第一部分

ICETEK DSP 教学实验箱说明书

(ICETEK-VC5416-USB-EDU 或 ICETEK-VC5416-PP-EDU)

第一章 ICETEK DSP 教学实验箱简介

ICETEK DSP 教学实验箱是由北京瑞泰创新技术有限责任公司于 2003 年推出的新一代 DSP 教学产品。它面向广大 DSP 初学者，提供 DSP 教学的一体化设备，为 DSP 教学提供整体解决方案，它还为 DSP 设备的研制者提供了一个较为完备的测试平台。尤其适用于开设 DSP 教学课程的学校和各类初学者。

一、ICETEK DSP 教学实验箱的特点：

1、完备性：提供完整的 DSP 实验环境。硬件上包括 DSP 仿真器、评估板、信号源、控制模块；软件上提供仿真软件、完全使用手册和实验例程。可以进行与 DSP 应用相关的大部分实验和测试。

2、易用性：完备的使用说明和实验手册使使用者可以轻松上手、尽快熟悉 DSP 使用的相关操作,多功能的控制模块提供从图象到声音、从输入到输出多种形象直观的显示、控制手段，使用户的知识得到感性的结果，从而加深对 DSP 的理解。

3、直观性：提供液晶图象显示、发光二级管阵列显示、电机指示等视觉实验效果，信号源也提供了容易控制、简单明了的测试手段，使实验现象能更加直观、具体、明确地展示出来。

4、灵活性：支持使用 ICETEK5100PP 和 ICETEK5100USB 仿真器；在接口相同的前提下，支持多种系列的评估板，如：ICETEK2407-A 板、ICETEK5416-A 板、ICETEKVC33-AE 板、ICETEK6711-A 板和已经或即将推出的多种评估板。各模块更换操作简单、安装容易，可以适用各种教学需求。

5、适用性：

针对 DSP 能同时进行多路信号处理的特点；提供两个独立的信号源，可单独设置，四路波形输出，充分测试 DSP 的并行数字信号处理能力；

针对 DSP 实验经常需要做 A/D、D/A 实验的特点，实验箱将 DSP 评估板上相关接口引出，在底板上扩展成专用插座供实验者联接使用，实验箱具备同时输入四路 A/D 和四路输出 D/A 的能力，结合信号源，可同时进行四路 A/D、D/A 实验；

对于 DSP 评估板输入、输出能力不足的特点，利用显示/控制模块提供充足的输入、输出手段；

针对实验箱的使用者，提供完备、容易上手的学习、实验资料，而且设计了循序渐进的、丰富完整的实验，同时还为教师提供指导材料和教材。

二、ICETEK DSP 教学实验箱的功能：

1、两个独立的信号发生器，可同时提供两种波形、四路输出；信号的波形、频率、幅度可调；

2、多种直流电源输出。支持对仿真器和评估板的直流电源连接插座；

3、显示输出：液晶图象显示器（LCD），可显示从 DSP 发送来的数据；发光二级管阵列(LEDArray)；发光二级管；马达指针 0-360 度指示；

4、音频输出：可由 DSP I/O 脚控制的蜂鸣器；D/A 输出提供音频插座，可直接接插耳机；

5、键盘输入：可由 DSP 回读扫描码；

6、步进电机：四相步进电机，可由 DSP I/O 端口控制旋转和方向、速度；

7、底板提供插座，可使用插座完成 DSP 评估板上的 A/D 信号输入和 D/A 输出；

8、软件资料：相关 DSP 设计编程使用教材、实验教程、使用说明、实验程序。

三、ICETEK DSP 教学实验箱的组成：

如图所示，ICETEK DSP 教学实验箱由以下几个部分组成。

1、箱盖：保护实验箱设备；保存教材、使用手册、实验指导书、各种实验用的连线；可拆卸，在实验中可从箱体上拆下；带锁，可在关闭时用钥匙锁住。

2、箱体：装载实验箱设备；左侧外壁上有一个标准外接电源线插孔；通过固定螺丝与实验箱底板连为一体。

3、底板：固定各模块；提供电源开关、实验用直流电源插座(5)、A/D D/A 输入输出插座(16)、各模块直流供电插座(5)、信号插座(4)、信号源(2)；实现显示控制模块和 DSP 评估板模块的信号互连。

4、信号源：两组、四路输出，可使用专门开关启动；提供切换选择输出方波、三角波和正弦波，可选择输出频率范围(10Hz-100Hz，100Hz-1KHz，1KHz-10KHz，10KHz-100KHz)，可进行频率和幅度(0-3.3V)的微调。

5、仿真器模块：固定 ICETEK 仿真器，支持 PP 型和 USB 型；提供 PP 型仿真器供电 5V 电源插座；仿真器可从底板上拆下更换。

6、显示控制模块(可选)：通过信号线连接到底板；从底板提供的 5V 和 12V 直流电源插座输入电源；提供液晶图形显示(128x64 象素)，发光二极管阵列显示(8x8 点)，指示灯，四相步进电机，键盘(16 按键)，蜂鸣器。显示控制模块可从底板上拆下更换。

7、DSP 评估板模块：固定各种 DSP 评估板；提供 5V 直流电源插座(两个位置)；34Pin 信号线插座(3 个)和 36Pin 信号线插座，用于连接 DSP 评估板和实验箱底板。DSP 评估板模块可从底板上拆下更换。

四、ICETEK DSP 教学实验箱性能指标：

1、**直流电源：**+5V(5A)，-5V(0.5A)，+12V(1A)，-12V(0.5A)，+3.3V(3A)，**地**

2、**信号源**(A、B)：

-**双路输出**

-**频率范围：分为** 4 **段**(10Hz—100Hz，100Hz—1KHz，1KHz—10KHz，10KHz—100KHz)，**可通过拨动开关进行选择**

-**频率微调：在每个频率段范围内进行频率调整**

-**波形切换：提供** 3 **种波形(方波，三角波，正弦波)，可通过拨动开关进行选择**

-**幅值微调：**0—3.3V **平滑调整**

3、**信号接插孔：**4 **路** A/D **输入**(ADCIN0-ADCIN3)，4 **路** D/A **输出**(DACOUT1—DACOUT4)，**每路均提供信号和地**

4、**显示/控制模块(可选)：**

-**液晶显示**(LCD)：128×64 **点阵图形显示屏，可调整显示对比度**

-**发光二极管显示阵列：**8×8 **点阵**

-**发光二极管**

-**蜂鸣器**

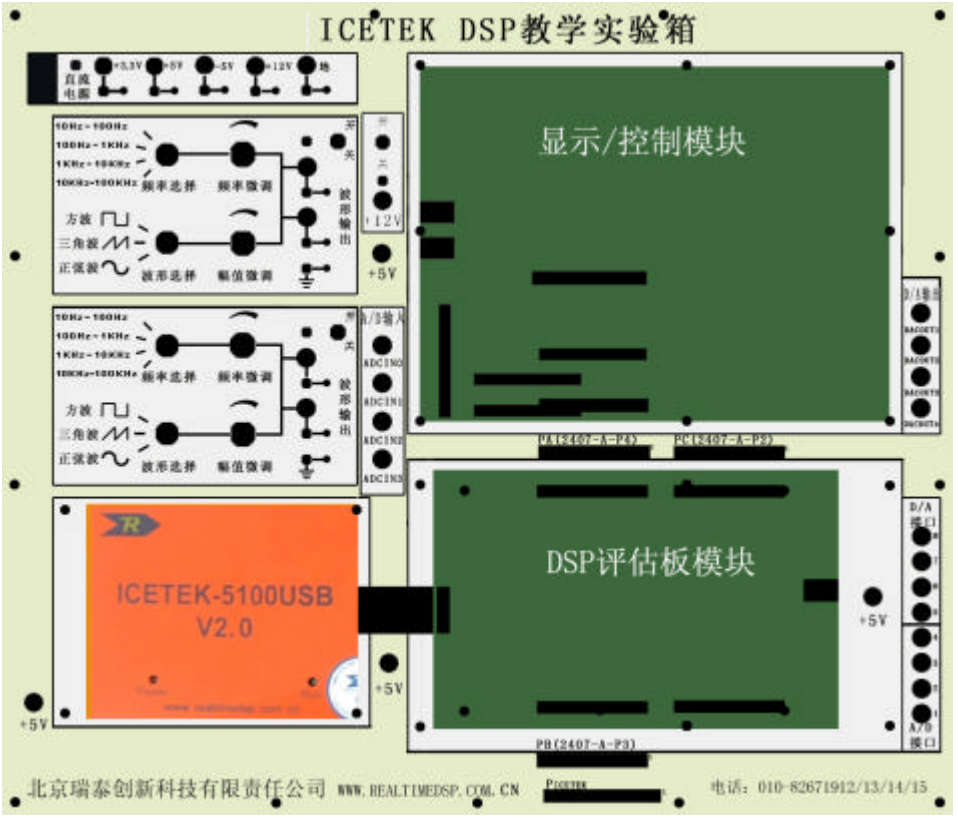
-**步进电机：四相八拍，步距角** 5.625，**起动频率≥**300PPS，**运行频率≥**900PPS

-**键盘：**4×4 **按键**

-**拨动开关**(DIP)：4 **路，可实现复位和设置** DSP **应用板参数**

5、**电源输入：**220V **交流**

五、ICETEK DSP 教学实验箱结构图：



第二章　ICETEK DSP 教学实验箱硬件接口和编程说明

一、CETEK DSP 教学实验箱的外围接口：

-外围接口 PA：

5416-A 扩展接口 P4(其中连接到实验箱底板的引脚加下划线，其他未连接)

1	Vcc	2	Vcc
3	$\overline{DS}$	4	$\overline{PS}$
5	$\overline{IS}$	6	
7	$\overline{IOWE}$	8	$\overline{IORD}$
9	$\overline{MSTRB}$	10	$R / \overline{W}$
11	$\overline{T\_RDY}$	12	$\overline{IOSTRB}$
13	$\overline{RESET}$	14	$\overline{EXT\_RESET}$
15	$\overline{NMI}$	16	$\overline{INT}$ 1
17	**GND**	18	**GND**
19	$\overline{INT}$ 2	20	$\overline{INT}$ 3
21	**BDR0**	22	BDR1
23	**BDX0**	24	**BDX1**
25	**BFSR0**	26	**BFSR1**
27	**BFSX0**	28	**BFSX2**
29	**BCLKXR0**	30	**BCLKXR1**
31		32	**CLKOUT**
33	**GND**	34	**GND**

-外围接口 PB：

5416-A 扩展接口 P3(其中连接到实验箱底板的引脚加下划线，其他未连接)

1	**A0**	2	**A1**
3	**A2**	4	**A3**
5	A4	6	A5
7	A6	8	A7
9	A8	10	A9
11	A10	12	A11
13	A12	14	**A13**
15	**A14**	16	**A15**
17	**GND**	18	**GND**
19	**IOD0**	20	**IOD1**
21	**IOD2**	22	**IOD3**
23	**IOD4**	24	**IOD5**
25	**IOD6**	26	**IOD7**
27	D8	28	D9
29	D10	30	D11
31	D12	32	D13
33	D14	34	D15

-外围接口 PC：

5416-A 扩展接口 P2(其中连接到实验箱底板的引脚加下划线，其他未连接)

1 VccA	2 VccA
3	4
5 **ADCIN1**	6 **ADCIN2**
7 ADCIN3	8 ADCIN4
9 ADCIN5	10 ADCIN6
11	12
13	14
15	16
17 **DAGND**	18 **DAGND**
19	20
21	22
23	24
25 **DACOUT1**	26 **DACOUT2**
27 **DACOUT3**	28 **DACOUT4**
29	30
31	32
33 **DAGND**	34 **DAGND**

二、ICETEK DSP 教学实验箱硬件编程：

(1) 液晶显示模块编程

液晶显示模块的访问、控制是由 5416DSP 对扩展 I/O 接口的操作完成。

控制 I/O 口的寻址：命令控制 I/O 接口的地址为 0x8001，数据控制 I/O 接口的地址为 0x8003 和 0x8004，辅助控制 I/O 接口的地址为 0x8002。

显示控制方法：

-液晶显示模块中有两片显示缓冲存储器，分别对应屏幕显示的象素，向其中写入数值将改变显示，写入“1”则显示一点，写入“0”则不显示。其地址与象素的对应方式如下：

左侧显示内存						右侧显示内存					
Y=	0	1	···	62	63	0	1	···	62	63	行号
X=0	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	0 ↓ 7
↓	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	8 ↓ 55
X=7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	56 ↓ 63

-发送控制命令：向液晶显示模块发送控制命令的方法是通过向命令控制 I/O 接口写入命令控制字，然后再向辅助控制接口写入 0。下面给出的是基本命令字、解释和 C 语言控制语句举例。

.显示开关：0x3f 打开显示；0x3e 关闭显示；

```
port8001=0x3f; port8002=0; //将液晶显示打开
port8001=0x3e; port8002=0; //将液晶显示关闭
```

.设置显示起始行：0x0c0+起始行取值，其中起始行取值为 0 至 63；

```
port8001=0x0c0; port8002=0; //设置液晶显示从存储器第 0 行开始显示
port8001=0x0c8; port8002=0; //设置液晶显示从存储器第 8 行开始显示
```

.设置操作页：0x0b8+页号，其中页号取值为 0-7；

```
port8001=0x0b0; port8002=0; //设置即将操作的存储器第 0 页
port8001=0x0b2; port8002=0; //设置即将操作的存储器第 2 页
```

.设置操作列：0x40+列号，其中列号为取值为 0-63；

```
port8001=0x40; port8002=0; //设置即将操作的存储器第 0 列
port8001=0x44; port8002=0; //设置即将操作的存储器第 4 列
```

-写显示数据：在使用命令控制字选择操作位置(页数、列数)之后，可以将待显示的数据写入液晶显示模块的缓存。将数据发送到相应数据控制 I/O 接口即可。C 语言语句举例说明：

```
port8003=0x80; port8002=0;    //向左侧屏幕缓存存入数据 0x80，如果
                              //显示行、页号和列号均为 0 时，屏幕上
                              //第 8 行第 1 列将显示黑色象素
port8004=0x01; port8002=0;    //向右侧屏幕缓存存入数据 1，如果
                              //显示行、页号和列号均为 0 时，屏幕上
                              //第 1 行第 65 列将显示黑色象素
```

(2) 发光二极管编程

显示/控制模块上的发光二极管是由 DSP 芯片上的通用 I/O 管脚直接控制的，对应于 5416 的 BFSX0，将 BFSX0 设置成通用 I/O 管脚后，将其置成高使发光二极管灭，置成低发光二极管亮。

(3) 发光二极管显示阵列编程

发光二极管显示阵列的显示是由 I/O 扩展端口控制，DSP 须将显示的图形按列的顺序存储起来(8×8 点阵，8 个字节，高位在上方，低位在下方)，然后定时刷新控制显示。具体方法是，将以下控制字按先后顺序，每两个为一组发送到端口 0x8005，发送完毕后，隔不太长的时间(以人眼观察不闪烁的时间间隔)再发送一遍。由于位值为“0”时点亮，所以需要将显示的数据取反。

0x01,第 1 列数据取反，0x02,第 2 列数据取反，
0x04,第 3 列数据取反，0x08,第 4 列数据取反，
0x10,第 5 列数据取反，0x20,第 6 列数据取反，
0x40,第 7 列数据取反，0x80,第 8 列数据取反，

(4) 步进电机编程

步进电机是由 DSP 通用 I/O 管脚输出直接控制。步进电机的起动频率大于 500PPS(拍每秒)，空载运行频率大于 900PPS。5416 的通用 I/O 口 BFSR1 控制电

机的转动频率，BCLKXR0 控制转动方向。

控制的方法是使用 DSP 通用定时器设置 BFSR1 以一定的频率改变高低状态，输出方波，设置 BCLKXR0 为高电平则顺时针转动，低电平为逆时针转动。

(5) 蜂鸣器编程

蜂鸣器由 DSP 通用 I/O 管脚输出控制，可将此管脚上的频率输出转换成声音输出。5416 的通用 I/O 口 BDX0 控制蜂鸣器的输出频率。

控制的方法是使用 DSP 通用定时器设置 BDX0 以一定的频率改变高低状态，输出方波。

(6) 键盘输入编程

键盘的扫描码由 DSP 的 I/O 扩展地址 0x8001 给出，当有键盘输入时，读此端口得到扫描码，当无键被按下时读此端口的结果为 0。各按键的扫描码排列如下所示。

0x18,0x14,0x12,0x11
0x28,0x24,0x22,0x21
0x48,0x44,0x42,0x41
0x88,0x84,0x82,0x81

第三章 ICETEK-VC5416-A 硬件用户手册

本文件描述 ICETEK-VC5416-A 型 DSP 用户板的外部接口、外部扩展资源的定义和其它相关的硬件接口的信息，本文件分为以下三个部分：

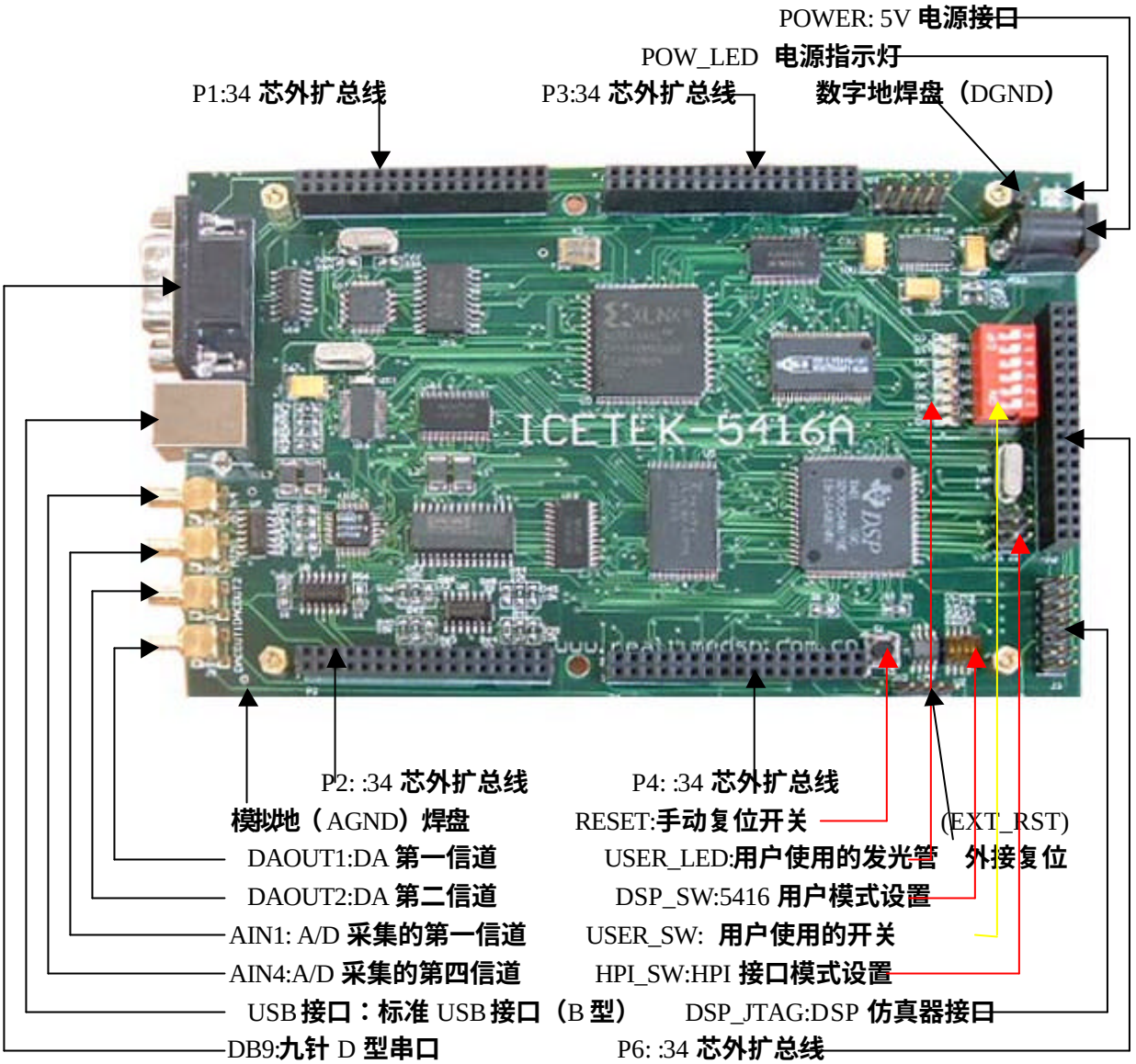
一、 用户板的外围接口

二、 用户板的外部扩展寄存器定义和功能

一、用户板的外围接口

下面是 ICETEK-VC5416-A 板的实物图，我们将通过此图描述整个板子的外部接口。

ICETEK-VC5416-A 板的实物图



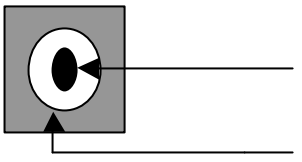
下面，我们将详细说明这些接口的功能和特征定义。首先，表一归纳总结了这些跳线和功能分类.

功能分类	接口名称	接口定义
电源接口	POWER	5V 电源输入
外设接口	DAOUT1	DA 第一信道
	DAOUT2	DA 第二信道
	AIN1	A/D 采集的第一信道
	AIN4	A/D 采集的第四信道
	DB9	九针 D 型串口
	USB 接口	标准 USB 接口（B 型）
总线接口	P1	34 芯外扩总线
	P2	34 芯外扩总线
	P3	34 芯外扩总线
	P4	34 芯外扩总线
	P6	34 芯外扩总线
指示灯	POW_LED	电源指示灯
	USER_LED	用户使用的发光管
辅助接口	DSP_JTAG	:DSP 仿真器接口
	AGND	模拟地（AGND）焊盘
	DGND	数字地（DGND）焊盘
跳线和开关	EXT_RST	外接复位
	DSP_SW	5416 用户模式设置
	HPI_SW	HPI 接口模式设置
	USER_SW	用户使用的开关
	RESET	手动复位开关

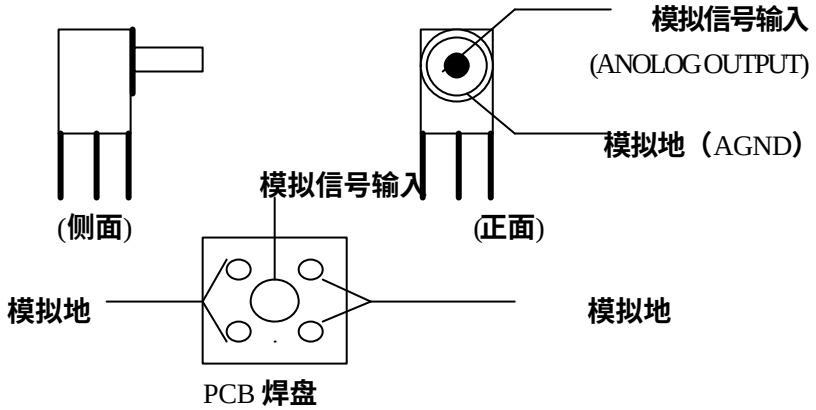
表一：接口和功能分类

下面将分别介绍这些接口：

1．POWER 这个接口用于接入为整个板子供电的电源，电源电压为+5V,标准配置的电源电流为 1A，如果不使用随板提供的电源，请注意电源的正负极性和电流的大小。下面是这个接口的插孔示意图：



2．DAOUT1:街头规格为 SMB-JWE 高频插头,接头是 ICETEK-VC5416-A 板的 DA 通道 1 的输出接口，接口输出 0 到 +5V 的电压。下面是接口的示意图：

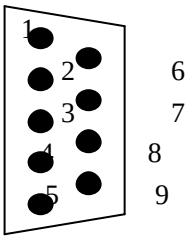


3．DAOUT2:街头规格为 SMB-JWE 高频插头,接头是 ICETEK-VC5416-A 板的 DA 通道 2 的输出接口，接口输出 0 到 +5V 的电压。连接器的规格同 1。

4．AIN1：模拟输入(ANOLOG INPUT)通道 1，采集速率最大为 500KHz，输入模拟电压为 0 到+5V。连接器规格同 1。

5．AIN2：模拟输入(ANOLOG INPUT)通道 2，采集速率最大为 500KHz，输入模拟电压为 0 到+5V。连接器规格同 1。

6．DB9:9 针 D 型连接器，异步串口连接器，符合 RS-232 规范，输出电平为正负 12V. 下面是 9 针连接器的管脚定义：



管脚号	管脚定义	说明
1	NC	无连接
2	TxD	数据输出引脚，与对方的输入脚连接
3	RxD	数据输入引脚，与对方的输出脚连接
4	NC	无连接
5	GND	共地端
6	NC	无连接
7	NC	无连接
8	NC	无连接
9	NC	无连接

DB9 管脚定义表

7．USB 接口：评估板支持 USB 通讯，在通讯时，评估板将作为计算机的一个外设使用，同时，在板子上接出一个 USB 接口（B 型）。
为了方便用户调试，我们接出一套扩展总线，以便于用户扩展。包括 P1、P2、P3、P4 和 P6 在内，这些外部扩展的引脚可以方便的用于对评估板进行扩展，下面是这些接口的定义和特点。

8．P1:34 芯扩展总线接口。P1 接口主要是扩展评估板上空闲的 DSP 外设引脚，以便于定制用户的硬件环境。注意：由于这组引脚是直接来自于 5416 DSP 芯片，因此，这些引脚为 TTL 3.3V 标准，其输出最高电压为 3.3V。如果要接入 5V 器件，请在外接时注意电平转换。（在扩展板上 使用 3.3-5V 兼容器件与扩展接口连接）。

下面是 P1 的管脚定义和说明：

管脚号	管脚名	说明
1	+5V 电源	由 POWER 提供的+5V 电源
2	+5V 电源	由 POWER 提供的+5V 电源
3	TOUT	5416 的定时器输出引脚
4	A16	高位地址总线，5416 程序空间的扩展地址总线
5	BDR2	5416 的 MCBSP2 同名引脚
6	A17	高位地址总线，5416 程序空间的扩展地址总线
7	BDX2	5416 的 MCBSP2 同名引脚
8	A18	高位地址总线，5416 程序空间的扩展地址总线
9	BFSR2	5416 的 MCBSP2 同名引脚
10	A19	高位地址总线，5416 程序空间的扩展地址总线
11	BFSX2	5416 的 MCBSP2 同名引脚
12	A20	高位地址总线，5416 程序空间的扩展地址总线
13	BCLKR2	5416 的 MCBSP2 同名引脚
14	A21	高位地址总线，5416 程序空间的扩展地址总线
15	BCLKX2	5416 的 MCBSP2 同名引脚
16	A22	高位地址总线，5416 程序空间的扩展地址总线
17	GND	地线
18	GND	地线
19	XF	5416 的同名引脚
20	BIO	5416 的同名引脚
21	NC	保留
22	NC	保留
23	NC	保留
24	NC	保留
25	NC	保留
26	NC	保留
27	NC	保留
28	NC	保留
29	NC	保留
30	NC	保留
31	NC	保留
32	NC	保留
33	GND	地线
34	GND	地线

P1:管脚定义和说明

9． P2:34 芯扩展总线接口。P2 接口主要是 AD 和 DA 接口，P2 中扩展了所有的 AD 和 DA 引脚，包括 AIN1、AIN4、DAOUT1、DAOUT2 和没有连接器连接的 AD 和 DA 引脚。请注意评估板的对采集信号的要求

下面是 P1 的管脚定义和说明：

管脚号	名称	说明
1	VCCA	模拟电源+5V
2	VCCA	模拟电源+5V
3	NC	保留
4	NC	保留
5	AIN1	模拟输入通道 1（与 AIN1 接口相同使用时可以选择其中一个）
6	AIN2	模拟输入通道 2
7	AIN3	模拟输入通道 3
8	AIN4	模拟输入通道 4（与 AIN1 接口相同使用时可以选择其中一个）
9	AIN5	模拟输入通道 5
10	AIN6	模拟输入通道 6
11	NC	保留
12	NC	保留
13	NC	保留
14	NC	保留
15	NC	保留
16	NC	保留
17	AGND	模拟地
18	AGND	模拟地
19	NC	保留
20	NC	保留
21	NC	保留
22	NC	保留
23	NC	保留
24	NC	保留
25	DAOUT1	模拟输出通道 1
26	DAOUT2	模拟输出通道 2
27	DAOUT3	模拟输出通道 3
28	DAOUT4	模拟输出通道 4
29	NC	保留
30	NC	保留
31	NC	保留
32	XINT2	外部中断 2
33	AGND	模拟地
34	AGND	模拟地

P2:管脚定义和说明

10．P3:34 芯扩展总线接口。P3 接口是外扩的 5416 总线，包含 16 根地址线和 16 根数据线，当程序访问 I/O 空间 8000h-0FFFFh 地址时这个总线被打开，可以用于读入和输出并行的数据。这个总线是受软件控制的，且加入了电平转换和三态控制。注意：这个插座上的地址线是由 5416 芯片提供的，如果您在外部扩展的话，请注意 5416 的地址线只能输出 3.3V 的电平。下面是 P3 的管脚定义和说明：

管脚号	名称	说明
1	A0	5416 地址线 A0
2	A1	5416 地址线 A1
3	A2	5416 地址线 A2
4	A3	5416 地址线 A3
5	A4	5416 地址线 A4
6	A5	5416 地址线 A5
7	A6	5416 地址线 A6
8	A7	5416 地址线 A7
9	A8	5416 地址线 A8
10	A9	5416 地址线 A9
11	A10	5416 地址线 A10
12	A11	5416 地址线 A11
13	A12	5416 地址线 A12
14	A13	5416 地址线 A13
15	A14	5416 地址线 A14
16	A15	5416 地址线 A15
17	GND	数字地
18	GND	数字地
19	D0	5416 数据线 D0，双向总线，当访问 I/O 空间 8000h-0FFFFh 地址时，总线有效。
20	D1	5416 数据线 D1
21	D2	5416 数据线 D2
22	D3	5416 数据线 D3
23	D4	5416 数据线 D4
24	D5	5416 数据线 D5
25	D6	5416 数据线 D6
26	D7	5416 数据线 D7
27	D8	5416 数据线 D8
28	D9	5416 数据线 D9
29	D10	5416 数据线 D10
30	D11	5416 数据线 D11
31	D12	5416 数据线 D12
32	D13	5416 数据线 D13
33	D14	5416 数据线 D14
34	D15	5416 数据线 D15

P3:管脚定义和说明

11．P4:：34 芯扩展扩展总线接口。P4:是 5416 的功能引脚和外设引脚，包括 5416 外部扩展总线的控制线、McBSP 接口线、外部中断和外部复位等重要的引脚信号。注意：这里的引脚都是由 DSP 直接引出的，在和外部设备连接时注意电平转换。

管脚号	管脚定义	管脚说明
1	VCC	+5V 电源
2	VCC	+5V 电源
3	DS	数据存储空间片选信号
4	PS	程序存储空间片选信号
5	IS	I/O 空间片选信号
6	NC	保留
7	IOWE	I/O 空间写信号，仅在 8000-0FFFFh 地址空间有效，此信号经过译码，可以直接作为外部扩展时的写信号
8	IORD	I/O 空间读线号，仅在 8000-0FFFFh 地址空间有效，此信号经过译码，可以直接作为外部扩展时的读信号
9	MSTRB	程序/数据空间总线操作信号
10	RW	5416 的读写信号
11	NC	保留
12	IOSTRB	I/O 空间总线操作引脚
13	RESET	DSP 评估板输出的复位信号
14	EXT_RESET	外部输入 DSP 评估板的复位信号
15	NMI	不可屏蔽中断
16	INT1	外部中断 1
17	GND	数字地
18	GND	数字地
19	INT2	外部中断 2
20	INT3	外部中断 3
21	BDR0	McBSP0 的数据接收脚
22	BDR1	McBSP1 的数据接收脚
23	BDX0	McBSP0 的数据发送脚
24	BDX1	McBSP1 的数据发送脚
25	BFSR0	McBSP0 的数据接收帧同步脚
26	BFSR1	McBSP1 的数据接收帧同步脚
27	BFSX0	McBSP0 的数据发送帧同步脚
28	BFSX1	McBSP1 的数据发送帧同步脚
29	BCLKXR0	McBSP0 的数据接收/发送时钟脚
30	BCLKXR1	McBSP1 的数据接收/发送时钟脚
31	NC	保留
32	CLKOUT	5416 的时钟输出
33	GND	数字地
34	GND	数字地

P4:管脚定义和说明

12．P6:34 芯片外扩展总线接口。P6 的功能是针对 HPI 接口的。HPI 接口是 5416DSP 芯片的特有接口，它允许外部的处理器使用 5416 片内的存储器作为缓冲来实现数据交换。由于它的接口引脚与 5416 的片外扩展总线独立，因此使用这个接口可以提高 5416 的总线吞吐能力，有关的信息，请参考用户手册（TMS320C54x DSP Enhanced Peripherals spru302.pdf）.注意，注意：这里的引脚都是由 DSP 直接引出的，在和外部设备连接时注意电平转换。

管脚号	管脚定义	管脚说明
1	HD0	由于 HPI 的工作方式比较特殊，这里不做说明，所有引脚信号均与 DSP 芯片上的管脚同名，关于 HPI 的详细说明,请参考（TMS320C54x DSP Enhanced Peripherals spru302.pdf）
2	HD1	
3	HD2	
4	HD3	
5	HD4	
6	HD5	
7	HD6	
8	HD7	
9	NC	保留
10	NC	保留
11	NC	保留
12	NC	保留
13	NC	保留
14	NC	保留
15	NC	保留
16	NC	保留
17	GND	数字地
18	GND	数字地
19	HCS	HPI 管脚
20	HCNTL0	
21	NC	保留
22	HCNTL1	HPI 管脚
23	HRW	
24	HDS2	
25	HRDY	
26	HDS1	
27	HINT	
28	HAS	
29	HBIL	
30	NC	保留
31	NC	保留
32	NC	保留
33	NC	保留
34	NC	保留

P4:管脚定义和说明

13．POW_LED（D1,D2）：电源指示灯，如果评估板工作正常，此灯常亮。其中指示灯 D1 为 5V 指示，若外接电源工作正常，此灯常亮。指示灯 D2 为+3.3V 指示，若评估板供电芯片工作正常，此灯常亮。

14．USER_LED(LED0..LED7):用户指示灯，在板上有 8 个可编程的指示灯，分别为 LED0..LED7，这 8 个指示灯的开关由 5416 编程控制。

15．DSP_JTAG：5416 的仿真接口，由于连接 ICETEK-5100 系列的仿真器或兼容产品。注意，使用的仿真器必须支持 3.3V 仿真。

16．AGND：模拟地焊盘，此处可作为模拟信号测量的基准地信号

17．DGND：数字地焊盘，此处可作为数字信号测量的基准地信号

18．EXT_RST(J10,J9)：外接复位信号和看门狗允许信号。J9:看门狗允许信号，当此跳线短路时，看门狗启动工作，当此跳线断开时，看门狗禁止工作。J10:如果需要外接控制 DSP 的复位信号，可以在此处连接，当两个信号连通时 DSP 处于复位状态，当两个信号分开时，DSP 正常工作。

19．DSP_SW:5416 芯片的配置开关。共有四位，如下表：（具体含义参见 TMS320VC5416 Datasheet）：

信号名	信号功能	信号定义
MP/MC	处理器方式选择	断开状态，即 OFF 状态，为高电平；连接状态，即 ON 状态，为低电平
CLKMD1	5416 硬件倍频选择。	断开状态，即 OFF 状态，为高电平；连接状态，即 ON 状态，为低电平
CLKMD2		
CLKMD3		

20．HPI_SW（J1,J2）:HPI 接口方式选择。这两个接口用于控制 5416 的 HPI16 和 HPIENA 信号的状态。下面列表如下：

跳线名	状态	含义
J1:HPI16（丝印 J1 处为 1 脚）	1,2,3 断开	HPI16 悬空
	12 短接	HPI16 高电平
	23 短接	HPI16 低电平
J2:HPIENA（丝印 J2 处为 1 脚）	1,2,3 断开	HPIENA 悬空
	12 短接	HPIENA 高电平
	23 短接	HPIENA 低电平

21．USER_SW（SW0..SW6）：7 个用户开关输入。可以用作 DSP 的输入信号。软件可以读取它的状态。当开关处于断开状态，即 OFF 状态时，开关输出高电平，DSP 读到逻辑“1”；而当开关处于连通状态，即 ON 状态时，开关输出低电平，DSP 读到逻辑“0”。

22．RESET:手动复位开关。

二、用户板的外部扩展资源的定义和功能

这部分介绍 5416 评估板的软件编程接口，说明板上外设的扩展寄存器。下面是评估板的外部扩展寄存器的说明：

分类	名称	地址（I/O 空间）	说明
存储器控制寄存器	Port0	0000h@IO	存储器控制寄存器
看门狗	Port1	0001h@IO	看门狗控制寄存器
A/D	Port2	0002h@IO	A/D 数据寄存器
	Port3	0003h@IO	A/D 通道选择寄存器
	Port4	0004h@IO	A/D 采集控制寄存器
D/A	Port1000	1000h@IO	DA 通道 1 数据寄存器
	Port1001	1001h@IO	DA 通道 2 数据寄存器
	Port1002	1002h@IO	DA 通道 3 数据寄存器
	Port1003	1003h@IO	DA 通道 4 数据寄存器
	Port1004	1004h@IO	DA 控制寄存器
串口控制寄存器	Port2000	2000h@IO	RBR/THR/DLL
	Port2001	2001h@IO	IER/DLM
	Port2002	2002h@IO	IIR/FCR
	Port2003	2003h@IO	LCR
	Port2004	2004h@IO	MCR
	Port2005	2005h@IO	LSR
	Port2006	2006h@IO	MSR
	Port2007	2007h@IO	SCR
USER_LED	Port3002	3002h@IO	LED0..LED7 控制寄存器
USER_SW	Port3003	3003h@IO	SW0..SW6 控制寄存器
中断控制寄存器	Port3004	3004h@IO	中断控制寄存器

寄存器说明：

1． 存储器控制寄存器：寄存器在 I/O 空间的地址 0 处，

7	1	0
保留	RAM_ENA	
	W-0	

RAM_ENA：最低位，此位是 1 时，数据存储空间 8000-0FFFFh 扩展为 RAM, 此位是 0 时，数据存储空间 8000-0FFFFh 扩展为 ROM, 复位值是 0。

2、看门狗控制寄存器：寄存器在 I/O 空间的地址 0 处，数据位无效，对这个地址执行写操作将会复位狗电路。

3、中断控制寄存器：寄存器在 I/O 空间的地址 3004h 处，

15	1	0
保留		INT_OE
		W-1

INT_OE 位，此位是外部中断寄存器允许位，当此位置 1 时，禁止所有外部硬件可屏蔽中断，当此位置 0 时，允许所有外部硬件可屏蔽中断。

4、SW0..SW6 控制寄存器：寄存器在 I/O 空间的地址 3003h 处，

15	6	5	0
保留		SWx	
		R-X	

SWx：共 7 位，对应 SW7..SW1 各位，当开关置 ON 时，读入数据 0，反之，为 1。

5、LED0..LED7 控制寄存器：寄存器在 I/O 空间的地址 3002h 处，

15	7	6	0
保留		LEDx	
		W-X	

SWx: 共 6 位，对应 LED7..LED0 各位，写 1 时显示灯亮，反之，灭

第四章　ICETEK DSP 教学实验箱操作手册

一、 CETEK DSP 教学实验箱的拆卸和安装：

注意：首先关闭实验箱总电源、拔掉外接电源线。

1、更换显示/控制模块：

拆卸：拔掉模块左侧的两个供电电源连线；卸除模块四角的固定螺钉；拔出连接的信号线(3 根排线)，这时模块将完全从底板脱离，取下模块。

安装：先连接模块和底板的信号连线(3 根排线)；用固定螺钉将模块固定在底板上；连接底板和模块的相应电源插座，注意+12V(位于上侧)和+5V(位于下侧)不要连错，安装完成。

2、更换 DSP 评估板模块：

拆卸：拔掉模块的供电电源连线；拔掉与仿真器的仿真插头(左侧)；拔掉评估板与底板连接的信号排线；卸除模块四角的固定螺钉，取下模块。

安装：先用固定螺钉将模块固定在底板上；连接仿真器的仿真插头到评估板上的 JTAG 插座；连接信号排线到底板,PA 连接 5416-A 板 P 4，PB 连接 5416-A 板 P 3，PC 连接 5416-A 板 P2；连接评估板与底板的+5V 电源插座；将显示/控制模块上左上角的拨动开关的 1 拨到“OFF”状态，安装完成。

3、更换仿真器模块：

拆卸：对于 PP 型仿真器，拔掉模块的供电电源连线；拔掉连接 DSP 评估板的仿真插头；卸除模块四角的固定螺钉，取下模块。

安装：安装模块四角的固定螺钉，将模块固定到底板上；将仿真插头插到 DSP 评估板上的 JTAG 端口；对于 PP 型仿真器，连接仿真器和底板上相应+5V 电源插座，安装完成。

4、拆卸和安装底板：

拆卸：拆除底板四周的固定螺钉；掀起底板；使用“十”字改锥将电源连线从箱底的开关电源的接线柱上松开拔出，取下底板。

安装 :连接底板上的 5 根电源线到箱底开关电源上的相应接线柱上,用改锥固定；将底板放置妥当后安装四周的固定螺钉，安装完毕。

二、 CETEK DSP 教学实验箱的使用：

1、连接电源：打开实验箱，取出三相电源连接线；将电源线的一端插入实验箱外部左恻箱壁上的电源插孔中；确认实验箱面板上电源总开关(位于实验箱底板左上角)处于“关”的位置；连接电源线的另一端至 220V 交流供电插座上，保证稳固连接。

2、连接各模块电源：确认断开实验箱总电源；使用电源连接线(两端均为带孔的插头)连接显示/控制模块上边插座到实验箱底板上+12V 电源插座；使用电源连接线(两端均为带孔的插头)连接显示/控制模块下边插座到实验箱底板上+5V 电源插座；如使用 PP(并口)型仿真器，则使用电源连接线(两端均为带孔的插头)连接仿真器上插座到实验箱底板上+5V 电源插座 ;使用电源连接线(两端均为带孔的插头)连接 DSP 评估板模块电源插座到实验箱底板上+5V 电源插座。注意各插头要插到底，防止虚接或接触

不良。

3、连接 DSP 评估板信号线：当需要连接信号源输出到 A/D 输入插座时，使用信号连接线(两端均为单声道耳机插头)分别连接相应插座。

4、接通电源：在连接电源线以后，检查各模块供电连线是否正确连接，打开实验箱上的电源总开关(位于实验箱底板左上角)，使开关位于“开”的位置，实验箱底板上的电源指示灯亮。

三、CETEK DSP 教学实验箱使用注意事项：

1、拆卸各模块时请务必将实验箱总电源；

2、不使用显示/控制模块的电机时将相关+12V 电源开关关闭；

3、220V 交流电源线连接须牢靠，勿使发生虚接或接触不良，并保证良好地连接地线；

4、实验箱底板上不同的直流电源不能直接跨接；

5、实验箱底板上直流电源不能直接跨接地线；

6、不要将直接连接电源和信号插座；

7、显示/控制模块上的两个电源插座不要连接错误，上边插座为+12V，下面的为+5V；

8、连接不同类型的插座时，请再三确认无误后进行；

9、不要带电拔插 DSP 评估板模块使用的信号排线；

10、不要带电拔插仿真器和 DSP 评估板上 JTAG 插头的连接电缆；

11、如无特殊情况，请勿打开实验箱底板。

四、ICETEK DSP 教学实验箱故障判断及排除：

1、无法接通电源：请检查外接电缆是否完好；电缆是否与实验箱边插座连接妥当；电缆是否与外接插座连接紧密。

2、信号源没有输出：请确认相应信号源的开关是否打开；检查相应信号源的“幅值微调”旋钮是否处在最小位置；信号连接线是否连接好。

3、显示/控制模块上电机不转：请检查显示/控制模块的+12V 电源是否连接，开关是否打开；摸一下电机指针看是否有振动，如有拨动一下是否能恢复转动。

4、显示/控制模块上液晶没有显示：请调节显示/控制模块上液晶对比度调节电位器 R2。

5、显示/控制模块上发光二极管显示阵列不能正常显示：复位显示/控制模块，将显示/控制模块左上角四路拨动开关的第 4 路拨动到“OFF”的位置再拨回到“ON”；或将实验箱电源重新开关一次；或将显示/控制模块上+5V 电源拔下再重新插好。

第五章　教学实验系统技术指标

（型号：ICETEK-VC5416-USBEDU 或 ICETEK-VC5416-PP-EDU）

一、教学实验箱：ICETEK-EDU-A

1、实验箱自带双信号发生器，产生 10HZ-100K 信号，包括方波，正弦波和三角波；

2、自带电源转换设计，直接输入 220V 电源，分别产生多路+5V 和+3.3V 电源；

3、设计 AD 和 DA 的直接输入输出接口，实验连接方便，并可以连接示波器。

特点：双信号发生器设计，更加贴近 DSP 的实际应用。因为许多实际的情况都是需要对两个信号进行相关分析。

二、通用 DSP 开发系统：ICETEK-5100PP 或 ICETEK-5100USB

1、开发系统支持 C2000/VC33/C5000/C6000 DSP 的开发；

2、并口或 USB 口连接，适合于目前的各种计算机接口。

特点：通用开发系统和 DSP 控制板分离，有利于将来 DSP 的升级。同时，也可以脱离实验箱单独从事科研开发使用。

三、DSP 控制板：ICETEK-VC5416-A（USB）

1、模数转换（A/D）：

精度：12bit

同相位采集路数（最多）：6

设计采集数率（最高）：450K/每路（2 路工作），150K/每路（6 路工作）

信号输入范围：0～5V（特殊输入要求可以为客户定制）

信号耦合方式：直流/交流

2、数模转换（D/A）：

精度：12bit

路数：4

变换数率：100K（10us）

输出范围：0～5V

3、VC5416 周边设计：

主处理器：TMS320VC5416PGE160

内部存贮空间：128K*16bit

外部存贮空间：64K*16bit

看门狗功能设计：有

系统自启动功能设计：8Mbit FLASH

数字 I/O 设计：有

4、上位机接口：

RS232 串行数据接口：有

5、其他：

DSP 扩展总线：有

尺寸：160mm×100mm

特点：采用的 TMS320VC5416PGE160 芯片是目前为止，最快的 VC54XX 系列产品，速度是 VC5410 的 1.6 倍，为 160MIPS。内存是 VC5410 的 2 倍，为 128K。

TMS320VC5416PGE160 与 TMS320VC5410、TMS320VC5402PGE100 的比较如下：

	TMSV320C5416	TMS32VC5410	TMS320VC5402
处理速度	160MIPS	100MIPS	100MIPS
片上RAM	128k*16Bit	64k*16Bit	4k*16Bit

四、通用控制板：ICETEK-CTR

1、液晶：128*64 图形显示

2、键盘：4*4　16 个

3、LED 阵列：8*8

4、发光二极管控制：1 个

5、蜂鸣器：无源，1 个

6、步进电机：12V，四相八拍，>500PPS，1 个

特点：这是一个通用的控制模块，可以和多种 DSP 控制板连接，如 C2000 系列，C5000 系列，VC33 系列，C6000 系列等。

五、配套教材与资料：

1、TMS320C54X 结构原理及应用：北京航空航天大学出版社

2、实验教程与指导书：

3、实验例程（光盘）：

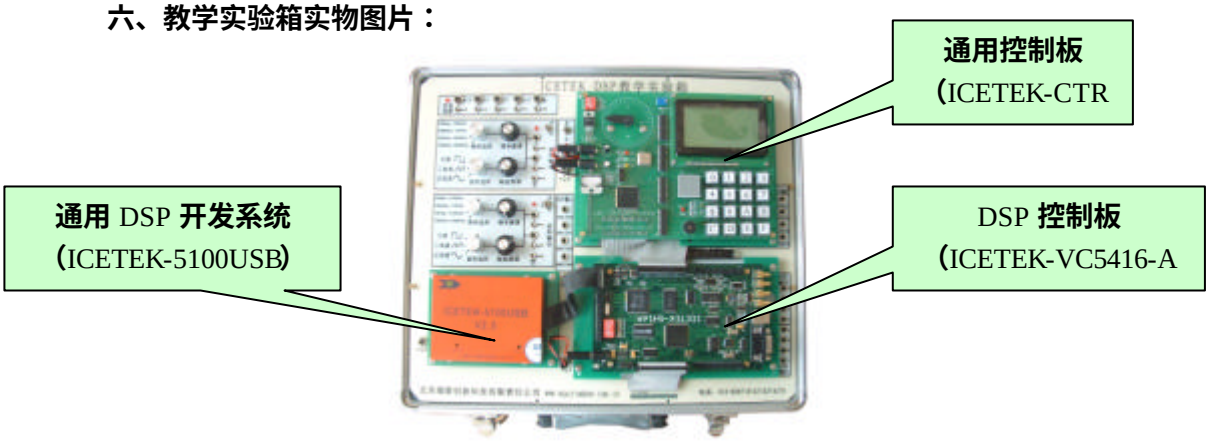
4、教学实验箱：ICETEK-EDU-A 使用说明

5、通用 DSP 开发系统 ICETEK-5100PP 或 ICETEK-5100USB 使用说明

6、DSP 控制板：ICETEK-VC5416-A（USB）使用说明

7、通用控制板：ICETEK-CTR 使用说明

六、教学实验箱实物图片：



ICETEK-VC5416-USB-EDU 教学实验箱

第六章　ICETEK DSP 实验箱实验指导

（型号：ICETEK-VC5416-USB-EDU，ICETEK-VC5416-PP-EDU）

实验一：Code Composer Studio 入门实验

一. 实验目的

1. 掌握 Code Composer Studio 2.0 的安装和配置。
2. 了解 DSP 开发系统和计算机与目标系统的连接方法。
3. 了解 Code Composer Studio 2.0 软件的操作环境和基本功能，了解 TMS320C5xxx 软件开发过程。
   ⑴. 学习创建工程和管理工程的方法。
   ⑵. 了解基本的编译和调试功能。
   ⑶. 学习使用观察窗口。
   ⑷. 了解图形功能的使用。

二. 实验设备

1. PC 兼容机一台；操作系统为 Windows2000 (或 WindowsNT、Windows98、WindowsXP，以下假定操作系统为 Windows2000)。Windows 操作系统的内核如果是 NT 的应安装相应的补丁程序（如：Windows2000 为 Service Pack3，WindowsXP 为 Service Pack1）。
2. ICETEK-VC54167-USB-EDU(或 ICETEK-VC5416-PP-EDU)实验箱一台。如无实验箱则配备 ICETEK-ICETEK-USB 仿真器或 ICETEK-ICETEK-PP 仿真器和 ICETEK-VC5416-A 或 ICETEK-VC5416-C 系统板，5V 电源一只。
3. USB 连接电缆一条(如使用 PP 型仿真器换成并口电缆一条)。

三. 实验原理

★开发 TMS320C5xxx 应用系统一般需要以下几个调试工具来完成：

.软件集成开发环境(Code Composer Studio 2.0)：完成系统的软件开发，进行软件和硬件仿真调试。它也是硬件调试的辅助手段。

.开发系统(ICETEK 5100 USB 或 ICETEK 5100 PP)：实现硬件仿真调试时与硬件系统的通信，控制和读取硬件系统的状态和数据。

.评估模块(ICETEK VC5416-A 或 ICETEK VC5416-C 等)：提供软件运行和调试的平台和用户系统开发的参照。

★Code Composer Studio 2.0 主要完成系统的软件开发和调试。它提供一整套的程序编制、维护、编译、调试环境，能将汇编语言和 C 语言程序编译连接生成 COFF (公共目标文件)格式的可执行文件，并能将程序下载到目标 DSP 上运行调试。

#用户系统的软件部分可以由 Code Composer Studio 建立的工程文件进行管理，工程文件一般包含以下几种文件：

.源程序文件：C 语言或汇编语言文件(*.ASM 或*.C)

.头文件(*.H)

.命令文件(*.CMD)

.库文件(*.LIB,*.OBJ)

四. 实验步骤

1. 实验准备

⑴. 连接实验设备

① 检查并设置 ICETEK-LF2407 实验箱的各电源开关均处于关闭状态；连接实验箱提供的三相电源线，保证接地良好；

② 如使用 USB 型仿真器，将提供的 USB 电缆的扁平端连接到计算机的 USB 接口上，另一端不接；

③ 如使用 PP 型仿真器，首先确认计算机电源处于关闭状态，然后将提供的并口电缆的一端连接到计算机的并行接口上；

④ 连接仿真器的仿真电缆接头到 DSP 系统板上的 JTAG 接头。注意仿真器接头上的一个插孔中有一个封针，DSP 系统板上的 JTAG 接口的相应插针是被空开的，这样保证了仿真接头的方向不会接反。

⑤ 如使用 PP 型仿真器，连接并口电缆的未连接端到仿真器上相应接头；

⑥ 如使用 ICETEK-VC5416-A 的 DSP 系统板，关闭 DSP 系统板上的电源开关；

⑦ 将 5V 电源连接到 DSP 系统板上；

⑵. 开启设备

① 接通计算机电源，进入 Windows 操作系统。

② 打开实验箱电源开关，实验箱上的电源指示灯亮。

③ 如使用 ICETEK-VC5416-A 的 DSP 系统板，打开 DSP 系统板电源开关；DSP 系统板上电源指示灯亮。

⑶. 安装 Code Composer Studio 2.0(可选做)

① 将实验箱附带的教学光盘插入计算机光盘驱动器；

② 利用桌面上‘我的电脑’打开教学光盘的 CC C5000 目录，双击‘Setup.exe”，进入安装程序；

③ 选择“Code Composer Studio”，按照安装提示进行安装，并重新启动计算机；

④ 安装完毕，桌面上出现两个新的图标“Setup CCS 2(‘C5000)”、“CCS 2(‘C5000)”；

⑷. 安装 DSP 开发系统驱动程序(可选做)

★安装 USB 型仿真器的驱动程序：

① 连接计算机上 USB 接口电缆的方形接口一端到仿真器上相应接口；仿真器上红色电源灯亮，表示 USB 接口连通；计算机提示发现新的设备；

② 指定驱动程序的路径到教学光盘的 ICETEKDriver\C5000\USBDevice 目录，选择“mdpjtag.inf”，并完成安装；

③ 选择“开始”、“设置”、“控制面板”，双击列表中“Blackhawk Control Panel”项，打开“Blackhawk”窗口；

④ 观察其中显示的设备名称为“Blackhawk USB2.0 JTAG Emulator (S/N: BD1050)”；

⑤ 观察仿真器上绿色指示灯亮，表示驱动程序开始工作；

★安装 PP 型仿真器的驱动程序：

① 利用桌面上“我的电脑”打开教学光盘的 ICETEKDriver\C5000 目录，双击“Setup.exe”，进入安装程序；

② 按顺序进行安装，注意驱动程序所安装的隐含路径为 C:\ICETEK\5xxPP；

③ 如果仿真器的工作环境是以 NT 为内核的操作系统，如：WindowsNT、Windows2000、WindowsXP，还需要安装 WindowsNTDriver 驱动，运行教学光盘的 ICETEKDriver 目录中的 WndowsNTDriver.exe 并按照步骤完成安装即可，安装完毕按照提示需要重新启动计算机；

2. 设置 Code Composer Studio2.0 在软件仿真(Simulator)方式下运行(可选做)
   (1). 双击桌面上“Setup CCS 2(‘C5000)”，启动“Code Composer Studio Setup”。
   (2). 在“Import Configuration”对话框中单击“Clear”按钮，在接下来的对话框中选择“是”，清除原先的系统设置；窗口“Code Composer Studio Setup”中左侧“System Configuration”栏中“My System”项被清空。
   (3). 在“Available Configurations”列表中，单击选择“C5416 Simulator”驱动，并单击“Import”按钮；窗口“Code Composer Studio Setup”中左侧“System Configuration”栏中“My System”项中被加入“C54x Simulator”项。
   (4). 单击“Close”按钮，退出“Import Configuration”对话框。
   (5). 选择“Code Composer Studio Setup”窗口“File”菜单中“Exit”项推出，并在接下来显示的对话框中选择“是”，保存设置；选择“否”，不启动 CCS。
3. 设置 Code Composer Studio 2.0 在硬件仿真(Emulator)方式下运行
   (1). 双击桌面上“Setup CCS 2(‘C5000)”，启动“Code Composer Studio Setup”。
   (2). 在“Import Configuration”对话框中单击“Clear”按钮，在接下来的对话框中选择“是”，清除原先的系统设置；窗口“Code Composer Studio Setup”中左侧“System Configuration”栏中“My System”项被清空。
   (3). 对于 USB 型仿真器(如使用 PP 型仿真器则跳过此步)，在“Available Configurations”列表中，单击选择“ICETEK-5100 USB Emulator”驱动，并单击“Import”按钮；窗口“Code Composer Studio Setup”中左侧“System Configuration”栏中“My System”项中被加入“C54x XDS”项。
   (4). 单击“Close”按钮，退出“Import Configuration”对话框。
   (5). 对于 PP 型仿真器，在“Code Composer Studio Setup”窗口中间的“Available Board/Simulator Types”窗口中查找名为“Itk5xxpp”的驱动程序，如果没有或此驱动程序前有禁止符号则：选择右侧窗口中“Install a Device Driver”，在随后出现的“Select Device Driver File”对话框中，选择 C:\ICETEK\5xxPP 目录中的“Itk5xxpp.dvr”驱动程序；双击“Code Composer Studio Setup”窗口中间的“Available Board/Simulator Types”中的“Itk5xxpp”，选择“Board Properties”卡片，将其中“I/O Port”的取值改为 0x378，选择“Next”，单击“Add Signal”，单击“Startup GEL Files”中“CPU_1”项末尾的浏览按钮，选择 C:\5416EDULab 目录中的 VC5416A.gel 文件，单击“打开”；单击“Finish”；
   (6). 选择“Code Composer Studio Setup”窗口“File”采单中“Exit”项推出，并在接下来显示的对话框中选择“是”，保存设置；
4. 启动 Code Composer Studio 2.0
   双击桌面上“CCS 2(‘C5000)”，启动 Code Composer Studio 2.0；可以看到显示出的 C54X Code Composer Studio 窗口；
5. 创建工程
   (1). 创建新的工程文件：
   选择菜单“Project”的“New…”项；在“Project Creation”对话框中，在“Project”项输入 volume；单击“Location”项末尾的浏览按钮，改变目录到 C:\5416EDULab\Lab1-UseCC，单击“OK”；单击“完成”；这时建立的是一个空的

工程文件；展开主窗口左侧工程管理窗口中“ Projects ”下新建立的“ volume.pjt ”，其中各项均为空。

⑵. 在工程文件中添加程序文件：

选择菜单“ Project ”的“ Add Files to Project… ”项；在“ Add Files to Project ”对话框中选择文件目录为 C:\5416EDULab\Lab1-UseCC，改变文件类型为“ C Source Files(*.c;*.ccc) ”,选择显示出来的文件‘ volum.c ”,重复上述各步骤,添加 volume.cmd 文件到 volum 工程中；添加 C:\ti\C5400\cgtools\lib\rts.lib。

⑶. 编译连接工程：

选择菜单‘ Project ’的‘ Rebuild All ’项 ,注意编译过程中 CCS 主窗口下部的‘ Build ”提示窗中显示编译信息，最后将给出错误和警告的统计数。

6. 编辑修改工程中的文件

⑴. 查看工程文件

展开 CCS 主窗口左侧工程管理窗中的工程各分支，可以看到“ volume.pjt ”工程中包含“ volume.h ”、“ rts.lib ”、“ volume.c ”和“ volume.cmd ”文件，其中第一个为程序在编译时根据程序中的“ include ”语句自动加入的。

⑵. 查看源文件

双击工程管理窗中的“ volume.c ”文件，可以查看程序内容。

双击工程管理窗中的“ volume.h ”文件，打开此文件显示，可以看到其中有主程序中要用到的一些宏定义如“ BUF_SIZE ”等。

“ volume.cmd ”文件定义程序所放置的位置，此例中描述了 5416 的存储器资源，指定了程序和数据在内存中的位置。

⑶. 编辑修改源文件

打开“ volume.c ”，找到“ main() ”主函数，将语句“ input = &inp_buffer[0]; ”最后的分号去掉，这样程序中就出现了一个语法错误；重新编译连接工程，可以发现编译信息窗口出现发现错误的提示，双击红色错误提示，CC 自动转到程序中出错的地方；将语句修改正确(这里是将语句末尾的分号加上)；重新编译；注意，重新编译时修改的文件被 CC 系统自动保存。

⑷. 修改工程文件的设置

选择“ Project ”菜单中的“ Build Options… ”项，打开“ Build Options for volume.pjt ”对话框，选择“ Linker ”卡片，在“ Stack Size ”项后输入 1024；单击“ 确定 ”完成设置；通过此设置，重新编译后，程序中的堆栈的尺寸被设置成 1024 个字。

7. 基本调试功能

⑴. 执行 File→Load Program ,在随后打开的对话框中选择刚刚建立的 C:\5416EDULab\Lab1-UseCC\Debug\volume.out 文件。

⑵. 在项目浏览窗口中,双击 volume.c 激活这个文件,移动光标到 main()行上,右击鼠标选择 Toggle Breakpoint 或按 F9 设置断点。

⑶. 选择 Debug→Run 或按 F5 运行程序,程序会自动停在 main()函数头上。

① 按 F10 执行到 write_buffer()函数上。

② 再按 F8,程序将转到 write_buffer 函数中运行。

③ 此时,为了返回主函数,按 shift-F7 完成 write_buffer 函数的执行。

④ 再次执行到 write_buffer 一行,按 F10 执行程序,对比与 F8 执行的不同。

注意: 在执行 C 语言的程序时,为了快速的运行到主函数调试自己的代码,可以使用 Debug→Go main 命令，上述实验中的使用的是较为繁琐的一种方法。

8. 使用观察窗口

(1). 执行 View→Watch Window 打开观察窗口。

(2). 在 volume.c 中,选中任意一个变量,右击鼠标,选择“ Quick Watch”，CCS 将打开 Quick Watch 窗口并显示选中的变量。

(3). 在 volume.c 中,选中任意一个变量,右击鼠标,选择“ Add to Watch Window ”，CCS 将把变量添加到观察窗口并显示选中的变量值。

(4). 在观察窗口中双击变量,则弹出修改变量窗口,此时,可以在这个窗口中改变程序变量的值。

(5). 把 str 变量加到观察窗口中,点击变量左边的"+",观察窗口可以展开结构变量,并且显示结构变量的每个元素的值。

(6). 把 str 变量加到观察窗口中；执行程序进入 write_buffer 函数，此时 num 函数超出了作用范围，可以利用 Call Stack 窗口察看在不同作用范围的变量：

① 执行 View→Call Stack 打开堆栈窗口。

② 双击堆栈窗口的 main()选项,此时可以察看 num 变量的值。

9. 文件输入/输出

这一节介绍如何从 PC 机上加载数据到目标机上。可用于使用已知的数据流测试算法的正确性。

在完成下面的操作以前，先介绍 Code Composer Studio 的 Probe（探针）断点,这种断点允许用户在指定位置提取/注入数据。Probe 断点可以设置在程序的任何位置，.当程序运行到 Probe 断点时，与 Probe 断点相关的事件将会被触发，当事件结束后，程序会继续执行。在这一节里，Probe 断点触发的事件是：从 PC 机的数据文件加载数据到目标系统的缓冲区中。

(1). 在真实的系统中，read_signals 函数用于读取 A/D 模块的数据并放到 DSP 缓冲区中。在这里，代替 A/D 模块完成这个工作的是 Probe 断点。当执行到函数 read_signals 时，Probe 断点完成这个工作。

① 在程序行 read_signals(int *input)上单击鼠标右键，选择“ Toggle breakpoint”，设置软件断点。

② 单击鼠标右键，选择“ Toggle Probe Point”，设置 Probe 断点。

(2). 执行 File→File I/O，打开对话框。

(3). 点击 Add File 把 sine2.dat 文件加到对话框中。

(4). 完成设置：

① 在 Address 中，输入 inp_buffer

② 在 Length 中,输入 100

③ 保证 warp around 被选中；

(5). 关联事件和 Probe 断点：

① 点击 Add Probe Point 按钮,打开对话框；

② 点击 Probe Point 列表中的内容,使之被选中；

③ 在 Connect 中选择 sine2.dat 文件；

④ 点击 Replace 按钮确认设置；

⑤ 点击"确定"关闭对话框。

(6). 点击"确定"关闭对话框,此时,已经配置好了 Probe 断点和与之关联的事件.进一步的结果在下面实验中显示；

10. 图形功能简介

下面我们使用 CC 的图形功能检验上一节的结果

⑴. 执行 View→Graph→Time/Frequency 打开 Graph Property Dialog 窗口；

⑵. 修改属性为如下值并确定：

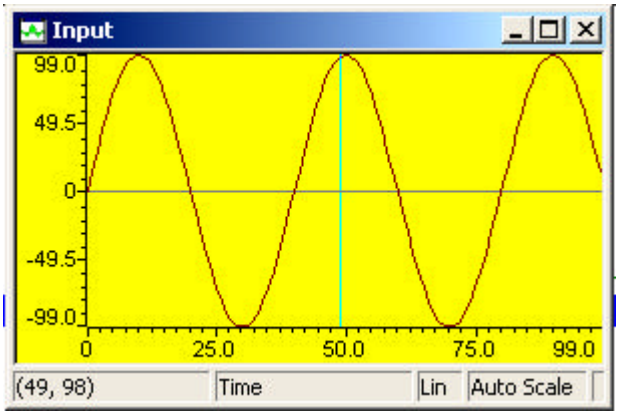
Graph Title：	Input
Satrt Address：	inp_buffer
Acquisition Buffer Size：	100
Display Data Size	100
DSP Type：	16-bit signed integer

-在弹出的图形窗口中单击鼠标右键，选择“Clear Display”。

⑶. 按 F12 运行程序.观察 input 窗口的内容。

五. 实验结果

通过对工程文件“volume”的编译、执行后得到结果的图形显示如下：



六. 问题与思考

实验二：编制链接控制文件

一. 实验目的

1. 学习用汇编语言编制程序；了解汇编语言程序与 C 语言程序的区别和在设置上的不同；
2. 学习编制命令文件控制代码的连接；
3. 学会建立和改变 map 文件，以及使用它观察内存使用情况的方法。
4. 熟悉使用软件仿真方式调试程序。

二. 实验设备

PC 兼容机一台，操作系统为 Windows2000(或 Windows98，WindowsXP，以下默认为 Windows2000)，安装 Code Composer Studio 2.0 软件。

三. 实验原理

1. 汇编语言程序

汇编语言程序除了程序中必须使用汇编语句之外，其编译选项的设置与 C 语言编制的程序也稍有不同。其区别为：

① 汇编语言程序在执行时直接从用户指定入口开始，常见的入口标号为“ start ”，而 C 语言程序在执行时，先要调用 C 标准库中的初始化程序(入口标号为“ _c_init00 ”)，完成设置之后，才转入用户的主程序 main()运行；为了支持 C 初始化代码的连接，C 程序在编译时要包含 C 语言库和与之相配的头文件，这需要用户将库添加到工程中。

② 由于 Code Composer Studio 的代码链接器默认支持 C 语言，在编制汇编语言程序时，需要设置链接参数，选择非自动初始化，注明汇编程序的入口地址。

2. 命令文件的作用

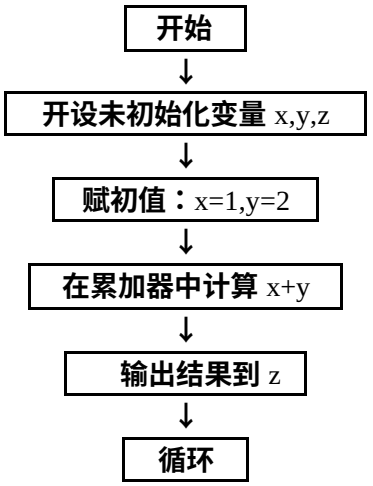
命令文件(文件名后缀为 cmd)为链接程序提供程序和数据在具体 DSP 硬件中的位置分配信息。通过编制命令文件，我们可以将某些特定的数据或程序按照我们的意图放置在 DSP 所管理的内存中。命令文件也为链接程序提供了 DSP 外扩存储器的描述。

3. 内存映射(map)文件的作用

一般地，我们设计、开发的 DSP 程序在调试好后，要固化到系统的 ROM 中。为了更精确地使用 ROM 空间，我们就需要知道程序的大小和位置，通过建立目标程序的 map 文件可以了解 DSP 代码的确切信息。当需要更改程序和数据的大小和位置时，就要适当修改 cmd 文件和源程序，再重新生成 map 文件来观察结果。

4. 源程序分析

汇编语言源程序 UseCMD.asm 框图：



四. 实验步骤

1. 实验准备

   设置软件仿真模式：

   ① 启动 CC 驱动设置窗口：双击桌面上“Setup CCS 2(‘C5000)”图标。

   ② 清除原先驱动设置：单击“Clear”按钮。

   ③ 安装软件仿真驱动（Simulator）：单击“C5416 Simulator”驱动名，单击“Import”按钮。

   ④ 完成设置：单击“Close”，菜单“File”、“Exit”，“是”。

2. 打开工程文件

   ⑴. 双击桌面上“CCS 2(‘C5000)”，启动 Code Composer Studio 2.0。

   ⑵. 打开菜单‘Project’的‘New…’项，在‘Project’项中输入 UseCMD，在‘Location’中选择 C:\5416EDULab\Lab2-UseCMD 目录，单击“完成”建立 UseCMD.pjt。

3. 设置工程文件

   ⑴. 打开设置窗口：选择菜单“Project”的“Build Options…”项。

   ⑵. 选择链接设置：单击“Linker”属性页。

   ⑶. 观察汇编语言程序的特殊设置：

   .“Autoinit Model”项设置成“No Autoinitialization”

   .“Code Entry Point”项中输入“start”。

   ⑷. 退出设置窗口：单击“确定”按钮。

4. 编译源文件，下载可执行程序

   ⑴. 单击菜单“Project”、“Rebuild All”；

   ⑵. 执行 File→Load Program，在随后打开的对话框中选择刚刚建立的 UseCMD.out 文件。完成后，系统自动打开源程序文件 UseCMD.asm。

5. 打开观察窗口

   ⑴. 开启 CPU 寄存器观察窗口：单击菜单“View”、“CPU Registers”、“CPU Registers”。

   ⑵. 在内存观察窗口中观察变量的值：

   选择“View”菜单中“Memory…”项，在“Memroy Window Options”窗口中的“Adress”项中输入 x，单击“OK”完成设置；在随后显示的“Memory”窗口中单击鼠标右键，选择“Float In Main Window”项。

6. 观察程序运行结果

   这时，代表程序运行位置的黄色光标条停在 start 标号下面语句上，程序将从此开始执行。

   ⑴. 单步执行程序（按 F10 键）2 次，可观察到 CPU 寄存器窗口中 DP 和 ST0 的值有变化。

   ⑵. 单步运行 2 次，在变量窗口中观察到变量 x、y 被赋值。

   ⑶. 单步执行到 xh 标号后面的语句，观察 ACC 寄存器和变量 z 值的变化。

7. 生成内存映像文件

   ⑴. 单击菜单“Project”、“Options…”，启动“Build Options”工程设置对话框。

   ⑵. 单击“Linker”属性页，在“Map Filename”项中输入需要生成的 map 文件名，比如可以输入 UseCMD.map

   ⑶. 单击“确定”，完成设置。

   ⑷. 选择菜单“Project”、“Rebuild All”，重新编译工程，生成新设置的 map 文件。

8. 对照观察 map 文件和 cmd 文件的内容

⑴. 选择菜单“ File ”、“ Open… ”，将找到 C:\2407EDULab\Lab2-UseCMD 目录，将文件类型改为“ Memory Map Files ”，选择刚刚生成的 UseCMD.map 文件、打开。

⑵. 展开工程管理窗中的 UseCMD.pjt，双击其中的 UseCMD.cmd 文件。

⑶. 程序的入口地址：cmd 文件的 SECTION 中指定.text 段放到程序区（PAGE 0）的 PRAM 中，在 MEMORY 中指定 PRAM 从内存地址 100h 开始，长度为 1f00h；再看 map 文件中“ ENTRY POINT SYMBOL ”中说明了“ start ”标号的地址为十六进制 0000100，两者相符。

⑷. 内存的占用情况：通过观察 map 文件中的“ MEMORY CONFIGURATION ”段可以了解内存的使用情况，可以看到，程序所占用的长度为十六进制 b，即 11 个字长，而数据区因开设了 3 个变量，所以占用了 3 个字的地址空间

9. 改变内存分配

修改 cmd 文件中的

PRAM : o=100h,l=1f00h

改为

PRAM : o=200h,l=1e00h

重新编译工程，观察 map 文件中有何变化。

五. 实验结果

通过实验可以发现，修改 cmd 文件可以安排程序和数据在 DSP 内存资源中的分配和位置；map 文件中描述了程序和数据所占用的实际尺寸和地址。

实验程序中计算变量的取值之和，由于取值较小，所以结果仍为 16 位数，程序中仅考虑保存 acc 的低 16 位作为结果。但如果计算中有进位等问题就需要考虑保存 acc 的高 16 位结果了。

六. 问题与思考

请修改程序完成无符号数 0f000h+0e000h 的计算。

实验三：数据存取实验

一、实验目的

了解 TMS320VC5416 的内部存储器空间的分配及指令寻址方式；
了解 ICETEK-VC5416-A 板扩展存储器空间寻址方法，及其应用；
了解 ICETEK-VC5416-EDU 实验箱扩展存储器空间寻址方法，及其应用；
学习用 Code Composer Studio 修改、填充 DSP 内存单元的方法；
学习操作 TMS320C5xxx 内存空间的指令。

二、实验设备

计算机，ICETEK-VC5416-EDU 实验箱（或 ICETEK 仿真器+ICETEK-VC5416-A 系统板+相关连线及电源）。

三、实验原理

1．TMS320VC54x DSP 内部存储器资源介绍：

C54x 片内有 8 条 16 位主总线：4 条程序/数据总线和 4 条地址总线，功能如下：

-程序总线（PB）传送取自程序存储器的指令代码和立即操作数。

-3 条数据中线（CB，DB，EB）将内部各单元连接在一起。

-4 个地址总线（PAB、CAB、DAB、EAB）传送执行指令所需的地址。

‘C54x 的总存储空间为 192K 字，分成 3 个可独立选择的空间：程序存储空间（64K 字），数据存储空间（64K 字），输入/输出（I/O）空间（64K 字）。

*程序区：

0000-007Fh： OVLY 位= 1，保留空间
OVLY 位=0，片外扩展存储器

0080-7FFFh： OVLY 位= 1，片上 DARAM0-3
OVLY 位=0，片外扩展存储器

8000-BFFFh： 片外扩展存储器

C000-FEFFh： MP/MC 位=0 片上 ROM，4Kx16 位
MP/MC 位=1 片外扩展存储器

FF00-FF7Fh： MP/MC 位=0，保留空间
MP/MC 位=1，片外扩展存储器

FF80-FFFFh： MP/MC 位=0，片上中断向量表
MP/MC 位=1，片外扩展中断向量表

*数据区：

0000-005Fh： 寄存器映射地址

0060-007Fh： 暂存器 SPRAM

0080-7FFFh： 片上 DARAM0-3，32Kx16 位

8000-FFFFh： DROM 位=1，片上 DARAM4-7
DROM 位=0，片外扩展存储器

*I/O 区：

0000-FFFFh： 片外扩展区

*扩展程序存储器空间：

扩展程序存储器空间采用分页扩展方法，使程序空间可扩展到 8192K 字。

2．ICETEK-VC5416-A 板对 TMS320VC5416 DSP 存储空间的扩展

程序区：未扩展所有片外扩展存储器
数据区：扩展所有片外扩展存储器
I/O 区：见前边第三章 二

3．ICETEK-VC5416-EDU 实验箱对 TMS320VC5416 DSP 存储空间的扩展
I/O 区：
8001-8001h： 读-键盘扫描值，写-液晶控制寄存器
8002-8002h： 液晶控制寄存器
8003-8004h： 液晶显示数据寄存器
8005-8005h： 发光二极管显示阵列控制寄存器
8006-9FFFh： 保留

4．MS320C54x 数据寻址方式介绍
’C54x 共有 7 种有效的寻址方式：
*立即寻址：主要用于初始化。
例如：LD #10,A ;将立即数 10 传送至累加器 A
*绝对寻址：利用 16 位地址寻址存储单元。
例如：STL A,*(y) ;将累加器的低 16 位存放到变量与 y 所在的存储单元中
*累加器寻址：把累加器的内容作为地址。
例如：READA x ;按累加器 A 作为地址读程序存储器，并存入变量 x 所在的数据存储器单元
*直接寻址：利用数据页指针或堆栈指针寻址。
例如：LD @x,A ;（DP+x 的低 7 位地址）→A
*间接寻址：利用辅助寄存器作为地址指针。
例如：LD *AR1,A ;（(AR1)）→A
*存储器映象寄存器寻址：快速寻址存储器映象器存器。
例如：LDM ST1,B ;（ST1）→B
*堆栈寻址：压入/弹出数据存储器和 MMR（存储器映象寄存器）。
例如：PSHM AG ;（SP）-1→SP，（AG）→TOS

5．实验程序分析
源程序 Memory.asm

```
        .global start    ; 定义全局标号
        .mmregs
        .text
start:
        nop
        ld      #4,dp    ;直接寻址，装载 DP 值，页指针指向片内数据区 DARAM B0
        st      #1,1     ; 绝对地址 201H 开始的四个单元存 1，2，3，4
        st      #2,2
        st      #3,3
        st      #4,4
```

```
        stm     #205h,ar1       ; 间接寻址，使用辅助寄存器 1
        rpt     #3              ; 循环重复执行下条语句 4 次
        st      #1234h,*ar1+    ; 将绝对地址 205H 开始的 4 个单元存成 1234H

                                ; 下面将 201H 开始的 8 个数读出
                                ;    存到 2000H 开始的 8 个单元
        stm     #7h,ar3         ; 循环计数器=7（8 次循环）
        stm     #201h,ar1       ; 源起始地址
        stm     #2000h,ar2      ; 目的起始地址
loop:
        ld      *ar1+,T  ; 将 ar1 指向单元内容读入 T 寄存器，ar1 的值+1
        st      T,*ar2+  ; 将 T 寄存器的值转存到 ar2 指向的目的地址，ar2 的值+1
        banz loop,*ar3-  ; ar3 的值-1，循环计数不等于 0 则循环
xh:
        b       xh       ; 空循环
        .end
```

四.实验步骤

1. **实验准备**

⑴. 连接设备：

① **关闭计算机和实验箱电源。**

② **如使用** PP **型仿真器则用附带的并口连线连接计算机并口和仿真器相应接口；**

③ **检查** ICETEK-VC5416-A **板上拨动开关** MP/MC **的位置，应设置在位置（靠近复位按钮一侧），即设置** DSP **工作在** MP **方式。**

④ **关闭实验箱上的三个开关。**

⑵. 开启设备：

① **打开计算机电源。**

② **打开实验箱电源开关，注意** ICETEK-VC5416-A **板上指示灯** D1 **和** D2 **亮。**

③ **如使用** USB **型仿真器用附带的** USB **电缆连接计算机和仿真器相应接口，注意仿真器上两个指示灯均亮。**

⑶. 设置 Code Composer Studio **为** Emulator **方式：**

参见“ Code Composer Studio **入门实验”之四**.3**。**

⑷. 启动 Code Composer Studio

双击桌面上“ CCS 2(‘C5000)**”图标，启动** Code Composer Studio 2.0**。**

2. **打开工程文件**

打开菜单“ Project**”的“** Open**”项；选择** C:\5416EDULab\Lab3-Memory **目录中的“** Memory.pjt**”。**

3. **观察修改程序区**

⑴.显示程序：选择菜单“ View**”的“** Memory…**”项；在“** Title**”中输入** PROG**，在“** Address**”项中输入** 0x1000**，选择“** Page**”项为“** Program**”；单击“** OK**”按钮；“** PROG **”窗口中显示了从地址** 1000H **开始的程序内存，由于** ICETEK-VC5416-A **板设置** DSP **工作在** MP **方式，窗口中显示的是片外扩展程序存储单元的内容；根据** cmd **文件中的设置，下载后的机器代码的入口应从** 1000H **处存放。**

⑵.修改程序区存储单元

程序区单元的内容由 CCS **的下载功能填充，但也能用手动方式修改；双击“** PROG **”**

窗口地址“0x1000:”后的第一个数，显示“Edit Memory”窗口，在“Data”中输入0x1234，单击“Done”按钮，观察“PROG”窗口中相应地址的数据被修改。

（3）观察修改数据区

① 显示片内数据存储区 DARAM0：选择菜单“View”的“Memory…”项，在“Title”中输入 DARAM0，在“Address”项中输入 0x200，单击“OK”按钮；“DARAM0”窗口中显示了从地址 200H 开始的数据内存；这片地址属于片内 DARAM0。

② 显示片内数据存储区 DARAM1：按照步骤⑴打开“DARAM1”窗口显示从地址 0x2000 开始的片内 DARAM1 区的数据单元。

③ 修改数据单元：数据单元也可以单个进行修改，只需双击想要改变的数据单元即可；选择菜单“Edit”、“Memory”、“Fill…”，在“Address”项中输入 0x200，在“Length”中输入 16，在“Fill”中输入 0x11，单击“OK”按钮，可在 200H 开始的数据区中的头 16 个单元填充统一的数 0x11；观察“DARAM0”窗口的变化；同样请将 0x2000 开始的头 8 个单元的值用 0 填充。

④ 访问未扩展的区域：当访问未扩展的存储单元时，将不能正确修改内容；选择菜单“View”的“Memory…”项；在“Title”中输入 NO EXIST，在“Address”项中输入 0xA000，选择“Page”项为“I/O”；单击“OK”按钮；“NO EXIST”窗口中显示了未扩展而不存在的 I/O 空间内存；试着修改其中的单元，然后在窗口中单击鼠标右键选择“Refresh Window”，可发现其他地址的内容也改变了，这说明此区域没有相对应的存储器存在。

4. 运行程序观察结果

⑴ 编译和下载程序：单击菜单“Option”、“Customuze…”，选择“Program Load Options”卡片，在“Load Program After Build”之前加上选择符号，单击“OK”按钮，此设置完成在每次编译完成后将程序自动下载到 DSP 上；选择菜单“Project”、“Rebuild All”，编译、连结和下载程序；观察“PROG”窗口中的变化。

⑵ 打开 CPU 寄存器观察窗口：选择菜单“View”、“CPU Registers”、“CPU Register”。

⑶ 单步执行程序并观察结果：按 F10 键单步运行，直到程序尾部的空循环语句；观察 CPU 寄存器窗口中 DP、ACC、ST0、AR0、AR1、AR2 的变化；观察“DARAM0”和“DARAM1”中的显示；体会用程序修改数据区语句的使用方法。

五.实验结果

实验程序运行之后，位于数据区地址 201H 开始的 8 个单元的数值被复制到了数据区 2000H 开始的 8 个单元中。

程序中使用了立即寻址、直接寻址和间接寻址方式。

*寻址方式的运用还有许多方法，可以完成复杂的寻址。

*如果在下载程序的过程中，Code Composer Studio 报告 xxxx 地址出现校验错误，可以使用内存显示和修改的方法，验证一下是否是该单元不能正确读写，以确定错误原因。

*通过改写内存单元的方式，我们可以手工设置 DSP 的一些状态位，从而改变 DSP 工作的状态。

六.问题与思考

实验四：定点数除法实验

一. 实验目的

1. 熟悉’C54x 指令系统，掌握常用汇编指令，学习设计程序和算法的技巧。
2. 学习用减法和移位指令实现除法运算。

二. 实验设备

PC 兼容机一台，操作系统为 Windows2000(或 Windows98，WindowsXP，以下默认为 Windows2000)，安装 Code Composer Studio 2.0 软件。

三. 实验原理

由内置的硬件模块支持，数字信号处理器可以高速的完成加法和乘法运算。但 TMS320 系列 DSP 不提供除法指令，为实现除法运算，需要编写除法子程序来实现。二进制除法是乘法的逆运算。乘法包括一系列的移位和加法，而除法可分解为一系列的减法和移位。本实验要求编写一个 16 位的定点除法子程序。

1. 除法运算的过程

设累加器为 8 位，且除法运算为 10 除以 3，除的过程包括与除数有关的除数逐步移位，然后进行减法运算，若所得商为正，则在商中置 1，否则该位商为 0

例如：4 位除法示例：

(1) 数的最低有效位对齐被除数的最高有效位

```
  00001010
- 00011000
----------
  11110010
```

(2) 由于减法结果为负，丢弃减法结果，将被除数左移一位再减

```
  00010100
- 00011000
----------
  11111000
```

(3) 结果仍为负，丢弃减法结果，将被除数左移一位再减

```
  00101000
- 00011000
----------
  00010000
```

(4) 结果为正，将减法结果左移一位后把商置 1，做最后一次减

```
  00100001
- 00011000
----------
  00001001
```

(5) 结果为正，将减法结果左移一位加 1 得最后结果，高 4 位是余数，低 4 位商：00010011

2. 除法运算的实现

为了尽量提高除法运算的效率，’C54x 系列提供了条件减指令 SUBC 来完成除法操作。SUBC 指令的功能如下：

若（ACC）≥0 且（数据存储器地址）≥0

PC+1 然后

（ACC）-[（数据存储器地址）×2^{15}] → ALU 输出

如果 ALU 输出≥0

则：（ALU 输出）×2+1 → ACC

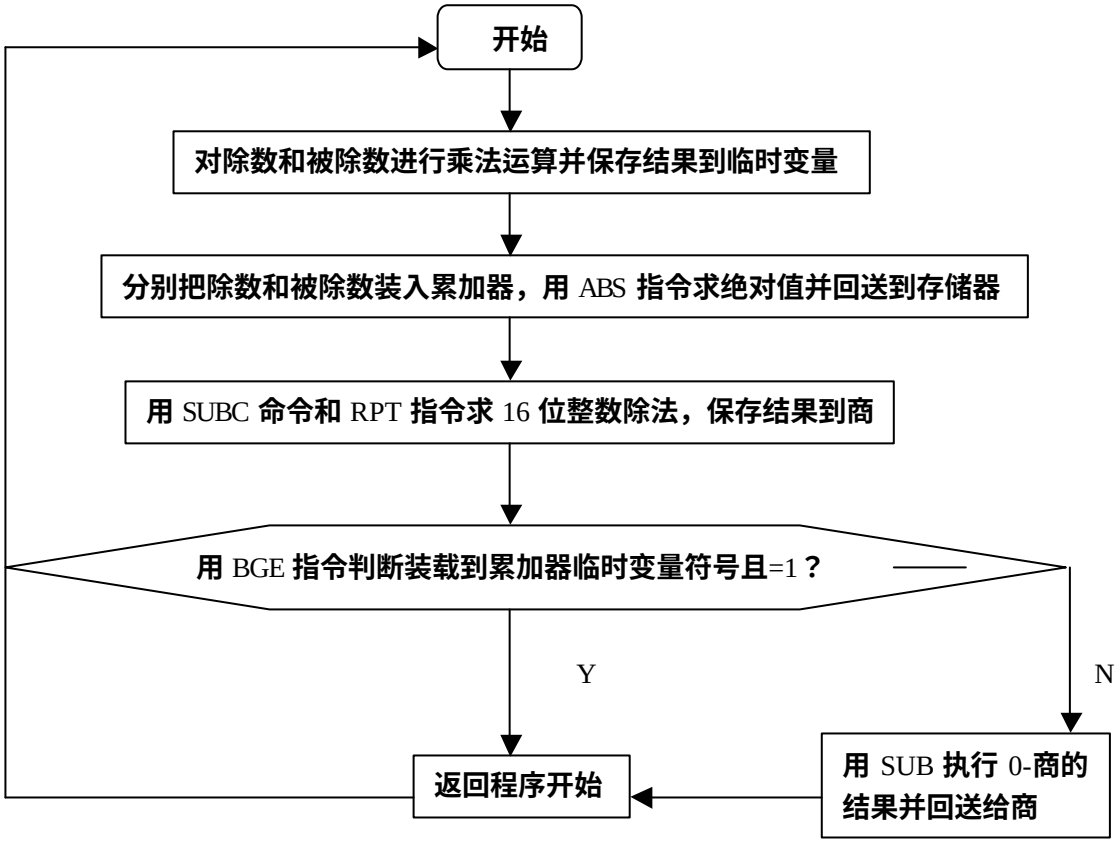
否则：（ACC×2） → ACC

实际上，SUBC 指令完成的是除法中的减除数求商的过程，即余数末位补 0，减去除数，若结果为正，该位商为 1，否则商为 0。

SUBC 指令实现条件减，可以用于如下除法：把 16 位的正被除数放在累加器的低 16 位，累加器的高 16 位清 0，16 位的正除数放在数据存储单元中。执行 SUBC 指令 16 次，最后一次 SUBC 指令完成后，累加器的低 16 位是除法的商，高 16 位是余数。若累加器和/或数据存储单元的内容为负，则不能用 SUBC 指令实现除法。为了完成多次的 SUBC 指令，还需要用到循环指令 RPT，它可以使 RPT 后的一条指令重复 1—256 次。

SUBC 指令仅能对正数除法进行运算，因此，要扩展到所有数值的除法，还需要：在程序开头对被除数和除数做乘法，并保存到临时变量，除数和被除数分别取绝对值，在除法运算完成后，根据临时变量的值修改商的符号。

3. 除法运算程序流程：



四. 实验步骤

1. 用 Simulator 方式启动 Code Composer。

2. 执行 Project→New 建立新的项目，输入 div 作为项目的名称，将程序定位在 C:\5416EDULab\Lab4-Division 目录。

3. 执行 File→New→Source File 建立新的程序文件，为创建新的程序文件命名为 div.asm 并保存；执行 Project→Add Files to Project，把 div.asm 加入项目中。

4. 执行 File→New→Source File 建立新的文件并保存为 div.cmd；执行 Project→Add Files to Project，把 div.cmd 加入项目中。

5. 编辑 div.asm 加入如下内容:

```
      .mmregs
stacksize  .set  256                          ; 堆栈尺寸
stack      .usect ".MYSTACK",stacksize        ; 堆栈空间
    .data
    .bss   NUMERA,1
    .bss   DENOM,1
    .bss   QUOT,1
    .bss   ARIT,1
    .bss   TEMSGN,1
    .text
start:
    stm        #stack+stacksize,SP            ; 初始化堆栈指针
    nop
    nop
next:
    call   DIV
    b          next
; 除法子程序
; 输入：NUMERA 被除数，DENOM 除数
; 输出：QUOT 商，ARIT 余数
DIV:ld     #NUMERA,dp
    ld     @NUMERA,T       ; 将被除数装入 T 寄存器
    mpy    @DENOM,A        ; 除数与被除数相乘，结果放入 A 寄存器
    ld     @DENOM,B        ; 将除数装入 B 寄存器的低 16 位
    abs    B               ; 求绝对值
    stl    B,@DENOM        ; 保存 B 寄存器的低 16 位
    ld     @NUMERA,B       ; 将被除数装入 B 寄存器低 16 位
    abs    B               ; 求绝对值
    rpt    #15             ; 重复下条 subc 指令 16 次
    subc   @DENOM,B        ; 完成除法运算
    bcd    done,AGT        ; 延时条转，先执行下面两条指令，然后判断 A,
                           ; 若 A>0 则跳转到标号 done,结束除法运算
    stl    B,@QUOT         ; 保存商（B 寄存器的低 16 位）
    sth    B,@ARIT         ; 保存余数（B 寄存器的高 16 位）
    xor    B               ; 若两数相乘的结果为负，则商也应为负
                           ; 先将 B 寄存器清 0
    sub    @QUOT,B         ; 将商反号
    stl    B,@QUOT         ; 存回反号后的商值
done:
    ret
```

6. 编辑 div.cmd 加入如下内容

```
MEMORY
{
  PAGE 0:
        VECT    :  o = 0ff80h , l = 0040h
       EX_ROM :  o = 1000h , l = 1000h
  PAGE 1:
        B2      :  o = 0060h , l = 0020h
        B1      :  o = 0300h , l = 0100h
}
SECTIONS
{
        .text     : {}> EX_ROM PAGE 0
        .data     : {}> EX_ROM PAGE 0
        .bss      : {}> B2     PAGE 1
       .MYSTACK : {}> B1    PAGE 1
}
```

7. 执行 Project→Rebuild All 编译链接；编译错误如下:

   warning：entry point symbol _c_int0 undefined

   出错原因：缺省时 Code Composer Studio 设置项目程序为 C 语言编译，因此当我们编译汇编程序时,要对项目作适当配置。

8. 执行 Project→Build Options…打开编译选项；在 linker 属性页上单击，把 Autoinit Model 栏选择为 No Autoinitialization；按“确定”保存对配置的修改。

9. 执行 File→Load Program 装载程序，装载完程序后，Code Composer Studio 把指针指向程序区 ff80H 处。为了执行我们的程序代码，需要修改 DSP 的 PC 值，执行 View→CPU Registers→CPU Registers 打开寄存器窗口；双击窗口中的 PC 标号，CC 弹出修改对话框供修改寄存器；在对话框中输入”start“,程序将处于我们的程序入口点上。

10. 在观察窗口中添加变量：

    ⑴ 在 div.asm 窗口中 NUMERA 变量名上双击鼠标左键，再单击鼠标右键，选择“Add to Watch Window”，在观察窗口中出现 NUMERA 变量，请将窗口中变量名（Name）由“NUMERA”改成“(int*)NUMERA”;“Radix”由“hex”改成“dec”;

    ⑵ 重复⑴的操作，加入 DENOM、QUOT 和 ARIT 四个变量。

    这样，我们就添加了为完成除法操作而需要的输入和输出变量。其中 NUMERA 是被除数，DENOM 是除数，QUOT 是商,而 ARIT 是余数。

11. 展开观察窗口上的 NUMERA 变量，在“Value”区输入数据“10”;同样输入 DENOM 变量的值“3”。

12. 按 F10 执行程序到”b next”，观察程序的 QUOT 和 ARIT 两个变量是否是正确的结果。此外，还可以多运行几次程序，分别用不同的数据测试除法程序。

13. 为汇编程序程序添加入口地址：在刚才的程序中，当我们装载完程序后，PC 指针指向程序的入口地址。现在，我们要为自己的汇编语言添加入口地址。

14. 双击项目浏览窗口中的 div.asm 激活源程序窗口；在程序的开头加入下述代码：

    .global start

    ⑴．执行 Project→Build Option 打开对话框；在对话框中单击 Linker 属性页，在 Code Entry Point 中输入 start；按”确定”保留对设置的修改。执行”Project→Rebuild

All”选项，重新编译链接整个项目。

⑵．执行”File→Reload Program”，Code Composer Studio 将会自动把上次选中的文件装载到目标系统中.观察这次装载与上次是否不同。

⑶．View→CPU Registers→CPU Register”打开寄存器观察窗口，察看 PC 指针的值是否有些不同。

⑷．构造完备的应用程序：’C54x 系列的芯片在上电复位后将 PC 机的值置为程序区 ff80H，这里实际上是复位向量的地址。一般地，要在这里添加一个无条件跳转指令，跳到程序真正的入口地址去。双击项目浏览窗口中的 div.asm 激活源程序窗口，在程序中.end 指令前加入如下指令:

```
.sect “.vectors”
      b     start
```

保存对 div.asm 的修改

五. 双击项目浏览窗口中的 div.cmd,激活该窗口；在.text :{}> PROG PAGE0 的下一行加入如下行：

```
.vectors :{}>VECT PAGE 0
```

执行 Project→Build 重新编译源程序

六. 执行 File→Exit 退出 Code Composer；在桌面上双击 CCS 2('C5000)图标重新启动 Code Composer Studio 2.0；执行 File→Load Program 装载 div.out。

七. 按 F5 运行程序；按 Shift-F5 结束程序运行；执行 Debug→Reset DSP 命令,观察程序的运行过程。

八. 实验结果

试验中用定点数计算代替除法运算，而在浮点 DSP 上可选用被除数承以除数的倒数的方法进行除法替代运算。

九. 问题与思考

实验五：I/O 端口实验

一 实验目的

1. 了解 ICETEK-VC5416-A 板在 I/O 空间上的扩展；
2. 掌握 I/O 端口的控制方法；
3. 学习在 C 语言中控制 I/O 端口读写的方法。

二 实验设备

计算机，ICETEK-VC5416-EDU 实验箱（或 ICETEK 仿真器+ICETEK-VC5416-A 系统板+相关连线及电源）。

三 实验原理

1. I/O 空间的扩展及使用：

-’C54x DSP 的 I/O 空间被保留用于外部扩展。由于在程序中访问 I/O 空间的语句只有 in 和 out 指令，所以在扩展时一般将带有控制能的寄存器或分离地址访问的存储单元的地址映射到 I/O 空间，访问这部分的单元又称 I/O 端口访问。例如：可将控制指示灯组的寄存器或锁存器映射到一个 I/O 端口地址上；A/D、D/A 等专用芯片控制端和状态寄存器也常映射到 I/O 端口上。总之，在 I/O 空间中扩展的设备一般重点用于控制，而使用大片连续存储空间的存储器单元一般映射到数据空间。

-ICETEK-VC5416-A 板将指示灯、DIP 开关、A/D、D/A、异步串行通信接口和 WatchDog 的控制端等映射在 I/O 空间，具体地址见：第三章 二。

0001-0001h：WatchDog 控制寄存器
0002-0004h：A/D 转换控制寄存器
1000-1004h：D/A 转换控制寄存器
2000-2007h：异步串口通信控制寄存器
3002-3002h： 板上指示灯控制寄存器
3003-3003h： 板上 DIP 拨动开关控制寄存器

-ICETEK-VC5416-EDU 实验箱上控制模块也使用 I/O 端口控制大部分设备：

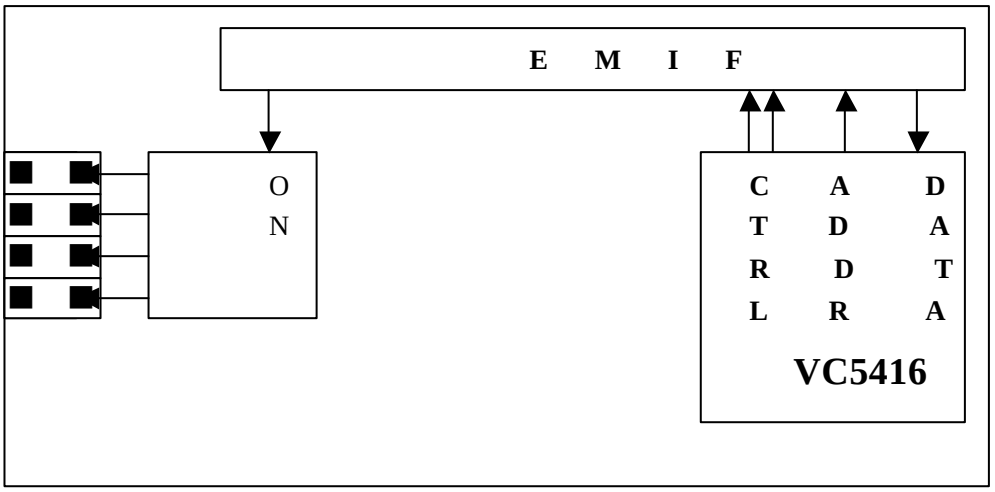
8001-8001h： 读-键盘扫描值，写-液晶控制寄存器
8002-8002h： 液晶控制寄存器
8003-8004h： 液晶显示数据寄存器
8005-8005h： 发光二极管显示阵列控制寄存器
8006-9FFFh： 保留

-在程序中，访问 I/O 端口的语句较为简单。对于汇编语言程序，可用 PORTR 和 PORTW 指令，例如，从端口 0008h 读入一个字到变量 x 的指令为 portr 8,x，而向端口 000Ch 输出 x 变量的值的指令为 portw #0Ch,x ;在 C 语言中访问 I/O 端口则必须首先声明 I/O 端口的类型，然后才能访问，以下语句仍完成上面汇编语言所完成的功能：

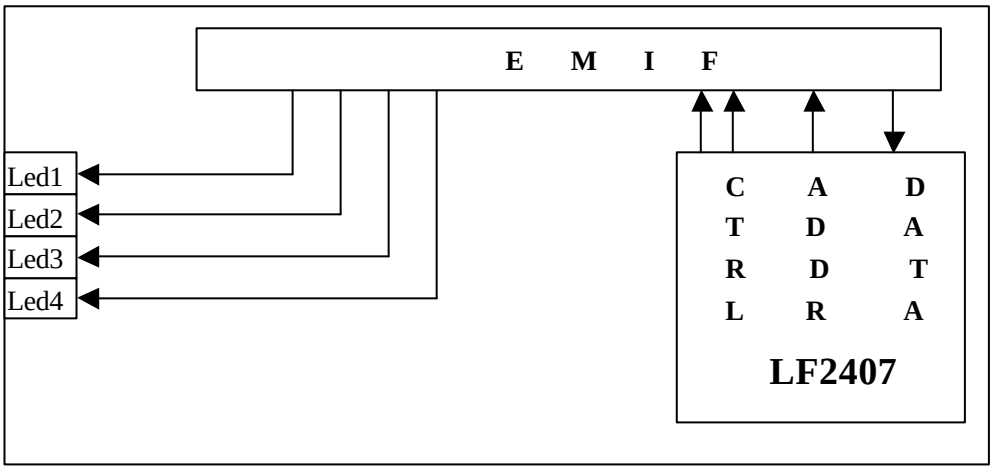
```
/*  端口定义  */
ioport unsigned int port0008;
ioport unsigned int port000c;
/*  在程序中使用：  */
   x=port0008;
```

```
port000c=(unsigned int)x;
```

2. **工程** 1 **操作相关硬件原理图**



3. **工程** 2 **硬件原理图**



四 实验步骤

1. **实验准备**

(1) **连接设备：**

① **关闭计算机和实验箱电源。**

② **如使用** PP **型仿真器则用附带的并口连线连接计算机并口和仿真器相应接口；**

③ **检查** ICETEK-VC5416-A **板上** DIP **开关** MP/MC **的位置，应设置在“** OFF **”位置（靠近复位按钮），即设置** DSP **工作在** MP **方式。**

④ **关闭实验箱上的三个开关。**

(2) **开启设备：**

①计算机电源

②打开实验箱电源开关，注意 ICETEK-VC5416-A **板上指示灯** D1 **和** DS2 **亮。**

③如使用 USB **型仿真器，用附带的** USB **电缆连接计算机和仿真器相应接口，注意仿真器上两个指示灯均亮。**

(3) 设置 Code Composer Studio 2.0 为 Emulator 方式：

参见“Code Composer Studio 入门实验”之四.3。

(4) 启动 Code Composer Studio

双击桌面上“CCS 2(‘C5000)”图标，启动 Code Composer Studio 2.0。

2. 打开工程文件

打开菜单“Project”的“Open”项；选择 C:\5416EDULab\Lab5-IOPort 目录中的“IOPort1.pjt”。

3. 浏览程序

在项目浏览器中，双击 led.c，激活 led.c 文件，浏览该文件的内容，理解各语句作用。

4. 编译工程

打开“Project”选单，选择“Rebuild All”选项，Code Composer Studio 重新编译和链接这个工程项目，整个的处理过程在屏幕下方的“Message”窗口中返回信息。

5. 下载程序

打开“File”选单，选择“Load Program”选项，在“Load Program”对话框中，选中 C:\5416EDULab\Lab5-IOPort\Debug 目录下的 ioport1.out 文件；此时，Code Composer Studio 将把这个目标文件装载到 ICETEK-VC5416-A 板上，同时，Code Composer Studio 打开反汇编窗口显示被加载程序的汇编指令码。

6. 运行程序观察结果

打开“Debug”菜单，选择“Run”选项运行程序（或按 F5 键），设置 ICETEK-VC5416-A 板上的 DIP 开关 U20 就可以定制指示灯是否为亮。

7. 结束运行，关闭工程

(1)打开“Debug”选单，选择“Halt”选项或按 Shift-F5 终止实验。

(2)选择菜单“Project”、“Close”关闭工程 1。

8. 打开另一个工程文件

(1) 打开菜单“Project”的“Open”项；选择 C:\5416EDULab\Lab5-IOPort 目录中的“IOPort2.pjt”。

(2) 编译工程，下载程序，观察结果。

五 实验结果

*工程 1 实验的最后现象：可以看到哪一个用户开关设置在 ON 状态上哪一盏指示灯就为亮。

*工程 2 实验的最后现象：可以看到指示灯依次闪烁。

*I/O 端口操作的地址和数据均通过总线完成，与读写单独的数据存储单元的方式一样，只是使用的地址空间不同。访问外部 I/O 空间时，DSP 的 IS 管脚变低。

*请参照程序，总结使用 I/O 端口控制简单外围设备的方式和方法。

六 问题与思考

*I/O 端口操作与内存读写操作的效果一样吗？各自的优势是什么？

*如果我需要扩展一片 Flash，应在地址上如何进行映射才方便程序使用？如果扩展的是 FIFO 呢？

实验六 I/O 控制模块实验

一. 实验目的

1. 了解 TMS320VC5416 DSP 的数字 I/O 控制模块的使用方法；
2. 学习使用 I/O 管脚控制外围设备；
3. 学会用程序驱动简单外围设备。

二. 实验设备

计算机，ICETEK-VC5416-EDU 实验箱（或 ICETEK 仿真器+ICETEK-VC5416-A 系统板+相关连线及电源）。

三. 实验原理

1. TMS320VC5416 DSP 的数字 I/O 控制模块介绍

   ’C54x 除了 64K 字 I/O 存储空间外，还有 2 个受软件控制的专用引脚 BIO 和 XF。

   5416DSP 还有一些管脚可以设置成通用输入/输出功能。这些管脚是：18 个 McBSP 引脚，8 个 HPI 数据引脚。这些引脚的使用可通过专用寄存器的设置完成。

2. ICETEK-VC5416-A 板引出的 I/O 管脚及使用方法

   ICETEK-VC5416-A 板使用了一些 I/O 管脚对 DSP 进行控制。例如：DIP 拨动开关 U2 的 MP/MC 位连接 DSP 上 MP/MC 管脚，在 DSP 复位时，DSP 可读回这一管脚的设置，当管脚接高电平时，DSP 采用微处理器（MP）方式工作，否则设置成微控制器（MC）方式。

   ICETEK-VC5416-A 板在扩展插头上将未使用的 I/O 引脚接出，提供给用户连接使用。其定义见 ICETEK-VC5416-A 板说明。这些管脚支持 0-3.3V 逻辑电平操作，用户在进行相应设置后可以在 I/O 管脚上进行输入或输出操作，使用时须注意根据引脚本身的负载能力驱动相关设备。

3. ICETEK-VC5416-EDU 实验箱及控制模块使用的 I/O 管脚

   ICETEK-VC5416-EDU 实验箱将引脚 AIN1-AIN2 连接到了实验箱底板上“A/D 输入”的 ADCIN2-ADCIN3 两个插座上（ADCIN0 和 ADCIN1 未使用）。

   实验箱控制模块 ICETEK-CTR 使用如下引脚：

   BFSX0--指示灯

   BFSR1 和 BCLKXR0—步进电机

   BDX0—蜂鸣器

4. 实验程序分析

   实验程序通过相关寄存器设置，使用 BFSX0 作为输出，控制实验相控制模块上指示灯 J5 进行有规律地闪烁。方法是用程序定时地修改 BFSX0 引脚的状态。

   注：BFSX0 引脚未连接驱动设备，通过一个 650Ω电阻限流后，直接连接到指示灯 J5。

   源程序及注释：

```
#define SPSA0 *(unsigned int *)0x38
#define SPSD0 *(unsigned int *)0x39
main()
{
   unsigned int uWork;
```

```
    SPSA0=1;          // 设置 McBSP0 的 SPCR2 控制寄存器
    uWork=SPSD0;
    uWork&=0xfffe;    // 标志 XRST=0
    SPSD0=uWork;
    SPSA0=0x0e;       // 设置 McBSP0 的 PCR1 寄存器
    uWork=SPSD0;
    uWork|=0x2800;    // 设置：XIOEN=1，          FSXM=1
                      //       使能通用 I/O 功能，FSX 用于输出
    SPSD0=uWork;
    while ( 1 )
    {
        SPSA0=0x0e; // 设置 McBSP0 的 PCR1 寄存器
        uWork=SPSD0;
        uWork|=0x2800;
        uWork^=0x8;  // FSXP=~FSXP，BFSX0 引脚状态取反
        SPSD0=uWork;
        Delay(4096);
    }
}
```

四. 实验步骤

1. **实验准备**

⑴ 连接设备：

① 关闭计算机和实验箱电源。

②如使用 PP 型仿真器则用附带的并口连线连接计算机并口和仿真器相应接口。

③检查 ICETEK-VC5416-A 板上 DIP 开关 MP/MC 的位置，应设置在“ OFF ”位置（靠近复位按钮），即设置 DSP 工作在 MP 方式。

④ 关闭实验箱上的三个开关。

⑵ 开启设备：

① 打开计算机电源。

② 打开实验箱电源开关，注意 ICETEK-VC5416-A 板上指示灯 D1 和 DS2 亮。

③ 如使用 USB 型仿真器，用附带的 USB 电缆连接计算机和仿真器相应接口，注意仿真器上两个指示灯均亮。

⑶ 设置 Code Composer Studio 2.0 为 Emulator 方式：

参见“ Code Composer Studio 入门实验 ”之四.3。

⑷ 启动 Code Composer Studio

双击桌面上“ CCS 2(‘C5000) ”图标，启动 Code Composer Studio 2.0。

⑸ 打开工程文件

打开菜单“ Project ”的“ Open ”项；选择 C:\5416EDULab\Lab6-IOPin 目录中的“ IOPin.pjt ”。

2. **浏览程序**

在项目浏览器中，双击 led.c，激活 led.c 文件，浏览该文件的内容，理解各语句作用。打开 led.cmd，浏览并理解各语句作用。

3. **编译工程**

单击“ Project ”菜单，“ Rebuild All ”项，编译工程中的文件，生成 IOPin.out 文件。

4. 下载程序

单击“ File ”菜单，“ Load program… ”项，选择 C:\5416EDULab\Lab6-IOPin\Debug 目录中的 IOPin.out 文件，通过仿真器将其下载到 5416 DSP 上。

5. 运行程序观察结果

.单击“ Debug ”菜单，“ Run ”项，运行程序。

.观察实验箱控制模块上指示灯 J5 闪烁情况。

.单击“ Debug ”菜单，“ Halt ”项，停止程序运行。

6. 修改程序重新运行

适当改变程序中“ Delay(4096); ”语句中的延时参数，重复步骤 3-5，使指示灯约 1 秒闪烁一次。

五. 实验结果

实验程序可控制指示灯闪烁。通过 DSP 的通用 I/O 引脚可以输出状态，从而直接控制外围设备。

六. 问题与思考

*如果需要控制较大电流驱动的设备或控制电平大于 3.3V 的设备应如何设计？

*如果需要精确控制指示灯闪烁的时间，有什么方法？

实验七：定时器实验

一. 实验目的

1. 通过实验熟悉 5416 的定时器。
2. 掌握 5416 定时器的控制方法。
3. 掌握 5416 的中断结构合对中断的处理流程。
4. 学会 C 语言中断程序设计，以及运用中断程序控制程序流程。
5. 学习用 C 语言同汇编语言混合编程的技术。

二. 实验设备

计算机，ICETEK-VC5416-EDU 实验箱（或 ICETEK 仿真器+ICETEK-VC5416-A 系统板+相关连线及电源）。

三. 实验原理

1. 通用定时器介绍及其控制方法

片内定时器是一个软件可编程定时器，可以用来产生周期的中断信号。

定时器主要由 3 个寄存器所组成：定时器寄存器（TIM）、定时器周期寄存器（PRD）和定时器控制寄存器（TCR）。这 3 个寄存器都有映象寄存器，它们在数据存储器中的地址分别为 24H、25H 和 26H。TIM 是一个递减计数器；PRD 中存放计数值；TCR 中有定时器的控制位和状态位：

15—12 保留

11--10 soft free

9—6 PSC 定时器预定标计数器

5 TRB 定时器重新加载位，用来复位片内定时器

4 TSS 定时器停止状态位，用于停止或启动定时器

3—0 TDDR 定时器分频系数

在正常工作情况下，当 TIM 减到 0 后，PRD 中的时间常数自动地加载到 TIM。

复位后，定时器控制寄存器（TCR）的停止状态位 TSS=0，定时器启动工作，时钟信号 CLKOUT 加到预定标计数器 PSC。PSC 也是一个递减计数器，每当复位或其减到 0 后，自动地将定时器分频系数 TDDR 加载到 PSC。PSC 在 CLKOUT 作用下，作减 1 计数。当 PSC 减到 0，产生一个借位信号，令 TIM 作减 1 计数。TIM 减到 0 后，产生定时器中断信号 TINT，传送到 CPU 和定时器输出引脚 TOUT。

定时器中断的周期为：CLKOUT ×（TDDR+1）×（PRD+1）

其中，CLKOUT 位时钟周期，TDDR 和 PRD 分别为定时器的分频系数和时间常数。

对定时器初始化的步骤如下：

⑴．先将 TCR 中的 TSS 位置 1，关闭定时器。

⑵．加载 PRD。

⑶．重新加载 TCR（使 TDDR 初始化；令 TSS 位=0，以接通 CLKOUT；TRB 位值 1，以使 TIM 减到 0 后重新加载定时器时间常数），启动定时器。

对中断的处理：

⑴．设置 INTM=1

⑵．将 IFR 中的 TINT 位置 1，清除尚未处理完的定时器中断。

⑶．将 IMR 中的 TINT 位置 1，开放定时器中断。

⑷．将 ST1 中的 INTM 位请 0，开放所有可屏蔽中断。

2. TMS320VC5416 中断结构

以下是 5416 的 IMR 和 IFR 寄存器的结构，其中包含了可响应的中断：

15–14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resvd	DMAC5	DMAC4	XINT1	RINT1	HINT	INT3	XINT2	RINT2	XINT0	RINT0	TINT	INT2	INT1	INT0

INT0-3 为外部引脚产生的中断，TINT 为定时器中断，RINT0-2 和 XINT0-2 对应 McBSP 口的接收和发送中断，HINT 对应 HPI 接口中断，另外还有 DMA 中断。

3. 中断响应过程

外设事件要引起 CPU 中断，必须保证：IMR 相应位被使能（置 1），ST1 寄存器中的 INTM 使能（置 0）。

当 CPU 响应中断时，PC 指针指向中断向量表中对应中断的地址，进入中断服务子程序。中断向量表是 DSP 存放中断服务程序的一段内存区域，大小为 80H。在中断向量表中，每一个中断占用 4 个字的空间，一般情况是将一条跳转或延时跳转指令存放于此。中断向量表的位置是可以改变的，修改 PMST 寄存器中的中断向量表基地址可以实现这一点。

4. 中断程序设计

-程序中应包含中断向量表，5416 默认向量表从程序区 FF80 地址开始存放。

-向量表中每项为 4 个字，存放一个跳转指令，跳转指令中的地址为相应服务程序入口地址；第一个向量表的首项为复位向量，即 CPU 复位操作完成后自动进入执行的程序入口；

-程序中包含相应的中断服务程序，应将其入口地址加入相应中断向量表中。

5. 实验程序分析

本实验设计的程序是在上一个实验基础上修改得来，由于上一实验控制指示灯闪烁的延时控制是用循环计算方法得到的，延时不精确也不均匀，采用中断方式可以实现指示灯的定时闪烁，时间更加准确。

对于定时器的周期寄存器为计数 f423H，分频系数定为 15，即 1000000 个 CPU 时钟计数一次，由于 DSP 工作在 8MHz 主频（ICETEK-VC5416-A 板上 DIP 开关 U2 的 CLKMD1-3 均为 OFF 时），正好是 125ms 中断一次，所以在中断服务程序中计算中断 4 次时改变指示灯状态，实现指示灯亮 0.5 秒再灭 0.5 秒，即每秒闪烁 1 次。

实验程序的工程中包含了两种源代码，主程序采用 C 语言编制利于控制，中断向量表在 vector.asm 汇编语言文件中，利于直观地控制存储区分配。在工程中只需将它们添加进来即可，编译系统会自动识别分别处理完成整合工作。

实验程序的 C 语言主程序中包含了内嵌汇编语句，提供一种在需要更直接控制 DSP 状态的方法，同样的方法也能提高 C 语言部分程序的计算效率。

四. 实验步骤

1. 实验准备

（1）.连接设备

① 关闭计算机和实验箱电源；

② 如使用 PP 型仿真器则用附带的并口连线连接计算机并口和仿真器相应接口；

③ 检查 ICETEK-VC5416-A 板上 DIP 开关 MP/MC 的位置，应设置在“OFF”位置（靠近复位按钮），即设置 DSP 工作在 MP 方式。

④ 关闭实验箱上的三个开关。

⑵ 开启设备：

① 打开计算机电源

② 打开实验箱电源开关，注意 ICETEK-VC5416-A 板上指示灯 D1 和 DS2 亮。

③ 如使用 USB 型仿真器，用附带的 USB 电缆连接计算机和仿真器相应接口，注意仿真器上两个指示灯均亮。

⑶ 设置 Code Composer Studio 为 Emulator 方式：

参见“CodeComposer Studio 入门实验”之四.3。

⑷ 启动 Code Composer Studio

双击桌面上“CCS 2(’C5000)”图标，启动 Code Composer 2.0。

2. 打开工程文件,浏览程序

打开菜单“Project”的“Open”项；选择 C:\5416EDULab\Lab7-Timer 目录中的“Timer.pjt”。

在项目浏览器中，双击 led.c，激活 led.c 文件，浏览该文件的内容，理解各语句作用。打开 led.cmd，浏览并理解各语句作用，对照 C 源程序学习中断向量表的写法。

3. 编译工程

单击“Project”菜单，“Rebuild all”项，编译工程中的文件，生成 Timer.out 文件。

4. 下载程序

单击“File”菜单，“Load program…”项，选择 C:\5416EDULab\Lab67-Timer\Debug 目录中的 Timer.out 文件，通过仿真器将其下载到 5416 DSP 上。

5. 运行程序观察结果

.单击“Debug”菜单，“Run”项，运行程序。

.观察实验箱控制模块上指示灯 J5 闪烁情况。

.单击“Debug”菜单，“Halt”项，停止程序运行。

6. 修改程序重新运行

适当改变程序中的延时参数，重复步骤 3-5，使指示灯约 1 秒闪烁两次、三次、四次。

五. 实验结果

-指示灯在定时器的定时中断中按照设计定时闪烁。

-使用定时器和中断服务程序可以完成许多需要定时完成的任务，比如 DSP 定时启动 A/D 转换，日常生活中的计时器计数、空调的定时启动和关闭等。

-在调试程序时，有时需要指示程序工作的状态，可以利用指示灯的闪烁来达到，指示灯灵活的闪烁方式可表达多种状态信息。

六. 问题与思考

实验八：模数转换实验

一. 实验目的

1. 掌握 A/D 转换的基本过程；
2. 熟悉 ICETEK-VC5416-A 板上使用 ADS7864 技术指标和操作方法。

二. 实验设备

计算机，ICETEK-VC5416-EDU 实验箱（或 ICETEK 仿真器+ICETEK-VC5416-A 系统板+信号源+示波器+相关连线及电源）。

三. 实验原理

1. ADS7864 模数转换模块特性

ADS7864 是 TI 公司的一种 500K、12 位、6 通道模数转换芯片。每通道信号可采用插分方式输入。

ICETEK-VC5416-A 板上使用的方式是非插分方式，即所有 IN-端均与参考地相接，采用输入信号的范围为 0～+5V。

由于 ADS7864 芯片为 5V 器件，而 5416DSP 为 3.3V 器件，所以在进行硬件连接设计时采用电平转换芯片对信号线和数据线进行隔离。

ICETEK-VC5416-A 板上将 ADS7864 的控制映射到 I/O 空间，使用 I/O 空间地址 2H 传送数据，3H 进行通道选择，4H 发送转换信号。

由于 ADS7864 的 6 个通道转换是分成 3 路完成的（A、B、C），在每个转换周期可选择启动 2、4、6 个转换通道，选择的方法是在 3H 地址的相应位上置低电平。3H 地址输出的最低位（第 0 位）控制 A 路，次低位（第 1 位）控制 B 路，第 2 位控制 C 路，所以如果需要采集 A0 和 A1 两路信号则可在 3H 地址上输出 6（110b），B0 和 B1 输出 5（101b），C0 和 C1 输出 3（011b）。以下是控制字同相应选择通道的列表：

06H	A0,A1	04H	A0,A1,B0,B1	00H	A0,A1,B0,B1,C0,C1
05H	B0,B1	01H	B0,B1,C0,C1		
03H	C0,C1	02H	A0,A1,C0,C1		

如果只需要进行单通道的转换，可以只进行 1 路输入而保存相应通道的数据即可。

2. 模数转换工作过程

-模数转换模块接到启动转换信号后，按照设置进行相应通道的数据采样转换。

-经过一个采样时间的延迟后，将采样结果放入相应通道的 FIFO 保存。

-转换结束，设置标志。

-等待下一个启动信号。

3. 模数转换的程序控制

模数转换相对于计算机来说是一个较为缓慢的过程。一般采用中断方式启动转换或保存结果，这样在 CPU 忙于其他工作时可以少占用处理时间。设计转换程序应首先考虑处理过程如何与模数转换的时间相匹配，根据实际需要选择适当的触发转换的手段，也要能及时地保存结果。

由于 ADS7864 芯片的 A/D 转换精度是 12 位的，转换结果的最高位表示转换值是否有效（1 有效），第 14-12 位表示转换的通道号，低 12 位为转换数值，所以在保留时应注意取出结果的低 12 位，再根据高 4 位进行相应保存。

4. 源程序及注释

本实验程序采用中断程序设计，定时器设置采样时间为 15.625KHz（64 **微秒），采样通道设置为** A0 **和** A1（ICETEK-VC5416-EDU **实验箱上** ADCIN2 **和** ADCIN3）。

```
#defineTIM      *(int *)0x24
#definePRD       *(int  *)0x25
#defineTCR      *(int *)0x26
#defineIMR      *(int *)0x0
#defineIFR      *(int *)0x1
#definePMST     *(int *)0x1d

ioport unsigned int port3,port4,port2;

#define AD_DATA     port2
#define AD_SEL    port3
#define AD_HOLD port4

void interrupt time(void);
int *ptr,k;
unsigned int uWork;

main()
{
  int i,j;

  asm("     ssbx INTM"); // 关闭可屏蔽中断
  k=0;
  ptr=(int *)0x3000;  // 转换数据的保存区，从数据区 3000H 开始
                      // 3000H-3200H 保存第 1 通道（AIN1）的转换结果
                      // 3200H-3400H 保存第 2 通道（AIN2）的转换结果
  for(i=0;i<0x400;i++)     // 将转换数据的保存区清 0
                      *(ptr+i)=0;
  j= PMST;
  PMST = j&0xff;
  IMR = 0x8;
  TCR = 0x412;        // 计数器分频系数=2
  TIM = 0;
  PRD = 0x100;        // 定时器周期=256，采样周期=周期*分频系数*时钟周期
  TCR = 0x422;        //                                  =512 时钟=64 微秒
  IFR = 0x100;        //     其中，时钟周期为 8MHz

  AD_SEL=6;           // 通道选择  A0，A1
  asm("     rsbx INTM"); // 开中断进行转换

  while ( 1 );
}
```

```
// 定时器中断服务程序，完成：保存转换结果、启动下次转换
void interrupt time(void)
{
  uWork=AD_DATA;            // 从 FIFO 中读取转换结果
  uWork&=0x0fff;            // 去掉高 4 位
  *(ptr+k)=uWork;           // 保存结果

  uWork=AD_DATA;            // 从 FIFO 中读取转换结果
  uWork&=0x0fff;            // 去掉高 4 位
  *(ptr+0x200+k)=uWork;     // 保存结果

  k++;
  if ( k>=0x200 )
  {
      k=0;                  // 软中断位置
  }
  AD_HOLD =0;               // 送转换信号
  for ( uWork=0;uWork<10;uWork++ );
  AD_HOLD=1;
}
```

四. 实验步骤

1. 实验准备

⑴. 连接设备：

① 关闭计算机和实验箱电源。

② 如使用 PP 型仿真器则用附带的并口连线连接计算机并口和仿真器相应接口。

③ 检查 ICETEK-VC5416-A 板上拨动开关 MP/MC 的位置，应设置在位置（靠近复位按钮一侧），即设置 DSP 工作在 MP 方式。

④ 用实验箱附带的信号连接线（两边均为单声道耳机插头）连接第一信号源的波形输出端到“A/D 输入”的 ADCIN2 插座（位于实验箱底板中部，第二信号源旁边）；用信号连线连接第二信号源的输出端到“A/D 输入”的 ADCIN3 插座；

⑤ 关闭实验箱上三个开关。

⑵. 开启设备：

① 打开计算机电源。

② 打开实验箱电源开关，注意 ICETEK-VC5416-A 板上指示灯 D1 和 D2 亮。

③ 如使用 USB 型仿真器用附带的 USB 电缆连接计算机和仿真器相应接口，注意仿真器上两个指示灯均亮。

⑶. 设置 Code Composer Studio 为 Emulator 方式：

参见“Code Composer Studio 入门实验”之四.3。

⑷. 启动 Code Composer Studio

双击桌面上“CCS 2(‘C5000)”图标，启动 Code Composer Studio 2.0。

2. 打开工程并浏览程序

-打开菜单‘ Project ’的‘ Open ’项 选择 C:\5416EDULab\Lab8-AD 目录中的‘ adc.pjt ”。

-在项目浏览器中，双击 ad.c，激活 ad.c 文件，浏览该文件的内容，理解各语句作用。打开 ad.cmd，浏览并理解各语句作用。

3. 编译工程

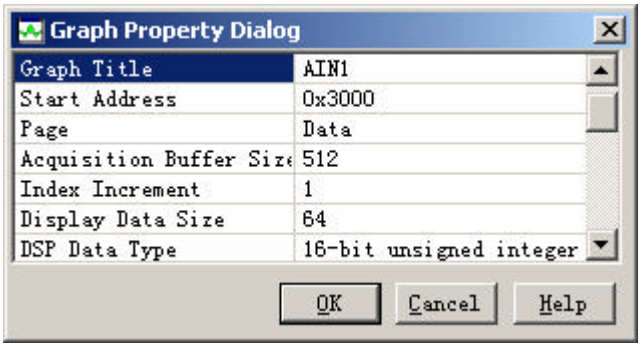
单击“ Project ”菜单，“ Rebuild All ”项，编译工程中的文件，生成 adc.out 文件。

4. 下载程序

单击“ File ”菜单，“ Load program… ”项，选择 C:\5416EDULab\Lab8-AD\Debug 目录中的 adc.out 文件，通过仿真器将其下载到 5416 DSP 上。

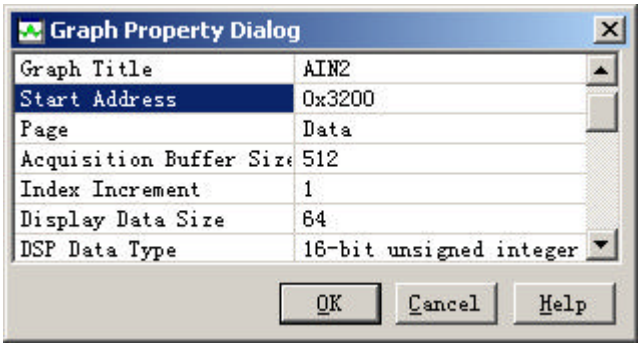
5. 打开观察窗口

-选择菜单“ View ”、“ Graph ”、“ Time/Frequency… ”做如下设置，然后单击“ OK ”按钮；



-在弹出的图形窗口中单击鼠标右键，选择“ Clear Display ”。

-选择菜单“ View ”、“ Graph ”、“ Time/Frequency… ”做如下设置，然后单击“ OK ”按钮；



-在弹出的图形窗口中单击鼠标右键，选择“ Clear Display ”。

-在有“ 软中断位置 ”注释的语句上加上软件跟踪断点（Toggle Breakpoint），即鼠标左键单击该语句后按 F9 键；

通过设置，我们打开了两个图形窗口观察两个通道模数转换的结果。

6. 设置信号源

由于模数输入信号未经任何转换就进入 DSP，所以必须保证输入的模拟信号的幅度在 0～+5V 之间。实验箱上信号源输出为 0～+3.3V。但如果使用外接信号源，则必须用示波器检测信号范围，保证最小值 0V 最大值+5V，否则容易损坏 DSP 芯片的模数采集模块。

首先设置一号信号源（上部）开关为“ 关 ”。设置实验箱上一号信号源的“ 频率选择 ”在“ 100Hz—1KHz ”档，“ 波形选择 ”在“ 三角波 ”档，“ 频率微调 ”选择较大

位置靠近最大值，“ 幅值微调 ” 选择最大。开启一号信号源开关，一号信号源电源指示灯亮。

首先设置二号信号源（下部）开关为 “ 关 ”。设置实验箱上二号信号源的 “ 频率选择 ” 在 “ 100Hz—1KHz ” 档，“ 波形选择 ” 在 “ 正弦波 ” 档，“ 频率微调 ” 选择适中位置，“ 幅值微调 ” 选择最大。开启二号信号源开关，二号信号源电源指示灯亮。

7. 运行程序观察结果

-单击 “ Debug ” 菜单，“ Run ” 项，运行程序；

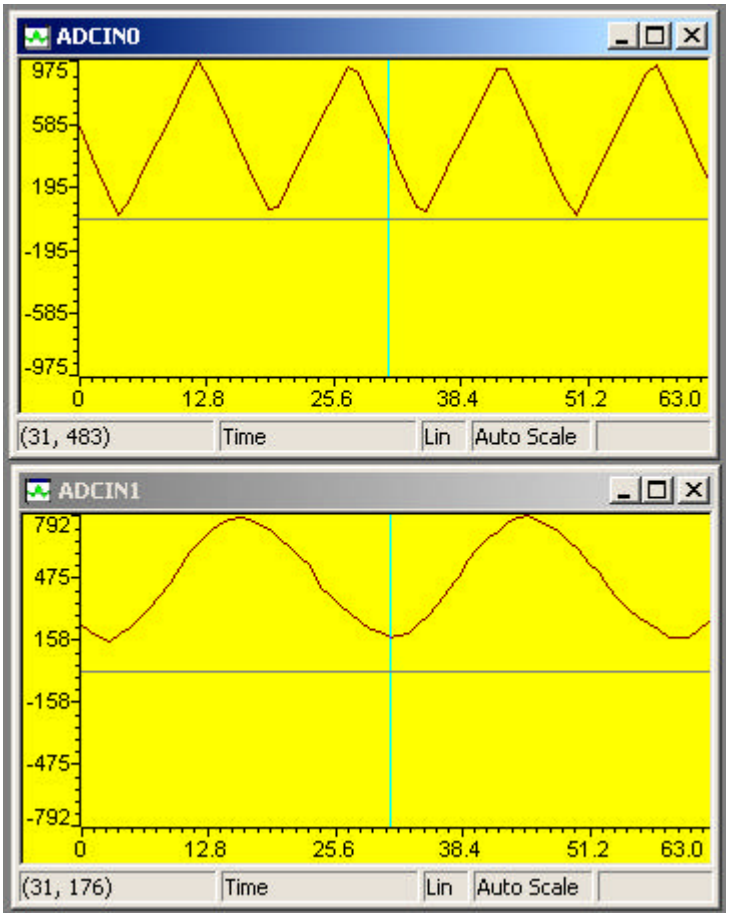
-当程序停在所设置的软件断点上时，观察 “ AIN1 ”、“ AIN2 ” 窗口中的图形显示；

-适当改变信号源的四个调节旋钮的位置，按 F5 健再次运行到断点位置，观察图形窗口中的显示。注意：输入信号的频率不能大于 7.5KHz，否则会引起混叠失真，而无法观察到波形，如果有兴趣，可以试着做一下，观察采样失真后的图形。

8. 停止运行结束实验

五. 实验结果

-用实验中的设置，我们可以看到如下结果



六. 问题与思考

请修改实验程序，实现采样频率为 200KHz 的转换

实验九：数模转换实验

一. 实验目的

1. 了解数模转换的基本操作；
2. 了解 ICETEK-VC5416-A 板扩展数模转换方式；
3. 掌握数模转换程序设计方法。

二. 实验设备

计算机，示波器，ICETEK-VC5416-EDU 实验箱（或 ICETEK 仿真器+ICETEK-VC5416-A 系统板+相关连线及电源）。

三. 实验原理

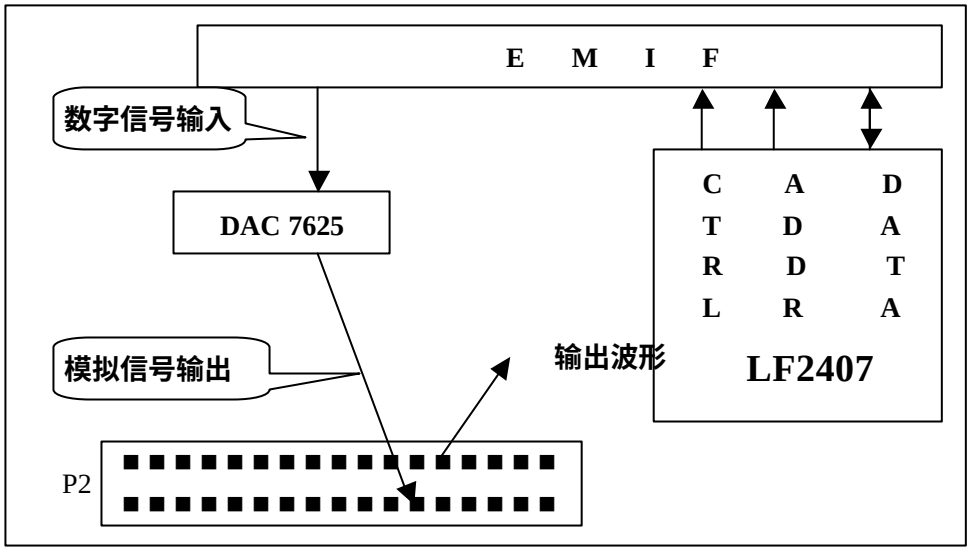
1. 数模转换操作

利用专用的数模转换芯片，可以实现将数字信号转换成模拟量输出的功能。在 ICETEK-VC5416-A 板上，使用的是 DAC7625 数模芯片，它可以实现同实转换四路模拟信号数出，并有 12 位精度，转换时间 10μs。其控制方式较为简单：首先将需要转换的数值通过数据总线传送到 DAC7625 上相应寄存器，再发送转换信号，经过一个时间延迟，转换后的模拟量就从 DAC7625 输出引脚输出。

2. DAC7625 与 TMS320VC5416 的连接

由于 TMS320VC5416 DSP 没有数模转换输出设备，采用外扩数模转换芯片的方法。在 ICETEK-VC5416-A 板上选用的是 DAC7625。DAC7625 的转换寄存器被映射到了 DSP 的 I/O 空间，地址是 1001H-1003H，控制转换由 I/O 端口地址 1004H 的写信号控制，这部分在硬件上由译码电路（CPLD）完成。在 DAC7625 的输出端，为了增加输出功率，经过一级运放再输出到板上插座上。输出范围为 0～+5V。

*硬件原理图



四. 实验步骤

1. 实验准备

⑴. 连接设备

① 关闭计算机和实验箱电源。

② 如使用 PP 型仿真器则用附带的并口连线连接计算机并口和仿真器相应接口。

③ 检查 ICETEK-VC5416-A 板上拨动开关 MP/MC 的位置，应设置在位置（靠近复位按钮一侧），即设置 DSP 工作在 MP 方式。

④ 关闭实验箱上三个开关。

⑵. 开启设备：

① 打开计算机电源。

② 打开实验箱电源开关，注意 ICETEK-VC5416-A 板上指示灯 D1 和 D2 亮。

③ 如使用 USB 型仿真器用附带的 USB 电缆连接计算机和仿真器相应接口，注意仿真器上两个指示灯均亮。

⑶. 设置 Code Composer Studio 为 Emulator 方式：

参见“Code Composer Studio 入门实验”之四.3。

⑷. 启动 Code Composer Studio

双击桌面上“CCS 2(‘C5000)”图标，启动 Code Composer Studio 2.0。

2. 打开工程并浏览程序

-打开菜单‘Project’的‘Open’项 选择 C:\5416EDULab\Lab9-DA 目录中的‘dac.pjt”。

-在项目浏览器中，双击 da.c，激活 da.c 文件，浏览该文件的内容，理解各语句作用。

3. 编译工程

单击“Project”菜单，“Rebuild All”项，编译工程中的文件，生成 dac.out 文件。

4. 下载程序

单击“File”菜单，“Load program…”项，选择 C:\5416EDULab\Lab9-DA\Debug 目录中的 dac.out 文件，通过仿真器将其下载到 5416 DSP 上。

5. 连接示波器

用信号线从实验箱底板上右侧“D/A 输出”的四个插座引线到示波器。

6. 运行并观察结果

-单击“Debug”菜单，“Run”项，运行程序；

-观察示波器上的波形。

7. 停止运行，将程序中“uDA0+=0x10; uDA0%=4096;”中的 0x10 改为 1，重新编译运行程序，观察 DACOUT1 通道上的输出波形有何改变。

8. 结束运行退出 Code Composer Studio 2.0。

五. 实验结果

-四路输出均为 0—+5V，示波器显示波形为三角波。

-由于各通道的起始值不同，所以各通道的波形的象位不同。

六. 问题与思考

请改变设计使第 2 通道输出正弦波，第 3 通道输出余弦波（可利用 C 语言的三角函数）。

实验十：自启动实验

一. 实验目的

1. 了解 TMS320VC5416 DSP 芯片的微处理器工作方式；
2. 了解 TMS320VC5416 DSP 芯片的自启动方式；
3. 学习了解 ICETEK-VC5416-A 板上 Flash 的扩展方式和特点；
4. 掌握 ICETEK-VC5416-A 板上 Flash 的烧写过程；
5. 学习自启动程序的设计。

二. 实验设备

计算机，ICETEK-VC5416-EDU 实验箱（或 ICETEK 仿真器+ICETEK-VC5416-A 系统板+相关连线及电源）。

三. 实验原理

1. MP/MC 方式

   TMS320VC5416 DSP 芯片有两种工作方式，一种为微控制器（MC）方式，一种为微处理器方式（MP）。

   两种工作方式的存储器映射有所不同：在 MC 方式下，程序区地址从 C000-FEFFh 为片内 ROM 空间，从 FF80h 开始则是位于 ROM 上的固定中断向量表。而在 MP 方式下这一空间为片外扩展存储器。

2. ICETEK-VC5416-A 板上扩展 Flash 的方式和特点

   -Flash 扩展在数据空间，地址从 8000h 开始，容量为 1M 字节，采用 8 位方式。

   -如果需要访问 Flash 存储器，MP/MC 必须为低电平。

3. 5416 自启动加载器（BootLoader）

   自启动加载器是一段固化在 5416 片内 ROM 中的程序，它主要完成在上电时从外部加载并执行用户的程序代码。加载的途径有：

   ⑴从一个外部 8 位或 16 位 EPROM 加载。

   ⑵由主处理器通过以下途径加载即

   *HPI 总线

   *8 位或 16 位并行 I/O 口

   *任何一个串行口

   *从用户定义的地址热启动

4. 自启动加载器从 EPROM 的加载过程

   ⑴在硬件复位其间，如果’C54x 的 MP/MC 引脚为低电平，则从片内 ROM 的 FF80 起执行程序，选择自启动方式。

   ⑵初始化：

   *INTM=1，关闭可屏蔽中断。

   *OVLY=1，将片内双寻址 RAM 和单寻址 RAM 映象到程序/数据空间。

   *SWWSR=7FFFh，所有程序和数据空间的操作都插入 7 个等待状态。

   *BSCR=FFFFh，设定外部存储区分区为 4K 字，当程序和数据空间切换时，插入一个等待周期。

   ⑶检查 INT2，决定是否从主机接口（HPI）加载。

   ⑷从 I/O 空间 FFFFh 读入自启动程序选择字（BRS）。BRS 的低 8 位决定了自启动加载的方式。如果低两位为 01 则选择 8 位并行 EPROM 方式，本实验采取这种方式。

⑸BRS 的高 6 位规定了 EPROM 源地址的 16 位起始地址（BRS 右移 2 位再左移 10 位得到）。按照 ICETEK-VC5416-A 板的设计，这一地址应为 8000h，即 BRS=81h。

⑹读入自启动表：在 8 位方式下，BootLoader 程序从 EPROM 的起始地址读入 4 个字节的自启动表，存放数据的顺序是：目的地址高字节，目的地址低字节，代码长度高字节，代码长度低字节。这里代码长度为实际代码字数-1。

⑺BootLoader 接着将 EPROM 中存放的代码从数据区转移到程序区，并从目的地址的首地址开始执行程序。

5. 自启动程序编制要点

程序一般先在外部扩展的 RAM 中调试好，应注意调整程序的尺寸，而且一般要连续存放，以利于进行 BootLoad。程序中的常量部分应分配到程序区存储。从生成的内存映射文件（map 文件）中读取程序的入口地址，以便在自启动模式中转移到此处运行。

四. 实验步骤

1. 实验准备

⑴. 连接设备

① 关闭计算机和实验箱电源。

② 如使用 PP 型仿真器则用附带的并口连线连接计算机并口和仿真器相应接口。

③ 检查 ICETEK-VC5416-A 板上拨动开关 MP/MC 的位置，应设置在位置（靠近复位按钮一侧），即设置 DSP 工作在 MP 方式。

④ 关闭实验箱上三个开关。

⑵. 开启设备

① 打开计算机电源。

② 打开实验箱电源开关，注意 ICETEK-VC5416-A 板上指示灯 D1 和 D2 亮。

③ 如使用 USB 型仿真器用附带的 USB 电缆连接计算机和仿真器相应接口，注意仿真器上两个指示灯均亮。

⑶. 设置 Code Composer Studio 为 Emulator 方式：

参见“Code Composer Studio 入门实验”之四.3。

⑷. 启动 Code Composer Studio 2.0

2. 打开工程并运行程序

-打开 C:\2407EDULab\Lab5-IOPort 中的工程 IOPort2.pjt

-修改工程选项：选择菜单“Project”、“Build Options”；在“Build Options for IOPort2.pjt”窗口中的“Processor Version”项后输入 548；单击“Linker”卡片，在“Map Filename”项中输入 IOPort2.map。

-重新编译工程，生成.out 文件。

-单击“File”菜单的“Open…”;将“文件类型”改成“Memory Map Files（*.map）”，选择 C:\5416EDULab\Lab5-IOPort 目录中的 mled.map 文件，“打开”；注意到文件中的“ENTRY POINT SYMBOL: "_c_int00" address: 00000175”行，其中标明了程序的入口地址为 175H。

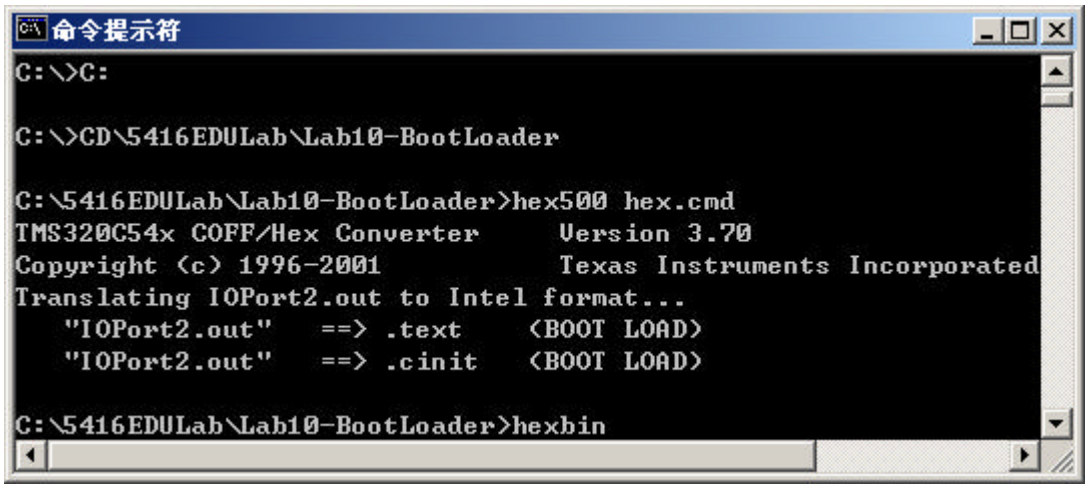
-单击“Project”、“Close”关闭工程。

-使用桌面上“我的电脑”将 C:\2407EDULab\Lab5-IOPort\Debug 目录中的 IOPort2.out 复制到 C:\2407EDULab\Lab10-BootLoader 目录中。

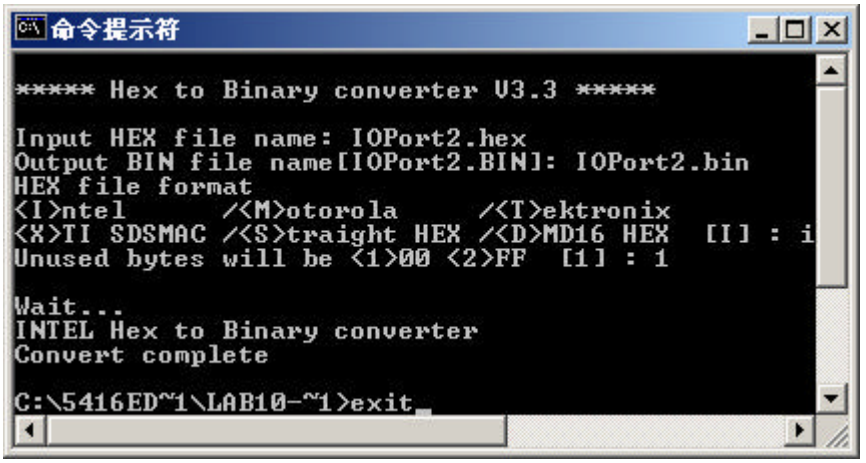
3. 制作 BIN 文件

-启动“命令提示符”：单击“开始”菜单、“程序”、“附件”、“命令提示符”。

-输入命令进行如下操作：



再进行如下操作，生成 BIN 文件：



这时在 C:\2407EDULab\Lab10-BootLoader 目录中生成了 IOPort2.bin 文件，下面将利用这个文件烧写到 ICETEK-VC5416-A 板上的 Flash，以实现自启动功能。

4. 浏览程序

-在项目浏览器中，双击 flshprog.c，激活 flshprog.c 文件，浏览该文件的内容，理解各语句作用。在有“open”的语句中应写上要烧写的.bin 文件及其路径。

-在“exit(0)”程序行上设置软件断点。

5. 编译工程并下载程序。
6. 运行程序，当程序运行到断点时，Flash 就烧写成功了。
7. 退出 CCS，关闭实验箱电源。
8. 设置 MC 方式

将 ICETEK-VC5416-A 板上 U2 的 MP/MC 设置到 “ON”的位值（远离复位按钮一侧）。

9. 测试自启动

拔掉仿真电缆（白色的），让仿真器和计算机脱开；重新打开实验箱电源；观察板上指示灯闪烁；表明烧入 Flash 的程序正在运行；按一下板上复位按钮，程序将从新运行。

10. 关闭实验箱电源，将 DSP 系统板上 MP/MC 设置到“OFF”，即 MP 方式。

五. 实验结果

在脱离计算机和仿真器的情况下，自启动程序正常工作在 MC **方式。**

六. 问题与思考

实验十一：外设控制实验—发光二极管阵列显示实验

一．实验目的

通过实验学习使用 5416 DSP 的扩展 I/O 端口控制外围设备的方法，了解发光二极管阵列的控制编程方法。

二．实验设备

计算机，ICETEK-VC5416-EDU 实验箱。

三．实验原理：

ICETEK-VC5416-A 是一块以 TMS320VC5416DSP 为核心的 DSP 扩展评估板，它通过扩展接口与实验箱的显示/控制模块连接，可以控制其各种外围设备。

发光二极管显示阵列的显示是由 I/O 扩展端口控制，DSP 须将显示的图形按列的顺序存储起来(8×8 点阵，8 个字节，高位在下方，低位在上方)，然后定时刷新控制显示。具体方法是，将以下控制字按先后顺序、每两个为一组发送到端口 0x8005，发送完毕后，隔不太长的时间(以人眼观察不闪烁的时间间隔)再发送一遍。由于位值为“0”时点亮，所以需要将显示的数据取反。

0x01,第 8 列数据取反，0x02,第 7 列数据取反，
0x04,第 6 列数据取反，0x08,第 5 列数据取反，
0x10,第 4 列数据取反，0x20,第 3 列数据取反，
0x40,第 2 列数据取反，0x80,第 1 列数据取反，

四．实验步骤

1．实验准备

⑴. 连接设备

① 关闭计算机和实验箱电源。
② 如使用 PP 型仿真器则用附带的并口连线连接计算机并口和仿真器相应接口。
③ 检查 ICETEK-VC5416-A 板上 DIP 开关 MP/MC 的位置，应设置在“OFF”位置（靠近复位按钮），即设置 DSP 工作在 MP 方式。
④ 关闭实验箱上三个开关。
⑤ 设置 ICETEK-CTR 板上拨动开关（位于板子左上角）的第 1 位 BS3 为“OFF”状态。

⑵. 开启设备

① 打开计算机电源；
② 打开实验箱电源开关，ICETEK-CTR 板上 J2、J3 灯亮；注意 ICETEK-VC5416-A 板上指示灯 D1 和 DS2 亮。
③ 如使用 USB 型仿真器用附带的 USB 电缆连接计算机和仿真器相应接口，注意仿真器上两个指示灯均亮。

⑶. 设置 Code Composer Studio 为 Emulator 方式

⑷. 启动 Code Composer Studio 2.0 for ’C5000

2．打开工程并浏览程序，工程目录为 C:\5416EDULab\Lab11-LedArray

3．编译并下载程序

4．运行程序，观察结果

如果显示出现乱码，可退出 CC 后关闭实验箱，再打开重新做一次。

5．停止程序运行并退出

五．实验结果与分析

实验结果：可以观察到发光二极管阵列显示从 0 到 9 的计数。

分析：本程序使用循环延时的方法，如果想实现较为精确的定时，可使用通用计时器，在通用计时器中断中取得延时，改变显示内容。另外本程序中 DSP 一直在做刷新显示的工作，如果使用通用计时器定时刷新显示，将能减少 DSP 用于显示的操作。适当更新显示可取得动画效果。

六.问题与思考

试设计用定时器定时刷新的程序，并显示秒计数的最低位。

实验十二：外设控制实验—液晶显示器控制显示实验

一．实验目的

通过实验学习使用 5416DSP 的扩展 I/O 端口控制外围设备的方法，了解液晶显示器的显示控制原理及编程方法。

二．实验设备

计算机，ICETEK-VC5416-EDU 实验箱。

三．实验原理

ICETEK-VC5416-A 是一块以 TMS320VC5416DSP 为核心的 DSP 扩展评估板，它通过扩展接口与实验箱的显示/控制模块连接，可以控制其各种外围设备。

液晶显示模块的访问、控制是由 5416DSP 对扩展 I/O 接口的操作完成。

控制 I/O 口的寻址：命令控制 I/O 接口的地址为 0x8001，数据控制 I/O 接口的地址为 0x8003 和 0x8004，辅助控制 I/O 接口的地址为 0x8002。

显示控制方法：

-液晶显示模块中有两片显示缓冲存储器，分别对应屏幕显示的象素，向其中写入数值将改变显示，写入“1”则显示一点，写入“0”则不显示。其地址与象素的对应方式如下：

左侧显示内存						右侧显示内存					
Y=	0	1	···	62	63	0	1	···	62	63	行号
X=0	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	0 ↓ 7
↓	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	8 ↓ 55
X=7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	DB0 ↓ DB7	56 ↓ 63

-发送控制命令：向液晶显示模块发送控制命令的方法是通过向命令控制 I/O 接口写入命令控制字，然后再向辅助控制接口写入 0。下面给出的是基本命令字、解释和 C 语言控制语句举例。

.显示开关：0x3f 打开显示；0x3e 关闭显示；

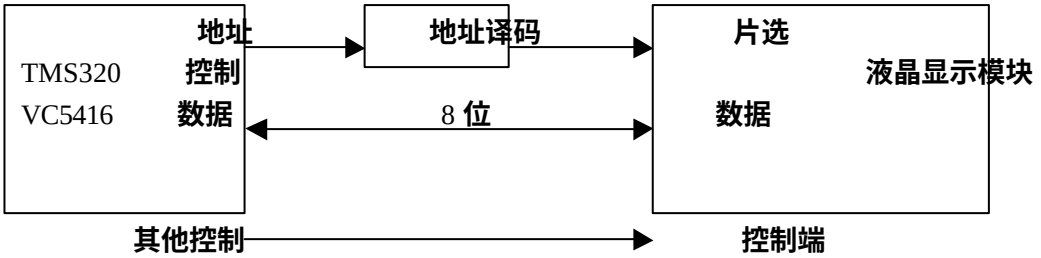
.设置显示起始行：0x0c0+起始行取值，其中起始行取值为 0 至 63；

.设置操作页：0x0b8+页号，其中页号取值为 0-7；

.设置操作列：0x40+列号，其中列号为取值为 0-63；

-写显示数据：在使用命令控制字选择操作位置(页数、列数)之后，可以将待显示的数据写入液晶显示模块的缓存。将数据发送到相应数据控制 I/O 接口即可。

*液晶显示器与 DSP 的连接：



*数据信号的传送：由于液晶显示模块相对运行在 8MHz 主频下的 DSP 属于较为慢速设备，连接时需要考虑数据线上信号的等待问题；

*电平转换：由于 DSP 为 3.3V 设备，而液晶显示模块属于 5V 设备，所以在连接控制线、数据线时需要加电平隔离和转换设备，如：ICETEK-CTR 板上使用了 74LS245。

四．实验步骤：

1．实验准备

⑴连接设备

① 关闭计算机和实验箱电源。

② 如使用 PP 型仿真器则用附带的并口连线连接计算机并口和仿真器相应接口。

③ 检查 ICETEK-VC5416-A 板上 DIP 开关 MP/MC 的位置，应设置在“ OFF ”位置（靠近复位按钮），即设置 DSP 工作在 MP 方式。

④ 关闭实验箱上三个开关。

⑤ 设置ICETEK-CTR板上拨动开关（位于板子左上角）的第 1 位 BS3 为‘ OFF ”状态。

⑵开启设备

①打开计算机电源；

②打开实验箱电源开关，ICETEK-CTR 板上 J2、J3 灯亮；

③打开 ICETEK-LF2407-A 板上电源开关，注意板上指示灯 DS1 灭、DS2 和 DS3 亮；

④如使用 USB 型仿真器用附带的 USB 电缆连接计算机和仿真器相应接口，注意仿真器上两个指示灯均亮。

⑶设置 Code Composer Studio 为 Emulator 方式。

⑷启动 Code Composer Studio 2.0

2．打开工程并浏览程序，工程目录为 C:\5416EDULab\Lab12-LCD

3．编译并下载程序

4．运行程序，观察结果

5．将内层循环中的“ port8003=ledkey[nCount][i]; ”语句改为“ port8004=ledkey[nCount][i];”，重复步骤 3-4，实现在屏幕右侧显示。

6．更改程序中对页、列的设置，实现不同位置的显示。

7．自己设计一些控制语句，实现不同显示效果。

8．停止程序运行并退出

五．实验结果与分析：

实验结果：可以观察到液晶显示从 0 到 9 的计数。

分析：灵活使用控制字，可以实现复杂多变的显示。当使用点阵图形显示时需要在 DSP 内存中建立图形存储缓冲；适当更新显示可取得动画效果。在实际生活中观察点阵显示的霓虹灯广告、交通指示牌、报站牌等领会这种控制的具体应用。

六．问题与思考

试设计程序在液晶显示屏上显示计时时钟，精确到秒，形式为“ 时时：分分：秒秒 ”。

实验十三： 外设控制实验—键盘输入实验

一．实验目的

通过实验学习使用 5416DSP 的扩展 I/O 端口接收外围设备信息的方法，了解键盘的使用原理及编程方法。

二．实验设备

计算机，ICETEK-VC5416-EDU 实验箱。

三．实验原理：

ICETEK-VC5416-A 是一块以 TMS320VC5416DSP 为核心的 DSP 扩展评估板，它通过扩展接口与实验箱的显示/控制模块连接，可以控制其各种外围设备，也可以接收外设发送的各种数据、信息。

键盘的扫描码由 DSP 的 I/O 扩展地址 0x8001 给出，当有键盘输入时，读此端口得到扫描码，当无键被按下时读此端口的结果为 0。各按键的扫描码排列如下所示。

0x18,0x14,0x12,0x11
0x28,0x24,0x22,0x21
0x48,0x44,0x42,0x41
0x88,0x84,0x82,0x81

四．实验步骤：

1．实验准备

⑴连接设备

①关闭计算机和实验箱电源。

②如使用 PP 型仿真器则用附带的并口连线连接计算机并口和仿真器相应接口。

③检查 ICETEK-VC5416-A 板上 DIP 开关 MP/MC 的位置，应设置在“OFF”位置（靠近复位按钮），即设置 DSP 工作在 MP 方式。

④关闭实验箱上三个开关；

⑤设置 ICETEK-CTR 板上拨动开关（位于板子左上角）的第 1 位 BS3 为“OFF”状态。

⑵开启设备

①打开计算机电源；

②打开实验箱电源开关，ICETEK-CTR 板上 J2、J3 灯亮；注意 ICETEK-VC5416-A 板上指示灯 D1 和 DS2 亮。

③如使用 USB 型仿真器用附带的 USB 电缆连接计算机和仿真器相应接口，注意仿真器上两个指示灯均亮。

⑶设置 Code Composer Studio 为 Emulator 方式

⑷启动 Code Composer Studio

2．打开工程并浏览程序，工程目录为 C:\5416EDULab\Lab13-Key

3．编译并下载程序

4．运行程序，观察结果

按下键盘上一些健，观察显示是否正确。

5．停止程序运行并退出

五．实验结果

实验结果：可以观察到发光二极管阵列显示键盘输入字符。

分析：在程序中加入分支语句实现对不同键盘输入值的处理或支持控制型按键；修改程序中键值查找表可实现按键的重新布局或修改。

六．问题与思考

实验十四：外设控制实验—音频信号发生实验

一．实验目的

通过实验学习使用 5416DSP 的扩展 I/O 端口控制外围设备信息的方法，掌握使用 5416DSP 通用计时器的控制原理及中断服务程序的编程方法；了解蜂鸣器发声原理和音乐发生方法。

二．实验设备

计算机，ICETEK-VC5416-EDU 实验箱。

三．实验原理：

ICETEK-VC5416-A 是一块以 TMS320VC5416DSP 为核心的 DSP 扩展评估板，它通过扩展接口与实验箱的显示/控制模块连接，可以控制其各种外围设备。

蜂鸣器由 DSP 通用 I/O 管脚 BDX0 输出控制，可将此管脚上的频率输出转换成声音输出。5416 的通用 I/O 口控制蜂鸣器的输出频率。

控制的方法是使用 DSP 通用定时器设置 BDX0 以一定的频率改变高低状态，输出方波。对于通用定时器周寄存器的设置，计数值为所需频率计数值的二分之一。

音乐的频率(C 调)：

C	D	E	F	G	A	B	^C
1	2	3	4	5	6	7	^1

C: 264, 297, 330,352, 396, 440,495, 528

*蜂鸣器的连接：由于选用的蜂鸣器所需电流较小，所以采用将 DSP 通用 I/O 引脚直接驱动的方式。

四．实验步骤

1.实验准备

⑴ 连接设备

①关闭计算机和实验箱电源。

②如使用 PP 型仿真器则用附带的并口连线连接计算机并口和仿真器相应接口。

③检查 ICETEK-VC5416-A 板上拨动开关 MP/MC 的位置，应设置在位置（靠近复位按钮一侧），即设置 DSP 工作在 MP 方式。

④关闭实验箱上三个开关。

⑵ 开启设备

①打开计算机电源。

② 打开实验箱电源开关，ICETEK-CTR 板上 J2、J3 灯亮；注意 ICETEK-VC5416-A 板上指示灯 D1 和 D2 亮。

③如使用 USB 型仿真器用附带的 USB 电缆连接计算机和仿真器相应接口，注意仿真器上两个指示灯均亮。

⑶ 设置 Code Composer Studio 为 Emulator 方式

⑷ 启动 Code Composer Studio 2.0

2.打开工程并浏览程序，工程目录为 C:\5416EDULab\Lab14-Speaker

3.编译并下载程序

4.运行程序

5.将语句“ delay(music[nCount][1]); ”改为“ delay(music[nCount][1]/2); ”，重复步骤 5-7，体会音乐的节奏快了一倍。

五．实验结果

实验结果：可以听到蜂鸣器发出的音乐声(6123216)。

分析：程序中使用循环延时的方法掌握节拍，可考虑使用定时器计数改变音符，更复杂的方法可以产生语音效果。

六．问题与思考

结合键盘控制程序，设计一个按键“ 弹琴 ”的程序。

实验十五：外设控制实验—步进电机控制实验

一．实验目的

通过实验学习使用 5416DSP 的扩展 I/O 端口控制外围设备信息的方法，掌握使用 5416DSP 通用计时器的控制原理及中断服务程序的编程方法；了解步进电机的控制方法。

二．实验设备

计算机，ICETEK-VC5416-EDU 实验箱。

三．实验原理

ICETEK-VC5416-A 是一块以 TMS320VC5416DSP 为核心的 DSP 扩展评估板，它通过扩展接口与实验箱的显示/控制模块连接，可以控制其各种外围设备。

步进电机是由 DSP 通用 I/O 管脚输出直接控制。步进电机的起动频率大于 500PPS(拍每秒)，空载运行频率大于 900PPS。5416 的通用 I/O 口 BFSR1 控制电机的转动频率，BCLKXR0 控制转动方向。

控制的方法是使用 DSP 通用定时器设置 BFSR1 以一定的频率改变高低状态，输出方波，设置 BCLKXR0 为高电平则顺时针转动，低电平为逆时针转动。

四．实验步骤：

1.实验准备

⑴连接设备

①关闭计算机和实验箱电源。

②如使用 PP 型仿真器则用附带的并口连线连接计算机并口和仿真器相应接口。

③检查 ICETEK-VC5416-A 板上 DIP 开关 MP/MC 的位置，应设置在“OFF”位置（靠近复位按钮），即设置 DSP 工作在 MP 方式。

④关闭实验箱上三个开关。

⑵开启设备

①打开计算机电源。

②打开实验箱电源开关，ICETEK-CTR 板上 J2、J3 灯亮；打开位于实验箱中部的电机电源开关，ICETEK-CTR 板上 J4 灯亮；注意 ICETEK-VC5416-A 板上指示灯 D1 和 DS2 亮。

③如使用 USB 型仿真器用附带的 USB 电缆连接计算机和仿真器相应接口，注意仿真器上两个指示灯均亮。

⑶设置 Code Composer Studio 为 Emulator 方式

⑷启动 Code Composer Studio 2.0

2.打开工程并浏览程序，工程目录为 C:\5416EDULab\Lab15-Motor

3.编译并下载程序

4.运行程序，观察结果

电机转动时按下 ICETEK-CTR 板上的键盘“0”和“1” 键，控制电机转动方向。

5.停止程序运行并退出

五．实验结果

实验结果：可以看到显示/控制模块上的电机指针在转动，使用“0”和“1”键可控制

其转动方向。

分析：使用优化的程序可控制电机更平滑地转动，平滑地改变频率可使马达的摆动减到最小。

六．问题与思考

实验十六：有限冲击响应滤波器（FIR）算法实验

一、实验目的

1、握用窗函数法设计 FIR 数字滤波器的原理和方法；

2、熟悉线性相位 FIR 数字滤波器特性；

3、了解各种窗函数对滤波器特性的影响。

二、实验设备

计算机，ICETEK-VC5416-EDU 实验箱（或 ICETEK 仿真器+ICETEK-VC5416-A 系统板+相关连线及电源）。

三、实验原理

1.FIR 的原理和参数生成公式

⑴N 阶有限冲激响应滤波器（FIR）公式：

$$y(n)=\sum_{k=0}^{N/2-1}h(k)[x(n-k)+x(n-(N-1+k))] \qquad \text{公式1-1}$$

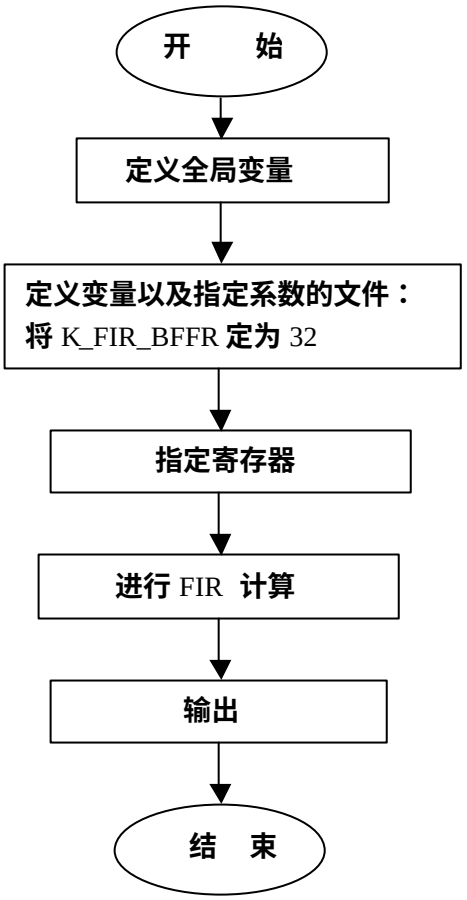
N=0,1,2...

⑵FIR 设计原理：根据系数 h 是偶对称为了简化运算产生如下计算方法

如果一个 FIR 滤波有一个冲激响应，h(0)，h(1)，…，h(N-1),和 x(n)描绘输入的时常滤波 n，输出滤波 y(n)的 n 给出以下方程式：

y(n)=h(0)x(n)+h(1)x(n-1)+h(2)x(n-2)+…+h(N-1)x[n-(n-1)] 公式1-2

2.程序流程图



3.程序的自编函数及其功能

1） .global start,fir 设定全局变量。

2）COFF_FIR_START:.sect "coff_fir"

.include "lowpass.inc" （设定系数文件）

提示：“lowpass.inc”提供低通系数，用户可参照说明手册最后的练习实现高通及带通。

3）K_FIR_BFFR .set 32 （滤波阶数）

4）d_filin （存放输入波形）

5）d_filout （存放输出波形）

6）指定寄存器：

指定 AR4 为 FIR_DATA_P 数据寄存器。

指定 AR6 为 INBUF_P 输入数据寄存器。

指定 AR7 为 OUTBUF_P 输出数据寄存器。

7）汇编程序部分说明：

start 部分：程序初始化部分指定寄存器，清空寄存器。

fir_loop 部分：循环调入输入数据，并调用子程序 fir 进行计算。

main_end 部分：跳转至循环部分。

fir 部分：子程序部分。

其中 fir_task 部分：进行计算并返回计算结果。

四、实验步骤

1.实验准备

⑴. 连接设备

① 关闭计算机和实验箱电源。

② 如使用 PP 型仿真器则用附带的并口连线连接计算机并口和仿真器相应接口。

③ 检查 ICETEK-VC5416-A 板上 DIP 开关 MP/MC 的位置，应设置在“ OFF ”位置（靠近复位按钮），即设置 DSP 工作在 MP 方式。

④ 关闭实验箱上三个开关。

⑵. 开启设备

① 打开计算机电源。

② 打开实验箱电源开关，注意 ICETEK-VC5416-A 板上指示灯 D1 和 DS2 亮。

③ 如使用 USB 型仿真器用附带的 USB 电缆连接计算机和仿真器相应接口，注意仿真器上两个指示灯均亮。

⑶. 设置 Code Composer Studio 为 Emulator 方式：

参见“ CodeComposer Studio 入门实验 ”之四.3。

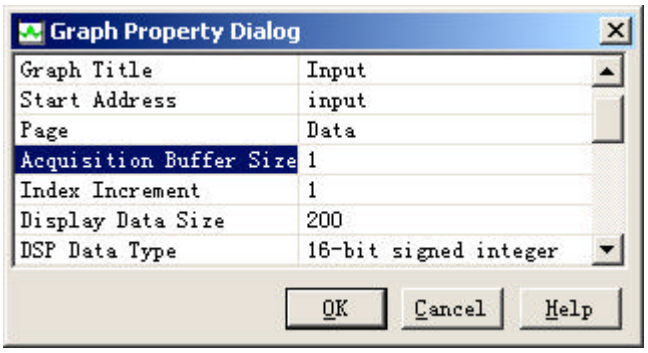
⑷. 启动 Code Composer Studio 2.0

2.打开工程，浏览程序，工程目录为 C:\5416EDULab\Lab16-FIR

3.编译并下载程序

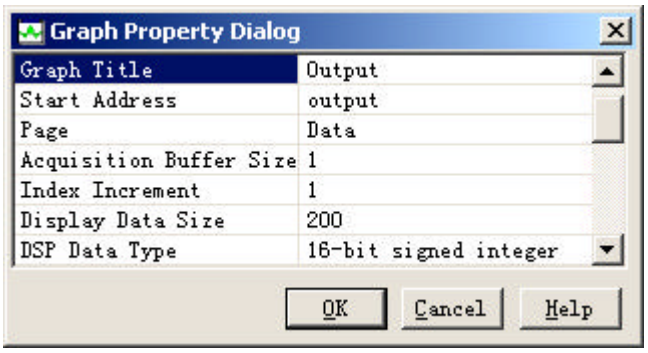
4.打开观察窗口

-选择菜单“ View ”、“ Graph ”、“ Time/Frequency… ”进行如下设置：



-在弹出的图形窗口中单击鼠标右键，选择“Clear Display”。

-选择菜单“View”、“Graph”、“Time/Frequency…”进行如下设置：



-在弹出的图形窗口中单击鼠标右键，选择“Clear Display”。

5.设置断点

-在标号“fir_loop”下面的“NOP”语句设置软件断点（Toggle breakpoint）和探针（Toggle Probe Point）。

-选择菜单“File”、“File I/O…”；单击“Add File”按钮，选择 C:\5416EDULab\Lab16-FIR\lowpass\64300.dat 文件，单击“打开”按钮；在“Adress”中输入 d_filin，在“Length”中输入 1；在“Warp Around”项前面加上选中符号；单击“Add Probe Point”按钮。

-单击“Probe Point”列表中的“FIR.asm line 38”行；在“Connect”项选择“FILE IN: C:\..\64300.dat”；单击“Replace”按钮；单击“确定”按钮。

-单击“确定”按钮。

6.运行并观察结果

(1) 选择“Debug”菜单的“Animate”项，或按 F12 键运行程序；

(2) 观察“Input”、“Output”窗口中时域图形；观察滤波效果；

(3) 鼠标右键单击“Input”和“Output”窗口，选择“Properties…”项，设置“Display Type”为“FFT Magitude”，再单击“OK”按钮结束设置；

(4) 按 F12 运行程序

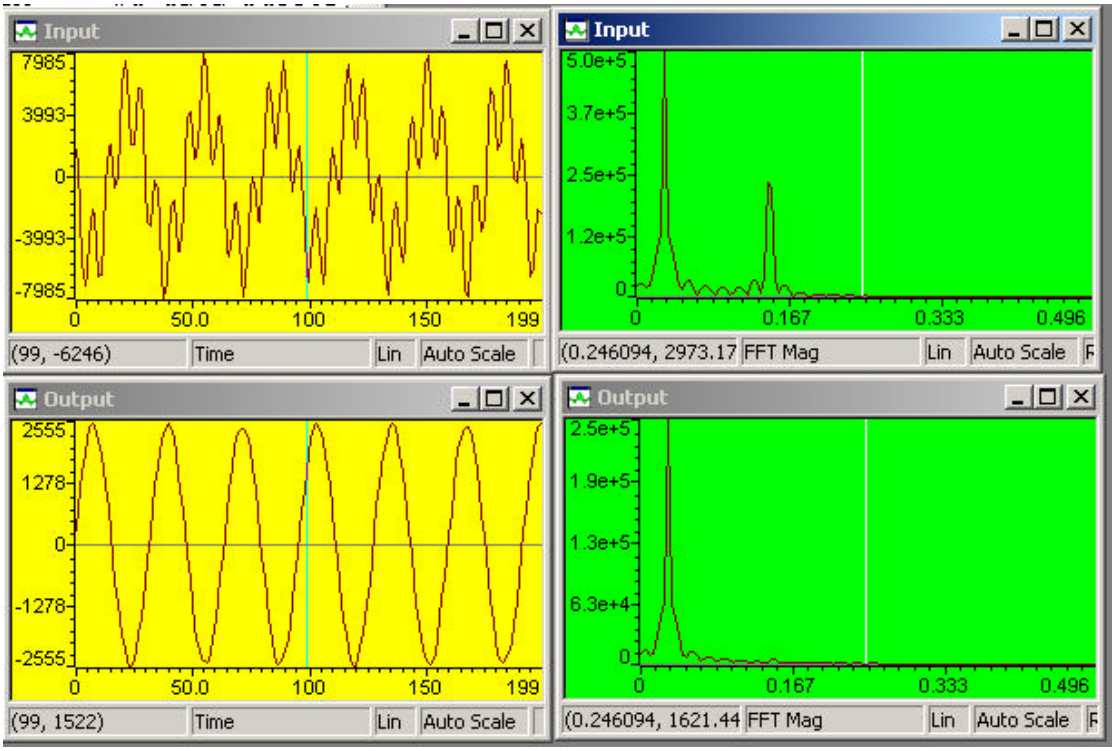
(5) 观察“Input”、“Output”窗口中频域图形；理解滤波效果。

7.停止程序运行并退出。

五、实验结果

输入波形为一个低频率的正弦波与一个高频的正弦波叠加而成。

实验现象图：



通过观察频域和时域图，得知：输入波形中的低频波形通过了滤波器，而高频部分则被滤除。

六、问题与思考

试选用合适的高通滤波参数滤掉实验的输入波形中的低频信号。

实验十七：快速命傅立叶变换（FFT）算法实验

一、实验目的

1、掌握用窗函数法设计 FFT 快速傅里叶的原理和方法；

2、熟悉 FFT 快速傅里叶特性；

3、了解各种窗函数对快速傅里叶特性的影响。

二、实验设备

计算机，ICETEK-VC5416-EDU 实验箱（或 ICETEK 仿真器+ICETEK-VC5416-A 系统板+相关连线及电源）。

三、实验原理

1.FFT 的原理和参数生成公式

$$x(k)=\sum_{r=0}^{\frac{N}{2}-1}x_1(r)W_{\frac{N}{2}}^{rk}+W_N^k\sum_{r=0}^{\frac{N}{2}-1}x_2(r)W_{\frac{N}{2}}^{rk}=X_1(k)+W_N^kX_2(k)$$

公式（1）FFT 运算公式

FFT 并不是一种新的变换，它是离散傅立叶变换（DFT）的一种快速算法。由于我们在计算 DFT 时一次复数乘法需用四次实数乘法和二次实数加法；一次复数加法则需二次实数加法。每运算一个 X（k）需要 4N 次复数乘法及 2N+2（N-1）=2（2N-1）次实数加法。所以整个 DFT 运算总共需要 4N^2 次实数乘法和 N*2(2N-1)=2N(2N-1)次实数加法。如此一来，计算时乘法次数和加法次数都是和 N^2 成正比的，当 N 很大时，运算量是可观的，因而需要改进对 DFT 的算法减少运算速度。

根据傅立叶变换的对称性和周期性，我们可以将 DFT 运算中有些项合并。

我们先设序列长度为 N=2^L，L 为整数。将 N=2^L 的序列 x(n)(n=0,1,……，N-1)，按 N 的奇偶分成两组，也就是说我们将一个 N 点的 DFT 分解成两个 N/2 点的 DFT，他们又从新组合成一个如下式所表达的 N 点 DFT：

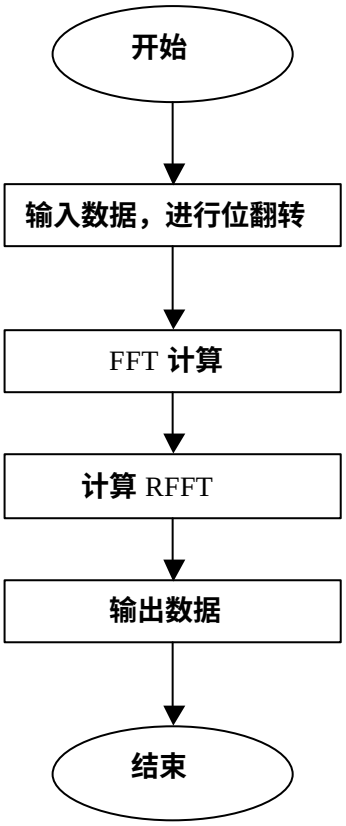
$$x(k)=\sum_{r=0}^{\frac{N}{2}-1}x_1(r)W_{\frac{N}{2}}^{rk}+W_N^k\sum_{r=0}^{\frac{N}{2}-1}x_2(r)W_{\frac{N}{2}}^{rk}=X_1(k)+W_N^kX_2(k)$$

一般来说，输入被假定为连续的。当输入为纯粹的实数的时候，我们就可以利用左右对称的特性更好的计算 DFT。

我们称这样的 RFFT 优化算法是包装算法：首先 2N 点实数的连续输入称为“进包”。其次 N 点的 FFT 被连续被运行。最后作为结果产生的 N 点的合成输出是“打开”成为最初的与 DFT 相符合的 2N 点输入。

使用这战略，我们可以划分 FFT 的大小，它有一半花费在包装输入 O（N）的操作和打开输出上。这样的 RFFT 算法和一般的 FFT 算法同样迅速，计算速度几乎都达到了两次 DFT 的连续输入。下列一部分将描述更多的在 TMS320C54x 上算法和运行的细节。

2.程序流程图



3.程序的自编函数及其功能

rfft_task();

功能：在 C 语言中进行汇编调用。

四、实验步骤

1、实验准备

(1) 连接设备

① 关闭计算机和实验箱电源。

② 如使用 PP 型仿真器则用附带的并口连线连接计算机并口和仿真器相应接口。

③ 检查 ICETEK-VC5416-A 板上 DIP 开关 MP/MC 的位置，应设置在“ OFF ”位置（靠近复位按钮），即设置 DSP 工作在 MP 方式。

④ 关闭实验箱上三个开关。

(2) 开启设备

①打开计算机电源。

②打开实验箱电源开关，注意 ICETEK-VC5416-A 板上指示灯 D1 和 DS2 亮。

③使用 USB 型仿真器用附带的 USB 电缆连接计算机和仿真器相应接口，注意仿真器上两个指示灯均亮。

(3) 设置 Code Composer Studio 为 Emulator 方式：

参见“ Code Composer Studio 入门实验 ”之四.3。

(4) 启动 Code Composer Studio

2、打开工程，浏览程序，工程目录为 C:\5416EDULab\Lab17-FFT

3、编译并下载程序

4、加入断点，指定输入数据文件

-打开工程中文件 rfft.asm

-在第一个“ NOP ”语句上加探针（Probe Point）

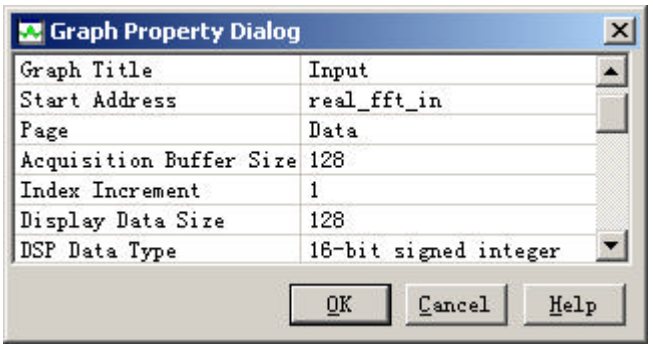
-在第二个“ NOP ”语句上加软件断点（Break Point）

-在最后一个“ NOP ”语句上加软件断点（Break Point）

-

5、打开观察窗口

-选择菜单“ View ”、“ Graph ”、“ Time/Frequency… ”进行如下设置：



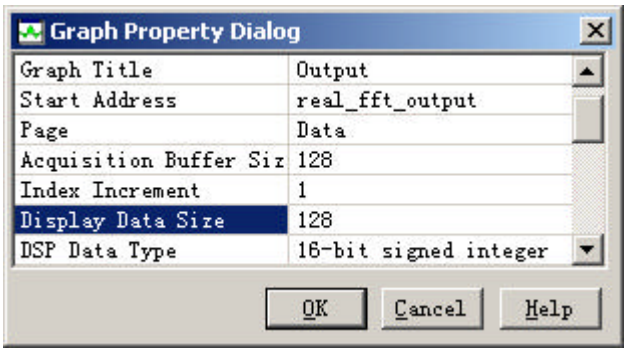
-在打开的窗口中单击鼠标右键，选择“ Clear Display ”

-选择菜单“ File ”、“ File I/O… ”；单击“ Add File ”按钮，选择 C:\5416EDULab\Lab17-FFT\1.DAT 文件，单击“打开”按钮；在“ Adress ”中输入 real_fft_in，在“ Length ”中输入 128；在“ Warp Around ”项前面加上选中符号；单击“ Add Probe Point ”按钮。

-单击“ Probe Point ”列表中的“ rfft.asm line 34 ”行；在“ Connect ”项选择“ FILE IN: C:\..\1.DAT ”；单击“ Replace ”按钮；单击“确定”按钮。

-单击“确定”按钮。

-再次选择菜单“ View ”、“ Graph ”、“ Time/Frequency… ”进行如下设置：



-在打开的窗口中单击鼠标右键，选择“ Clear Display ”

6．按 F5 或选择 debug 菜单下的 run 来运行程序；当程序停止在软件断点时可在“ Input ”窗口中观察到输入波形，为一正弦波。

7．按 F5 或选择 debug 菜单下的 run 运行程序；当程序停止在软件断点时可在“ Output ”窗口中观察到输出波形。

8．重复第 5 步，将输入波形文件改为 2.DAT，选择“ Debug ”菜单“ Restart ”项，程序

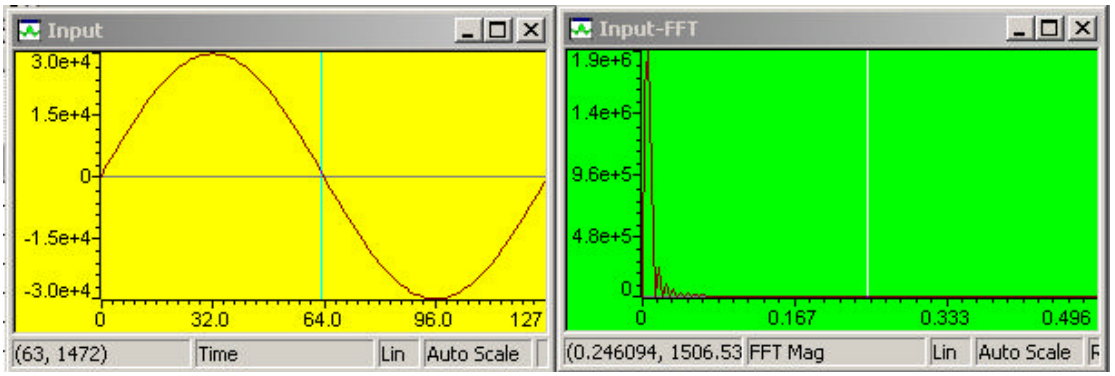
将重新开始，再做第 6，7 步。

五、实验结果

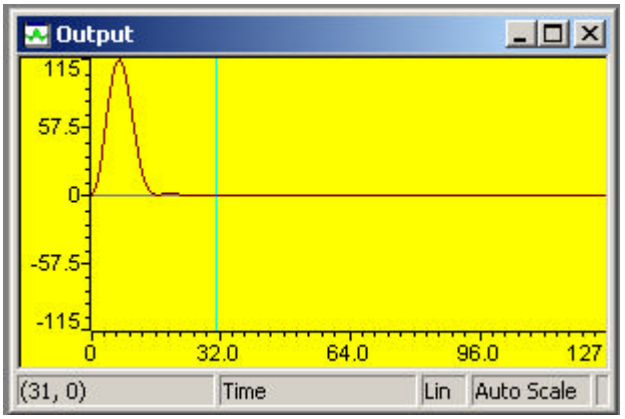
输入波形为一个低频率的正弦波，输出波形为此波形经 fft 运算后的显示。

*步骤 7 实验现象图：

-输入波形时域图和频域图：

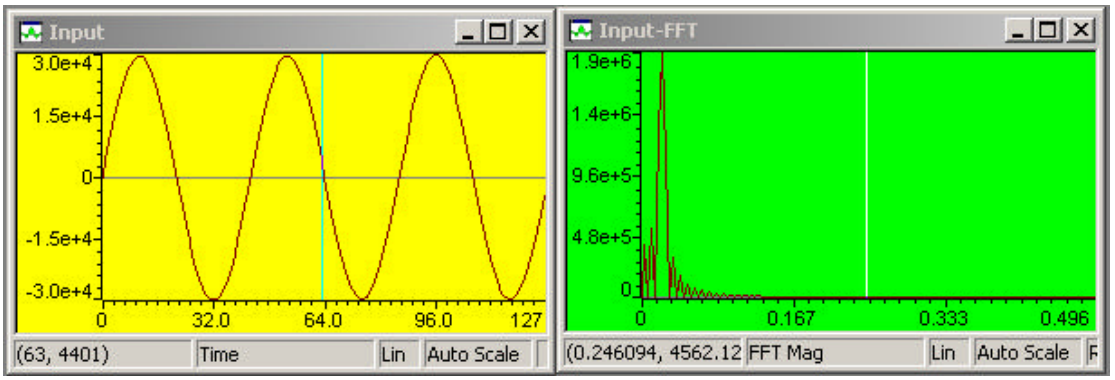


-输出波形图：

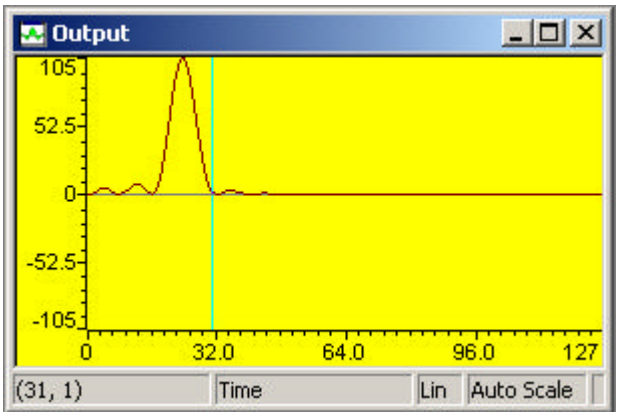


*步骤 8 实验现象图：

-输入波形时域图和频域图：



-输出波形图：



通过观察频域和时域图，可以验证 fft 运算后的结果是否正确。

六、问题与思考

试选用不同点数的 fft 运算法则来运算。

实验十八：卷积算法实验

一．实验目的

1、掌握用窗函数法设计卷积算法的原理和方法；

2、熟悉卷积算法特性；

3、了解各种窗函数对卷积算法的影响。

二．实验设备

计算机，Code Composer Studio 2.0 for ’C5000 系统。

三．实验原理

1.卷积的基本原理和公式

卷集和：对离散系统“卷积和”也是求线性时不变系统输出响应（零状态响应）的主要方法。

卷积和的运算在图形表示上可分为四步：

$$Y(n) = \sum_{m=-\infty}^{\infty} X(m)h(n-m) = X(n) * h(n)$$

1）翻褶　先在哑变量坐标 M 上作出 x(m)和 h(m)，将 m=0 的垂直轴为轴翻褶成 h(-m)。

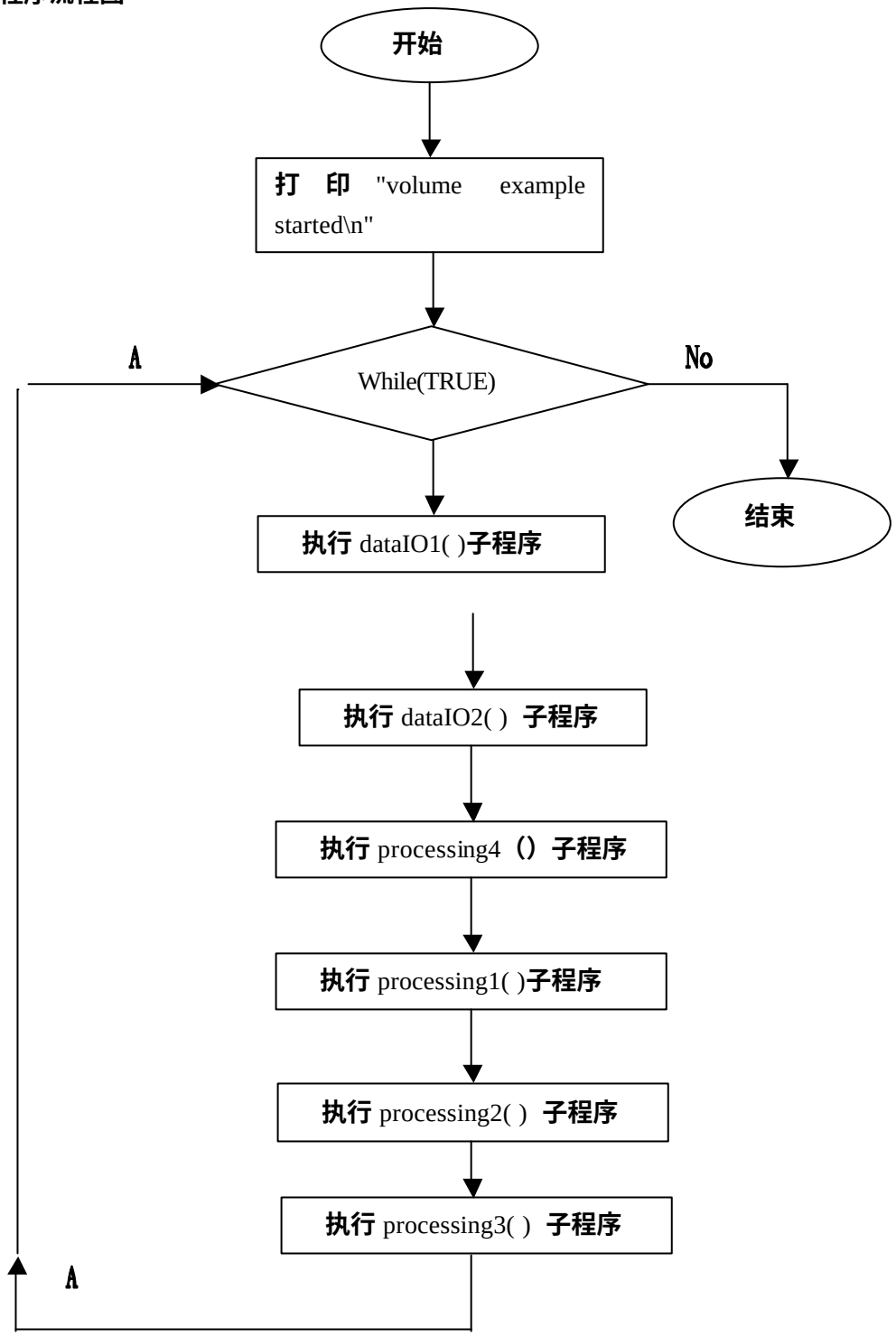
2）移位　将 h(-m)移位 n，即得 h(n-m)。当 n 为正整数时，右移 n 位。当 n 为负整数时，左移 n 位。

3）相乘　再将 h(n-m)和 x(m)的相同 m 值的对应点值相乘。

4）相加　把以上所有对应点的乘积叠加起来，即得 y(n)值。

依上法，取 n=…,-2,-1,0,1,2,3,…各值，即可得全部 y(n)值。

2.**程序流程图**



3.**程序的自编函数及其功能**

1）processing1(int *input2, int *output2)

调用形式：processing1(int *input2, int *output2)

参数解释：intput2、output2 **为两个整型指针数组。**

返回值解释：返回了一个“TREN”，**让主函数的** while **循环保持连续。**

功能说明：对输入的 input2 buffer **波形进行截取** m **点，再以零点的** Y **轴为对称轴进行翻褶，把生成的波形上的各点的值存入以** OUTPUT2 **指针开始的一段地址空间中。**

2）processing2(int *output2, int *output3)

调用形式：processing2(int *output2, int *output3)

参数解释：output2、output3 **为两个整型指针数组。**

返回值解释：返回了一个“ TREN **”，让主函数的** while **循环保持连续。**

功能说明：对输出的 output2 buffer **波形进行作** n **点移位，然后把生成的波形上的各点的值存入以** OUTPUT3 **指针开始的一段地址空间中。**

3） processing3(int *input1,int *output2,int *output4)

调用形式：processing3(int *input1,int *output2,int *output4)

参数解释：output2、output4、input1 **为三个整型指针数组。**

返回值解释：返回了一个“ TREN **”，让主函数的** while **循环保持连续。**

功能说明：对输入的 input2 buffer **波形和输入的** input1 buffer **作卷积和运算，然后把生成的波形上的各点的值存入以** OUTPUT4 **指针开始的一段地址空间中。**

4） processing4(int *input2,int *output1)

调用形式：processing4(int *input2,int *output1)

参数解释：output1、input2 **为两个整型指针数组。**

返回值解释：返回了一个“ TREN **”，让主函数的** while **循环保持连续。**

功能说明：对输入的 input2 buffer **波形截取** m **点，然后把生成的波形上的各点的值存入以** OUTPUT1 **指针开始的一段地址空间中。**

四、实验步骤

1、实验准备

(1) **连接设备**

①**关闭计算机和实验箱电源。**

②**如使用** PP **型仿真器则用附带的并口连线连接计算机并口和仿真器相应接口。**

③**检查** ICETEK-VC5416-A **板上** DIP **开关** MP/MC **的位置，应设置在“** OFF **”位置（靠近复位按钮），即设置** DSP **工作在** MP **方式。**

④**关闭实验箱上三个开关。**

(2) **开启设备**

① **打开计算机电源。**

② **打开实验箱电源开关，**

③ **如使用** USB**型仿真器用附带的** USB **电缆连接计算机和仿真器相应接口，注意仿真器上两个指示灯均亮。**

(3) **设置** Code Composer Studio **为** Simulator **方式：**

参见“ Code Composer Studio **入门实验”之四**.2**。**

(4) **启动** Code Composer Studio 2.0

2. **打开工程，浏览程序，工程目录为** C:\5416EDULab\Lab18-Convolve
3. **编译并下载程序**
4. **设置输入数据文件**

请在 c **程序中的如下两行上设置** probe point**：**

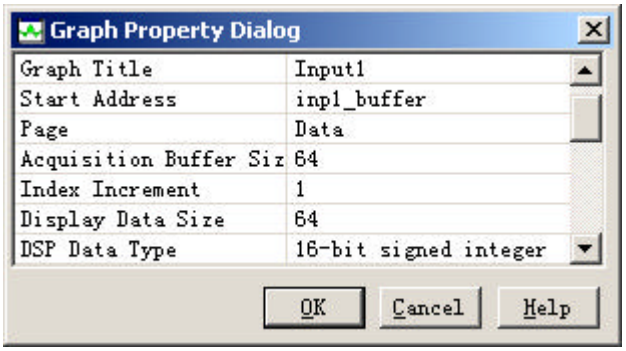
```
dataIO1();
 dataIO2();
```

设置方法是把光标指示到这一行上，按鼠标右键，从显示的菜单上分别选择 probe point。

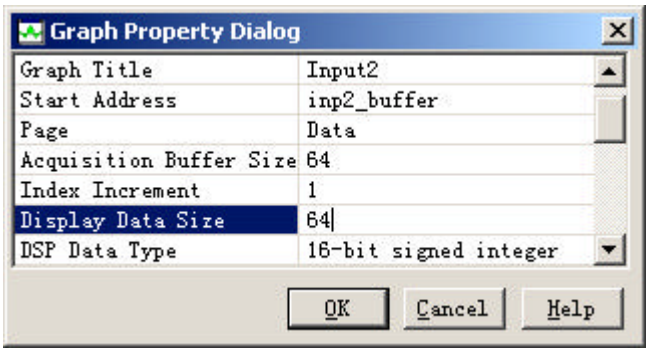
在 c 程序的“ dataIO1(); ”行上设置 break point。

5. 打开观察窗口

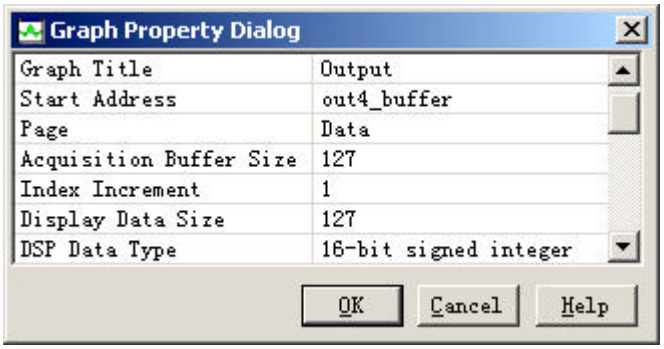
-选择菜单“ View ”、“ Graph ”、“ Time/Frequency… ”进行如下设置：



-选择菜单“ View ”、“ Graph ”、“ Time/Frequency… ”进行如下设置：



-选择菜单“ View ”、“ Graph ”、“ Time/Frequency… ”进行如下设置：



-在弹出的三个图形窗口中单击鼠标右键，选择“ Clear Display ”。

6. 设置波形输入文件

-选择“ File ”菜单中的“ File I/O… ”，打开“ File I/O ”窗口；单击“ Add File ”按钮，在“ File Input ”窗口中选择 C:\5416EDULab\Lab18-Convolve 目录下的 sin.dat 文件，单击“ 打开 ”按钮；在“ Address ”项中输入 inp1_buffer，在“ Length ”项中输入 32，在“ Warp Around ”项前加上选择标记，单击“ Add Probe Point ”按钮；

-在“ Break/Probe/Profile Points ”窗口中单击“ Probe Point ”列表中的“ Convolve.c line52 → No Connection ”，再单击“ Connect ”项尾部的展开按钮，在显示的展开式列表中选择列表末尾的“ FILE IN:C:\..\SIN.DAT ”，单击“ Replace ”按钮，单击“ 确定 ”按钮。

-在“File I/O”窗口中单击“确定”，完成设置。

-选择“File”菜单中的“File I/O…”，打开“File I/O”窗口；单击“Add File”按钮，在“File Input”窗口中选择 C:\5416EDULab\Lab19-Convolve 目录下的 sin.dat 文件，单击“打开”按钮；在“Address”项中输入 inp2_buffer，在“Length”项中输入 32，在“Warp Around”项前加上选择标记，单击“Add Probe Point”按钮；

-在“Break/Probe/Profile Points”窗口中单击“Probe Point”列表中的“Convolve.c line53 → No Connection”，再单击“Connect”项尾部的展开按钮，在显示的展开式列表中选择列表末尾的“FILE IN:C:\..\SIN.DAT”，单击“Replace”按钮，单击“确定”按钮。

-在“File I/O”窗口中单击“确定”，完成设置。

7. 运行程序，观察结果

-按 F5 键运行程序，待程序停留在软件断点；观察刚才打开的三个图形窗口，其中显示的是输入和输出的时域波形；

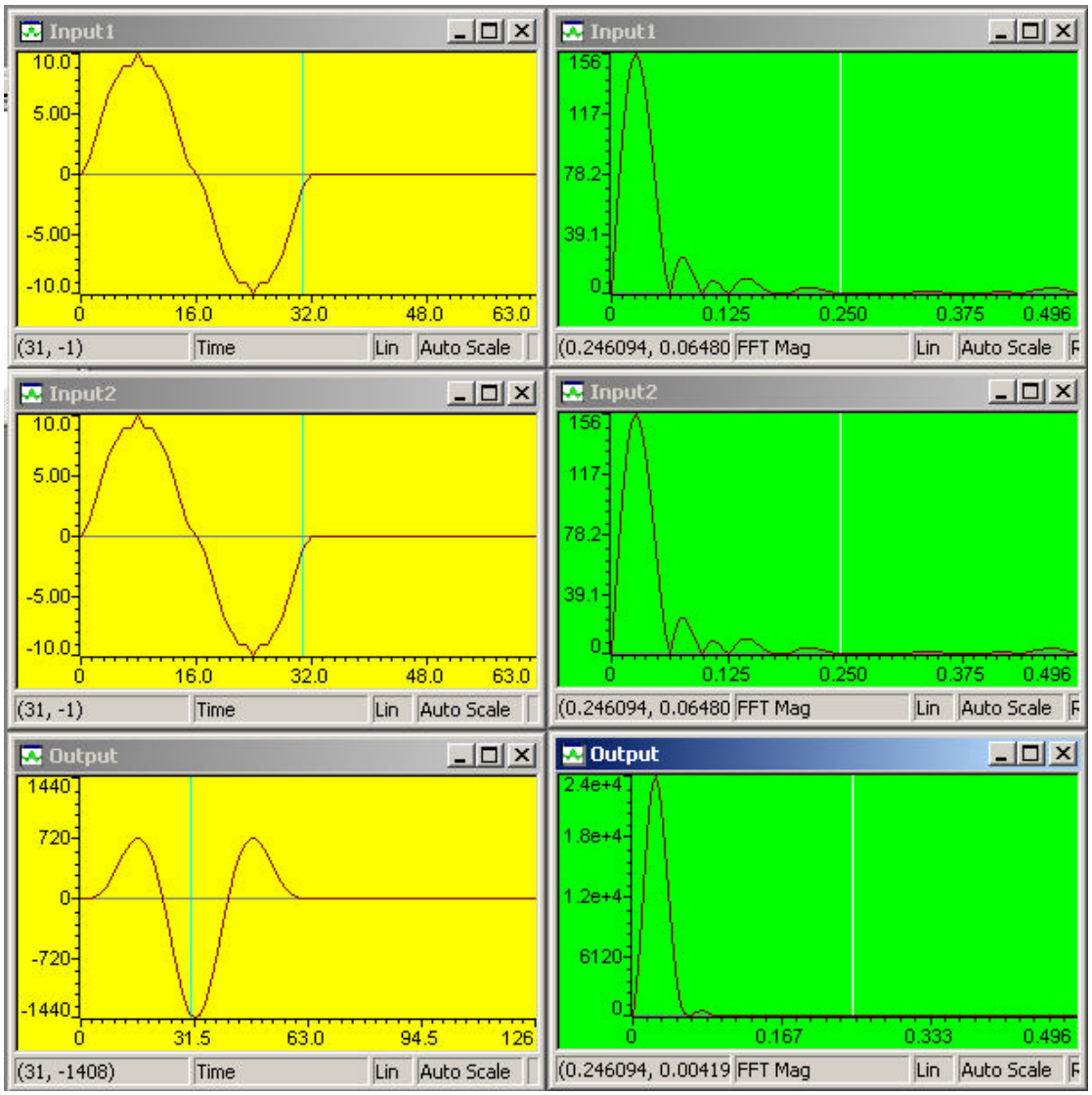
-观察频域波形：在各图形窗口中单击鼠标右键，选择“Properties…”，在“Graph Property Dialog”窗口中的第 1 项“Display Type”项中选择“FFT Magnitude”，单击“OK”完成；这时图形窗口中显示波形的频域图。（也可再打开显示频域图的窗口）

-验算结果：在各频域窗口中的波形上单击鼠标左键，将光标停到统一的位置（通过观察窗口状态栏中的第 1 个浮点数表示其坐标值），读取状态栏中的第 2 个浮点数，为卷积计算的输入和输出结果，请验算：Output≈Input1*Input2。

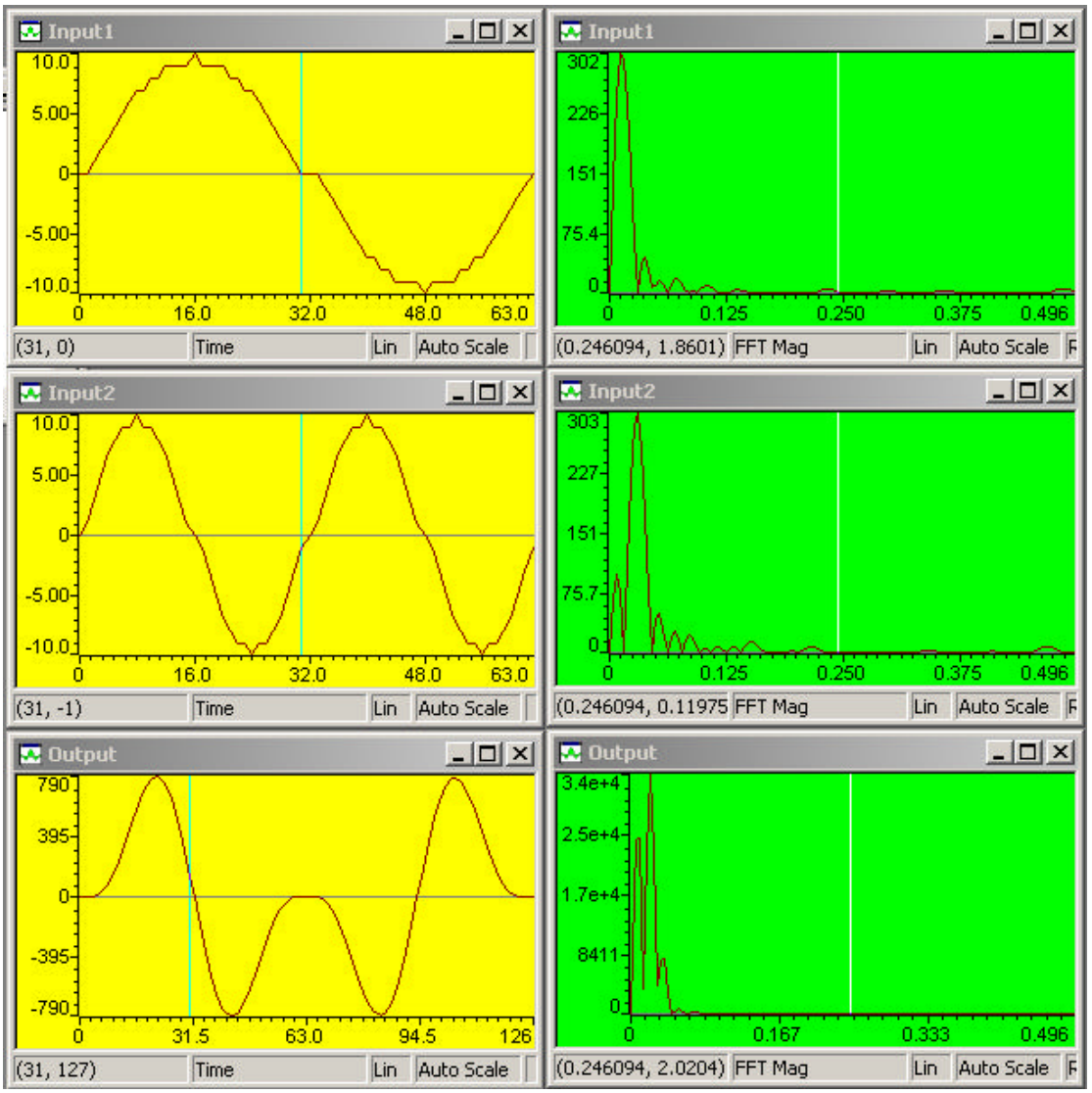
8. 将输入波形文件改成其他波形：选择“File”、“File I/O…”，将 2 个文件删除，将第 1 个文件换成 SIN11.DAT，讲第 2 个文件换成 SIN22.DAT；输入“Length”改为 64，其他不变。再按 F5 运行，停止后观察波形。
9. 将第 2 个输入波形改成 SIN33.DAT，观察卷积运算后的波形。
10. 将第 2 个输入波形改成 SIN44.DAT，观察卷积运算后的波形。
11. 将第 2 个输入波形改成方波.DAT，观察卷积运算后的波形。

五、试验结果

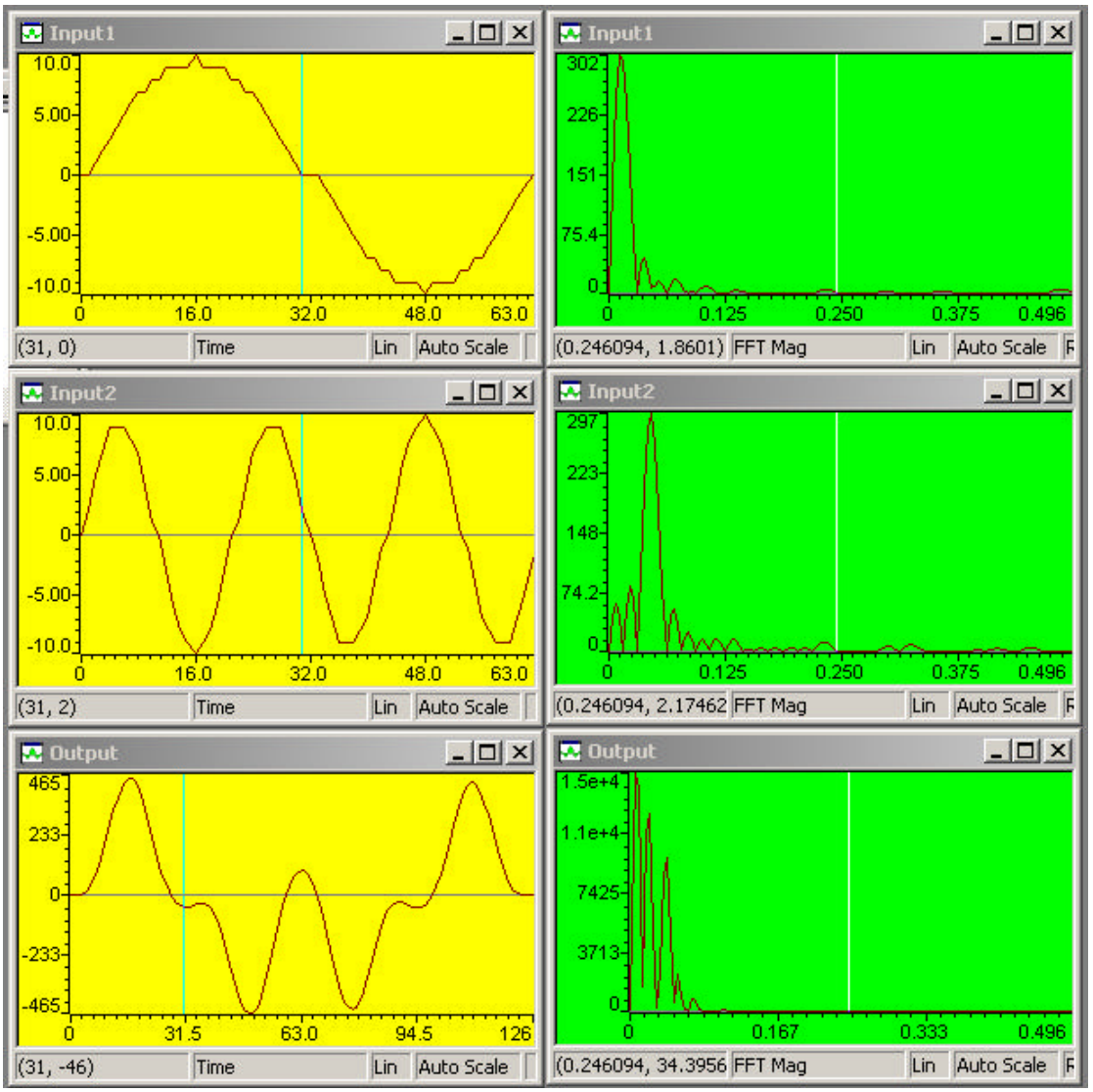
*实验第 7 步的结果图：图 1



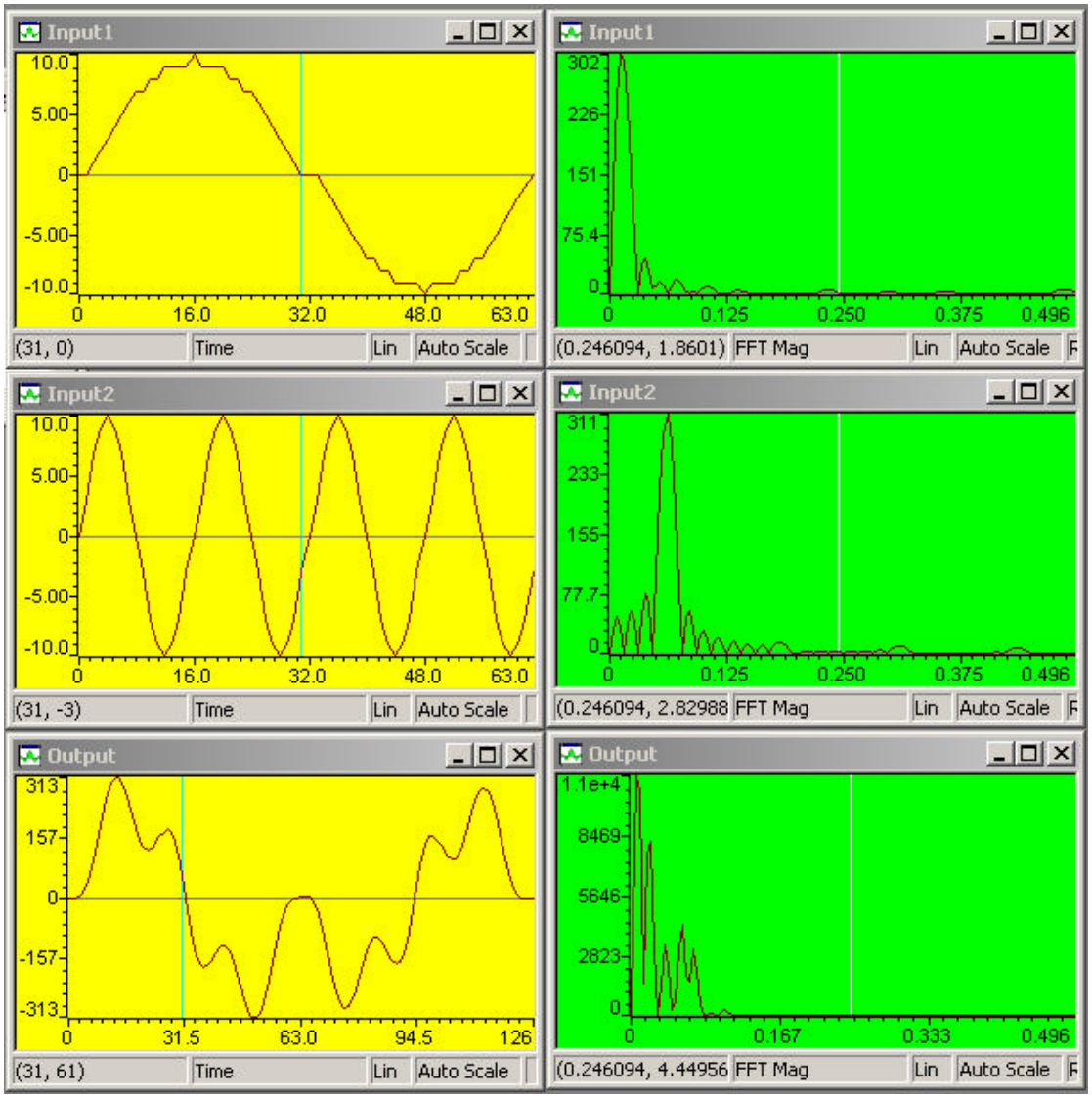
*实验第 8 步的结果图：图 2



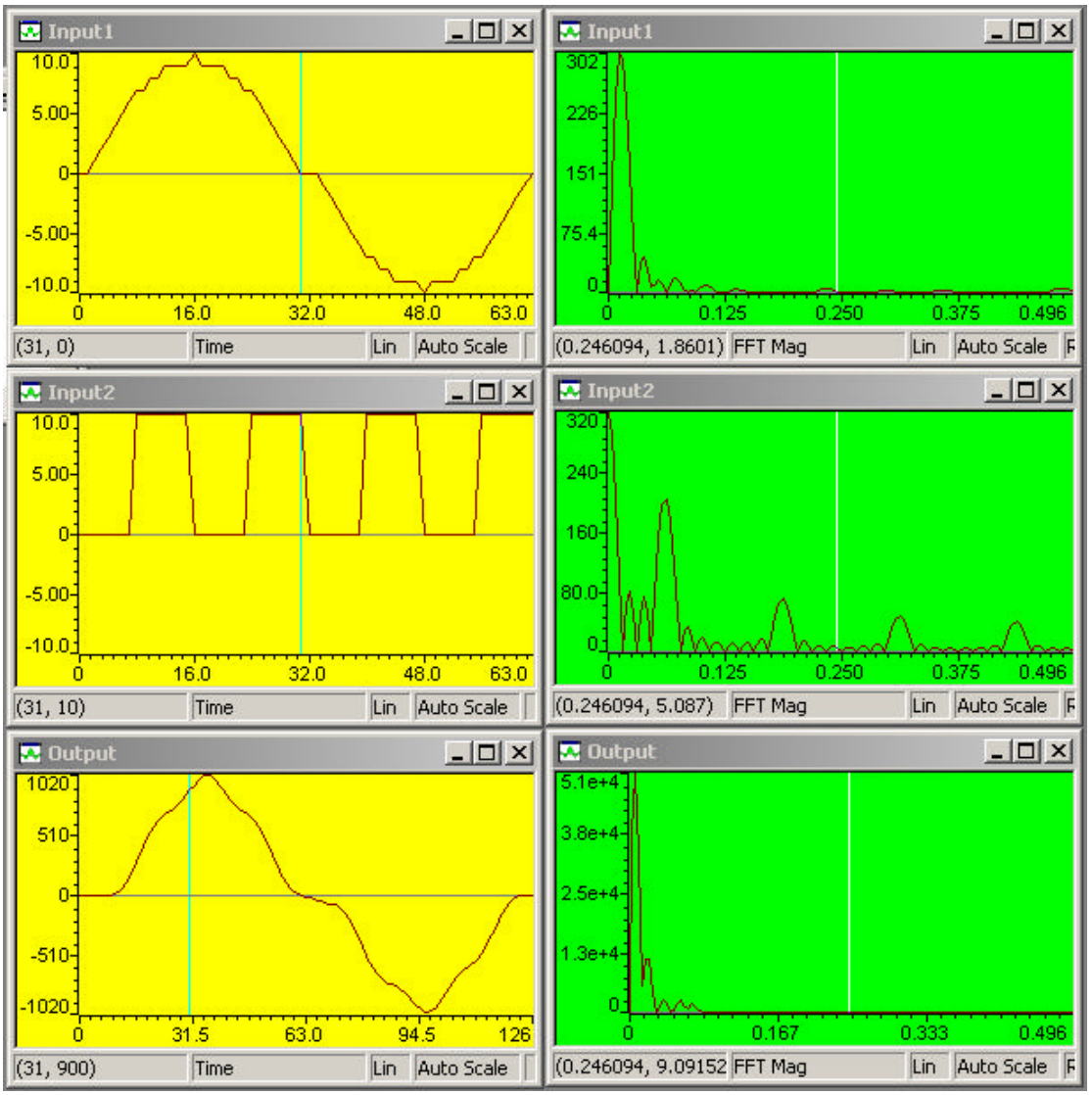
*实验第 9 步的结果图：图 3



*实验第 10 步的结果图：图 4



*实验第 11 步的结果图：图 5



名称	Inp1 上取样点值 (样点位置)	Inp2 上取样点值 (样点位置)	Out4 上取样点值(样点位置)	标准值
图 1	101.401(0.0117188)	101.401(0.0117188)	10279.34(0.0117188)	10282.16280
图 2	156.796(0.234375)	217.879(0.234375)	33642.7(0.234375)	34162.555684
图 3	156.796(0.234375)	83.1131(0.234375)	12414(0.234375)	13031.8026176
图 3	301.704(0.0117188)	1.90995(0.0117188)	14069.5(0.0117188)	13359.03
图 4	156.796(0.234375)	56.0769(0.234375)	8248.46(0.234375)	8792.634
图 4	301.704 (0.0117188)	35.2126(0.0117188)	10663(0.0117188))	10623.78227
图 5	296.69156.796(0.234375)	81.743 (0.234375)	12593.0(0.234375)	12817
图 5	301.704 (0.0117188)	100.378(0.0117188)	31667.2(0.0117188)	30284.4

-图表分析

输入图形频域图形采样通过频域采样取值比较，卷积后的结果与标准值只有很小的误差。所以说卷积实验结果正确，卷积程序正确无误。

六、问题与思考

实验十九：异步串口通信实验

一. 实验目的

1. 了解 TL16C550 异步串行通信收发器。
2. 学会设置异步串行通信接口进行通信。
3. 了解 ICETEK-VC5416-A 板上 DSP 对 TL16C550 的连接设计。
4. 学习设计异步通信程序。

二. 实验设备

计算机，ICETEK-VC5416-EDU 实验箱（或 ICETEK 仿真器+ICETEK-VC5416-A 系统板+相关连线及电源）。

三. 实验原理

1.TL16C550 异步串行通信收发器

TL16C550 有 11 个寄存器,通过 A2~A0 和线路控制寄存器中的 DLAB 位对它们进行寻址.

TL16C550 的寄存器如下表所示:

寄存器	DLAB	A2	A1	A0	地址	操作
接收缓冲寄存器 RBR	0	0	0	0	00H	只读
发送缓冲寄存器 THR	0	0	0	0	00H	只写
中断始能寄存器 IER	0	0	0	1	01H	读/写
中断标志寄存器 IIR	X	0	1	0	01H	只读
FIFO 控制寄存器 FCR	X	0	1	0	01H	只读
线路控制寄存器 LCR	X	0	1	1	03H	读/写
MODEM 控制寄存器 MCR	X	1	0	0	04H	读/写
线路状态寄存器 LSR	X	1	0	1	05H	读/写
MODEM 状态寄存器 MSR	X	1	1	0	06H	读/写
暂存寄存器 SCR	X	1	1	1	07H	读/写
低位除数寄存器 DLL	1	0	0	0	00H	读/写
低位除数寄存器 DLM	1	0	0	1	01H	读/写

1）线路控制寄存器:

D7	D6	D5	D4	D3	D2	D1	D0
DLAB	BREAK	SPB	EPS	PEN	STB	WLS1	WLS0

WLS1 WLS0:设置数据长度:

0 0 :5 位
0 1 :6 位
1 0 :7 位
1 1 :8 位

STB :设置停止位个数

0 ： 一个停止位

1 ： 1.5 个停止位(5 位数据长度时),2 个停止位(6,7,8 位数据长度时)

PEN:奇偶校验使能

0 ： 奇偶校验无效

1 ： 奇偶校验有效

EPS: 奇偶校验选择

0： 奇校验

1： 偶校验

DLAB:寄存器访问选择

0 :访问其余寄存器

1 :访问除数和功能切换寄存器

2）线路状态寄存器

D7	D6	D5	D4	D3	D2	D1	D0
FERR	TEMT	THRE	BI	FE	PE	OE	DR

DR:接收数据准备好标志

0: 接收数据缓冲器空

1: 接收数据缓冲器中有数据

OE: 溢出错误标志(上一个接收数据被当前接收数据覆盖)

0: 无溢出

1: 有溢出

PE: 奇偶校验错误标志

0: 无奇偶校验错误

1: 有奇偶校验错误

THRE:发送保持寄存器空标志

0: 非空

1: 空

TEMT:发送器空标志

0: 发送保持寄存器和发送移位寄存器非空

1: 发送保持寄存器和发送移位寄存器都空

3）中断使能寄存器

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	EMSI	ELSI	ETHREI	ERDAI

ERDAI: 接收中断使能

0 ： 接收中断禁止

1 ： 接收中断使能

ETHREI: 发送中断使能

0 ： 接收中断禁止

1 ： 接收中断使能

ELSI ： 接收错误中断使能

0 ： 接收错误中断禁止

1 ： 接收错误中断使能

EMSI ： MODEM 中断使能

0 ： MODEM 中断禁止

1 ： MODEM 中断使能

4）中断标志寄存器

中断标志寄存器				中断设置与清除			
D3	D2	D1	D0	优先级	中断类型	中断源	中断清除
0	0	0	1	---	无中断	无中断	
0	1	1	0	最高	接收错误	溢出,奇偶,帧错误	读线路状态寄存器
0	1	0	0	第二	接收	接收缓冲器中数据	读接受缓冲器
1	1	0	0	第二	FIFO 超时	FIFO 超时	读接收缓冲器
0	0	1	0	第三	发送	发送保持寄存器空	写发送保持寄存器
0	0	0	0	第四	MODEM	MODEM 状态	读状态寄存器

5）设置波特率

TL16C550 的波特率可通过除数寄存器 DLM,DLL 来设置,除数寄存器值和波特率之间的换算公式如下:除数值=输入频率÷(波特率×16),TL16C550 的输入频率为:3.6864MHz,波特率和除数之间的关系如表所示:

波特率	高位除数寄存器 DLM	低位除数寄存器 DLL
1200	00H	C0H
2400	00H	60H
4800	00H	30H
9600	00H	18H
19200	00H	0CH
38400	00H	06H

6）串口标准

RS232 标准

2.ICETEK-VC54167-A 板异步串口设计

16C550 芯片的控制寄存器映射到 5416 的 I/O 空间，通过 CPLD 进行译码选通。另外需要加上驱动电路。驱动电路主要完成将 DSP 输出的 0-3.3V 电平转换成异步串口电平的工作。转换电平的工作由 MAX232 芯片完成，但由于它是 5V 器件所以它同 DSP 间的信号线必须有电平转换，此板采用的是 74LS245。

3.串行通信接口设置

*CPU 进行串行通信时可以采用两种方式，一种是轮询方式，即 CPU 不断查询串口状态进行接收和发送，缺点是占用 CPU 时间太多；另一种是中断方式，接收和发送都可以产生中断信号，这样 CPU 可以在完成其他一些工作的同时进行串行通信。

*串行通信接口波特率计算

内部生成的串行时钟由系统时钟 SYSCLK 频率和波特率选择寄存器决定。串行通信接口使用 16 位波特率选择寄存器，数据传输的速度可以被编程为 65000 多种不同的方式。

不同通信模式下的串行通信接口异步波特率由下列方法决定：

-BRR=1—65535 时的串行通信接口异步波特率：

SCI 异步波特率=SYSCLK/ [(BRR+1)*8]

其中，BRR=SYSCLK/(SCI 异步波特率*8)-1；

-BRR=0 时的串行通信接口异步波特率：

SCI 异步波特率=SYSCLK/16

这里 BRR 等于波特率选择寄存器的 16 位值。

4.TL16C550 编程步骤如下:

1）初始化(包括:设置数据长度,有无奇偶校验,奇/偶校验选择,停止位个数及波特率),

2）开相应的中断,等中断

3）在中断服务程序中完成数据的发送和接收.

四. 实验步骤

1.实验准备

⑴. 连接设备

① 关闭计算机和实验箱电源。

② 如使用 PP 型仿真器则用附带的并口连线连接计算机并口和仿真器相应接口。

③ 检查 ICETEK-VC5416-A 板上 DIP 开关 MP/MC 的位置，应设置在“ OFF ”位置（靠近复位按钮），即设置 DSP 工作在 MP 方式。

④ 关闭实验箱上三个开关。

⑤ 用附带的串行通信电缆连接计算机 COM 端口和 ICETEK-VC5416-A 板上九针接头 DB1。

⑵. 开启设备

① 打开计算机电源。

② 打开实验箱电源开关，注意 ICETEK-VC5416-A 板上指示灯 D1 和 DS2 亮。

③ 如使用 USB 型仿真器用附带的 USB 电缆连接计算机和仿真器相应接口，注意仿真器上两个指示灯均亮。

⑶. 设置 Code Composer Studio 为 Emulator 方式：

参见“ Code Composer Studio 入门实验 ”之四.3。

⑷. 启动 Code Composer Studio 2.0

2.打开工程，浏览程序，工程目录为 C:\5416EDULab\Lab19-Serial

3.编译并下载程序

4.运行“ 串口调试助手 ”

利用桌面上“ 我的电脑 ”，找到 C:\5416EDULab\Lab19-Serial 目录中的程序“ 串口调试助手 V2.0B.exe ”，双击它启动；设置“ 串口调试助手 ”的串行端口为实际连接的计算机 COM 端口，设置波特率为 9600，设置传输方式为 8 位、无校验、1 个停止位。

5.运行程序观察结果

运行程序后，切换窗口到“ 串口调试助手 ”；在“ 串口调试助手 ”的接收窗口中可看到 DSP 通过 SCI 发送来的“ Hello PC!,Over| ”字样；在“ 发送的字符/数据 ”栏中输入一些要发送到 DSP 的字符串，以“ . ”字符结尾；然后单击“ 手动发送 ”按钮；DSP 在接收到 PC 机的信息后会自动进行回答。

6.结束程序运行退出。

五. 实验结果

通过 DSP 传送到 PC 机上的信息，可以看出：串行通信接口正确工作。

六. 问题与思考

请考虑用中断方式设计程序完成异步串行通信。