的数据是否正确。
2. 从存储器地址 10H 开始，连续从双端口 RAM 的左端口和右端口同时读出存储器的内容。

六、实验步骤及实验结果

第 1 步：将“控制转换”开关拨到中间位置既“独立”灯亮，双端口存储器实验；【操作模式：1110】，拨动编程开关到正常位置。

第 2 步：开关与控制信号之间的接线图：

K15	K14	K13	K12	K11	K10	K9
SBUS	ARINC	LAR	MEMW	MBUS	PCINC	LPC

第 3 步：实验的具体步骤和信号：

名称	操作	K15	K14	K13	K12	K11	K10	K9
序号	模式	SBUS	ARINC	LAR	MEMW	MBUS	PCINC	LPC
1	1110	1		1				1
2		1	1		1			
3		1	1		1			
4		1	1		1			
5		1		1				1
6			1		1		1	
7			1		1		1	
8			1		1		1	

名称	K8	SD	实验现象	备注
序号	ABUS		连线对应位置	
1	0	10	置 AR、PC/ AR=PC=10	
2		85	写第 1 个数 85/AR=11,PC=10,INS= IR =85	
3		60	写第 2 个数 60/AR=12,PC=10,INS= IR =60	
4		38	写第 3 个数 38/AR=13,PC=10,INS= IR =38	
5		10	重置 AR、PC/ AR=PC=10	
6			AR=PC=10,INS=85	
7			AR=PC=11,INS=60	
8			AR=PC=12,INS=38	

2.2.3 实验报告记录:

2.3 数据通路实验

2.3.1 数据通路实验：微程序控制器方式

一、实验类型 原理性+分析性

二、实验目的

- (1)进一步熟悉 TEC-8 模型计算机的数据通路的结构；
- (2)进一步掌握数据通路中各个控制信号的作用和用法；
- (3)掌握数据通路中数据流动的路径。

三、实验设备

序列号	名 称	数 量	备 注
1	实验系统	1 台	
2	双踪示波器	1 台	
3	直流万用表	1 块	
4	逻辑测试笔	1 支	在试验箱上

四、实验原理

数据通路实验电路图如图 2.4 所示。它由运算器部分、双端口存储器部分加上数据开关 SD7~SD0 连接在一起构成。

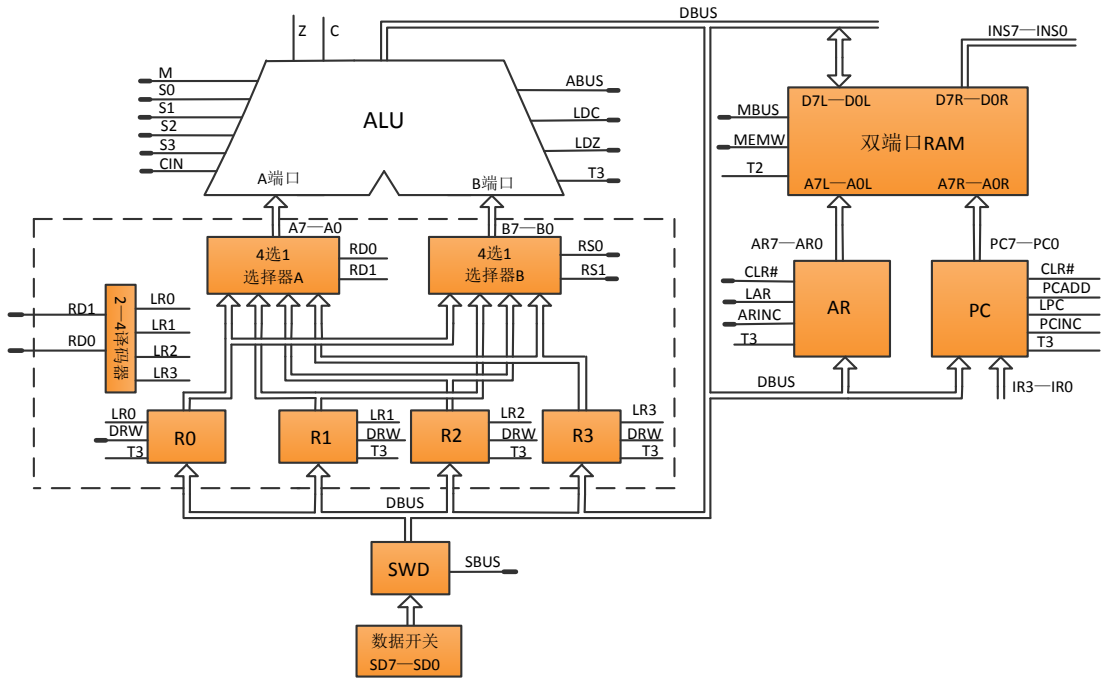


图 2.4 数据通路实验电路图

数据通路中各个部分的作用和工作原理在 2.1 节和 2.2 节已经做过详细说明，不再重

述。这里主要说明 TEC-8 模型计算机的数据流动路径和方式。

在进行数据运算操作时，由 RD1、RD0 选中的寄存器通过 4 选 1 选择器 A 送往 ALU 的 A 端口，由 RS1、RS0 选中的寄存器通过 4 选 1 选择器 B 送往 ALU 的 B 端口；信号 M、S3、S2、S1、S1 和 S0 决定 ALU 的运算类型，ALU 对 A 端口和 B 端口的两个数连同 CIN 的值进行算数逻辑运算，得到的数据运算结果在信号 ABUS 为 1 时送往数据总线 DBUS；在 T3 的上升沿，数据总线 DBUS 上的数据结果写入由 RD1、RD0 选中的寄存器。

在寄存器之间进行数据传送操作时，由 RS1、RS0 选中的寄存器通过 4 选 1 选择器 B 送往 ALU 的 B 端口；ALU 将 B 端口的数在信号 ABUS 为 1 时送往数据总线 DBUS；在 T3 的上升沿将数据总线上的数写入由 RD1、RD0 选中的寄存器。ALU 进行数据传送操作由一组特定的 M、S3、S2、S1、S0、CIN 的值确定。

在进行运算操作时，由 RS1、RS0 选中的寄存器通过 4 选 1 选择器 B 送往 ALU 的 B 端口；由 RD1、RD0 选中的寄存器通过 4 选 1 选择器 A 送往 ALU 的 A 端口；ALU 对数 A 和 B 进行运算，运算的数据结果在信号 ABUS 为 1 时送往数据总线 DBUS；在 T3 的上升沿将数据总线上的数写入由 RD1、RD0 选中的寄存器。ALU 进行何种运算操作由 M、S3、S2、S1、S0、CIN 的值确定。

在从存储器中取数操作中，由地址 AR7~AR0 指定的存储器单元中的数在信号 MEMW 为 0 时被读出；在 MBUS 为 1 时送数据总线 DBUS；在 T3 的上升沿写入由 RD1、RD0 选中的寄存器。

在写存储器操作中，由 RS1、RS0 选中的寄存器过 4 选 1 选择器 B 送 ALU 的 B 端口；ALU 将 B 端口的数在信号 ABUS 为 1 时送往数据总线 DBUS；在 MEMW 为 1 且 MBUS 为 0 时，通过左端口将数据总线 DBUS 上的数在 T2 为 1 期间写入由 AR7~AR0 指定的存储器单元。

在读指令操作时，通过存储器右端口读出由 PC7~PC0 指定的存储器单元的内容送 INST7~INST0，当信号 LIR 为 1 时，在 T3 的上升沿写入指令寄存器 IR。

数据开关 SD7~SD0 上的数在 SBUS 为 1 时送到数据总线 DBUS 上，用于给寄存器 R0、R1、R2 和 R3，地址寄存器 AR，程序计数器 PC 设置初值，用于通过存储器左端口向存储器写入测试程序。

数据通路实验中涉及到的信号如下：

M、S3、S2、S1、S0	控制 74LS181 的算术逻辑运算类型。
CIN	低位 74LS181 的进位输入。
SEL3 (RD1)	选择送 ALU 的 A 端口的寄存器和被写入的寄存器。
SEL2 (RD0)	
SEL1 (RS1)	选择送往 ALU 的 B 端口的寄存器。
SEL0 (RS0)	
DRW	=1 时，在 T3 上升沿对 RD1、RD0 选中的寄存器进行写操作，将数据总线 DBUS 上的数 D7~D0 写入选定的寄存器。
ABUS	=1 时，将运算结果送数据总线 DBUS， =0 时，禁止运算结果送数据总线 DBUS。
SBUS	=1 时，将运算结果送数据总线 DBUS， =0 时，禁止运算结果送数据总线 DBUS。
A7~A0	送往 ALU 的 A 端口的数。
B7~B0	送往 ALU 的 B 端口的数。
D7~D0	数据总线 DBUS 上的 8 位数。
MBUS	=1 时，将双端口 RAM 的左端口数据送到数据总线 DBUS。
MEMW	=1 时，在 T2 为 1 期间将数据总线 DBUS 上的 D7~D0 写入双端口 RAM，写入的存储器单元由 AR7~AR0 指定。
LPC	=1 时，在 T3 的上升沿，将数据总线 DBUS 上的 D7~D0 写入程序计数器 PC。
PCINC	=1 时，在 T3 的上升沿 PC 加 1。
LAR	=1 时，在 T3 的上升沿，将数据总线 DBUS 上的 D7~D0 写入地址寄存器 AR。
ARINC	=1 时，在 T3 的上升沿，AR 加 1。
SBUS	=1 时，数据开关 SD7~SD0 的数送数据总线 DBUS。
AR7~AR0	双端口 RAM 左端口存储器地址。
PC7~PC0	双端口 RAM 右端口存储器地址。
INS7~INS0	从双端口 RAM 右端口读出的指令，本实验中作为数据使用。
SETCTL	=1 时，实验系统处于实验台状态。 =0 时，实验系统处于运行程序状态。

上述信号都有对应的指示灯。当指示灯灯亮时，表示对应的信号为 1；当指示灯不亮时，对应的信号为 0。实验过程中，对每一个实验步骤，都要记录上述信号的值。另外 $\mu A5 \sim \mu A0$ 指示灯指示当前微地址。

五、实验任务

1. 将数 75H 写到寄存器 R0，数 28H 写到寄存器 R1，数 89H 写到寄存器 R2，数 32H 写到寄存器 R3。
2. 将寄存器 R0 中的数写入存储器 20H 单元，将寄存器 R1 中的数写入存储器 21H 单元，将寄存器 R2 中的数写入存储器 22H 单元，将寄存器 R3 中的数写入存储器 23H 单元。
3. 从存储器 20H 单元读出数到寄存器 R3，从存储器 21H 单元读出数到寄存器 R2，从存储器 21H 单元读出数到寄存器 R1，从存储器 23H 单元读出数到寄存器 R0。
4. 显示 4 个寄存器 R0、R1、R2、R3 的值，检查数据传送是否正确。

六、实验步骤

1. 实验准备

将控制器转换开关拨到微程序位置，将编程开关设置为正常位置。打开电源。

2. 进行数据通路实验

(1) 设置数据通路实验模式

首先将“控制转换”开关拨到最下方位置既“微程序”灯亮。按复位按钮 CLR，使 TEC-8 实验系统复位。指示灯 $\mu A5 \sim \mu A0$ 显示 00H。将操作模式开关设置为 SWC=1、SWB=1、SWA=1，准备进入数据通路实验。

按一次 QD 按钮，进入数据通路实验。

(2) 将数 75H 写到寄存器 R0、数 28H 写到 R1、数 89H 写到 R2、数 32H 写到 R3。

指示灯 $\mu A5 \sim \mu A0$ 显示 0FH。在数据开关 SD7~SD0 上设置数 75H。在数据总线 DBUS 指示灯 D7~D0 上可以看到数设置得正确不正确，发现错误需及时改正。数设置正确后，按一次 QD 按钮，将 SD7~SD0 上的数写入寄存器 R0，进入下一步。

依照写 R0 的方式，在指示灯 $\mu A5 \sim \mu A0$ 显示 32H 时，在指示灯 B7~B0 观测寄存器 R0 的值，将数 28H 写入 R1；在指示灯 $\mu A5 \sim \mu A0$ 显示 33H 时，在指示灯 B7~B0 上观测 R1 的值，将数 89H 写入 R2；在指示灯 $\mu A5 \sim \mu A0$ 显示 34H 时，在指示灯 B7~B0 上观测 R2 的值，将数 32H 写入 R3。

(3) 设置存储器地址 AR 和程序计数器 PC

指示灯 $\mu A5 \sim \mu A0$ 显示 35H。此时指示灯 B7~B0 显示寄存器 R3 的值。在数据开关 SD7~SD0 上设置地址 20H。在数据总线 DBUS 指示灯 D7~D0 上可以看到地址设置得正确不正确。地址设置正确后，按一次 QD 按钮，将 SD7~SD0 上的地址写入地址寄存器 AR 和程序计数器 PC，进入下一步。

(4) 将寄存器 R0、R1、R2、R3 中的数依次写入存储器 20H、21H、22H 和 23H 单元。

指示灯 $\mu A5 \sim \mu A0$ 显示 36H。此时指示灯 AR7~AR0 和 PC7~PC0 分别显示出存储器左、右两个端口的存储器地址。指示灯 A7~A0、B7~B0 和 D7~D0 都显示寄存器 R0 的值。按一次 QD 按钮，将 R0 中的数写入存储器 20H 单元，进入下一步。

依照此法，在指示灯 $\mu A5 \sim \mu A0$ 显示 37H 时，在 INS7~INS0 上观测存储器 20H 单元的值，将 R1 中的数写入存储器 21H 单元；在指示灯 $\mu A5 \sim \mu A0$ 显示 38H 时，在 INS7~INS0 上观测存储器 21H 单元的值，将 R2 中的数写入存储器 22H 单元；在指示灯 $\mu A5 \sim \mu A0$ 显示 39H 时，在 INS7~INS0 上观测存储器 22H 单元的值，将 R3 中的数写入存储器 23H 单元。

(5) 重新设置存储器地址 AR 和程序计数器 PC

指示灯 $\mu A5 \sim \mu A0$ 显示 3AH。此时指示灯 PC7~PC0 显示 23H，INS7~INS0 显示存储器 23H 单元中的数。在数据开关 SD7~SD0 上设置地址 20H。按一次 QD 按钮，将地址 20H 写入地址寄存器 AR 和程序计数器 PC，进入下一步。

(6) 将存储器 20H、21H、22H 和 23H 单元中的数依次写入寄存器 R3、R2、R1 和 R0。

指示灯 $\mu A5 \sim \mu A0$ 显示 3BH。此时指示灯 AR7~AR0 和 PC7~PC0 显示 20H，指示灯 D7~D0 和 INS7~INS0 同时显示存储器 20H 中的数，按一次 QD 按钮，将存储器 20H 单元中的数写入寄存器 R3，进入下一步。

依照此法，在指示灯 $\mu A5 \sim \mu A0$ 显示 3CH 时，在指示灯 B7~B0 上观测 R3 的值，将存储器 21H 单元中的数写入寄存器 R2；在指示灯 $\mu A5 \sim \mu A0$ 显示 3DH 时，在指示灯 B7~B0 上观测 R2 的值，将存储器 22H 单元中的数写入寄存器 R1；在指示灯 $\mu A5 \sim \mu A0$ 显示 3EH 时，在指示灯 B7~B0 上观测 R1 的值，将存储器 23H 单元中的数写入寄存器 R0。

(7) 观测 R0 的值

指示灯 $\mu A5 \sim \mu A0$ 显示 00H。此时指示灯 A7~A0 显示 R0 的值，指示灯 B7~B0 显示 R3 的值。

七、实验要求

1. 做好实验预习，掌握 TEC-8 模型计算机的数据通路及各种操作情况下的数据流动路径和流动方向。
2. 写出实验报告，内容是：
 - (1)实验目的。
 - (2)根据实验结果填写表 2.3。

表 2.3 数据通路实验结果表

$\mu A5 \sim \mu A0$	A7~A0	B7~B0	D7~D0	AR	PC	INS7	R0	R1	R2	R3
0FH										
32H										
33H										
34H										
35H										
36H										
37H										
38H										
39H										
3AH										
3BH										
3CH										
3DH										
3EH										
00H										

(3)结合实验现象，在每一实验步骤中，对下述信号所起的作用进行解释：SBUS、MBUS、LPC、PCINC、LAR、ARINC、MEMW、M、S0、S1、S2、S3、CIN、ABUS、SEL3、SEL2、SEL1、SEL0、DRW、SBUS。并说明在该步骤中，哪些信号是必需的，哪些信号不是必需的，哪些信号必需采用实验中使用的值，哪些信号可以不采用实验中使用的值。

(4)写出下列操作时，数据的流动路径和流动方向：给寄存器置初值、设置存储器地址、将寄存器中的数写到存储器中，从存储器中读数到寄存器。

八、可探索和研究的问题

1. 如果用 I-cache 和 D-cache 来代替双端口存储器，请提出一种数据通路方案。

2.3.2 实验报告记录:

2.3.3 数据通路实验：独立方式：

首先将“控制转换”开关拨到最中间位置既“独立”灯亮。【操作模式：1111】

名称	K15	K14	K13	K12	K11	K10	K9	K8	备 注
信号名	SBUS	DRW	RD1	RD0	RS1	RS0	MBUS	M	
序列号									
01	1	1							
02	1	1		1					
03	1	1	1						
04	1	1	1	1					
05	1								
06								1	
07						1		1	
08					1			1	
09					1	1		1	
10	1				1	1			
11		1	1	1			1		
12		1	1		1	1	1		
13		1		1	1		1		
14		1				1	1		
15									

名称	K7	K6	K5	K4	K3	K2	K1	K0	SD	备注
信号名	S3	S1	ABUS	MEMW	LAR	ARINC	LPC	PCINC		
序列号										
01									75	
02									28	
03									89	
04									32	
05					1		1		20	
06	1	1	1	1		1				
07	1	1	1	1		1		1		
08	1	1	1	1		1		1		
09			1	1		1		1		
10					1					
11						1				
12						1				
13						1				
14						1				
15										

This image shows a completely blank white page. It is surrounded by a thin black rectangular border, which appears to be the edge of a scanner or a frame. There are no markings, text, or illustrations on the page itself.

2.4 微程序控制器实验

2.4.1 微程序控制器实验部分

一、实验类型 原理性+设计性+分析性

二、实验目的

- (1)掌握微程序控制器的原理
- (2)掌握 TEC-8 模型计算机中微程序控制器的实现方法，尤其是微地址转移逻辑的实现方法。
- (3)理解条件转移对计算机的重要性。

三、实验设备

序列号	名 称	数 量	备 注
1	实验系统	1 台	
2	双踪示波器	1 台	
3	直流万用表	1 块	
4	逻辑测试笔	1 支	在试验箱上

四、实验电路

微程序控制器与硬连线控制器相比，由于其规整性、易于设计以及需要的时序发生器相对简单，在上世纪七、八十年代得到广泛应用。本实验通过一个具体微程序控制器的实现使学生从实践上掌握微程序控制器的一般实现方法，理解控制器在计算机中的作用。

1. 微指令格式

根据机器指令功能、格式和数据通路所需的控制信号，TEC-8 采用如图 2.5 所示的微指令格式。微指令字长 40 位，顺序字段 11 位(判别字段 P4~P0，后继微地址 NμA5~NμA0)，控制字段 29 位，微命令直接控制。

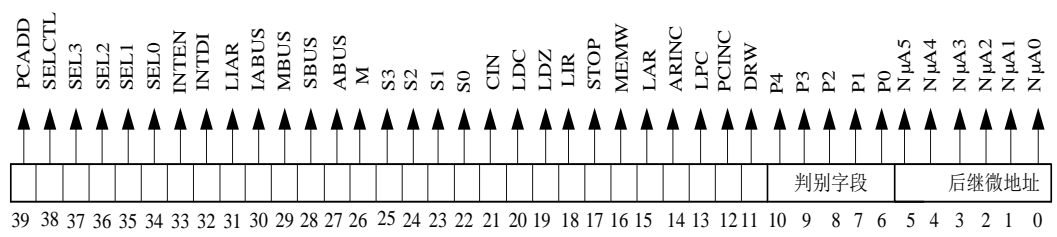


图 2.5 微指令格式

前面的 3 个实验已经介绍了主要的微命令(控制信号)，介绍过的微命令不再重述，这里介绍后继微地址、判别字段和其它的微命令。

N μ A5~N μ A0	下址，在微指令顺序执行的情况下，它是下一条微指令的地址
P0	=1 时，根据后继微地址 N μ A5~N μ A0 和模式开关 SWC、SWB、SWA 确定下一条微指令的地址。见图 2.6 微程序流程图
P1	=1 时，根据后继微地址 N μ A5~N μ A0 和指令操作码 IR7~IR4 确定下一条微指令的地址。见图 2.6 微程序流程图。
P2	=1 时，根据后继微地址 N μ A5~N μ A0 和进位 C 确定下一条微指令的地址。见图 2.6 微程序流程图。
P3	=1 时，根据后继微地址 N μ A5~N μ A0 和结果为 0 标志 Z 确定下一条微指令的地址。见图 2.6 微程序流程图。
P4	=1 时，根据后继微地址 N μ A5~N μ A0 和中断信号 INT 确定下一条微指令的地址。模型计算机中，中断信号 INT 由时序发生器在接到中断请求信号后产生。
STOP	=1 时，在 T3 结束后时序发生器停止输出节拍脉冲 T1、T2、T3。
LIAR	=1 时，在 T3 的上升沿，将 PC7~PC0 写入中断地址寄存器 IAR。
INTDI	=1 时，置允许中断标志(在时序发生器中)为 0，禁止 TEC-8 模型计算机响应中断请求
INTEN	=1 时，置允许中断标志(在时序发生器中)为 1，允许 TEC-8 模型计算机响应中断请求
IABUS	=1 时，将中断地址寄存器中的地址送数据总线 DBUS。
PCADD	=1 时，将当前的 PC 值加上相对转移量，生成新的 PC。

由于 TEC-8 模型计算机有微程序控制器和硬连线控制器 2 个控制器，因此微程序控制器产生的控制信号以前缀“A-”标示，以便和硬连线控制器产生的控制信号区分。硬连线控制器产生的控制信号以前缀“B-”标示。

2. 微程序流程图

根据指令系统和控制台功能和数据通路，TEC-8 模型计算机的微程序流程图如图 2.6 所示。图 2.6 中，为了简洁，将许多以“A-”为前缀的信号，省略了前缀。

需要说明的是，图 2.6 中没有包括运算器组成实验、双端口存储器实验和数据通路三部分。这三部分的微程序很简单，微程序都是顺序执行的，根据这三个实验很容易画出。

1. 微程序控制器电路

根据 TEC-8 模型计算机的指令系统、控制台功能、微指令格式和微程序流程图，TEC-8 模型计算机微程序控制器电路如图 2.7 所示。

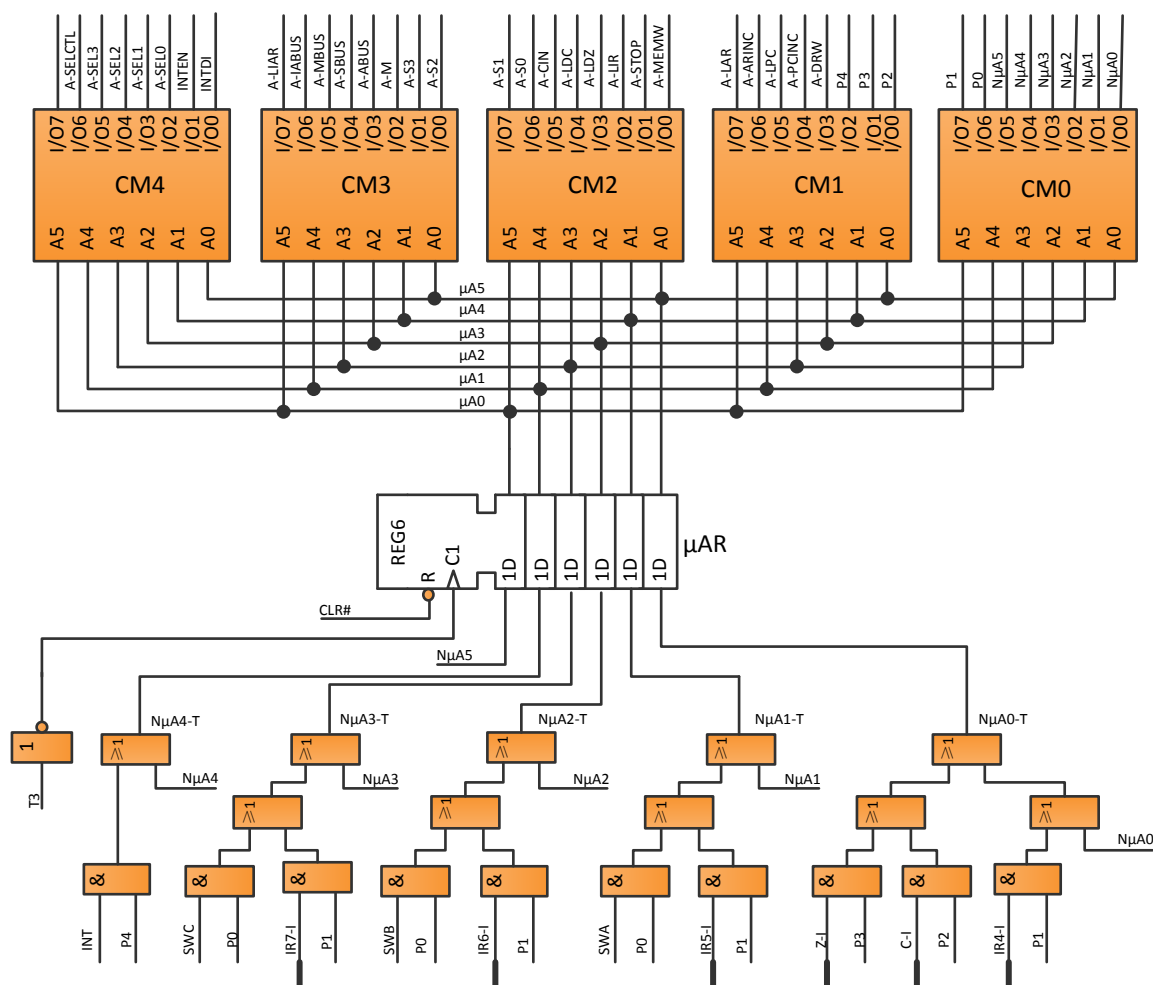


图 2.7 微程序控制器电路图

图 2.7 中，以短粗线标志的信号都有接线孔。信号 IR4-I、IR5-I、IR6-I、IR7-I、C-I 和 Z-I 的实际意义分别等同于 IR4、IR5、IR6、IR7、C 和 Z。INT 信号是时序发生器接收到中断请求脉冲 PULSE(高电平有效)后产生的中断信号。

(1) 控制存储器

控制存储器由 5 片 58C65 组成，在图 2.6 中表示为 CM4~CM0。其中 CM0 存储微指令最低的 8 位微代码，CM4 存储微指令最高的 8 位微代码。控制存储器的微代码必须与微指令格式一致。58C65 是一种 $8K \times 8$ 位的 E^2 PROM 器件，地址位为 A12~A0。由于 TEC-8 模型计算机只使用其中 64 个字节作为控制存储器，因此将 A12~A6 接地，A5~A0 接微地址 $\mu A5 \sim \mu A0$ 。在正常工作方式下，5 片 E^2 PROM 处于只读状态；在修改控制存储器内容时，5 片 E^2 PROM 处于读、写状态。

(2) 微地址寄存器

微地址寄存器 μAR 由 1 片 74LS174LS 组成，74LS174 是一个 6D 触发器。当按下复位按钮 CLR 时，产生的信号 CLR#(负脉冲)使微地址寄存器复位， $\mu A5 \sim \mu A0$ 为 00H，供读出第一条微指令使用。在一条微指令结束时，用 T3 的下降沿将微地址转移逻辑产生的下条微指令地址 $N\mu A5$ 、 $N\mu A4-T \sim N\mu A0-T$ 写入微地址寄存器。

(3)微地址转移逻辑

微地址转移逻辑由若干与门和或门组成,实现“与~或”逻辑。深入理解微地址转移逻辑,对于理解计算机的本质有很重要的作用。计算机现在的功能很强大,但是它是建立在两个很重要的基础之上,一个是最基本的加法和减法功能,一个是条件转移功能。设想一下,如果没有条件转移指令,实现 10000 个数相加,至少需要 20000 条指令,还不如用算盘计算速度快。可是有了条件转移指令后,一万个数相加,不超过 20 条指令就能实现。因此可以说,最基本的加法和减法功能和条件转移功能给计算机后来的强大功能打下了基础。本实验中微地址转移逻辑的实现方法是一个很简单的例子,但对于理解条件转移的实现方法大有益处。

下面分析根据后继微地址 $N\mu A5 \sim N\mu A0$ 、判别位 $P1$ 和指令操作码如何实现微程序分枝的。

微地址 $N\mu A5 \sim N\mu A0$ 中的微指令是一条功能为取指令的微指令,在 $T3$ 的上升沿,从双端口存储器中取出的指令写入指令寄存器 IR 。在这条微指令中,后继微地址为 $20H$,判别位 $P1$ 为 1、其他判别位均为 0。因此根据微地址转移逻辑,很容易就知道,下一条微指令的微地址是:

$$N\mu A5-T = N\mu A5$$

$$N\mu A4-T = N\mu A4$$

$$N\mu A3-T = N\mu A3 \text{ or } P1 \text{ and } IR7$$

$$N\mu A2-T = N\mu A2 \text{ or } P1 \text{ and } IR6$$

$$N\mu A1-T = N\mu A1 \text{ or } P1 \text{ and } IR5$$

$$N\mu A0-T = N\mu A3 \text{ or } P1 \text{ and } IR4$$

新产生的微地址 $N\mu A5-T \sim N\mu A0-T$ 在 $T3$ 的下降沿写入微地址寄存器 μAR ,实现了图 2.6 微程序流程图图所要求的根据指令操作码进行微程序分枝。

五、实验任务

1. 正确设置模式开关 SWC 、 SWB 、 SWC ,用单微指令方式(单拍开关 DP 设置为 1)跟踪控制台操作读寄存器、写寄存器、读存储器、写存储器的执行过程,记录下每一步的微地址 $\mu A5 \sim \mu A0$ 、判别位 $P4 \sim P0$ 和有关控制信号的值,写出这 4 种控制台操作的作用和使用方法。
2. 正确设置指令操作码 $IR7 \sim IR4$,用单微指令方式跟踪除停机指令 STP 之外的所有指令的执行过程。记录下每一步的微地址 $\mu A5 \sim \mu A0$ 、判别位 $P4 \sim P0$ 和有关控制信号的值。对于 JZ 指令,跟踪 $Z=1$ 、 $Z=0$ 两种情况;对于 JZ 指令,跟踪 $C=1$ 、 $C=0$ 两种情况。

六、实验步骤

1. 实验准备

将控制器转换开关拨到微程序位置,微程序灯亮,将编程开关设置为正常位置,将单拍开关设置为 1(朝上)。在单拍开关 DP 为 1 时,每按一次 QD 按钮,只执行一条微指令。

将信号 $IR4-I$ 、 $IR5-I$ 、 $IR6-I$ 、 $IR7-I$ 、 $C-I$ 、 $Z-I$ 依次通过接线孔与电平 $K0 \sim K5$ 连接。通过拨动开关 $K0 \sim K5$,可以对上述信号设置希望的值。

打开电源。

2. 跟踪控制台操作读寄存器、写寄存器、读存储器、写存储器的执行。

按复位按钮 CLR 后,拨动操作模式开关 SWC 、 SWB 、 SWA 到希望的位置,按一次 QD 按钮,则进入希望的控制台操作模式。控制台模式开关和控制台操作的对应关系如下:

操作模式	功能选择	备注
000	启动程序运行	
001	写存储器	
010	读存储器	
011	读寄存器	
100	写寄存器	

按一次复位按钮 CLR 按钮，能够结束本次跟踪操作，开始下一次跟踪操作。

3. 跟踪指令的执行

按复位按钮 CLR 后，设置操作模式开关 SWC=0、SWB=0、SWA=0，按一次 QD 按钮，则进入启动程序运行模式。设置电平开关 K3~K0，使其代表希望的指令操作码 IR7~IR4，按 QD 按钮，跟踪指令的执行。

按一次复位按钮 CLR 按钮，能够结束本次跟踪操作，开始下一次跟踪操作。

七、实验要求

1. 认真做好实验的预习，掌握 TEC-8 模型计算机微程序控制器的工作原理。
2. 写出实验报告，内容是：
 - (1)实验目的
 - (2)控制台操作的跟踪过程。写出每一步的微地址 $\mu A5 \sim \mu A0$ 、判别位 P4~P0 和有关控制信号的值。
 - (3)写出这 4 种控制台操作的作用和使用方法。
 - (4)指令的跟踪过程。写出每一步的微地址 $\mu A5 \sim \mu A0$ 、判别位 P4~P0 和有关控制信号的值。
 - (5)写出 TEC-8 模型计算机中的微地址转移逻辑的逻辑表达式。分析它和各种微程序分枝的对应关系。

八、可探索和研究的问题

1. 试根据运算器组成实验、双端口存储器实验和数据通路实验的实验过程，画出这部分的微程序流程图。
2. 你能将图 2.5 中的微指令格式重新设计压缩长度吗？

2.4.2 实验报告记录:

2.5 CPU 组成与机器指令的执行

2.5.1 CPU 组成与机器指令的执行：微程序控制器方式

一、实验类型 原理性+分析性+设计性

二、实验目的

(1)用微程序控制器控制数据通路，将相应的信号线连接，构成一台能运行测试程序的 CPU。

(2)执行一个简单的程序，掌握机器指令与微指令的关系。

(3)理解计算机如何取出指令、如何执行指令、如何在一条指令执行结束后自动取出下一条指令并执行，牢固建立的计算机整机概念。

三、实验设备

序列号	名 称	数 量	备 注
1	实验系统	1 台	
2	双踪示波器	1 台	
3	直流万用表	1 块	
4	逻辑测试笔	1 支	在试验箱上

四、实验电路

本实验将前面几个实验中的所有电路，包括时序发生器、通用寄存器组、算术逻辑运算部件、存储器、微程序控制器等模块组合在一起，构成一台能够运行程序的简单处理机。数据通路的控制由微程序控制器完成，由微程序解释指令的执行过程，从存储器取出一条指令到执行指令结束的一个指令周期，是由微程序完成的，即一条机器指令对应一个微程序序列。

在本实验中，程序装入到存储器中和给寄存器置初值是在控制台方式下手工完成的，程序执行的结果也需要用控制台操作来检查。TEC-8 模型计算机的控制台操作如下：【控制转换开关放置在最下方，微程序灯亮】

(1)写存储器

写存储器操作用于向存储器中写测试程序和数据。

按复位按钮 CLR，设置 SWC=0、SWB=0、SWA=1。按 QD 按钮一次，控制台指示灯亮，指示灯 $\mu A5 \sim \mu A0$ 显示 03H，进入写存储器操作。在数据开关 SD7~SD0 上设置存储器地址，通过数据总线指示灯 D7~D0 可以检查地址是否正确。按 QD 按钮一次，将存储器地址写入地址寄存器 AR，指示灯 $\mu A5 \sim \mu A0$ 显示 02H，指示灯 AR7~AR0 显示当前存储器地址。在数据开关上设置被写的指令。按 QD 按钮一次，将指令写入存储器。写入指令后，从指示灯 AR7~AR0 上可以看到地址寄存器自动加 1。在数据开关上设置下一条指令，按 QD 按钮一次，将第 2 条指令写入存储器。这样一直继续下去，直到将测试程序全部写入存储器。

(2)读存储器

读存储器操作用于检查程序的执行结果和检查程序是否正确写入到存储器中。

按复位按钮 CLR，设置 SWC=0、SWB=1、SWA=0。按 QD 按钮一次，控制台指示灯亮，指示灯 $\mu A5 \sim \mu A0$ 显示 05H，进入读存储器操作。在数据开关 SD7~SD0 上设置存储器地址，通过指示灯 D7~D0 可以检查地址是否正确。按 QD 按钮一次，指示灯 AR7~AR0 上显示出当前存储器地址，在指示灯 D7~D0 上显示出指令或数据。再按一次 QD 按钮，则在指示灯 AR7~AR0 上显示出下一个存储器地址，在指示灯 D7~D0 上显示出下一条指令。一直操作下去，直到程序和数据全部检查完毕。

(3)写寄存器

写寄存器操作用于给各通用寄存器置初值。

按复位按钮 CLR，设置 SWC=1、SWB=0、SWA=0。按 QD 按钮一次，控制台指示灯亮，指示灯 $\mu A5 \sim \mu A0$ 显示 09H，进入写寄存器操作。在数据开关 SD7~SD0 上设置 R0 的值，通过指示灯 D7~D0 可以检查地址是否正确，按 QD 按钮，将设置的数写入 R0。指示灯 $\mu A5 \sim \mu A0$ 显示 08H，指示灯 B7~B0 显示 R0 的值，在在数据开关 SD7~SD0 上设置 R1 的值，按 QD 按钮，将设置的数写入 R1。指示灯 $\mu A5 \sim \mu A0$ 显示 0AH，指示灯 B7~B0 显示 R1 的值，在在数据开关 SD7~SD0 上设置 R2 的值，按 QD 按钮，将设置的数写入 R2。指示灯 $\mu A5 \sim \mu A0$ 显示 0CH，指示灯 B7~B0 显示 R2 的值，在在数据开关 SD7~SD0 上设置 R3 的值，按 QD 按钮，将设置的数写入 R3。指示灯 $\mu A5 \sim \mu A0$ 显示 00H，指示灯 A7~A0 显示 R0 的值，指示灯 B7~B0 显示 R3 的值。

(4)读寄存器

读寄存器用于检查程序执行的结果。

按复位按钮 CLR，设置 SWC=0、SWB=1、SWA=1。按 QD 按钮一次，控制台指示灯亮，指示灯 $\mu A5 \sim \mu A0$ 显示 07H，进入读寄存器操作。指示灯 A7~A0 显示 R0 的值，指示灯 B7~B0 显示 R1 的值。按一次 QD 按钮，指示灯 $\mu A5 \sim \mu A0$ 显示 06H，指示灯 A7~A0 显示 R2 的值，指示灯 B7~B0 显示 R3 的值。

(5)启动程序运行

当程序已经写入存储器后，按复位按钮 CLR，使 TEC-6 模型计算机复位，设置 SWC=0、SWB=0、SWA=0，按一次启动按钮 QD，则启动测试程序从地址 00H 运行。如果单拍开关 DP=1，那么每按一次 QD 按钮，执行一条微指令；连续按 QD 按钮，直到测试程序结束。如果单拍开关 DP=0，那么按一次 QD 按钮后，程序一直运行到停机指令 STP 为止。如果程序不以停机指令 STP 结束，则程序将无限运行下去，结果不可预知。

五、实验任务

1. 将下面的程序手工汇编成二进制机器代码并装入存储器。

表 2.4 在预习时完成。表中地址 0FH、10H、11H 中存放的不是指令，而是数。此程序运行前要 R2 的值为 12H，R3 的值为 0FH。

表 2.4 预习时要求完成的手工汇编

地址	指令	机器 16 进制代码
00H	LD R0, [R3]	0101 0011 【53】
01H	INC R3	0100 1100 【4C】
02H	LD R1, [R3]	0101 0111 【57】
03H	SUB R0, R1	0010 0001 【21】
04H	JZ 0BH	1000 0110 【86】
05H	ST R0, [R2]	0110 1000 【68】
06H	INC R3	0100 1100 【4C】
07H	LD R0, [R3]	0101 0011 【53】
08H	ADD R0, R1	0001 0001 【11】
09H	JC 0CH	0100 0010 【72】
0AH	INC R2	0100 1000 【48】
0BH	ST R2, [R2]	0110 1010 【6A】
0CH	AND R0, R1	0011 0001 【31】
0DH	OUT R2	1010 0010 【A2】
0EH	STP	1110 0000 【E0】
0FH	85H	1000 0101 【85】
10H	23H	0010 0011 【23】
11H	0EFH	1110 1111 【EF】
12H	00H	0000 0000 【00】

2. 通过简单的连线构成能够运行程序的 TEC-8 模型计算机。
TEC-8 模型计算机所需的连线很少，只需连接 6 条线，具体连线见实验步骤。
3. 将程序写入寄存器，并且给 R2、R3 置初值，跟踪执行程序，用单拍方式运行一遍，用连续方式运行一遍。用实验台操作检查程序运行结果。

六、实验步骤

1. 实验准备

将控制器转换开关拨到微程序位置，将编程开关设置为正常位置。

将信号 IR4-I、IR5-I、IR6-I、IR7-I、C-I、Z-I 依次通过接线孔与信号 IR4-O、IR5-O、IR6-O、IR7-O、C-O、Z-O 连接。使 TEC-8 模型计算机能够运行程序的整机系统。

打开电源。

2. 在单拍方式下跟踪程序的执行

- (1)通过写存储器操作将程序写入存储器。
- (2)通过读操作将程序逐条读出，检查程序是否正确写入了存储器。
- (3)通过写寄存器操作设置寄存器 R2 为 12H、R3 为 0FH。
- (4)通过读寄存器操作检查设置是否正确。
- (5)将单拍开关 DP 设置为 1，使程序在单微指令下运行。
- (6)按复位按钮 CLR，复位程序计数器 PC 为 00H。将模式开关设置为 SWC=0、SWB=0、SWA=0，准备进入程序运行模式。
- (7)按一次 QD 按钮，进入程序运行。每按一次 QD 按钮，执行一条微指令，直到程序结束。

在程序执行过程中，记录下列信号的值：PC7~PC0、AR7~AR0、 μ A5~ μ A0、IR7~IR0、A7~A0、B7~B0 和 D7~D0。

(8)通过读寄存器操作检查 4 个寄存器的值并记录。

(9)通过读存储器操作检查存储单元 12H、13H 的值并记录。

3. 在连续方式下运行程序

由于单拍方式下运行程序并没有改变存储器中的程序。因此只要重新设置 R2 为 12H、R3 为 0FH。然后将单拍开关 DP 设置为 0，按复位按钮 CLR 后，将模式开关设置为 SWC=0、SWB=0、SWA=0，准备进入程序运行模式。按一次 QD 按钮，程序自动运行到 STP 指令。通过读寄存器操作检查 4 个寄存器的值并记录。通过读存储器操作检查存储单元 12H、13H 的值并记录。

七、实验要求

1. 认真做好实验的预习，在预习时将程序汇编成机器十六进制代码。

2. 写出实验报告，内容是：

(1)实验目的。

(2)填写表 2.4。

(3)填写表 2.5。

表 2.5 单拍方式下指令执行结果指令执行跟踪结果

指令	μ A	PC	AR	IR	A7~A0	B7~B0	D7~D0

(4)单拍方式和连续方式程序执行后 4 个寄存器的值、寄存器 12、13 单元的值。

(5)对表 2.5 中数据的分析、体会。

(6)结合第 1 条和第 2 条指令的执行，说明计算机中程序的执行过程。

(7)结合程序中条件转移指令的执行过程说明计算机中如何实现条件转移功能。

八、可探索和研究的问题

1. 如果需要全面测试 TEC-8 模型计算机的功能，需要什么样的测试程序？请写出测试程序，并利用测试程序对 TEC-8 模型计算机进行测试。

2.5.2 实验报告记录:

Patient Information	
Full Name	
Date of Birth	
Gender	
Address	
City	
State	
Zip	
Phone	
Medical History	
Allergies	
Current Medications	
Past Medical History	
Family History	
Social History	
Physical Examination	
Vital Signs	
Laboratory Tests	
Imaging Studies	
Diagnosis	
Treatment Plan	
Follow-up	

2.6 中断原理实验

2.6.1 中断原理实验部分：微程序控制器方式

一、实验类型 原理性+分析性

二、实验目的

- (1)从硬件、软件结合的角度，模拟单级中断和中断返回的过程；
- (2)通过简单的中断系统，掌握中断控制器、中断向量、中断屏蔽等概念；
- (3)了解微程序控制器与中断控制器协调的基本原理；
- (4)掌握中断子程序和一般子程序的本质区别，掌握中断的突发性和随机性。

三、实验设备

序列号	名 称	数 量	备 注
1	实验系统	1 台	
2	双踪示波器	1 台	
3	直流万用表	1 块	
4	逻辑测试笔	1 支	在试验箱上方

四、实验原理

1. TEC-8 模型计算机中的中断机构

TEC-8 模型计算机中有一个简单的单级中断系统，只支持单级中断、单个中断请求，有中断屏蔽功能，旨在说明最基本的工作原理。

TEC-8 模型计算机中有 2 条指令用于允许和屏蔽中断。DI 指令称作关中断指令。此条指令执行后，即使发生中断请求，TEC-8 也不响应中断请求。EI 指令称作开中断指令，此条指令执行后，TEC-8 响应中断。在时序发生器中，设置了一个允许中断触发器 EN_INT，当它为 1 时，允许中断，当它为 0 时，禁止中断发生。复位脉冲 CLR#使 EN_INT 复位为 0。使用 VHDL 语言描述的 TEC-8 中的中断控制器如下：

```
INT_EN_P: process (CLR#, MF, INTEN, INTDI, PULSE, EN_INT)
begin
    if CLR# = '0' then
        EN_INT <= '0';
    elsif MF'event and MF = '1' then
        EN_INT <= INTEN or (EN_INT and (not INTDI));
    end if;
    INT <= EN_INT and PULSE;
end process;
```

在上面的描述中，CLR#是按下复位按钮 CLR 后产生的低电平有效的复位脉冲，MF 是 TEC-8 的主时钟信号，INTEN 是执行 EI 指令产生的允许中断信号，INTDI 是执行 DI 指令产生的禁止中断信号，PULSE 是按下 PULSE 按钮产生的高电平有效的中断请求脉冲信号，INT 是时序发生电路向微程序控制器输出的中断程序执行信号。

为保存中断断点的地址，以便程序被中断后能够返回到原来的地址继续执行，设置了一个中断地址寄存器 IAR，参看第 1 章中的图 1.2。中断地址寄存器 IAR 是 1 片 74LS374(U44)。当信号 LIAR 为 1 时，在 T3 的上升沿，将 PC 保存在 IAR 中。当信号 IABUS 为 1 时，IABUS

中保存的 PC 送数据总线 DBUS，指示灯显示出中断地址。由于本实验系统只有一个断点寄存器而无堆栈，因此仅支持一级中断而不支持多级中断。

中断向量即中断服务程序的入口地址，本实验系统中由数据开关 SD7~SD0 提供。

2. 中断的检测、执行和返回过程

一条指令的执行由若干条微指令构成。TEC-8 模型计算机中，除指令 EI、DI 外，每条指令执行过程的最后一条微指令都包含判断位 P4，用于判断有无中断发生，参看本章图 2.6。因此在每一条指令执行之后，下一条指令执行之前都要根据中断信号 INT 是否为 1 决定微程序分支。如果信号 INT 为 1，则转微地址 11H，进入中断处理；如果信号 INT 为 0，则转微地址 01H，继续取下一条指令然后执行。

检测到中断信号 INT 后，转到微地址 11H。该微指令产生 INTDI 信号，禁止新的中断发生，产生 LIAR 信号，将程序计数器 PC 的当前值保存在中断地址寄存器(断点寄存器)中，产生 STOP 信号，等待手动设置中断向量。在数据开关 SD7~SD0 上设置好中断地址后，机器将中断向量读到 PC 后，转到中服务程序继续执行。

执行一条指令 IRET，从中断地址返回。该条指令产生 IABUS 信号，将断点地址送数据总线 DBUS，产生信号 LPC，将断点从数据总线装入 PC，恢复被中断的程序。

发生中断时，关中断由硬件负责。而中断现场(包括 4 个寄存器、进位标志 C 和结果为 0 标志 Z)的保存和恢复由中断服务程序完成。中断服务程序的最后两条指令一般是开中断指令 EI 和中断返回指令 IRET。为了保证从中断服务程序能够返回到主程序，EI 指令执行后，不允许立即被中断。因此，EI 指令执行过程中的最后一条微指令中不包含 P4 判别位。

五、实验任务

1. 了解中断每个信号的意义和变化条件，并将表 2.5 中的主程序和表 2.6 中的中断服务程序手工汇编成十六进制机器代码。此项任务在预习中完成。

表 2.5 主程序的机器代码

地址	指令	机器代码
00H	EI	
01H	INC R0	
02H	INC R0	
03H	INC R0	
04H	INC R0	
05H	INC R0	
06H	INC R0	
07H	INC R0	
08H	INC R0	
09H	JMP [R1]	

表 2.6 中断服务程序的机器代码

地址	指令	机器代码
45H	ADD R0, R0	
46H	EI	
46H	IRET	

为了保证此程序能够循环执行，应当将 R1 预先设置为 01H。R0 的初值设置为 0。

2. 将 TEC-8 连接成一个完整的模型计算机。
3. 将主程序和中断服务程序装入存储器，执行 3 遍主程序和中断服务程序。列表记录中断有关信号的变化情况。特别记录好断点和 R0 的值。
4. 将存储器 00H 中的 EI 指令改为 DI，重新运行程序，记录发生的现象。

六、实验步骤

1. 实验准备

将控制器转换开关拨到微程序位置，将编程开关设置为正常位置。

将信号 IR4-I、IR5-I、IR6-I、IR7-I、C-I、Z-I 依次通过接线孔与信号 IR4-O、IR5-O、IR6-O、IR7-O、C-O、Z-O 连接。使 TEC-8 模型计算机能够运行程序的整机系统。

打开电源。

2. 通过控制台写存储器操作，将主程序和中断服务程序写入存储器。

3. 执行 3 遍主程序和中断子程序

(1)通过控制台写寄存器操作将 R0 设置为 00H，将 R1 设置为 01H。

(2)将单拍开关 DP 设置为连续运行方式(DP=0)，按复位按钮 CLR，使 TEC-8 模型计算机复位。按 QD 按钮，启动程序从 00H 开始执行。

(3)按一次 PULSE 按钮，产生一个中断请求信号 PULSE，中断主程序的运行。记录下这时的断点 PC、R0(指示灯 A7~A0 上显示)的值。

(4)将单拍开关 DP 设置为单拍方式(DP=1)，在数据开关上设置中断服务程序的入口地址 45H。按 QD 按钮，一步步执行中断服务程序，直到返回到断点为止。

(5)按照步骤(1)~(4)，再重复做 2 遍。

4. 将存储器 00H 的指令改为 DI，按照步骤 3，重做一遍，记录发生的现象。

七、实验要求

1. 认真做好实验的预习，在预习时将程序汇编成机器十六进制代码。
2. 写出实验报告，内容是：
 - (1)实验目的。
 - (2)填写表 2.5。
 - (3)填写表 2.6。
 - (4)填写表 2.7。

表 2.7 中断原理实验结果

执行程序顺序	PC 断点值	中断时的 R0
第 1 遍		
第 2 遍		
第 3 遍		
第 4 遍		

- (4)分析实验结果，得到什么结论？
- (5)简述 TEC-8 模型计算机的中断机制。

八、可研究和探索的问题

1. 在 TEC-8 模型计算机中，采用的是信号 PULSE 高电平产生中断。如果改为信号 PULSE 的上升沿产生中断，怎么设计时序发生器中的中断机制？提出设计方案。

2.6.2 实验报告记录:

第 3 章 课程综合设计

4 个课程设计实验是大型的综合性研究课题。采用大容量的 ISP(在系统可编程)器件或,集中安排在小学期 2 周内独立完成。经验证明,课程综合设计是理论与实践相统一、培养学生研究途径的有效途径。学生根据自己情况选择其中 1~2 个课题,其中使用硬连线控制器的 CPU 设计必做。

3.1 模型机硬连线控制器设计

3.1.1 模型机硬连线控制器设计部分

一、教学目的

- (1) 融会贯通计算机组成与体系结构课程各章教学内容,通过知识的综合运用,加深对 CPU 各 模块工作原理及相互联系的认识。
- (2) 掌握硬连线控制器的设计方法。
- (3) 学习运用当代的 EDA 设计工具,掌握用 EDA 设计大规模复杂逻辑电路的方法。
- (4) 培养科学研究能力,取得设计和调试的实践经验。

二、实验设备

序列号	名 称	数 量	备 注
1	实验系统	1 台	
2	双踪示波器	1 台	
3	直流万用表	1 块	
4	逻辑测试笔	1 支	在试验箱上方
5	PC 机	1 台	

三、设计与调试任务

1. 设计一个硬连线控制器,和 TEC-8 模型计算机的数据通路结合在一起,构成一个完整的 CPU,该 CPU 要求:
 - (1) 能够完成控制台操作:启动程序运行、读存储器、写存储器、读寄存器和写寄存器。
 - (2) 能够执行表 3.1 中的指令,完成规定的指令功能。

表 3.1 新设计 CPU 的指令系统

名称	助记符	功 能	指令格式		
			IR (7-4)	IR (3-2)	IR (1-0)
加法	ADD Rd, Rs	$Rd \leftarrow Rd + Rs$	0001	Rd	Rs
减法	SUB Rd, Rs	$Rd \leftarrow Rd - Rs$	0010	Rd	Rs
逻辑与	AND Rd, Rs	$Rd \leftarrow Rd \text{ and } Rs$	0011	Rd	Rs
加 1	INC Rd	$Rd \leftarrow Rd + 1$	0100	Rd	XX
取数	LD Rd, [Rs]	$Rd \leftarrow [Rs]$	0101	Rd	Rs
存数	ST Rs, [Rd]	$Rs \rightarrow [Rd]$	0110	Rd	Rs
C 条件转移	JC addr	如果 C=1, 则 $PC \leftarrow @ + \text{offset}$	0111	offset	
Z 条件转移	JZ addr	如果 Z=1, 则 $PC \leftarrow @ + \text{offset}$	1000	offset	
无条件转移	JMP [Rd]	$PC \leftarrow Rd$	1001	Rd	XX
停机	STOP	暂停运行	1110	XX	XX

表 13.1 中, XX 代表随意值。Rs 代表源寄存器号, Rd 代表目的寄存器号。在条件转移指令中, @代表当前 PC 的值, offset 是一个 4 位的有符号数, 第 3 位是符号位, 0 代表正数, 1 代表负数。**注意: @不是当前指令的 PC 值, 是当前指令的 PC 值加 1。**

2. 在 Quartus II 下对硬连线控制器对设计方案进行编程和编译
3. 将编译后的硬连线控制器下载到 TEC-8 实验台上的 ISP 器件 EPM7128 中去, 使 EPM7128 成为一个硬连线控制器。
4. 根据指令系统, 编写检测硬连线控制器正确性的测试程序, 并用测试程序对硬布线控制器在单拍方式下进行调试, 直到成功。
5. 在调试成功的基础上, 整理出设计文件, 包括:
 - (1)硬连线控制器逻辑模块图
 - (2)硬连线控制器指令周期流程图
 - (3)硬连线控制器的 VHDL 源程序
 - (4)测试程序
 - (5)设计说明书
 - (6)调试总结

四、设计提示

1. 硬连线控制器的基本原理

硬连线控制器的基本原理, 每个微操作控制信号 S 是一系列输入量的逻辑函数, 即用组合逻辑来实现,

$$S = f(I_m, M_i, T_k, B_j)$$

其中 I_m 是机器指令操作码译码器的输出信号, M_i 是节拍电位信号, T_k 是节拍脉冲信号, B_j 是状态条件信号。

在 TEC-8 实验系统中, 节拍脉冲信号 $T_k(T_1 \sim T_3)$ 已经直接输送给数据通路。因为机器指令系统比较简单, 省去操作码译码器, 4 位指令操作码 $IR_4 \sim IR_7$ 直接成为 I_m 的一部分; 由于 TEC-8 实验系统有控制台操作, 控制台操作可以看作一些特殊的功能复杂的指令, 因此

SWC、SWB、SWA 可以看作是 I_m 的另一部分。 M_i 是时序发生器产生的节拍信号 $W_1 \sim W_3$ ； B_j 包括 ALU 产生的进位信号 C 、结果为 0 信号 Z 等等。

2. 机器指令周期流程图设计

设计微程序控制器使用流程图。设计硬连线控制器同样使用流程图。微程序控制器的控制信号以微指令周期为时间单位，硬连线控制器以节拍电位 (CPU 周期) 为时间单位，两者在本质上是一样的，1 个节拍电位时间和 1 条微指令时间都是从节拍脉冲 T_1 的上升沿到 T_3 的下降沿的一段时间。在微程序控制器流程图中，一个执行框代表一条微指令，在硬连线控制器流程图中，一个执行框代表一个节拍电位时间。

3. 执行一条机器指令的节拍电位数

在 TEC-8 实验系统中，采用了可变节拍电位数来执行一条机器指令。大部分指令的执行只需 2 个节拍电位 W_1 、 W_2 ，少数指令需要 3 个节拍电位 W_1 、 W_2 、 W_3 。为了满足这种要求，在执行一条指令时除了产生完成指令功能所需的微操作控制信号外，对需要 3 个电位节拍的指令，还要求它在 W_2 时产生一个信号 $LONG$ 。信号 $LONG$ 送往时序信号发生器，时序信号发生器接到信号 $LONG$ 后产生节拍电位 W_3 。

对于一些控制台操作，需要 4 个节拍电位才能完成规定的功能。为了满足这种情况，可以将控制台操作化成两条机器指令的节拍。为了区分写寄存器操作的 2 个不同阶段，可以用某些特殊的寄存器标志标。例如建立一个 $FLAG$ 标志，当 $FLAG=0$ 时，表示该控制台操作的第 1 个 W_1 、 W_2 ；当 $FLAG=1$ 时，表示该控制台操作的第 2 个 W_1 、 W_2 。

为了适应更为广泛的情况，TEC-8 的时序信号发生器允许只产生一个节拍电位 W_1 。当 1 条指令或者一个控制台在 W_1 时，只要产生信号 $SHORT$ ，该信号送往时序信号发生器，则时序信号发生器在 W_1 后不产生节拍电位 W_2 ，下一个节拍仍是 W_1 。

信号 $LONG$ 和 $SHORT$ 只对紧跟其后的第一个节拍电位的产生起作用。

在硬连线控制器中，控制台操作的流程图与机器指令流程图类似，图 3.1 画出了硬连线控制器的机器周期参考流程图。


```

begin
    LIR <= '0' ;
    M <= '0' ;
    S3 <= '0' ;
    S2 <= '0' ;
    S1 <= '0' ;
    case IR is
        when "0001" =>
            LIR <= W1;
            S3 <= W2;
        when "0010" =>
            LIR <= W1;
            S2 <= W2;
            S1 <= W2;
        when "0011" =>
            LIR <= W1;
            M <= W2;
            S3 <= W2;
            S1 <= W2;
        .....
    end case;
end;

```

很明显，这种方法省略了译码表，且不容易出错。

5. EPM7128 器件的引脚

TEC-8 实验系统中的硬连线控制器是用 1 片 EPM7128 器件构成的。为了使将主要精力集中在硬连线控制器的设计和调试上，硬连线控制器和数据通路之间不采用接插线方式连接，在印制电路板上已经用印制导线进行了连接。这就要求硬连线控制器所需的信号的输出、输入信号的引脚号必须符合表 3.3 中的规定。

表 3.3 作为硬连线控制器时的 EPM7128 引脚规定

信号	方向	引脚号	信号	方向	引脚号
CLR#	输入	1	MEMW	输出	27
T3	输入	83	STOP	输出	28
SWA	输入	4	LIR	输出	29
SWB	输入	5	LDZ	输出	30
SWC	输入	6	LDC	输出	31
IR4	输入	8	CIN	输出	33
IR5	输入	9	S0	输出	34
IR6	输入	10	S1	输出	35
IR7	输入	11	S2	输出	36
W1	输入	12	S3	输出	37
W2	输入	15	M	输出	39
W3	输入	16	ABUS	输出	40
C	输入	2	SBUS	输出	41
Z	输入	84	MBUS	输出	44
DRW	输出	20	SHORT	输出	45
PCINC	输出	21	LONG	输出	46
LPC	输出	22	SEL0	输出	48
LAR	输出	25	SEL1	输出	49
PCADD	输出	18	SEL2	输出	50
ARINC	输出	24	SEL3	输出	51
SELCTL	输出	52			

6. 调试

由于使用在系统可编程器件，集成度高，灵活性强，编程、下载方便，用于硬连线控制器将使调试简单。控制器内部连线集中在器件内部，由软件自动完成，其速度、准确率和可靠性都是人工接线难以比拟的。

用 EDA 技术进行设计，可以使用软件模拟的向量测试对设计进行初步调试。软件模拟和使用向量测试时，向量测试方程的设计应全面，尽量覆盖所有的可能性。

在软件模拟测试后，将设计下载到 EPM7128 器件中。将控制开关拨到硬连线控制器方式，硬连线灯亮。首先单拍 (DP=1) 方式检查控制台操作功能。第二步将测试程序写入存储器，以单拍方式执行程序，直到按照流程图全部检查完毕。在测试过程中，要充分利用 TEC-8 实验系统上的各种信号指示灯。

五、设计报告要求

1. 采用 VHDL 语言描述硬连线控制器的设计，列出设计源程序。
2. 测试程序。
3. 写出调试中出现的问题、解决办法、验收结果。
4. 写出设计、调试中遇到的困难和心得体会。

六、测试报告

一、写存储器模式：

序号	操作模式	动作	微址	SD	D	AR	备注
1	1001	按 CLR	00				初始化
2	1001	按 QD	03	00	00		进入写存储器
3	1001	按 QD	02	XX	XX	00	写入数据
4	1001	按 QD	02	XX	XX	01	
				XX 是用 SD 拨入的数据【以下同】			

二、读存储器模式：

序号	操作模式	动作	微址	SD	D	AR	备注
1	1010	按 CLR	00				初始化
2	1010	按 QD	05	00	00		进入读存储器
3	1010	按 QD	04		XX	00	读出数据
4	1010	按 QD	04		XX	01	

三、写寄存器模式：

序号	操作模式	动作	微址	SD	D	B	备注
1	1100	按 CLR	00				
2	1100	按 QD	09	01	01		置【R0】
3	1100	按 QD	08	XX	XX	XX	置【R1】
4	1100	按 QD	0A	XX	XX	XX	置【R2】
5	1100	按 QD	0C	XX	XX	XX	置【R3】
6	1100	按 QD	00	XX	XX	XX	

四、读寄存器模式：

序号	操作模式	动作	微址	A	B	备注
1	1011	按 CLR	00			
2	1011	按 QD	07	XX	XX	【R0】【R1】
3	1011	按 QD	06	XX	XX	【R2】【R3】
4	1011	按 QD	00	01	01	

This image shows a completely blank white rectangular area. It is surrounded by a thin, solid black border that frames the entire content. There are no markings, text, or illustrations on the page.

3.2 模型机流水微程序控制器设计

3.2.1 模型机流水微程序控制器设计部分

一、教学目的

- (1) 融会贯通计算机组成与体系结构课程各章教学内容，通过知识的综合运用，加深对 CPU 各模块工作原理及相互联系的认识。
- (2) 掌握流水微程序控制器的设计方法。
- (3) 培养科学研究能力，取得设计和调试的实践经验。

二、实验设备

序列号	名 称	数 量	备 注
1	实验系统	1 台	
2	双踪示波器	1 台	
3	直流万用表	1 块	
4	逻辑测试笔	1 支	在试验箱上方
5	PC 机	1 台	

三、调试与设计任务

1. 设计一个流水微程序控制器，和 TEC-8 模型计算机的数据通路结合在一起，构成一个完整的 CPU，该 CPU 要求：

(1) 能够完成控制台操作：启动程序运行、读存储器、写存储器、读寄存器和写寄存器。控制台操作不要求流水。

(2) 能够执行本章 3.1 节表 3.1 中的指令，完成规定的指令功能。

3. 根据指令系统，编写检测流水微程序控制器正确性的测试程序，并用测试程序对流水微程序控制器在单拍方式下进行调试，直到成功。

4. 在调试成功的基础上，整理出设计文件，包括：

- (1) 流水微程序控制器指令周期流程图
- (2) 微指令代码表
- (3) 5 个控制存储器 E²PROM 的二进制代码文件
- (4) 测试程序
- (5) 设计说明书
- (6) 调试总结

四、设计提示

1. 流水微程序控制器参考流程图

微指令格式见第 2 章第 2.6 节图 2.16。流水微程序控制器参考流程图见图 3.2。

5. 对流水微程序控制器进行调试

五、设计报告要求

1. 流水微程序控制器流程图。
2. 微指令代码表。
3. 测试程序
4. 写出调试中出现的问题、解决办法、验收结果。
5. 写出设计、调试中遇到的困难和心得体会。

[illegible]

3.3 模型机流水硬连线控制器设计

3.3.1 模型机流水硬连线控制器设计

一、教学目的

- (1) 融会贯通计算机组成与体系结构课程各章教学内容，通过知识的综合运用，加深对 CPU 各模块工作原理及相互联系的认识。
- (2) 掌握流水硬连线控制器的设计方法。
- (3) 学习运用当代的 EDA 设计工具，掌握用 EDA 设计大规模复杂逻辑电路的方法。
- (4) 培养科学研究能力，取得设计和调试的实践经验。

二、实验设备

序号	名称	数量	备注
1	实验系统	1 台	
2	双踪示波器	1 台	
3	直流万用表	1 块	
4	逻辑测试笔	1 支	在试验箱上方
5	PC 机	1 台	

三、设计与调试任务

1. 设计一个流水硬连线控制器，和 TEC-8 模型计算机的数据通路结合在一起，构成一个完整的 CPU，该 CPU 要求：
 - (1) 能够完成控制台操作：启动程序运行、读存储器、写存储器、读寄存器和写寄存器。
 - (2) 能够执行本章第 3.1 节的表 3.1 中的指令，完成规定的指令功能。
2. 在 Quartus II 下对硬连线控制器对设计方案进行编程和编译
3. 将编译后的流水硬连线控制器下载到 TEC-8 实验台上的 ISP 器件 EPM7128 中去，使 EPM7128 成为一个流水硬连线控制器。
4. 根据指令系统，编写检测流水硬连线控制器正确性的测试程序，并用测试程序对硬连线控制器在单拍方式下进行调试，直到成功。
5. 在调试成功的基础上，整理出设计文件，包括：
 - (1) 流水硬连线控制器逻辑模块图
 - (2) 流水硬连线控制器指令周期流程图
 - (3) 流水硬连线控制器的 VHDL 源程序
 - (4) 测试程序
 - (5) 设计说明书
 - (6) 调试总结

四、设计提示

1. 流水硬连线控制器参考流程图

流水硬连线控制器参考流程图见图 3.3。

3.3.2 实验报告记录:

3.4 含有阵列乘法器的 ALU 设计

3.4.1 含有阵列乘法器的 ALU 设计

一、教学目的

- (1) 掌握阵列乘法器的组织结构和实现方法。
- (2) 改进 74181 的内部结构设计, 仅实现加、减、乘、传送、与、加 1、取反、求补等 8 种操作。
- (3) 学习运用当代的 EDA 设计工具, 掌握用 EDA 设计大规模复杂逻辑电路的方法。
- (4) 培养科学研究能力, 取得设计和调试的实践经验。

二、实验设备

序列号	名 称	数 量	备 注
1	实验系统	1 台	
2	双踪示波器	1 台	
3	直流万用表	1 块	
4	逻辑测试笔	1 支	在试验箱上方
5	PC 机	1 台	

三、设计与调试任务

1. 设计 1 个 4 位 \times 4 位的阵列乘法器, 其积为 8 位。
2. 设计一个 ALU, 能够对 2 个 4 位操作数 A 和 B 进行 8 种运算, 由 3 位操作码控制进行何种运算。3 位操作码从电平开关 SWA、SWB、SWC 输入。要求:

操作模式	功能选择	备 注
000	A+B	
001	A-B	
010	A and B	
011	A or B	
100	not A	
101	A xor B	
110	not (A and B)	
111	乘	

操作数 A 从开关 K3~K0 输入, 操作数 B 从开关 K11~K8 输入。运算结果送指示灯 L7~L0 输出, L0 为最低位, L7 为最高位。在做加、减法时, L4 显示进位。在做与、或、求反、异或、与非运算时, L7~L4 显示 0。

2. 在 Quartus II 下对改进 ALU 的设计方案进行编程和编译
3. 将编译后的 ALU 下载到 TEC-8 实验台上的 ISP 器件 EPM7128 中去, 使 EPM7128 成为含有阵列乘法器的 ALU。
4. 测试方法和正确性验证。
5. 写出设计、调试报告总结。

四、设计提示

1. 无符号阵列乘法器结构

无符号阵列乘法器的结构框图如图 3.4 所示。它由一系列全加器 FA 用流水方法(时间并行)和资源重复方式(空间并行)有序组成。图 3.4 所示为 5×5 位的阵列乘法器。

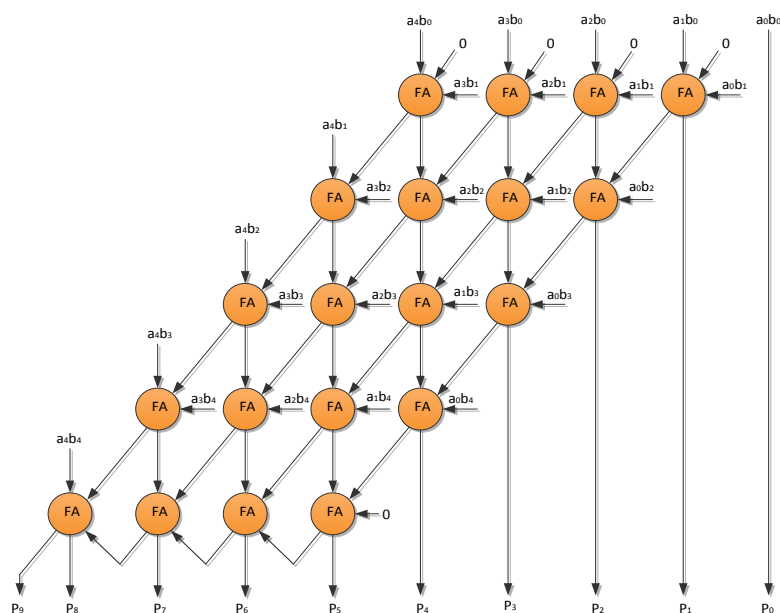


图 3.4 无符号阵列乘法器框图

2. EPM7128 和电平开关 K15~K0、指示灯 L11~L0 的连接

EPM7128 通过一条 34 芯扁平电缆和电平开关 K15~K0、指示灯 L15~L0 连接。连接时扁平电缆的 34 芯端插到插座 J6 上，扁平电缆的 12 芯端插到插座 J4 上(请注意：P12 的滑面朝下或者箭头朝在左边)，16 芯插到插座 J8 上。电平开关 K15~K0、指示灯 L11~L0 对应的 EPM7128 引脚如下表 3.5 所示。

表 3.5 电平开关、指示灯对应的 EPM7128 引脚号

电平开关	方向	引脚号	指示灯	方向	引脚号
S0	输入	54	L0	输出	37
S1	输入	81	L1	输出	39
S2	输入	80	L2	输出	40
S3	输入	79	L3	输出	41
S4	输入	77	L4	输出	44
S5	输入	76	L5	输出	45
S6	输入	75	L6	输出	46
S7	输入	74	L7	输出	48
S8	输入	73	L8	输出	49
S9	输入	70	L9	输出	50
S10	输入	69	L10	输出	51
S11	输入	68	L11	输出	52
S12	输入	67			
S13	输入	65			
S14	输入	64			
S15	输入	63			
SWA	输入	4			
SWB	输入	5			
SWC	输入	6			

3. 运算测试

(1) 用测试数据表 3.6 中的数据对乘法进行验证测试，运算结果正确。

表 3.6 乘法测试数据

组号 数据	1	2	3	4	5	6
被乘数 A	9	15	0	15	随机	随机
乘数 B	8	15	15	0	随机	随机
乘积 P	72	225	0	0		

(2) ALU 的其它 7 种操作验证测试与乘法类似，自行设计测试数据表。

五、设计报告要求

1. 采用 VHDL 语言或者原理图描述改进 ALU 的设计，列出源程序或者画出原理图。
2. 测试数据表及测试结果。
3. 写出调试中出现的问题、解决办法、验收结果。
4. 写出设计、调试中遇到的困难和心得体会。

六、实验完成后，必须要将硬连线控制器重新装入 EPM7128 中。以备以后做其他实验使用。

3.3.2 实验报告记录:

第 4 章 数字逻辑与数字系统综合设计实验

4.1 简易电子琴实验

4.1.1 简易电子琴实验部分

本实验是个大型实验，可作为课程设计使用，也可作为一个演示实验。

一、实验目的

- (1) 掌握简易电子音响的基本原理。
- (2) 通过简易电子的设计掌握数字逻辑系统的设计方法。
- (3) 掌握 EDA 软件 Quartus II 的基本使用方法。
- (4) 掌握用 VHDL 语言设计复杂数字电路的方法。

二、实验原理

1. 简易电子音响

电子音响是当今一个很时髦的物品，操作简单，声音动听。大家都知道一个基本的物理原理：振动发声。无论是何种声音，都是通过振动产生的。例如刮风时由于空气的振动产生了风声，用木棍敲击铜钟时，铜钟振动产生钟声。在 TEC-8 实验台上，通过喇叭的纸盆振动发声，只要控制纸盆的振动频率，就能控制声音的音调；只要控制纸盆振动的长短，就能控制声音的节拍；只要控制振动的幅度，就能控制声音的强度。本实验只控制声音的音调和节拍。

2. 小喇叭及其驱动电路

TEC-8 实验台上的喇叭及其驱动电路如图 4.1 所示。

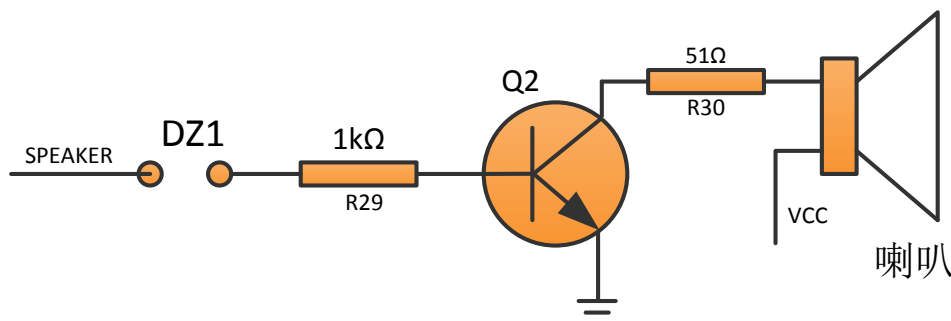


图 4.1 喇叭及其驱动电路

图 4.1 中，当短路子 DZ1 断开时，喇叭不受控制，因此不发声。喇叭的阻抗为 8 欧姆，R30 是防止喇叭烧毁的限流电阻。R29 是晶体三极管 Q2 的基极电阻，当控制信号 SPEAKER 为高电平时，Q2 饱和，电流流过喇叭；当控制信号 SPEAKER 为低电平时，Q2 截止，没有电流流过喇叭。控制了电流流过喇叭的频率，就控制了喇叭纸盆振动的频率。本实验中，我们使用方波信号 SPEAKER 控制喇叭纸盆的振动。方波信号虽然不是正弦波，但它的基波是正弦波，而且频率同方波频率一致。基波在控制喇叭振动中起主要作用。喇叭的纸盆振动也不可能突变，因此用方波信号控制喇叭纸盆的振动也能产生出清晰的声音。

3. 音调的频率

每个音调对应 1 个固定的频率，本实验中用到的 C 调的音符和对应频率见表 4.1。

表 4.1 C 调的部分音符对应频率表

音调	1	2	3	4	5	6	7	i
频率(Hz)	262	294	330	349	392	440	494	523

三、实验设备

序 号	名 称	数 量	备 注
1	个人计算机	1 台	
2	TEC-8 实验系统	1 台	
3	双踪示波器	1 台	
4	万用表	1 只	

四、实验要求

- (1) 利用 TEC-8 实验系统上的时钟信号 MF，对其进行分频，产生出表 4.1 中的 8 个音频信号。
- (2) 使用 TEC-8 实验系统上的 8 个开关 K8~K1 代表 8 个琴键。其中 K8 代表 i。当 K8~K1 中的某个开关拨到向上位置时，发出对应的音调，拨到向下位置时，停止发声。任何时候，只允许 K8~K1 中的一个开关处于向上位置。
- (3) 在个人计算机 PC 上 Quartus II 环境中用 VHDL 语言设计出程序，经过编译，然后下载到 EPM7128 器件中，构成一个电子琴。
- (4) 听喇叭发出的声音，如果不符合要求，重新修改程序、编译后下载，直到正确为止。

五、实验提示

1. EPM7128 引脚信号

喇叭控制信号 SPEAKER 由 TEC-8 实验台的 EPM7128 器件产生。本实验中，使用下列信号，对应 EPM7128 的引脚如下：

序号	信号名	管脚号	方向	信 号 意 义
1	MF	55	In	1MHz 的时钟频率。
2	CLR#	1	OUT	复位信号，低电平有效。
3	Speaker	52	Out	喇叭的控制信号
4	K1	81	IN	开关
5	K2	80	IN	开关
6	K3	79	IN	开关
7	K4	77	IN	开关
8	K5	76	IN	开关
9	K6	75	IN	开关
10	K7	74	IN	开关
11	K8	73	IN	开关

上述信号中的 MF、K8~K1，EMP7128 的引脚并没有直接和时钟、开关连接，它们是通过一条扁平电缆连接的，因此做实验时，需将扁平电缆的 34 芯端插到插座 J6 中，将 6 芯端插到插座 J5 中，将 16 芯端插到插座 J8 中。**注意：扁平电缆进行插接或者拔出必须在关电源后进行。**另外，做实验时，应将短路子 DZ1 左侧短接，以使喇叭受到控制；实验完成后，应将短路子 DZ1 右侧短接，使其断开。

2. 音调分频程序示例

音调 1 的频率是 262Hz，音调 i 的频率是 523Hz，如果直接用 1MHz 的主时钟信号分频，则产生 262Hz 的信号需要对 1MHz 信号进行 3816(十六进制 0EE8)分频，产生 523Hz 的信号需要对 1MHz 的信号进行 1912(十六进制 778)分频。这样产生 262Hz 信号的分频器需要 12 个宏单元，产生 523Hz 信号的分频器需要 11 个宏单元，占用资源太多。为了节省资源，首先对 1MHz 信号进行 10 分频得到 100KHz 信号，作为分频的基准，其他音调的信号由 100KHz 信号通过分频产生。

驱动喇叭发声的信号 SPEAKER 应当是占空比为 50%的方波。

对 100KHz 信号进行分频产生 261Hz 的设计程序如下：

```
p1:process(clr, f100k, f1_t, f1)
begin
    if clr = '0' then
        f1_t <= x"00";
        f1 <= '0';
    elsif f100k'event and f100k = '1' then
        if f1_t = x"be" then    --十六进制 be 相当于 190，因此是进行 191 分频。
            f1_t <= x"00";
            f1 <= not f1;      --2 分频
        else
            f1_t <= f1_t + '1';
            f1 <= f1;
        end if;
    end if;
end if;
```


end process;

上面的程序中首先对 100KHz 信号进行了 191 分频得到信号 f1_t, 然后对 f1_t 进行 2 分频得到信号 f1。f1 实际频率为 261.7Hz。。

3. 如果只进行演示性实验, 则可以直接把电子琴文件夹中的 .sof 文件下载到 EPM7128 中进行。

六、实验完成后, 必须要将硬连线控制器重新装入 EPM7128 中。以备以后做其他实验使用。

A large, empty rectangular frame with a thin black border, occupying the majority of the page below the header. The frame is intended for a drawing or illustration related to the text above.

4.2 简易频率计实验

4.2.1 简易频率计实验部分

一、实验目的

- (1) 掌握频率计的基本原理。
- (2) 通过简易频率计的设计掌握数字逻辑系统的设计方法。
- (3) 掌握 EDA 软件 Quartus II 的基本使用方法。
- (4) 掌握用 VHDL 语言设计复杂数字电路的方法。

二、实验原理

频率计是一种常用的仪器，用于测量一个信号的频率或者周期。与示波器相比，它测量频率更加准确、直观。

一个频率计总体上分为两部分：第一部分以被测信号作为计数时钟进行计数，第二部分将计数结果显示出来。频率的显示方法有两种，一种是用数码管显示，一种是用液晶显示屏显示。本实验采用的是数码管显示方法，用 6 个数码管显示，最多显示 6 位 10 进制数。LG1 显示个位数，LG2 显示十位数，LG3 显示百位数，LG4 显示千位数，LG5 显示万位数，LG6 显示十万位数。6 个数码管中，对 LG1 采用使用 8 位数据直接驱动各发光二极管的方法，对 LG6~LG2 采用 8-4-2-1 编码的方式驱动。具体驱动方式，参看本书第 4 章 4.9 节。

频率计中，被测信号作为计数时钟进行计数时需要一个时间闸门，只有在这个时间闸门允许的时间段内才能进行计数。例如时间闸门可以是 0.001 秒、0.01 秒、0.1 秒、1 秒、10 秒等等。如果时间闸门选用 1 秒，那么对被测信号计数得到的数就是该信号的实际频率；如果时间闸门选用 0.001 秒，那么以被测信号作为计数时钟进行计数得到的计数器的值是被测信号实际频率的千分之一；如果时间闸门选用 10 秒，那么计数器的值是被测信号实际频率的 10 倍。对于选用 10 秒的时间闸门而言，在显示频率的时候，要将小数点放在最低位之前，这样可以得到 0.1Hz 的分辨率。对于选用 0.001 秒的时间闸门，显示的是 KHz 而不是 Hz，用米字型数码管显示“KHz”也不难。不过 TEC-8 实验台上的数码管不能显示“K”字符。在本实验中，只选用时间闸门为 1 秒，只显示频率的数字，不显示单位。

时间闸门要求很高的精度，精度至少要在 10^{-5} 以上。因此产生时间闸门信号时一定用到高精度石英晶体振荡器。如果石英晶体振荡器的频率是 1MHz，对其进行 1000 分频，能得到 0.001 秒的时间闸门，对其进行 1000000 分频，得到 1 秒的时间闸门。

由于人眼不能分辨眼花缭乱的显示，因此需要在计数器停止计数后需要一段较长的时间显示频率。例如可以 1 秒时间计数，1 秒时间显示计数值。还有的做法是计数过程中随时显示，计数结束后用 1 秒时间显示计数结果。需要注意的是在时间闸门中计数前，首先要使计数器复位为 0，以保证计数值准确。这个计数器复位工作通常在按下复位按钮后进行一次，保证第一次计数准确；在显示用的每个时间段的最后通过内部产生的复位信号进行 1 次复位计数器工作，为下次计数做好准备。

影响频率计计数能力的主要是计数器的速度。如果一个计数器的计数速度不高，就无法对高频信号计数，测出的频率是不真实的。本实验中不考虑这个问题。

在测量一个信号的周期时，通常的做法是将信号的周期作为时间闸门，用一个频率精确的信号作为计数时钟脉冲，计算在时间闸门内通过的计数时钟脉冲个数。例如，采用 1MHz 的计数时钟脉冲时，如果在一个信号周期内有 678 个计数时钟脉冲通过，则该信号的周期就是 678 微秒。

三、实验设备

序 号	名 称	数 量	备 注
1	个人计算机	1 台	
2	TEC-8 实验系统	1 台	

3	双踪示波器	1 台	
4	万用表	1 只	

四、实验要求

- (1) 使用频率为 1MHz 的主时钟产生一个时间为 1 秒的时间闸门，用于对被测信号进行计数，计数时间为 1 秒，静态显示时间为 1 秒。计数过程中对计数器的值进行显示。
- (2) 对 TEC-8 试验台上的下列频率的信号测试其频率：MF、100KHz、10KHz、1KHz、100Hz、10Hz。
- (3) 用模式开关 SWB、SWA 选择被测信号，选择标准如下：

序号	SWB	SWA	被测信号	信号意义
1	0	0	MF	既作为主时钟，又作为被测信号。
2	0	1	CP1	DZ3:100KHz, DZ4:10KHz; DZ3 和 DZ4 不允许同时短接。
3	1	0	CP2	DZ5:1KHz, DZ6:100Hz 信号; DZ5 和 DZ6 不允许同时短接。
4	1	1	CP3	DZ7: 10Hz; DZ7 和 DZ8 不允许同时短接。

- (4) 频率计采用自顶向下或者自底向上的层次方法设计。
- (5) 在个人计算机 PC 上 Quartus II 用 VHDL 语言设计出程序，经过编译，然后下载到 EPM7128 器件中，构成一个频率计。
- (6) 观察实验结果。如果不符合要求，重新修改程序、编译后下载，直到正确为止。

五、实验提示

1. EPM7128 引脚信号

本实验中使用下列信号，对应 EPM7128 的引脚如下：

序号	信号名	管脚号	方向	信 号 意 义
1	MF	55	In	主时钟及被测信号
2	CP1	56	in	被测信号，频率为 100KHz 或者为 10KHz
3	CP2	57	in	被测信号，频率为 1KHz 或者 100Hz
4	CP3	58	in	被测信号，频率为 10Hz
5	CLR#	1	in	复位信号，低电平有效
6	SWA	4	in	选择被测信号
7	SWB	5	in	选择被测信号
8	LG1-D0	44	out	数码管 LG1 的驱动信号
9	LG1-D1	45	out	数码管 LG1 的驱动信号
10	LG1-D2	46	out	数码管 LG1 的驱动信号
11	LG1-D3	48	out	数码管 LG1 的驱动信号
12	LG1-D4	49	out	数码管 LG1 的驱动信号
13	LG1-D5	50	out	数码管 LG1 的驱动信号
14	LG1-D6	51	out	数码管 LG1 的驱动信号
15	LG1-D7	52	out	数码管 LG1 的驱动信号
16	LG2-A	37	out	数码管 LG2 的驱动信号
17	LG2-B	39	out	数码管 LG2 的驱动信号
18	LG2-C	40	out	数码管 LG2 的驱动信号
19	LG2-D	41	out	数码管 LG2 的驱动信号
20	LG3-A	35	out	数码管 LG3 的驱动信号
21	LG2-B	36	out	数码管 LG3 的驱动信号
22	LG3-C	17	out	数码管 LG3 的驱动信号
23	LG3-D	18	out	数码管 LG3 的驱动信号
24	LG4-A	30	out	数码管 LG4 的驱动信号
25	LG4-B	31	out	数码管 LG4 的驱动信号
26	LG4-C	33	out	数码管 LG4 的驱动信号
27	LG4-D	34	out	数码管 LG4 的驱动信号
28	LG5-A	25	out	数码管 LG5 的驱动信号
29	LG5-B	27	out	数码管 LG5 的驱动信号

30	LG5-C	28	out	数码管 LG5 的驱动信号
31	LG5-D	29	out	数码管 LG5 的驱动信号
32	LG6-A	20	out	数码管 LG6 的驱动信号
33	LG6-B	21	out	数码管 LG6 的驱动信号
34	LG6-C	22	out	数码管 LG6 的驱动信号
35	LG6-D	24	out	数码管 LG6 的驱动信号

对于上述信号中的 MF、CP1、CP2、CP3、LG1-D7~LG1-D0 需要用扁平电缆将 EPM7128 的引脚和 TEC-8 实验台上的对应信号进行连接。将扁平电缆的 34 芯端插到插座 J6 上，将扁平电缆的 12 芯端插到插座 J1 上，将扁平电缆的 6 芯端插到插座 J5 上。**注意：扁平电缆进行插接或者拔出必须在关电源后进行。**另外，做实验时，将 DZ2 左侧短接为连通，右侧短接为短开，以使数码管正极接到+5V 上；实验结束后，将短路子 DZ2 右侧短接，即为断开。

2. 异步十进制计数器

对被测信号的频率进行计数的计数器必须是十进制的计数器而不能是十六进制的计数器。一个典型的异步十进制计数器具有如下形式：

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity counter10A is port
    (clr:   in std_logic;
     clk:   in std_logic;
     enable: in std_logic;
     c_out: out std_logic;
     cnt:   buffer std_logic_vector(3 downto 0));
end counter10A;

architecture behav of counter10A is
begin
    process(clr,clk,cnt,enable)
    begin
        if clr = '0' then
            cnt <= "0000";
            c_out <= '0';
        elsif clk'event and clk = '1' then
            if enable = '1' then
                if cnt = "1001" then      cnt <= "0000" ;c_out <= '1';
                    else                  cnt <= cnt + '1';c_out <= '0';
                end if;
            end if;
        end if;
    end process;
end behav;
```

在上面的程序中，当复位信号 `clr` 为 0 时，十进制计数器 `cnt` 复位为 “0000”，进位信号 `c_out` 复位为 0。`enable` 信号是时间闸门，`clk` 是被测信号。在 `enable` 信号为 1 时，允许在时钟信号 `clk` 上升沿计数。当计数到 “1001”（十进制 9）时，下一个时钟脉冲 `clk` 的上升沿重新回到 “0000”（十进制 0），周而复始计数。信号 `c_out` 只有在 `cnt10` 的值为 “0000” 的时间段内为 1。在异步计数器中，低位计数器产生的 `c_out` 作为高位计数器的时钟信号。

3. 十进制计数器数码管的驱动

十进制计数器的输出除了个位之外，其他十进制位的 4 位二进制输出直接与数码管的驱动电路连接。十进制计数器的个位由于相应的数码管是按每个发光二极管驱动，因此必须进行转换。转换使用 `case` 语句，例如：

```
library ieee;
use ieee.std_logic_1164.all;

entity display is port
    (counter1: in std_logic_vector(3 downto 0);
     a,b,c,d,e,f,g,h: out std_logic );
end display;

architecture behave of display is
    signal s_out: std_logic_vector(7 downto 0);
begin
    a <= s_out(0);
    b <= s_out(1);
    c <= s_out(2);
    d <= s_out(3);
    e <= s_out(4);
    f <= s_out(5);
    g <= s_out(6);
    h <= s_out(7);

    process(counter1)
    begin
        case counter1 is
            when "0000" => --显示数字 0
                s_out <= "00111111";
            when "0001" => --显示数字 1
                s_out <= "00000110";
            when "0010" => --显示数字 2
                s_out <= "01011011";
            when "0011" => --显示数字 3
                s_out <= "01001111";
            when "0100" => --显示数字 4
                s_out <= "01100110";
            when "0101" => --显示数字 5
                s_out <= "01101101";
            when "0110" => --显示数字 6
                s_out <= "01111110";
            when "0111" => --显示数字 7
```

```

        s_out <= "00000111";
    when "1000" => --显示数字 8
        s_out <= "01111111";
    when "1001" => --显示数字 9
        s_out <= "01101111";
    when others =>
        s_out <= "00000000";
    end case;
end process;
end behave;

```

4. 如果只进行演示性实验，则可以直接把频率计文件夹中的.sof 文件下载到 EPM7128 中进行。

六、实验完成后，必须要将硬连线控制器重新装入 EPM7128 中。以备以后做其他实验使用。

A large, empty rectangular frame with a thin black border, occupying the majority of the page below the header. The frame is intended for a drawing or illustration related to the text above.

4.3 简易交通灯实验

4.3.1 简易交通灯实验

一、实验目的

- (1) 学习状态机设计。
- (2) 掌握数字逻辑系统的设计方法。
- (3) 掌握 EDA 软件 Quartus II 的基本使用方法。
- (4) 掌握用 VHDL 语言设计复杂数字电路的方法。

二、实验原理

交通灯控制是一种常见的控制，几乎在每个十字路口上都可以看到交通灯。本实验通过南北和东西 2 个方向上的 12 个指示灯 (4 个黄灯、4 个红灯、4 个绿灯) 模拟路口的交通灯控制情况。TEC-8 实验台上的交通灯电路如图 4.2 所示。

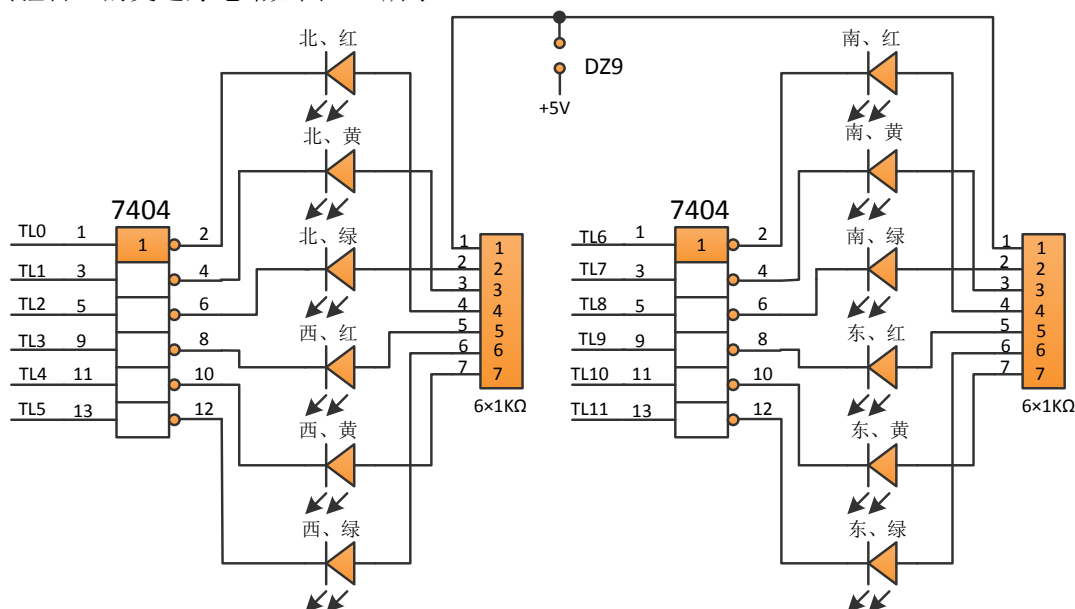


图 4.2 交通灯实验电路图

12 个发光二极管代表 12 个交通灯。2 个 7 引脚的排电阻向 12 个发光二极管提供电流。排电阻的引脚 1 为公共端，它和排电阻其他引脚之间的电阻值为 1K 欧姆。当短路子 DZ9 断开时，两个排电阻的引脚 1 悬空；当短路子 DZ3 短接时，两个排电阻的引脚 1 接 +5V，通过排电阻给 12 个发光二极管供电。控制信号 TL0~TL11 分别控制各发光二极管的负极。由于 2 个 7404 器件对控制信号 TL0~TL11 反相后接到发光二极管的负极，因此当 TL0~TL11 中的某一个信号为 1 时，对应的发光二极管有电流流过而被点亮。只要对信号 TL0~TL11 进行合适的控制，就能使 12 个发光二极管按要求亮、灭。

三、实验设备

序 号	名 称	数 量	备 注
1	个人计算机	1 台	
2	TEC-8 实验系统	1 台	
3	双踪示波器	1 台	
4	万用表	1 只	

四、实验任务

模拟十字路口交通灯的运行情况，完成下列功能。

- (1) 按下复位按钮 CLR 后，进入(2)。
- (2) 南、北方向的 2 个绿灯亮，允许车辆通行；东、西方向的 2 个红灯亮，禁止车辆通行。时间 10 秒。
- (3) 南、北的 2 个黄灯闪烁，已经过了停车线的车辆继续通行，没有过停车线的车辆停止通行；东、西方向的 2 个红灯亮，禁止车辆通行。时间 2 秒。
- (4) 南、北方向 2 个红灯亮，禁止车辆通行；东、西方向 2 个绿灯亮，允许车辆通行。时间 10 秒。
- (5) 南、北方向 2 个红灯亮，禁止车辆通行；东、西 2 个黄灯闪烁，已经过了停车线的车辆继续通行，没有过停车线的车辆停止通行。时间 2 秒。
- (6) 返回(2)，继续运行。
- (7) 如果在(2)状态情况下，按一次紧急按钮 QD，立即结束(2)状态，进入(3)状态，以使东、西方向车辆尽快通行。如果在(4)状态情况下，按一次紧急按钮，立即结束(4)状态，进入(5)状态，以使南、北方向车辆尽快通行。

五、实验提示

1. EPM7128 器件引脚信号

本实验中使用的信号对应的 EPM7128 引脚如下：

序号	信号名	信号属性	引脚号	信 号 说 明
1	MF	55	in	频率 1MHz 的时钟
2	CLR#	1	in	复位信号，低电平有效
3	QD	60	in	按下启动按钮 QD 后产生的 QD 脉冲
4	TL0	20	out	控制北方红灯
5	TL1	21	out	控制北方黄灯
6	TL2	22	out	控制北方绿灯
7	TL3	24	out	控制西方绿灯
8	TL4	25	out	控制西方黄灯
9	TL5	27	out	控制西方红灯
10	TL6	28	out	控制南方绿灯
11	TL7	29	out	控制南方黄灯
12	TL8	30	out	控制南方红灯
13	TL9	31	out	控制东方红灯
14	TL10	33	out	控制东方黄灯
15	TL11	34	out	控制东方绿灯

由于信号 MF、QD 的 EPM7128 引脚和实验台上的相应信号没有直接连接，因此在实验时首先要将扁平电缆的 34 芯端插到插座 J6 上，将扁平电缆的 6 芯端插到插座 J5 上，将 DZ9 左侧短接为连通，右侧短接为短开。**注意：扁平电缆进行插接或者拔出必须在关电源后进行。**

2. 状态机

本实验是典型的状态机实验。首先从 1MHz 的 MF 时钟信号经过 5 次 10 分频产生 0.1 秒的计数时钟，用以对计数器计数，用计数器的值控制状态机状态之间的转换。当按下一次 QD 按钮时，直接修改计数器的值，使状态转换提前产生。

本实验中有的状态机有 4 个状态，分别对应四中的(2)~(5)。状态机的示例程序如下：

```
state_p: process(clr, clk)
begin
    if clr = '0' then
        state <= "00";
    elsif clk'event and clk = '1' then    --clk 为 0.1 秒的时钟信号
        state <= next_state;
    end if;
end process;

state_trans: process(clk, state, count)
begin
    case state is
        when "00" =>    --0 状态
            if count = "1100011" then    --10 秒到了吗?
                next_state <= "01";    --转到 1 状态。
            else
                next_state <= "00";    --继续 0 状态。
            end if;

        when "01" =>    --1 状态
            if count = "1110111" then    --12 秒到了吗?
                next_state <= "11";    --转到 2 状态。
            else
                next_state <= "01";    --继续 1 状态。
            end if;

        when "11" =>    --2 状态
            if count = "1100011" then    --10 秒到了吗?
                next_state <= "10";    --转到 3 状态。
            else
                next_state <= "11";    --继续 2 状态。
            end if;

        when "10" =>    --3 状态
            if count = "1110111" then    --12 秒到了吗?
                next_state <= "00";    --转到 0 状态
            else
                next_state <= "10";    --继续 3 状态。
            end if;
```

```
        end case;  
    end process;
```

上述的状态机由 2 个 PROCESS 语句构成。第一个 PROCESS 语句给出了状态机中每 0.1 秒用 next_state 代替 state；第 2 个 PROCESS 语句根据时间控制 next_state 的产生。

2. 黄灯闪烁可通过连续亮 0.2 秒、灭 0.2 秒实现。
 3. 本实验中短路子 DZ9 需要左侧短接。实验完毕后，短路子 DZ9 右侧短界。
 4. 如果只进行演示性实验，则可以直接把交通灯文件夹中的 .sof 文件下载到 EPM7128 中进行。
- 六、实验完成后，必须要将硬连线控制器重新装入 EPM7128 中。以备以后做其他实验使用。**

A large, empty rectangular frame with a thin black border, occupying the majority of the page below the header. The frame is intended for a drawing or illustration related to the text above.

4.4 VGA 接口设计

4.4.1 VGA 接口设计部分

一、实验目的

- (1) 学习 VGA 接口的工作原理和在显示器上显示某种特定图形的方法。
- (2) 掌握数字逻辑系统的设计方法。
- (3) 掌握 EDA 软件 Quartus II 的基本使用方法。
- (4) 掌握用 VHDL 语言设计复杂数字电路的方法。

二、实验原理

1. VGA 接口

VGA 彩色显示器 (640×480/60Hz) 显示过程中所必需的信号, 除 R、G、B 三基色信号外, 行同步 HS 和场同步 VS 也是非常重要的两个信号。在显示器显示过程中, HS 和 VS 的极性可正可负, 显示器内可自动转换为正极性逻辑。

现以正极性为例, 说明 CRT 的工作过程: R、G、B 为正极性信号, 即高电平有效。当 VS=0, HS=0, CRT 显示的内容为亮的过程, 即正向扫描过程约为 26μs, 当一行扫描完毕, 行同步 HS=1, 约需 6μs; 其间, CRT 扫描产生消隐, 电子束回到 CRT 左边下一行的起始位置 (X=0, Y=1); 当扫描完 480 行后, CRT 的场同步 VS=1, 产生场同步使扫描线回到 CRT 的第一行第一列 (X=0, Y=0) 处 (约为两个行周期), HS 和 VS 的时序如图 4.3 所示。

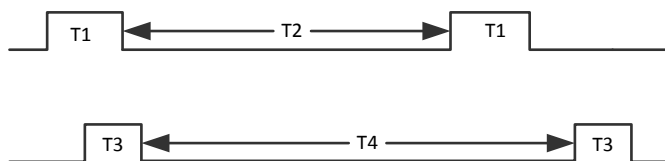


图 4.3 HS 和 VS 时序图

在图 4.3 中, T1 为行同步消隐 (约为 6μs); T2 为行显示时间 (约为 26μs); T3 为场同步消隐 (两行周期); T4 为场显示时间 (480 行周期)。

表 4.2 是各种颜色的编码表。

表 4.2 颜色编码表

颜色	黑	黄	红	品红	绿	青	黄	白
R	0	0	0	0	1	1	1	1
G	0	0	1	1	0	0	1	1
B	0	1	0	1	0	1	0	1

2. VGA 接口驱动

TEC-8 实验系统中, 对 VGA 接口的驱动如图 4.4 所示。

本实验中使用的信号对应的 EPM7128 引脚如下：

信号名	信号属性	引脚号	信号说明
VGA-R	out	34	VGA 接口的红色信号
VGA-G	out	35	VGA 接口的绿色信号
VGA-B	out	36	VGA 接口的蓝色信号
VGA-H	out	37	VGA 接口的行同步信号
VGA-V	out	39	VGA 接口的场同步信号
MF	In	55	频率为 1MHz 的主时钟信号
QD	in	60	模式计数器时钟，按 QD 按钮后产生，高电平有效
CLR#	in	1	复位信号，按 CLR 按钮后产生，低电平有效

由于信号 MF、QD 的 EPM7128 引脚和实验台上的相应信号没有直接连接，因此在实验时首先要将扁平电缆的 34 芯端插到插座 J6 上，将扁平电缆的 6 芯端插到插座 J5 上。**注意：扁平电缆进行插接或者拔出必须在关电源后进行。**

2. 主时钟 MF 的频率是 1MHz，因此很容易通过计数的办法产生 26 μ S 和 6 μ S 左右的时间长度。由于每台显示器参数上略有差别，实验时需要根据显示器的参数调整时间长度。
3. 可以使用行同步脉冲作为行计数器的计数时钟。
4. 把 26 μ S 时间段分为 8 个小时时间段，在每个小时时间段内向 VGA 接口输出一个固定的 VGA-R、VGA-G、VGA-B 值，就会在显示器上显示出希望的竖彩条。
5. 将 480 行分为 8 部分，在每一部分向 VGA 接口输出一个固定的 VGA-R、VGA-G、VGA-B 值，就会在显示器上显示出希望的横彩条。
6. 将竖彩条和横彩条异或，就能得到颜色方格。
7. 如果只进行演示性实验，则可以直接把 VGA 文件夹中的 .sof 文件下载到 EPM7128 中进行。

六、实验完成后，必须要将硬连线控制器重新装入 EPM7128 中。以备以后做其他实验使用。

[illegible]

第 5 章：Quartus II 软件简易使用

5.1 软件安装：

安装 Quartus II 软件，步骤如下：

第一步：打开试验箱所带光盘，双击 Quartus II 文件夹目录下 Setup,单击 Next 按钮；

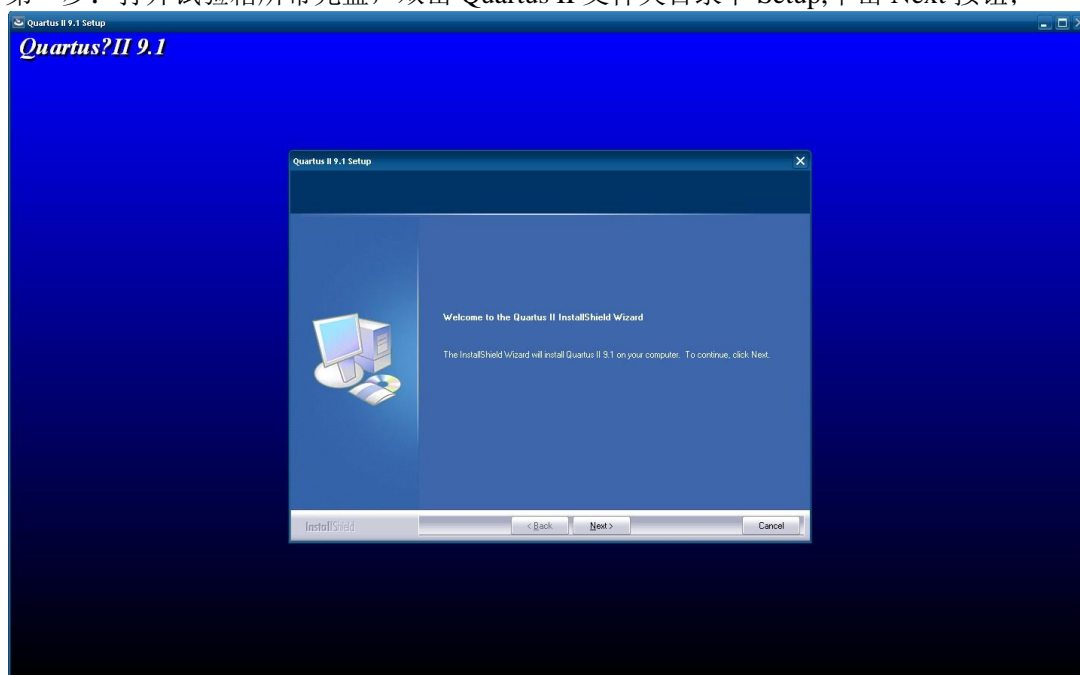


图 6.1

第二步：选择 I accept the terms of this license... 选项，并单击 Next 按钮；

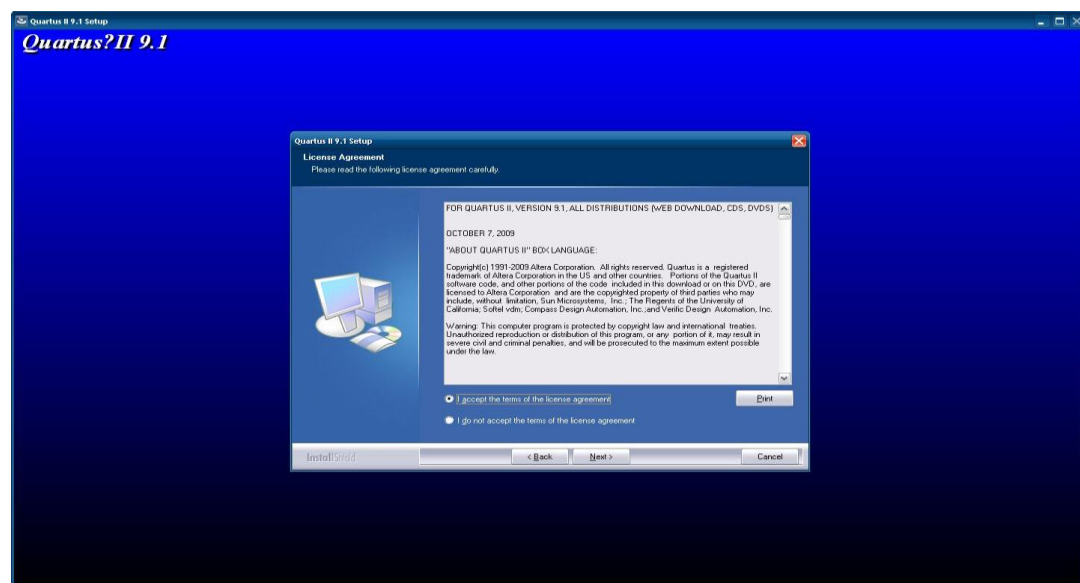


图 6.2

第三步：填写用户名和公司名称后，点击 Next 按钮；

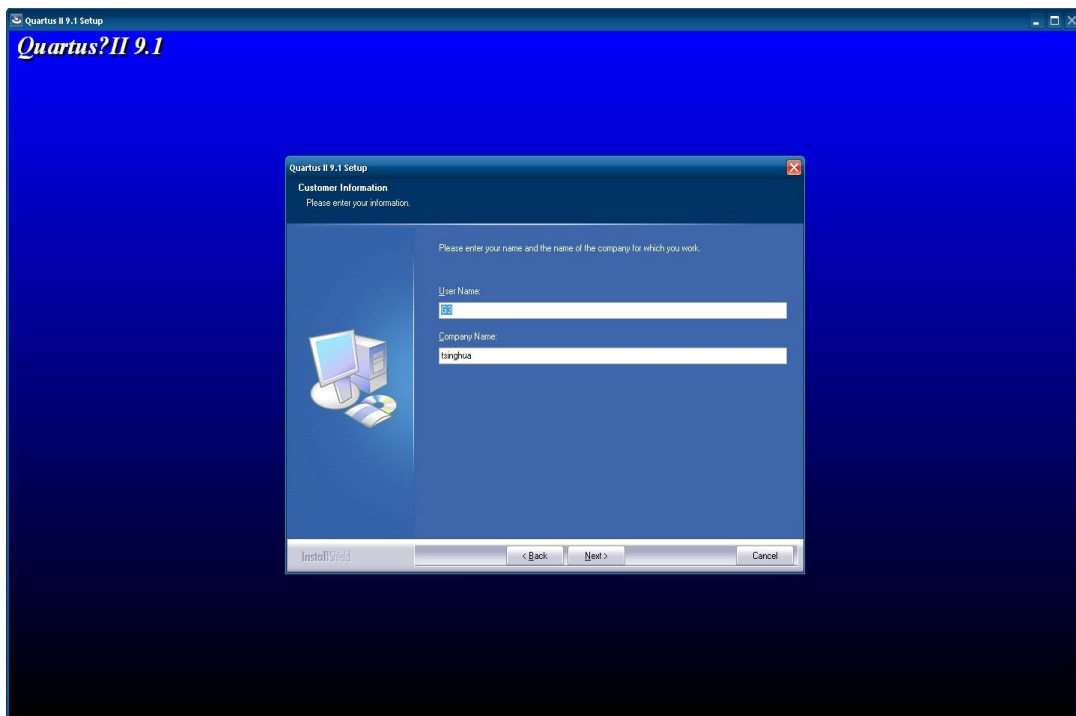


图 6.3

第四步：默认路径为：C:\altera\91,单击 Next 按钮；

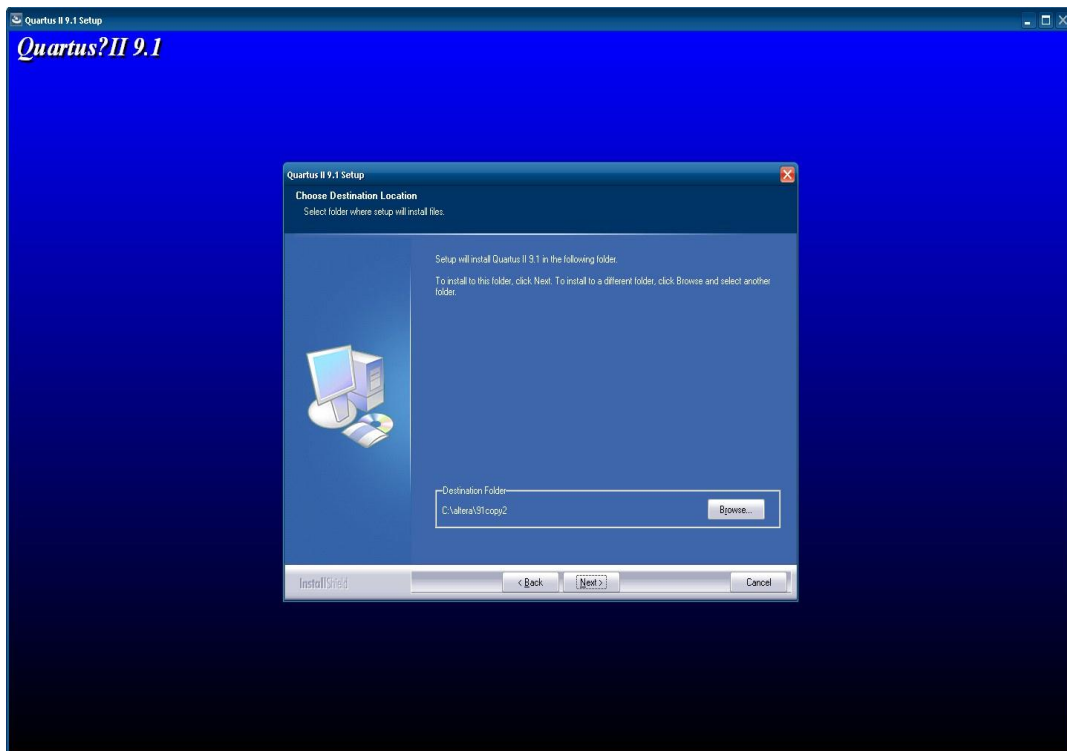


图 6.4

第五步：点击 Next 按钮；

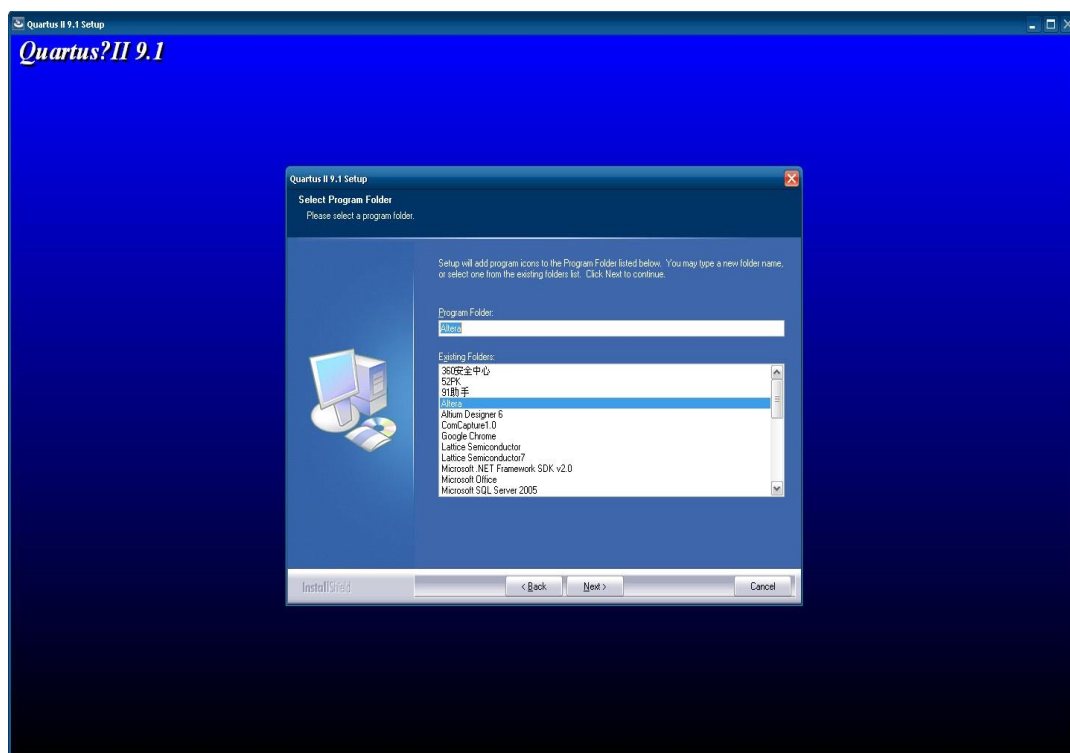


图 6.5

第六步：点击 Next 按钮；连续几次；点击 Finish 按钮，完成安装。

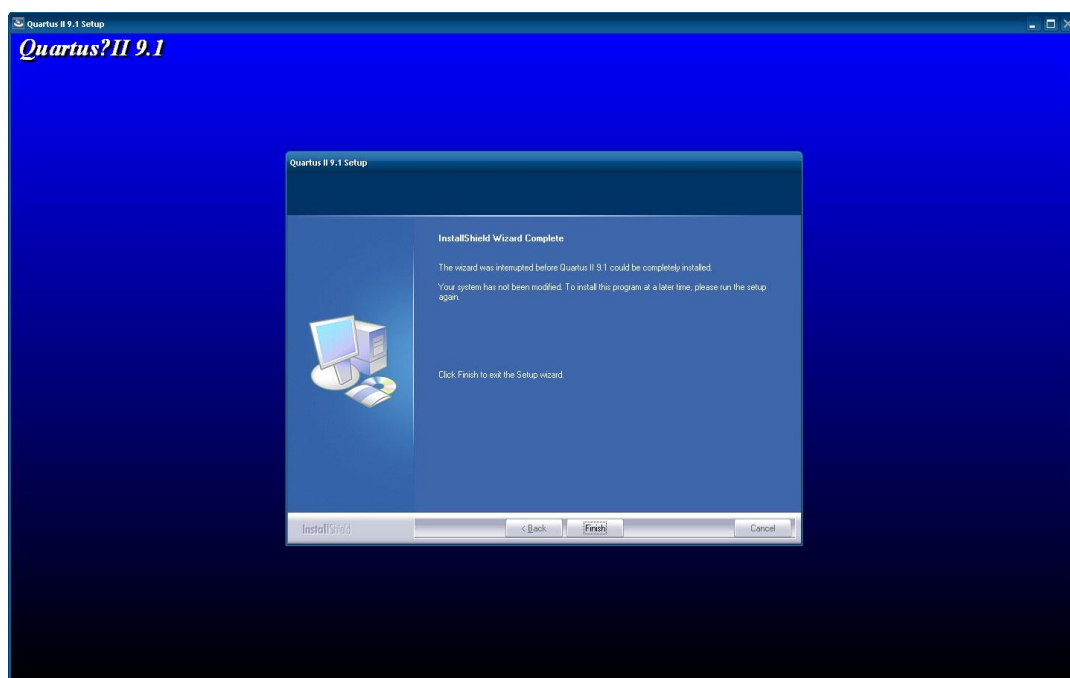


图 6.6

5.2 实验步骤

1)使用 Quartus II 建立工程

Quartus II 是以工程(project)的方式对设计过程进行管理的, Quartus II 工程中存放创建 FPGA 配置文件所需要的所有设置和设计文件。因此, 每个开发过程开始时都应建立一个 Quartus II 工程。

(1)选择“开始”→“程序”→Altera→Quartus II 9.1 或单击桌面快捷方式打开 Quartus II 软件, 软件界面如图 6.7 所示。

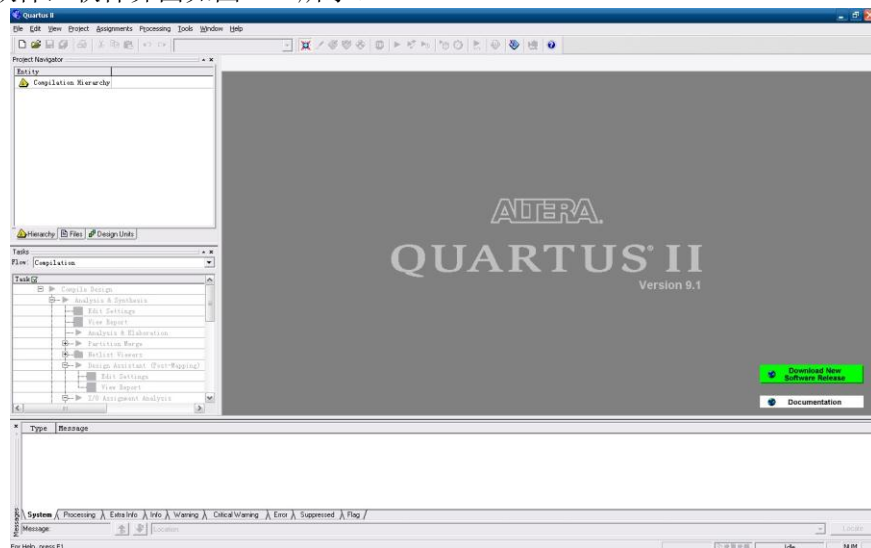


图 6.7 Quartus 软件图形用户界面

(2)在图 6.7 中选择 File → New Project Wizard 新建一项工程。新建工程向导对话框如图 6.8 所示。在该对话框中需要设置:

- ①工程文件夹保存目录。
- ②工程名称。
- ③工程顶层文件设计实体名称。

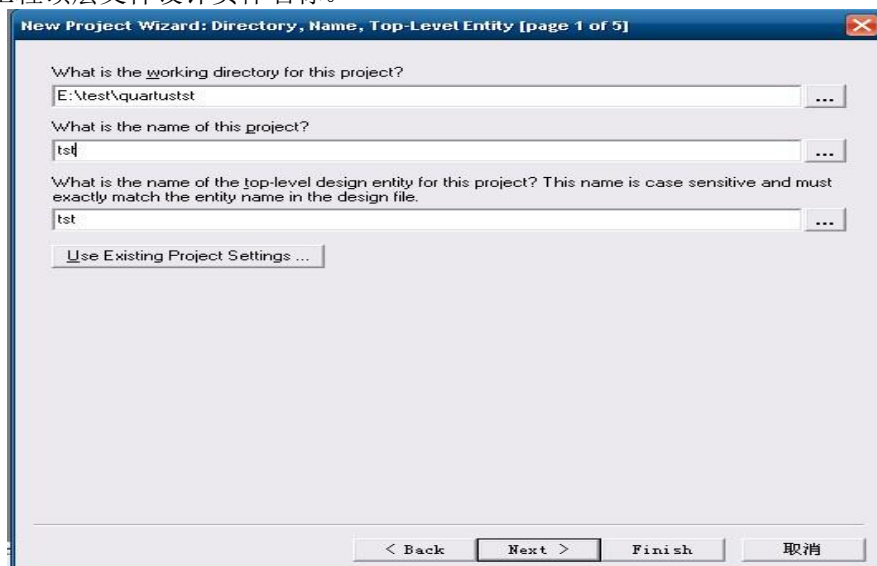


图 6.8 新建文本输入对话框

(3)单击 NEXT 进入图 6.9 所示对话框, 由于是新建工程, 暂无输入文件, 直接单击 NEXT。

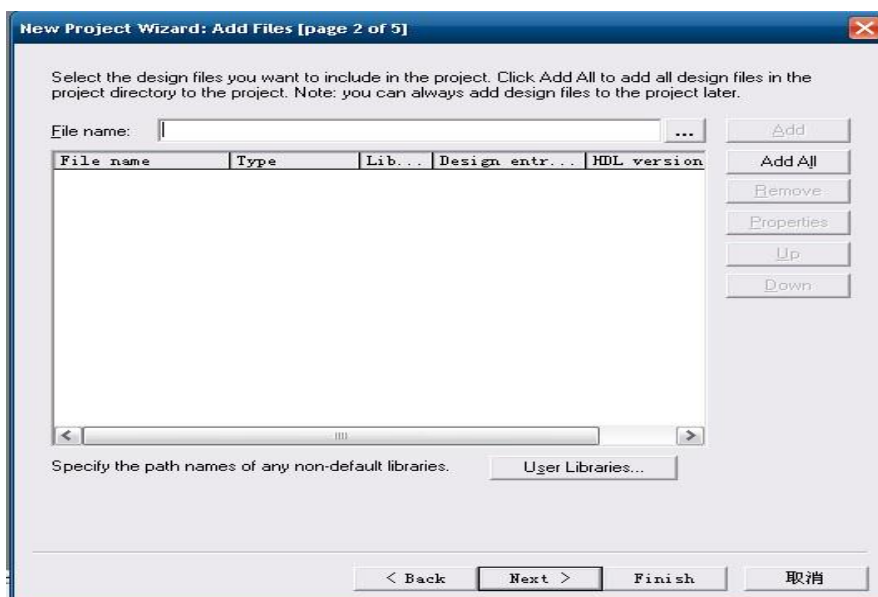


图 6.9 【New project wizard】对话框

(4) 弹出图 6.10 所示选择器件对话框。这里我们选择 TEC-8 核心板上所用的 Cyclone 系列的 EPM7128SLC84-15。

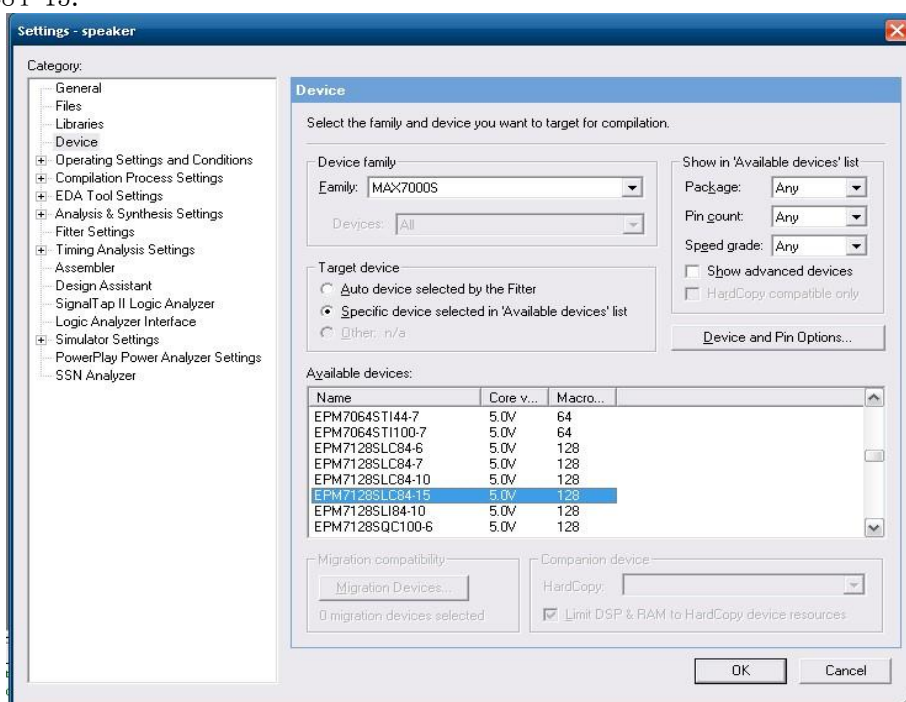


图 6.10 【Settings】对话框

(5) 指定完器件后，单击 NEXT，进入图 9 所示对话框，本实验利用 Quartus II 集成开发环境进行开发，不使用任何第三方 EDA 工具，因此这里不作任何改动。

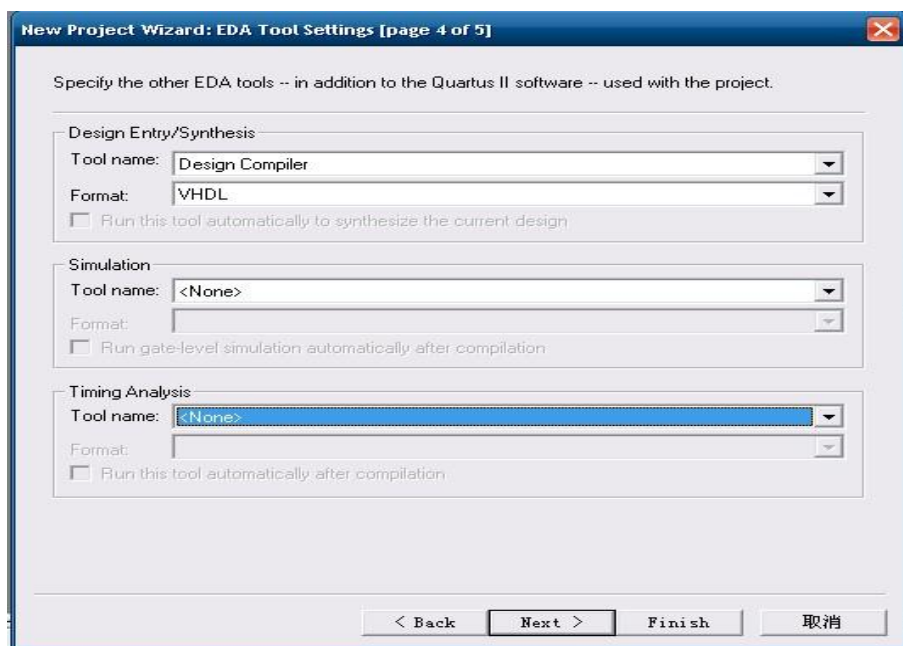


图 6.11

(6) 在图 6.11 中单击 NEXT，设计者可以看到工程文件的配置信息报告。单击 FINISH 完成新工程的建立。

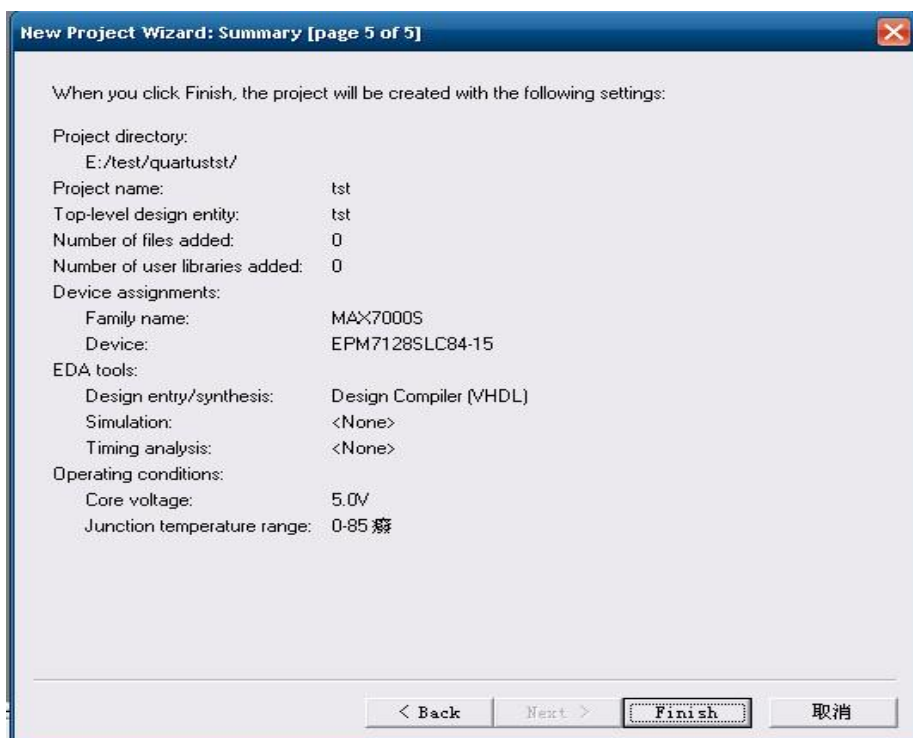


图 6.12

2) 使用 Quartus II 建立设计文件

单击 FILE → NEW, 弹出图 6.13 所示对话框。在该对话框中选择输入文件类型，在本书所列试验中均选择 VHDL File。此时主界面中出现设计文件输入窗口。

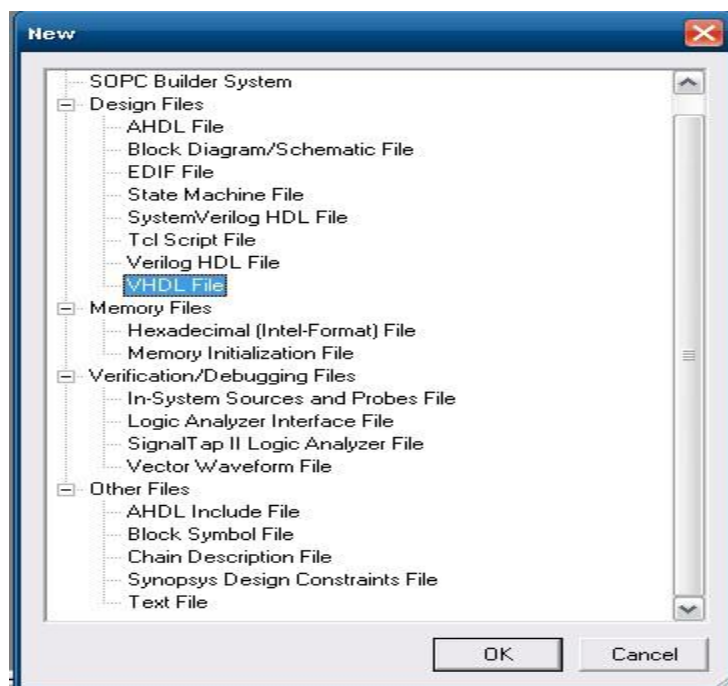


图 6.13

在该窗口中输入设计文件如图 6.13 输入结束选择 FILE → NEW 保存。

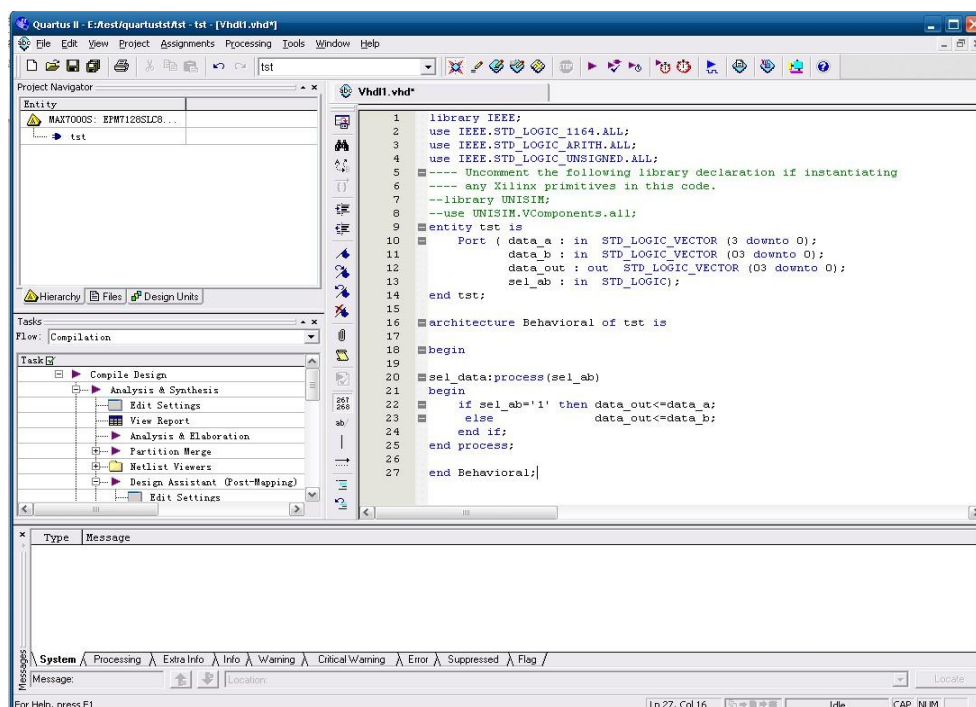


图 6.14

3)分配 FPGA 引脚

选择菜单 Assignments → Pins 打开引脚分配对话框如图 13 所示。在 TO 栏中填入与设计文件中输入输出端口名称相对应的各引脚名称，在 Location 栏中选择相应的引脚，也可以在 Location 下直接输入引脚号来快速定位。(具体引脚号参考各实验说明)选择 FILE → Save 来保存分配结果，然后关闭 Assignment Editor。

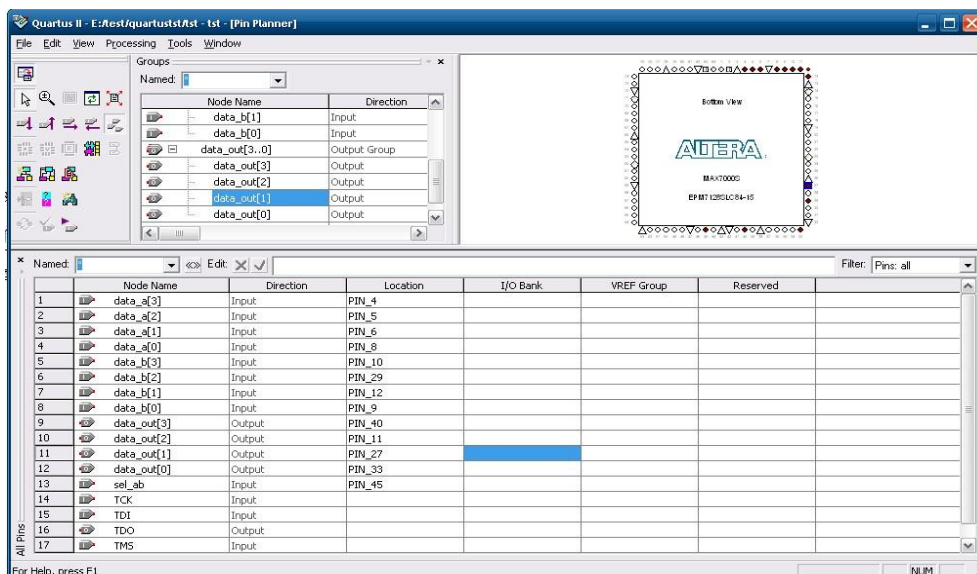


图 6.15 管脚分配对话框

4)编译整个工程

在编译过程中，编译器定位并处理所有工程文件，生成与编译相关的消息与报告，创建 SOF 文件及任何可选配置文件。

选择 Processing → Start Compilation 进行全程编译。在编译过程中，状态窗口显示整个编译进程及每个编译阶段所用时间。编译结果显示在 Compilation Report 窗口中。编译结束后，对话框显示消息 “Full compilation was successful”，单击 OK 确定。

5)下载硬件设计到目标 FPGA 芯片

(1) JTAG(下载后缀为 sof 的 FPGA 器件配置文件)

通过 ByteBlaster II 下载电缆连接试验箱 USB 口和计算机，接通试验箱电源。

在 Quartus II 软件中选择 Tools → Programmer，打开编程器窗口，并自动打开配置文件 sof，如图 15 所示。单击 sof 文件的 Program configure 方框选中下载文件。

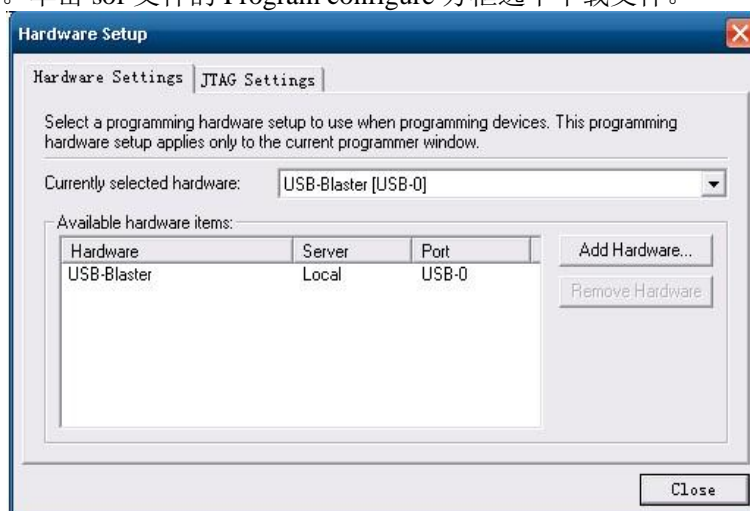


图 6.16

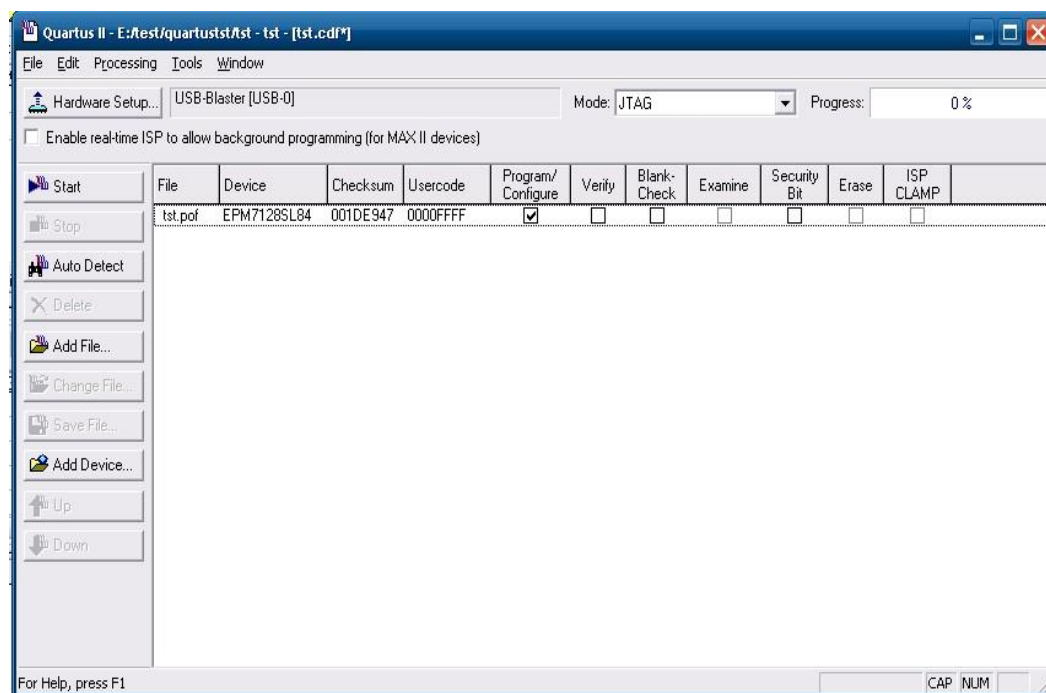


图 6.17

单击 **Start** 开始下载。进度条完成 100% 则完成下载。