《DSP 原理及应用》

实 验 指 导 书

邢素霞 编

北京工商大学计算机与信息工程学院

2012 年 9 月

目　录

实验一　基础实验.....3
实验二　汇编寻址方式.....15
实验三　汇编的算术逻辑运算.....19
实验四　汇编的循环与分支.....23
实验五　快速傅立叶变换（FFT）.....28
实验六　数字滤波器的设计.....32
实验七　定时器及中断实验.....37
实验八　A/D-D/A 实验.....41

实验一　基础实验

一、实验目的

1. 熟悉 CCS 开发环境
2. 了解 DSP 54X EVM 板的硬件环境

二、实验内容

1. 新建/打开工程文件（.pjt）；
2. 新建/打开源文件（.c, .asm），添加到工程文件中；
3. 新建/打开命令文件（.cmd），添加到工程文件中；
4. 编译工程文件；（BUILD）,汇编联接后生成 DSP 的可执行文件（.out）；
5. 使用 FILE 菜单中的 LOAD PROGRAM 命令调入可执行文件（.out）
6. 程序运行（RUN）
7. 打开 view 菜单中的 CPU Registers 观察 DSP 54X EVM 板存储器值和 DSP 寄存器值；
8. 打开 view 菜单中的 memory，设定要观察的单元地址，观察 DSP 54X 的内存值；
9. 打开 view 菜单中的 watch windows 窗口，观察变量值的变化；
10. 打开 debuge 菜单中的 breakpoints 在程序中建立断点，观察程序运行到断点是变量值、存储器的变化。

三、实验要求

通过编写一个小程序，熟悉 CCS 各个功能的使用，并掌握 DSP54X 的工程建立、编译方法。

四、实验环境

计算机，DSP 仿真器，DSP 54X EVM 板， CCS 软件， EVM54X 库文件

五、实验步骤

1. 将 DSP 仿真器与计算机并口连接好；
2. 将 DSP 仿真器 JTAG 头插入 DSP 5409EVM 板 JTAG 上；
3. 打开计算机电源，当计算机启动完毕后，打开电源仿真器和 DSP 54X EVM 板电源，板上 3.3V、1.8V 电源指示灯均亮，若不亮，请立即关闭

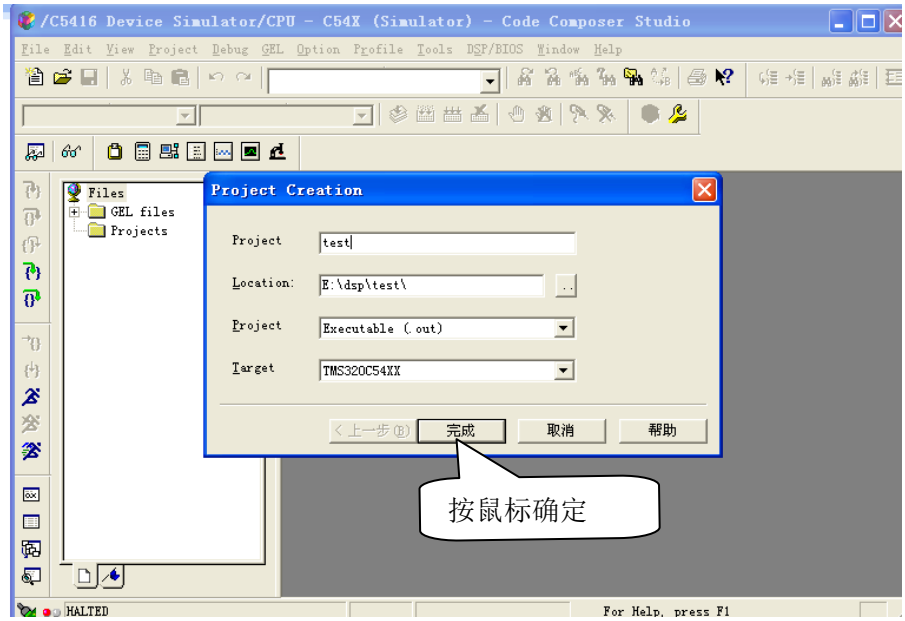
EVM 板电源，检查连接和电源电压；

4. 双击桌面上 CCS 图标，进入 CCS 操作环境；

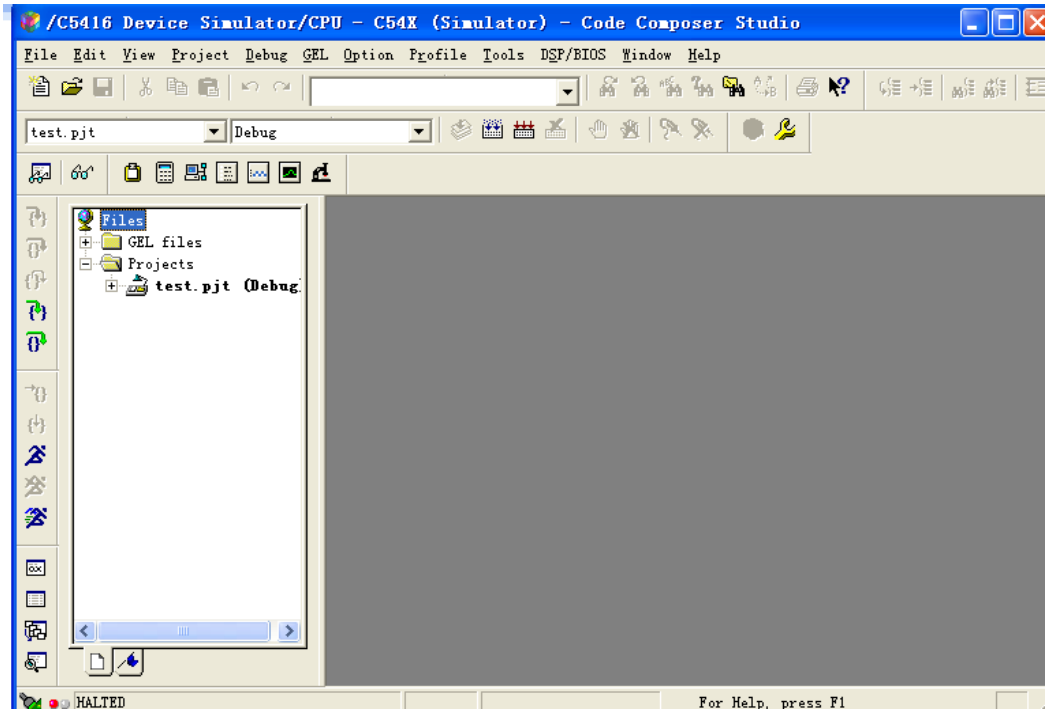
5. 建立工程文件（.pjt）



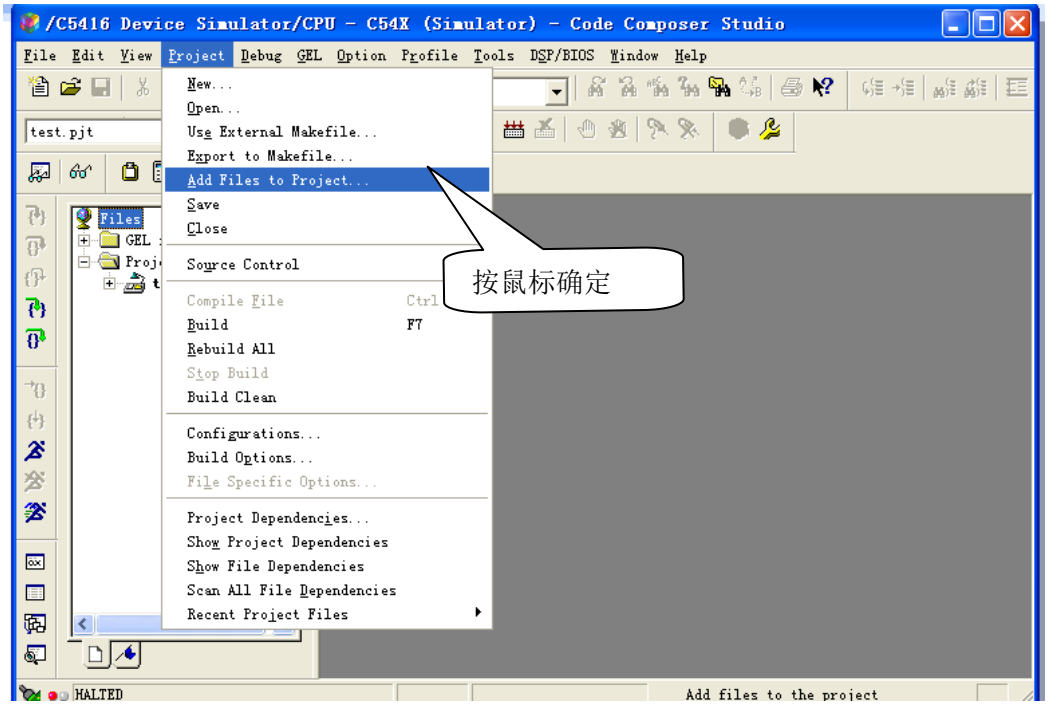
工程文件名及路径（文件名和路径可任意命名，但必须是唯一的）



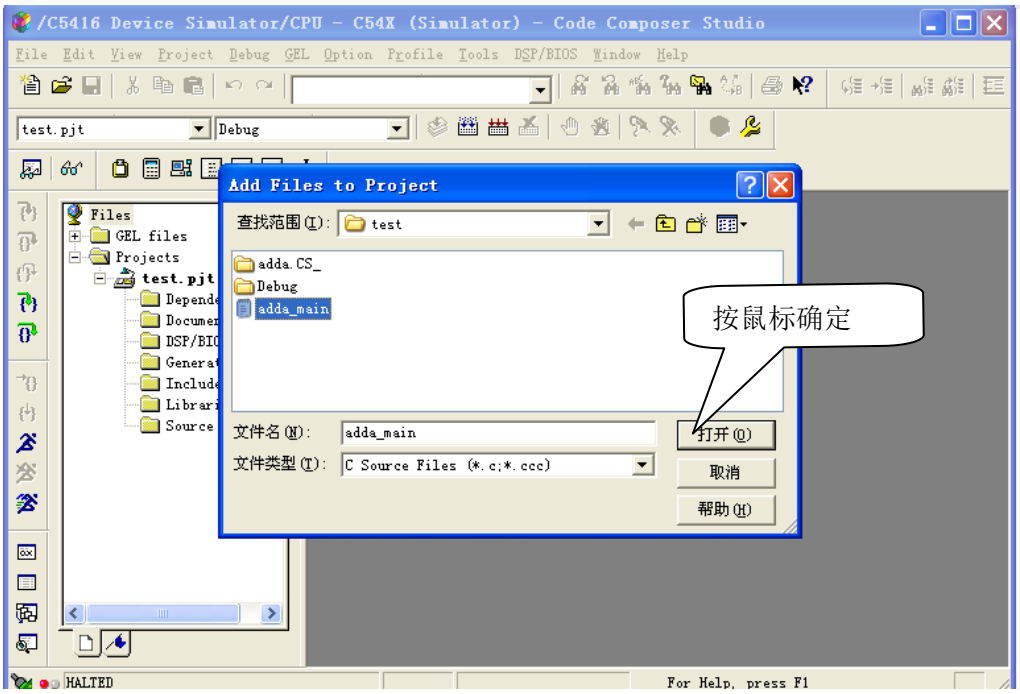
已新建一个名为 test.pjt 的工程文件，路径为 E:\dsp\test\



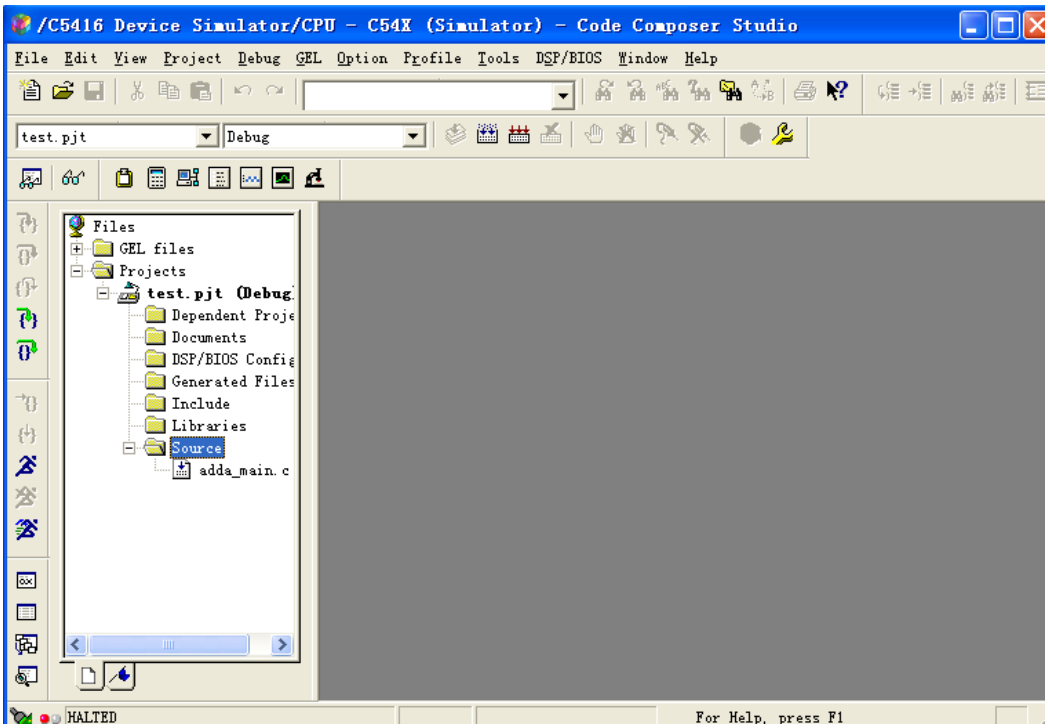
添加源文件到工程中（源文件可为*.c,*.cmd, *.s,……）

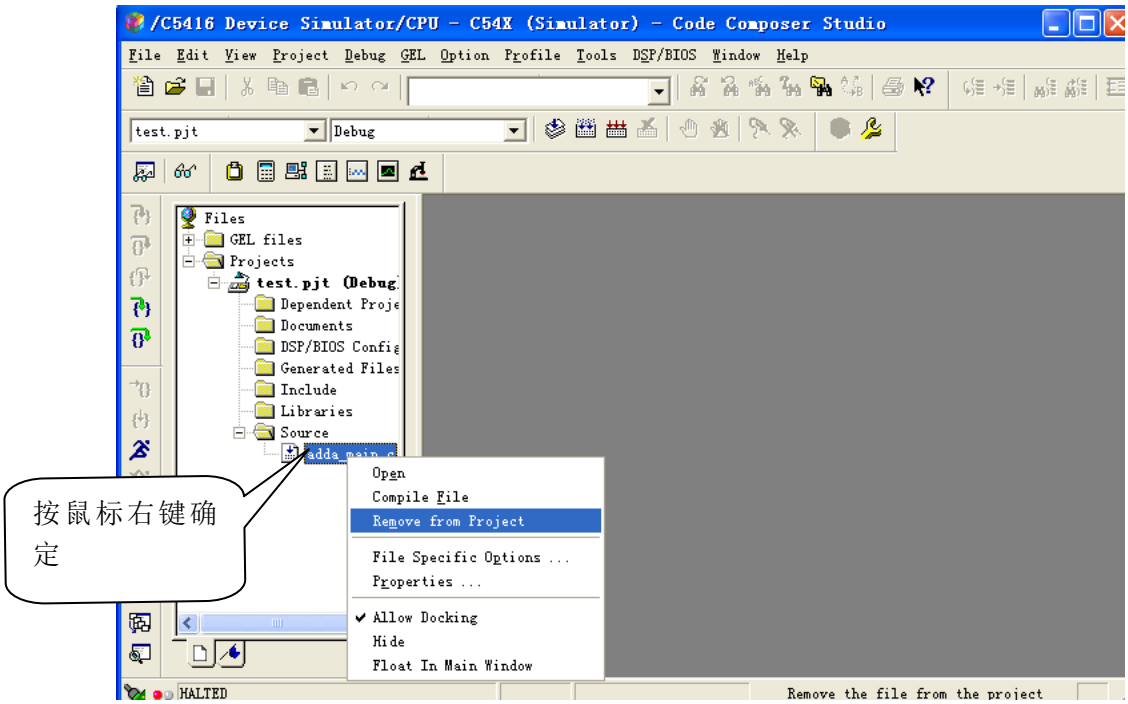


添加 c 文件到工程中

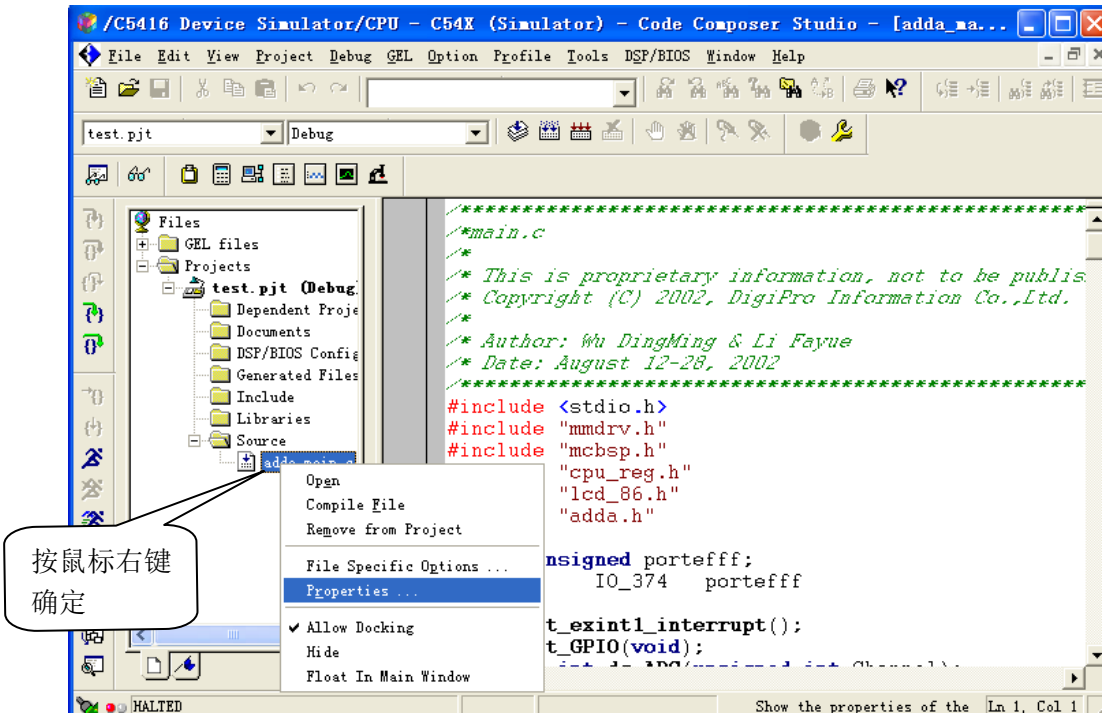


添加源文件完毕

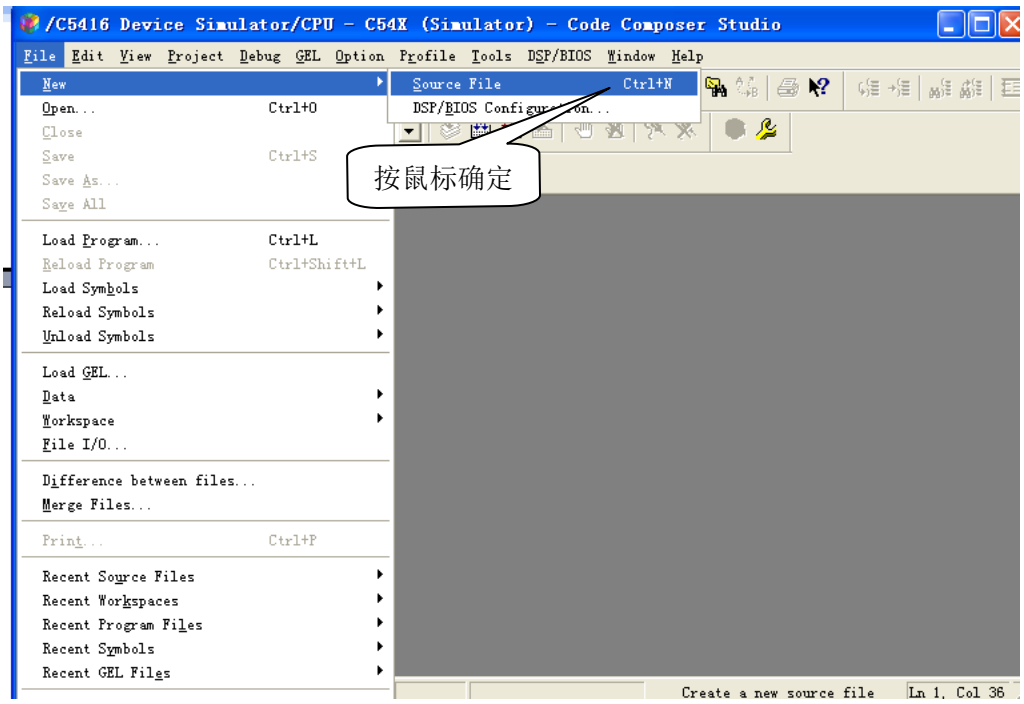




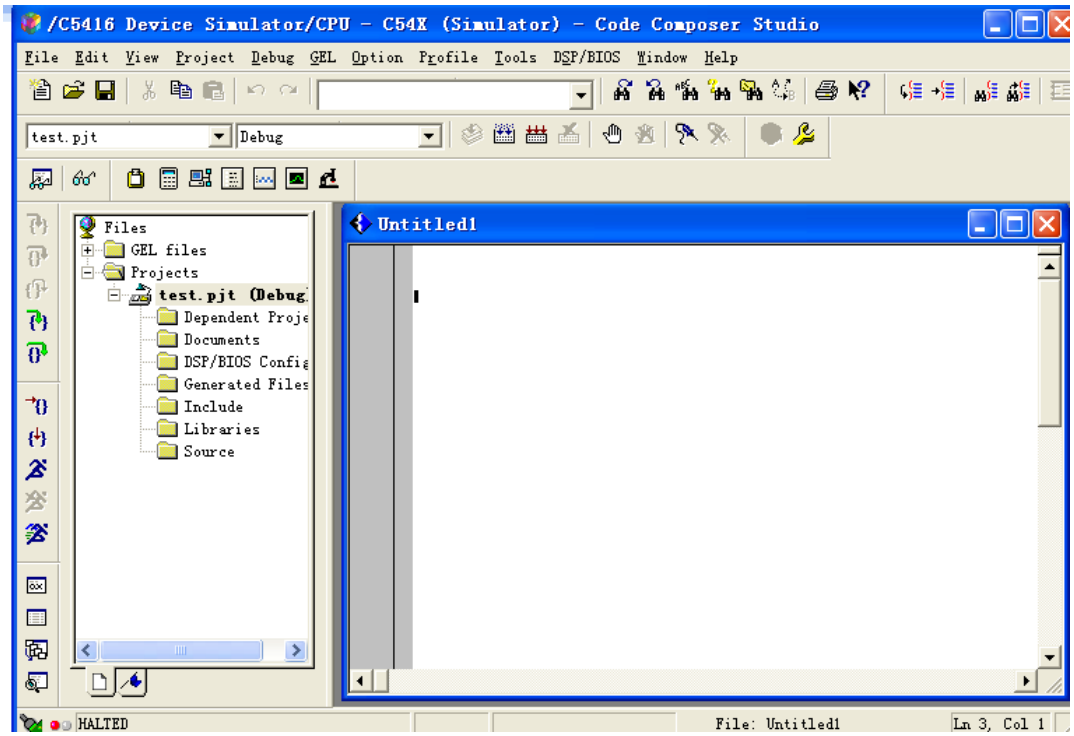
若想了解一个源文件的属性，



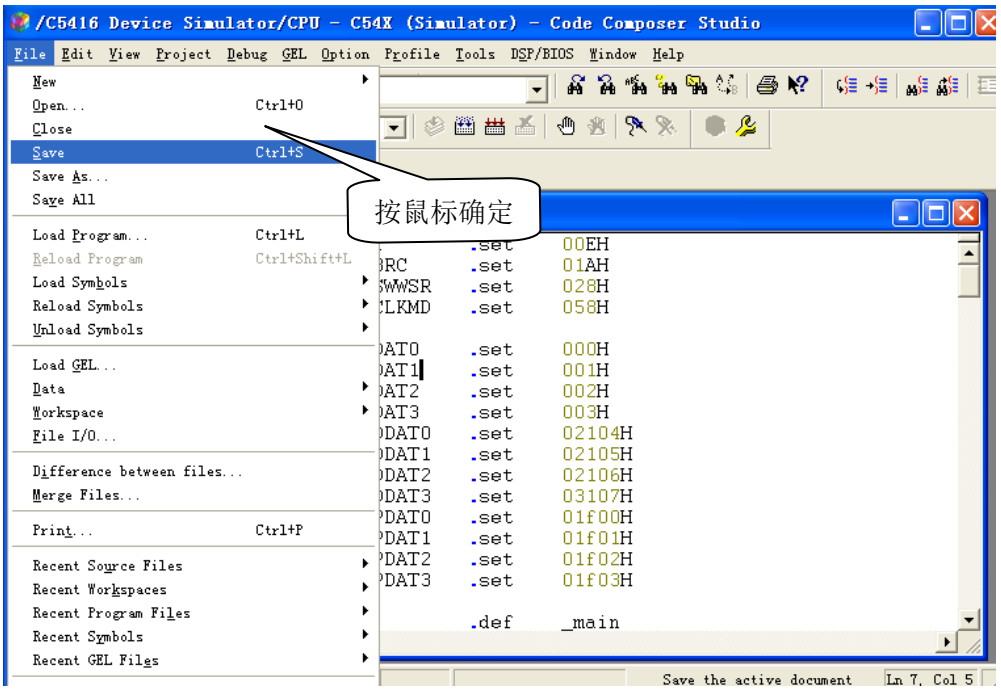
6. 新建源文件



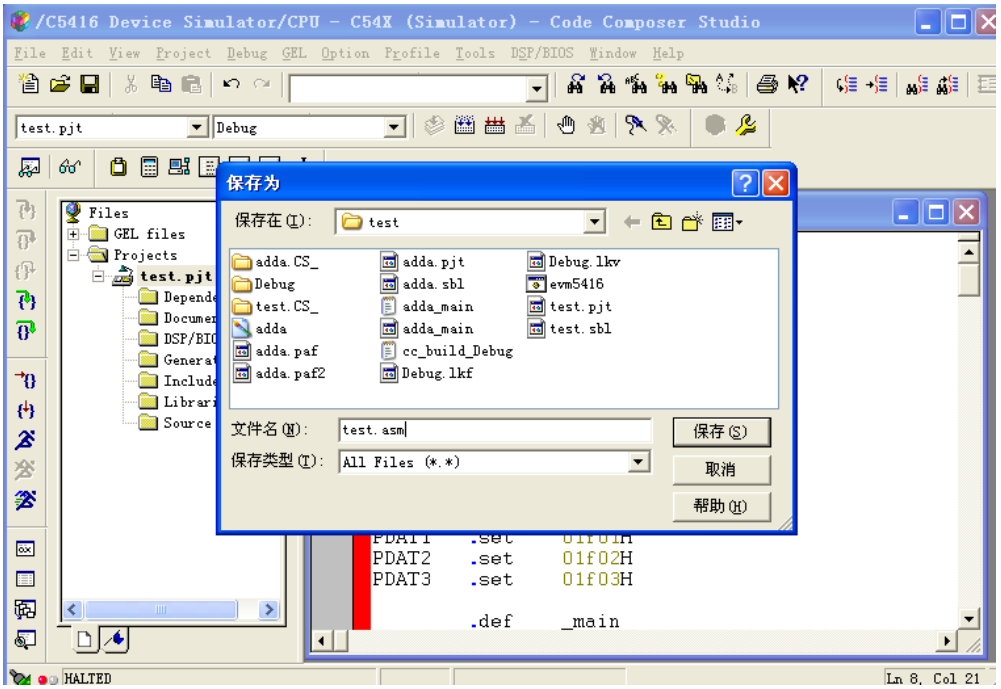
已建立一个未命名的源文件



编写源程序并保存



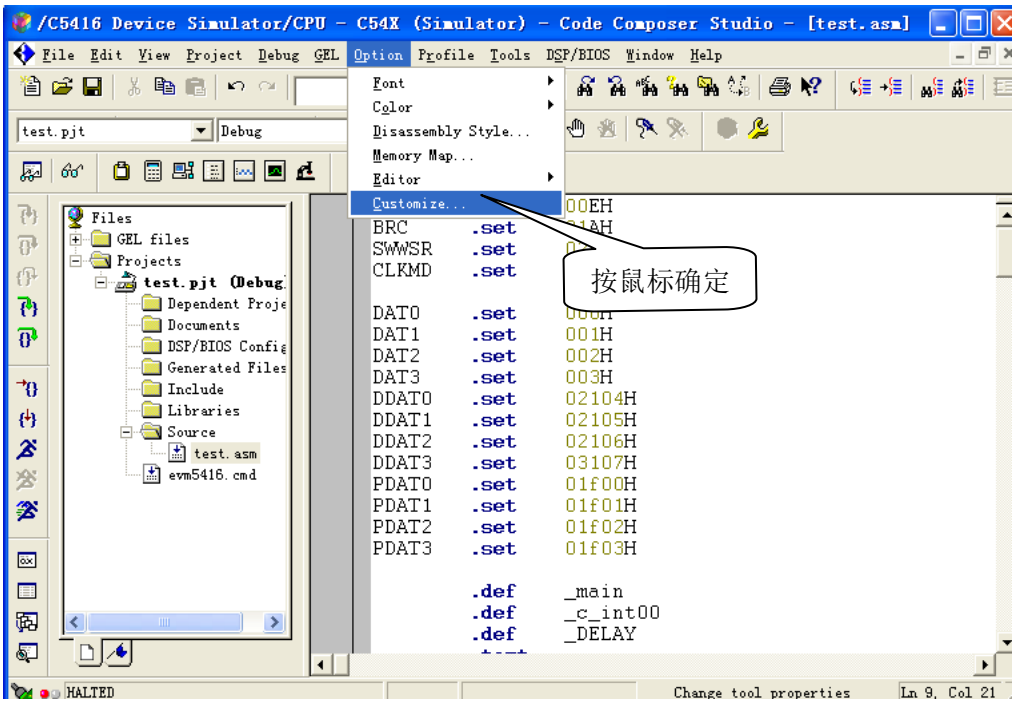
保存源程序名为 test.asm，路径为 E：\dsp\test



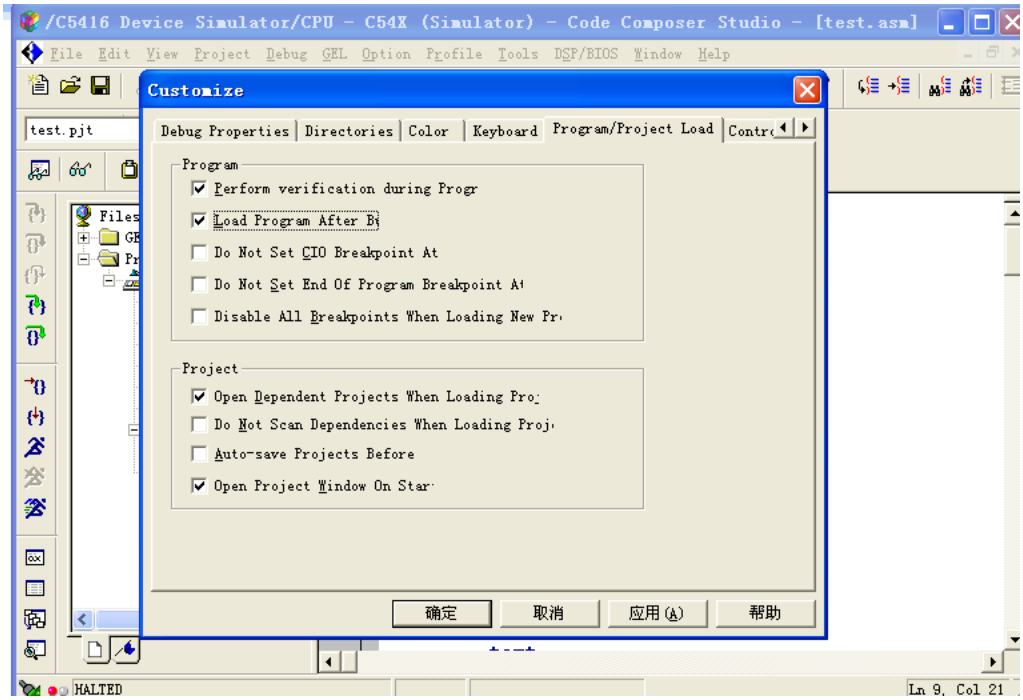
创建其他源程序（*.s, *.cmd……），可重复上述步骤，

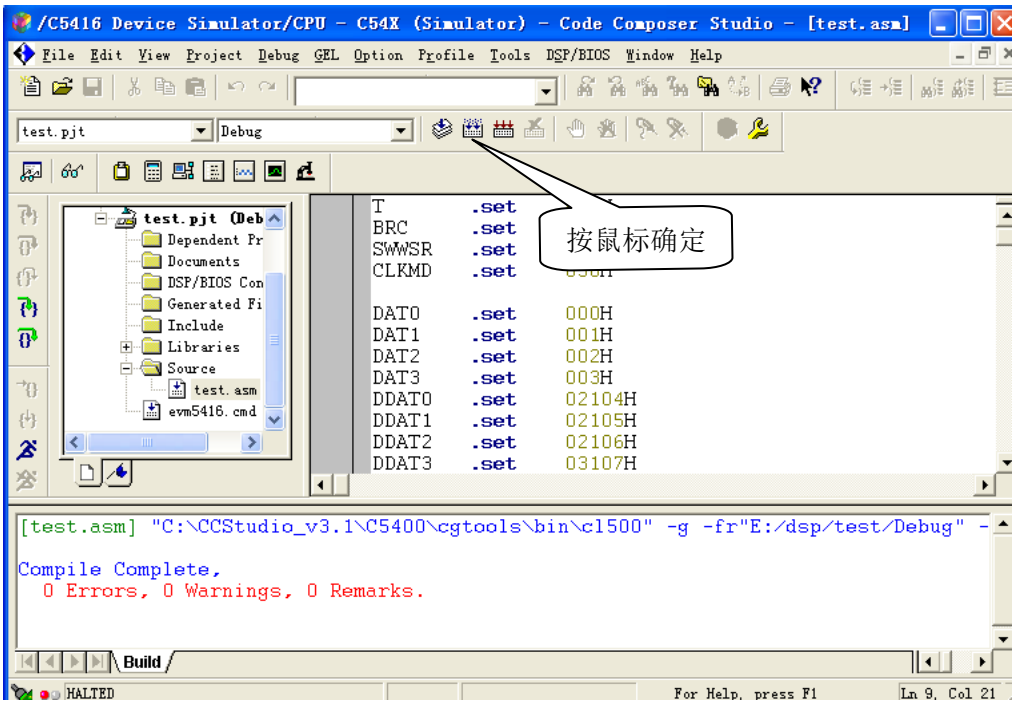
7. 编译文件（Build），产生一个名为 test.out 的可执行文件

在 Build 程序前的 option 设置

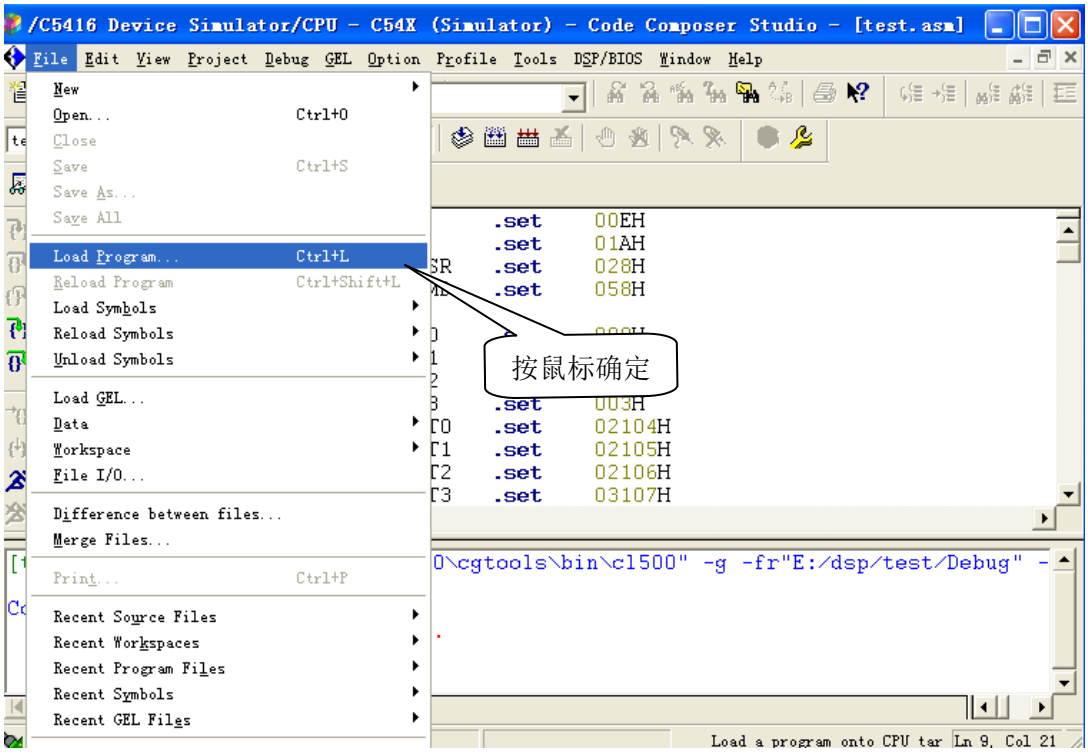


选中“Load Program After Bu”,编译后，执行文件自动装载到 EVM 板上的程序区

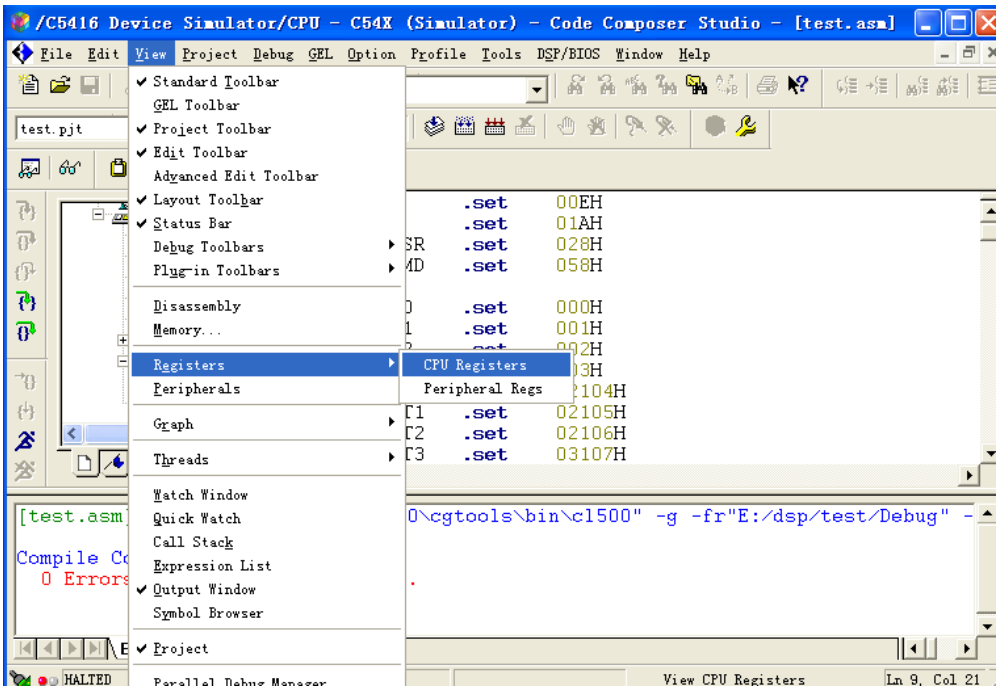




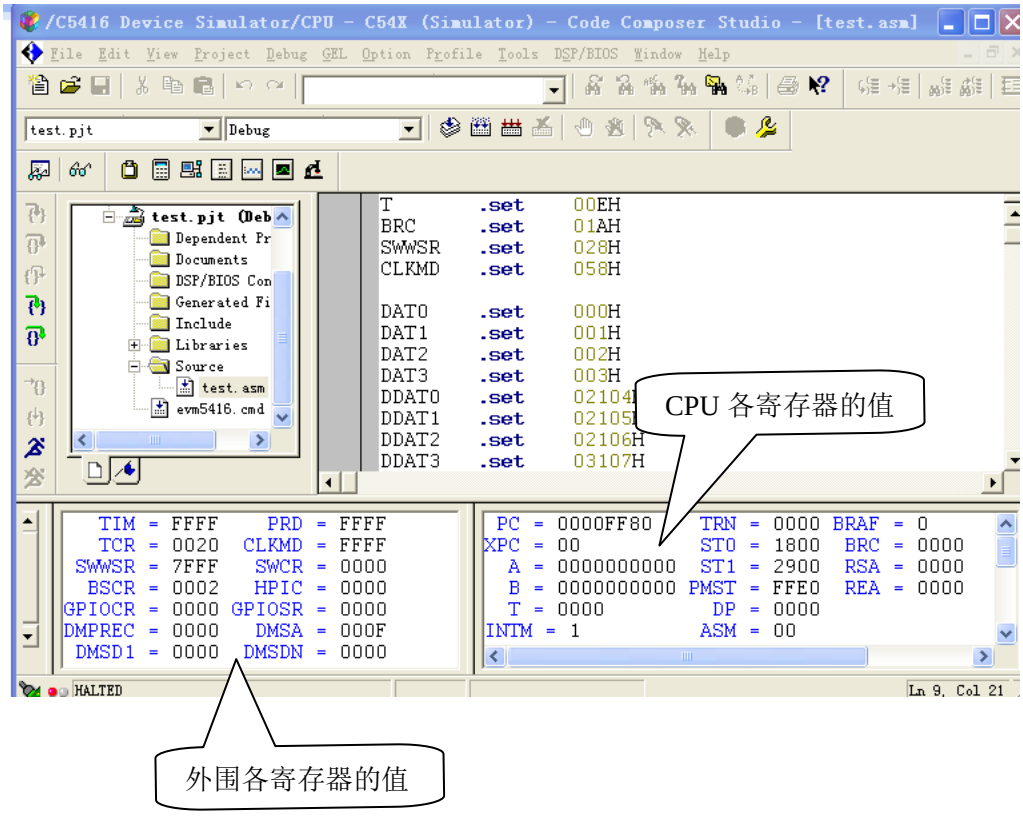
8. 装载（load）可执行文件（*.out）



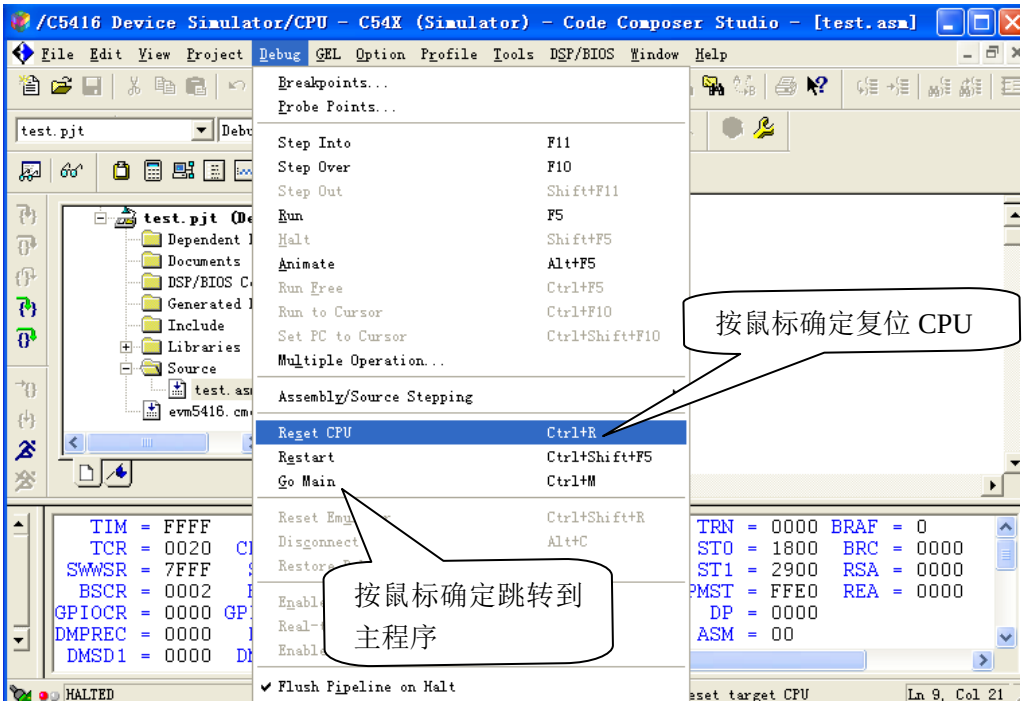
9. 观察 CPU 寄存器



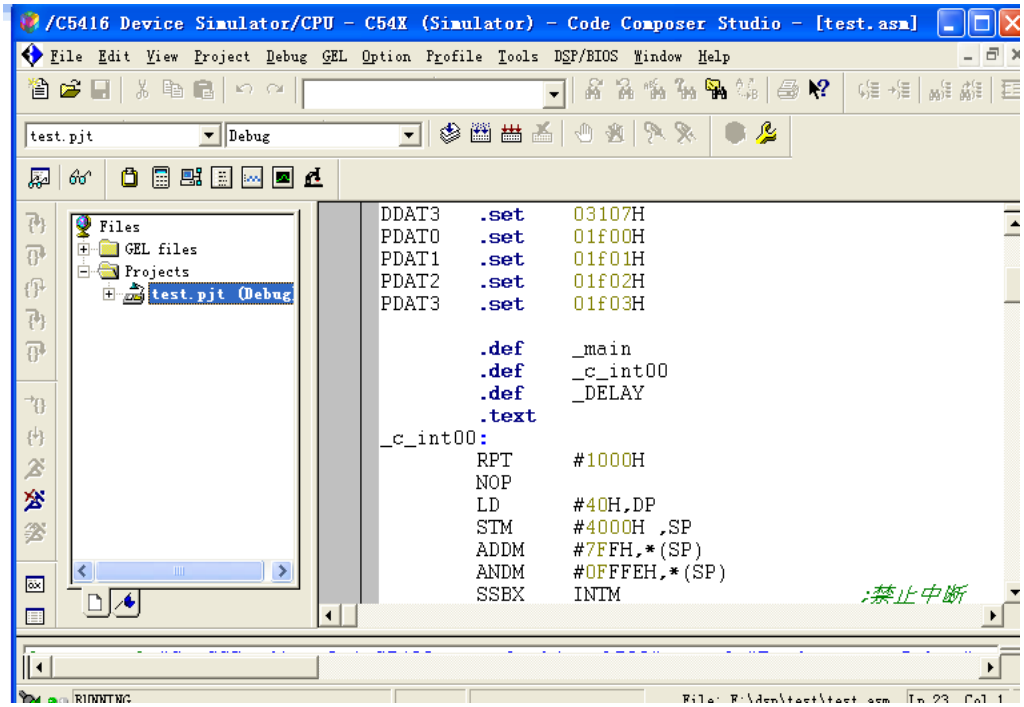
CPU 各寄存器的值



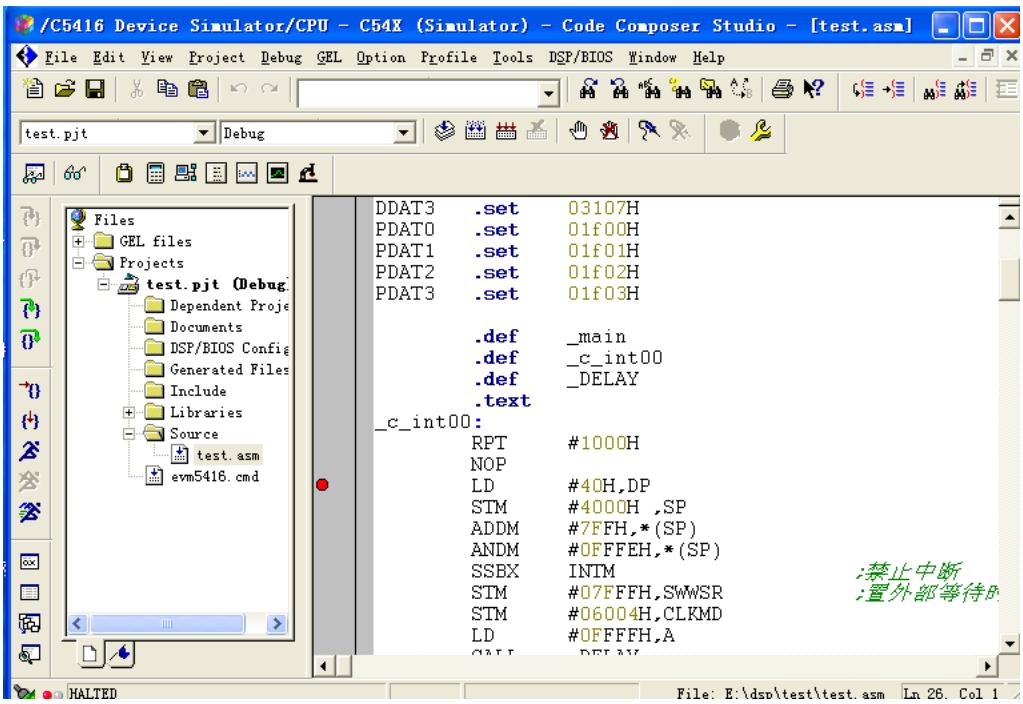
10.调试(debug)



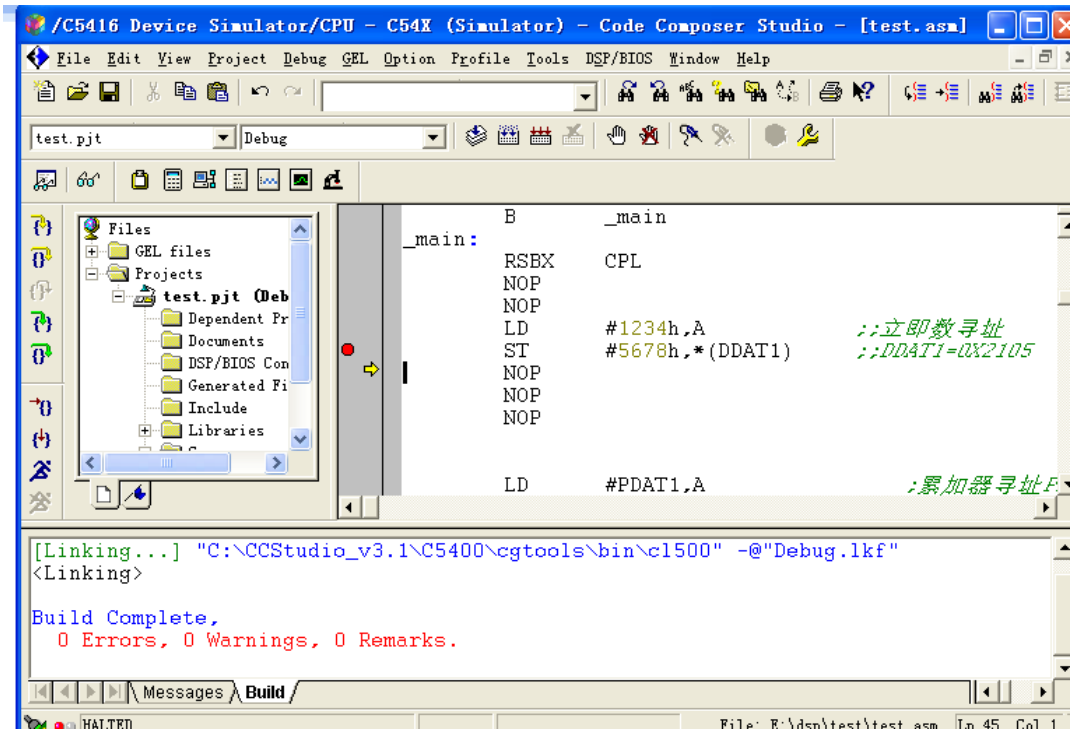
按 F5→运行，F8→单步运行且进入子程序，F9→设置断点，F10→单步运行，



F9 设置断点，按 F5 运行到该行



F10 单步运行到下一行



实验二　汇编寻址方式

一、　实验目的

1. 了解 DSP 结构
2. 熟悉 CCS 开发环境
3. 熟悉 DSP 54X 教学开发实验系统的硬件环境
4. 掌握汇编语言的寻址方法

二、　实验内容

1. DSP 54X 汇编初始化程序；
2. DSP 54X 各种汇编寻址方式的源程序。

三、　实验要求

运行程序，通过 CCS－> View -> CPU Registers, 观察各寄存器的值，通过 CCS－> View ->Memory， 观察片内各数据地址的值，上述值应与源程序的结果一致。

四、　实验环境

计算机、DSP 仿真器，DSP 54X 教学开发实验系统，CCS 软件，EVM54 库文件

五、　实验步骤

1.将 DSP 仿真器与计算机并口连接好；

2.将 DSP 仿真器 JTAG 头插入 DSP 5409EVM 板 JTAG 上；

3.打开计算机电源，当计算机启动完毕后，打开电源仿真器和 DSP 54X EVM 板电源，板上 3.3V、1.8V 电源指示灯均亮，若不亮，请立即关闭 EVM 板电源，检查连接和电源电压；

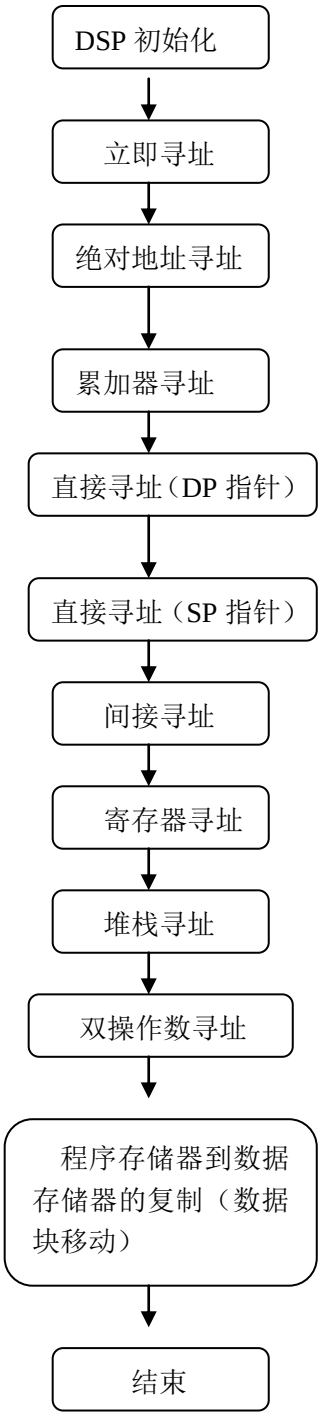
4.双击桌面上 CCS 图标，进入 CCS 操作环境；

5.按实验一的步骤创建一个工程和源文件，并调试。

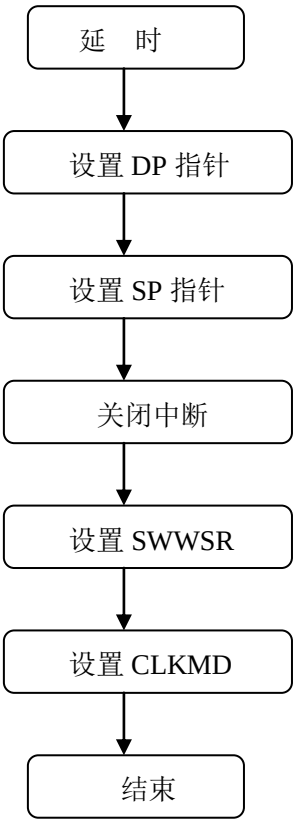
6.在 Project ->Build Options 中设置添加文件的路径。

六、实验框图

主程序



初始化程序



七、实验结果

1.立即数寻址

执行前：　　执行后

A=XXXX　　A=0x1234

(0X2105)=XXXX　　(0X2105)=0x5678

2.累加器寻址

执行前：　　执行后

A=XXXX　　A=0x1234

DP=0x40　　DP=Ox40

(0X1F01)=5679　　(0X1F01)=0x5678

(0X2002)=XXXX　　(0X2002)=0x5678

3.直接寻址（DP 指针）

执行前：　　执行后

CPL=0　　CPL=0

DP=0x40　　DP=0x40

(0X2003)=XXXX　　(0X2003)=0x3210

4.直接寻址（SP 指针）

执行前：　　执行后

CPL=0　　CPL=1

SP=0x47FE　　SP=0x47FD

(0X4801)=XXXX　　(0X4801)=0x9876

5.间接寻址

执行前：　　执行后

AR0=XXXX　　AR0=0x3107

(0X3107)=XXXX　　(0X3107)=0x9876

6.绝对寻址（数据区）

执行前：　　执行后

DP=0x40　　DP=OX40

(0X2000)=0x8000　　(0X2000)=0x8000

(0X2104)=0x7711　　(0X2000)=0x8000

(0X2105)=0x5678　　(0X2000)=0x5678

(0X2001)=0x7603　　(0X2000)=0x5678

7.绝对寻址（程序区）

执行前：　　执行后

DP=0x40　　DP=0x40

(0X2000)=0x8000　　(0X2000)=0x8000

(0X1F00)=0x000E　　(0X2000)=0x8000

(0X2001)=0x5678　　(0X2000)=0x5678

(0X1F01)=0x7711　　(0X2000)=0x5678

8.堆栈寻址

执行前：　　执行后

SP=0x47FE　　SP=0x47FD

(0X47FD)=0x061B　　(0X47FD)=0x8888

9.绝对寻址（程序区）

执行前：　　执行后

AR0=0x2107　　AR0=0x4000

(0X1000)=XXXX　　(0X4000)=0x1000

(0X1001)=XXXX　　(0X4001)=0x1001

(0X10FE)=XXXX　　(0X40FE)=0x10FE

(0X10FF)=XXXX　　(0X40FF)=0x10FF

10.双操作数寻址

执行前：　　执行后

A=0x0001　　A=0x0C4C10C0

AR2=0x2106　　AR2=0x2107

AR3=0x3107　　AR3=0x3108

T=0x8888　　T=0x1234

(0X2106)=0x1234　　(0X2106)=(0x1234)

(0X3107)=0x5678　　(0X3107)=(0x5678)

实验三　汇编的算术逻辑运算

一、 实验目的

5. 了解 DSP 结构
6. 熟悉 CCS 开发环境
7. 熟悉 DSP 54X 教学开发实验系统的硬件环境
8. 掌握汇编语言的逻辑和算术运算

二、实验内容

3. DSP 54X 汇编初始化程序；
4. DSP 54X 汇编语言的逻辑和算术运算源程序。

三、 实验要求

运行程序，通过 CCS－> View -> CPU Registers, 观察各寄存器的值，通过 CCS－> View ->Memory， 观察片内各数据地址的值，上述值应与源程序的结果一致。

四、 实验环境

计算机、DSP 仿真器，DSP 54X 教学开发实验系统，CCS 软件，EVM54 库文件

五、 实验步骤

1.将 DSP 仿真器与计算机并口连接好；

2.将 DSP 仿真器 JTAG 头插入 DSP 5409EVM 板 JTAG 上；

3.打开计算机电源，当计算机启动完毕后，打开电源仿真器和 DSP 54X EVM 板电源，板上 3.3V、1.8V 电源指示灯均亮，若不亮，请立即关闭 EVM 板电源，检查连接和电源电压；

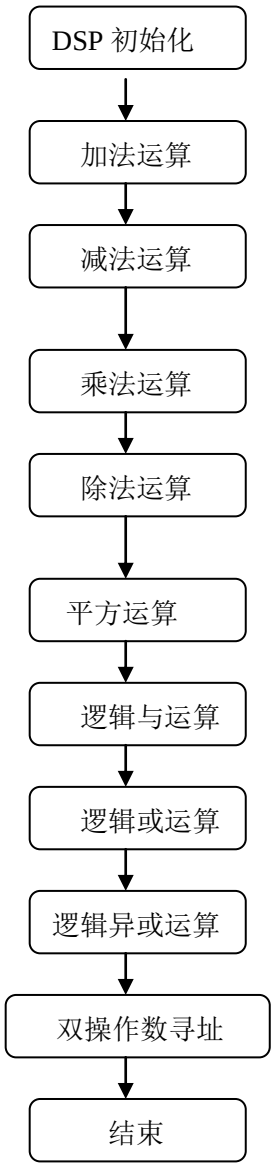
4.双击桌面上 CCS 图标，进入 CCS 操作环境；

5.按实验一的步骤创建一个工程和源文件，并调试。

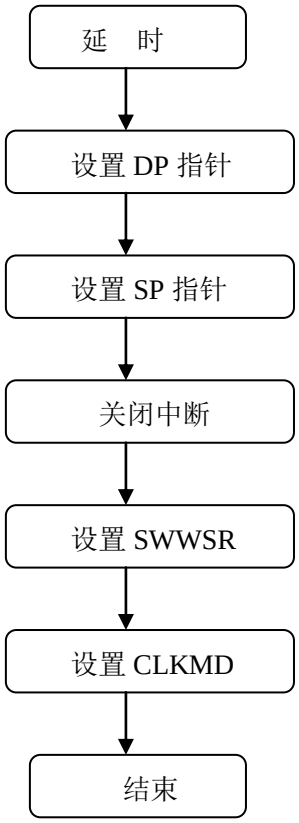
6.在 Project ->Build Options 中设置添加文件的路径。

六、实验框图

主程序



初始化程序



七 、实验结果

1.加法操作

执行前：	执行后
DP=0x40	**DP=0x40**
A=0x43	**A=0x98**
(0x2000)=0x55	**(0x2000)=0x55**
(0X2001)=0xxxx	**(0X2001)=0x98**

2.减法操作

执行前：	执行后
DP=0x40	**DP=0x40**
A=0xa2	**A=0x6a**
(0x2002)=0x38	**(0x2002)=0x38**
(0x2003)=0xxxx	**(0x2003)=0x6a**

3.乘法操作

执行前：	执行后
DP=0x40	**DP=0x40**
A=0x6a	**A=0x1b5aa**
T=0x421	**T=0x421**
(0x2003)=0x6a	**(0x2003)=0x6a**
(0x2004)=0xxxx	**(0x2004)=0xb5aa**
(0x2005)=0xxxx	**(0x2005)=0x1**

4.除法操作

执行前：	执行后
DP=0x40	**DP=0x40**
A=0x1000	**A=0x4007c**
(0x2006)=0x1000	**(0x2006)=0x1000**
(0x2007)=0x21	**(0x2007)=0x21**

(0x2008)=0xxxx **(0x2008)=0x007c(商)**

(0x2009)=0xxxx **(0x2009)=0x4（余数）**

5.平方操作

执行前： **执行后**

DP=0x40 **DP=0x40**

A=0xxxx **A=0x19a10**

(0x200a)=0x144 **(0x200a)=0x144**

(0x200b)=0xxxx **(0x200b)=0x9a10**

(0x200c)=0xxxx **(0x200c)=0x0001**

6.逻辑与操作

执行前： **执行后**

A=0xffff **A=0x00ff**

7.逻辑或操作

执行前： **执行后**

A=0xff00 **A=0xffff**

8.逻辑异或操作

执行前： **执行后**

A=0xff00 **A=0xffff**

实验四　汇编的循环与分支

一 、实验目的

1. 了解 DSP 结构
2. 熟悉 CCS 开发环境
3. 熟悉 DSP 54X 教学开发实验系统的硬件环境
4. 掌握汇编程序的循环和分支结构

二 、实验内容

1. DSP 54X 汇编初始化程序；
2. DSP 54X 数据区初始化程序；
3. DSP 54X 各种汇编循环源程序；
4. DSP 54X 各种汇编分支源程序。

三 、实验要求

在数据区预留 2 个数据块，每块大小为 1024 字，分别为源数据块和目标数据块，将源数据块初始化为表 1 所示，在初始化程序中需要用到循环和跳转指令，然后将源数据块复制到目标数据块，需用到跳转指令，运行程序，通过 CCS－> View -> CPU Registers, 观察各寄存器的值，通过 CCS－> View ->Memory， 观察片内各数据地址的值，上述值应与源程序的结果一致。

四 、实验环境

计算机、DSP 仿真器，DSP 54X 教学开发实验系统，CCS 软件，EVM54 库文件

五 、实验步骤

1.将 DSP 仿真器与计算机并口连接好；

2.将 DSP 仿真器 JTAG 头插入 DSP 5409EVM 板 JTAG 上；

3.打开计算机电源，当计算机启动完毕后，打开电源仿真器和 DSP 54X EVM 板电源，板上 3.3V、1.8V 电源指示灯均亮，若不亮，请立即关闭 EVM 板电源，检查连接和电源电压；

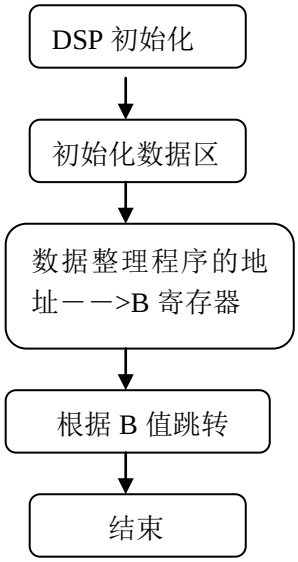
4.双击桌面上 CCS 图标，进入 CCS 操作环境；

5.按实验一的步骤创建一个工程和源文件，并调试。

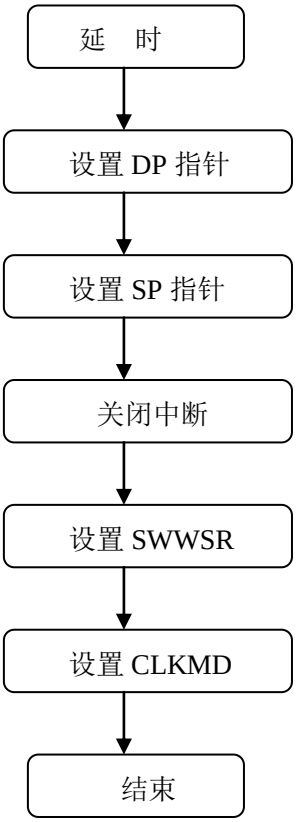
6.在 Project ->Build Options 中设置添加文件的路径。

六 、实验框图

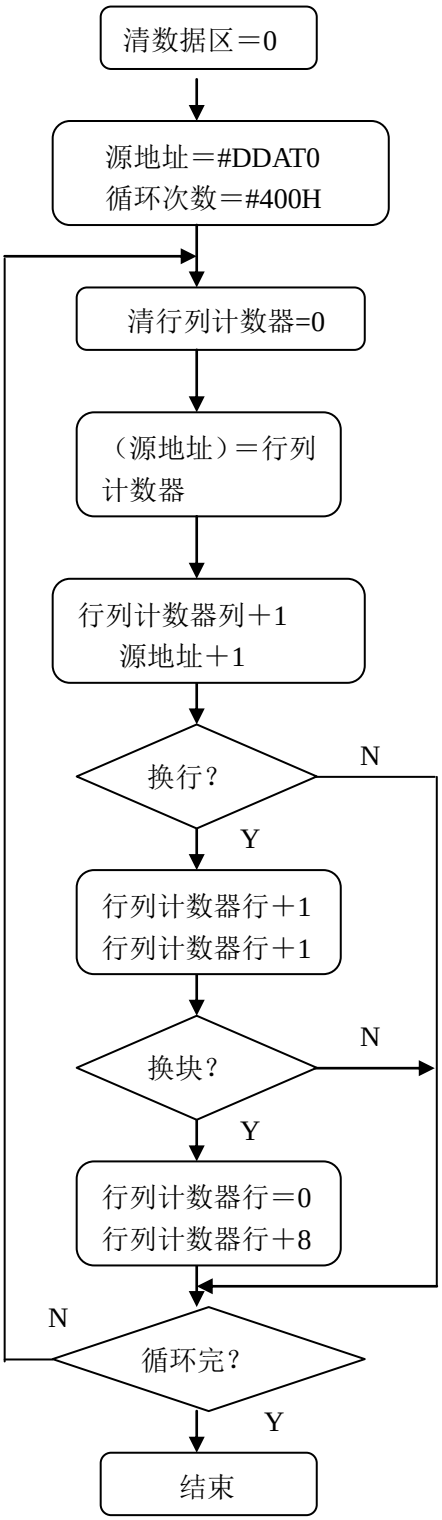
主程序



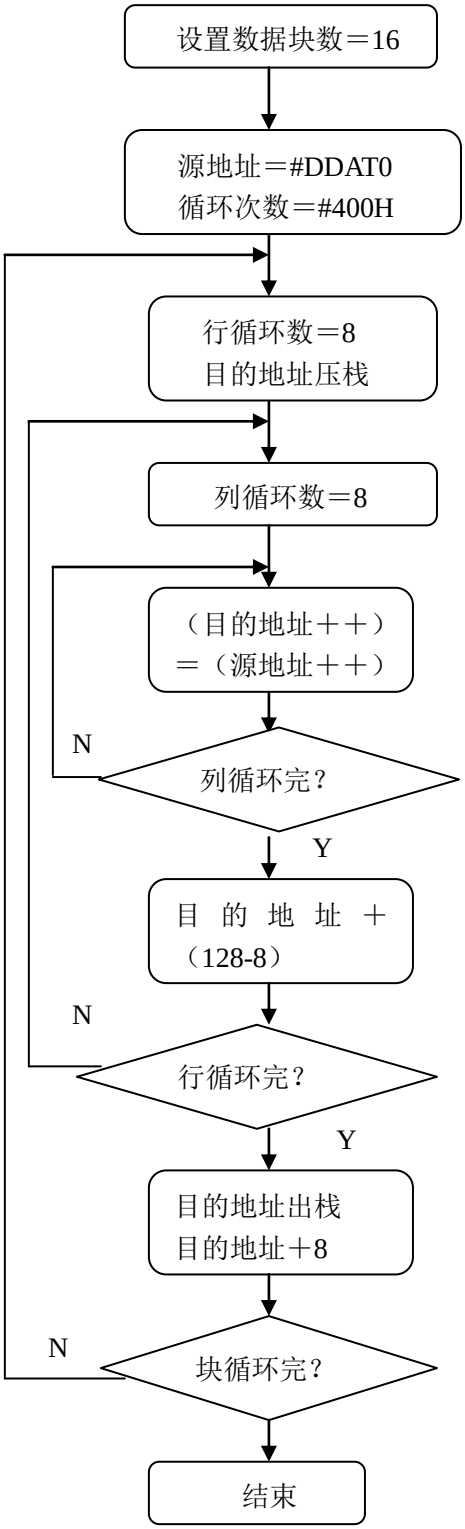
初始化程序



数据初始化



数据整理程序



七 、实验结果

1. 数据区初始化：

1#数据区地址＝0x2000，

2#数据区地址＝0x2400，

执行前：　　执行后

DP=0x40　　DP=0x40

(0x2000)=0xxxx　　(0x2000)=0x0

(0X2001)=0xxxx　　(0X2001)=0x1

……..

(0x2007)=0xxxx　　(0x2007)=0x7

(0x2008)=0xxxx　　(0x2008)=0x100

(0x2009)=0xxxx　　(0x2009)=0x101

……

(0x200f)=0xxxx　　(0x200f)=0x107

(0x2010)=0xxxx　　(0x2010)=0x200

(0x2011)=0xxxx　　(0x2011)=0x201

…….

(0x2017)=0xxxx　　(0x2017)=0x207

(0x2018)=0xxxx　　(0x2018)=0x300

(0x2019)=0xxxx　　(0x2019)=0x301

…….

(0x201f)=0xxxx　　(0x201f)=0x307

----

2. 数据区复制：

1#数据区地址＝0x2000，

2#数据区地址＝0x2400，

执行前：　　执行后

DP=0x40　　DP=0x40

(0x2400)=0x0　　(0x2400)=0x0

(0X2401)=0x0　　(0X2401)=0x1

……..

(0x247f)=0x0　　(0x247f)=0x7f

(0x2480)=0x0　　(0x2480)=0x100

(0x2481)=0x0　　(0x2481)=0x101

……

(0x24ff)=0x0　　(0x24ff)=0x17f

(0x2500)=0x0　　(0x2500)=0x200

(0x2501)=0x0　　(0x2501)=0x201

…….

(0x257f)=0x0　　(0x257f)=0x27f

实验五　快速傅立叶变换（FFT）

一 、实验目的

1. 了解 DSP 结构
2. 熟悉 CCS 开发环境
3. 熟悉 DSP 54X 教学开发实验系统的硬件环境
4. 掌握汇编程序的循环和分支结构

二 、实验内容

1. DSP 54X 汇编初始化程序；
2. 编写位码倒置程序；
3. 编写蝶形运算程序；
4. FFT 调用程序。

三 、实验要求

对输入数据进行 FFT 变换，并计算功率计算，并将功率谱结果用 CCS 的 View ->Graph 显示。

四 、实验环境

计算机、DSP 仿真器，DSP 54X 教学开发实验系统，CCS 软件，EVM54 库文件

五 、实验原理

1. 离散傅立叶变换 DFT 的定义：

2. $$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega}$$

3. DFT 的意义：将时域的采样变换为频域的周期性离散函数，频域的采样也可以变换为时域的周期性离散函数，从而把采样序列的频域与其连续时间信号的频与联系起来
4. FFT 是 DFT 的一种快速算法，将 DFT 的 N^2 步运算减少为$(N/2)\log_2 N$ 步，极大的提高了运算的速度
5. FFT 蝶型运算规律

6. FFT 码位倒置的原理
7. DSP 的特殊的间接寻址功能：比特翻转寻址功能
8. FFT 的两种实现方法：时间抽取法与频域抽取法

六 、实验步骤

1.初始化 DSP，调用 DSP 初始化程序；

2.编写位码倒置程序；

3.编写蝶形运算程序；

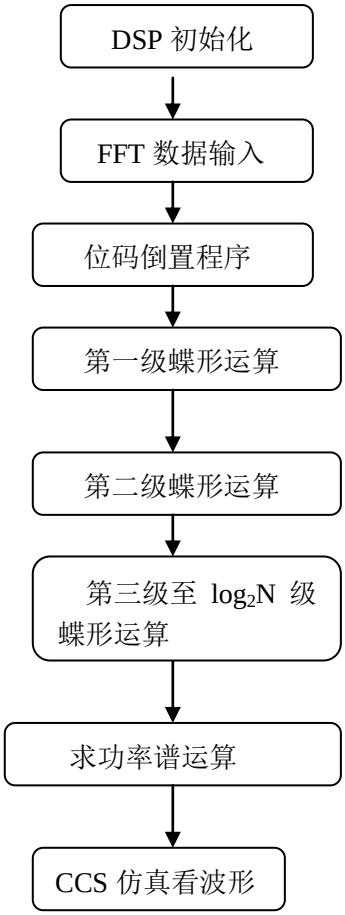
4.编写功率谱计算程序；

5.建立工程，编译调试程序；

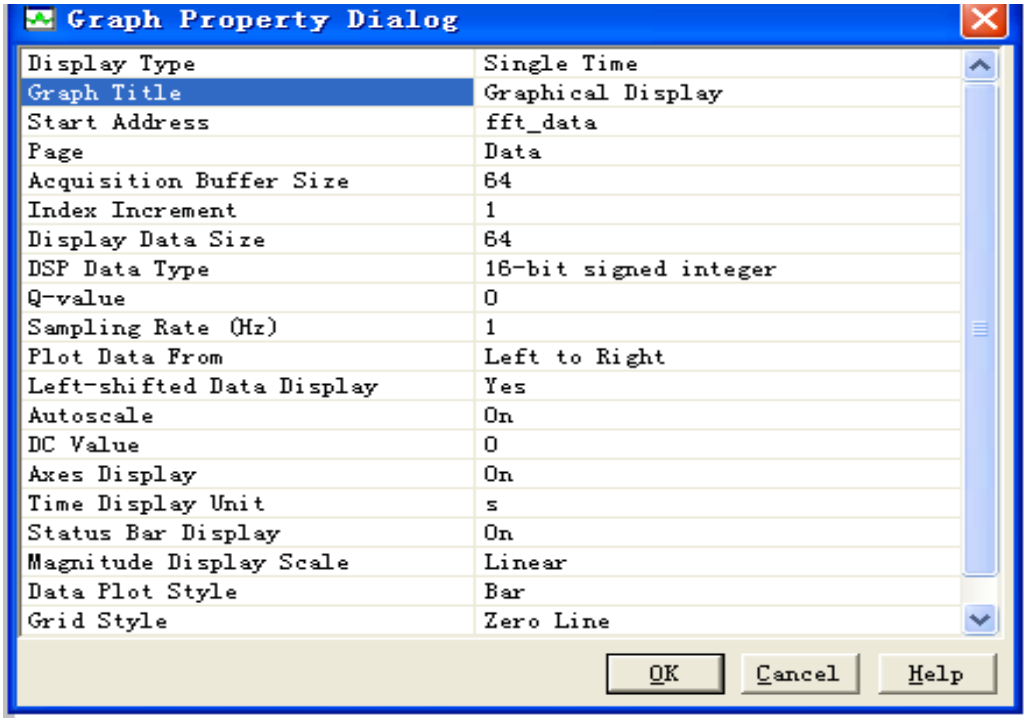
6.通过 CCS 的 View→ Graph → Time/Frequency 察看输出功率谱；

7. .在 Project ->Build Options 中设置添加文件的路径。

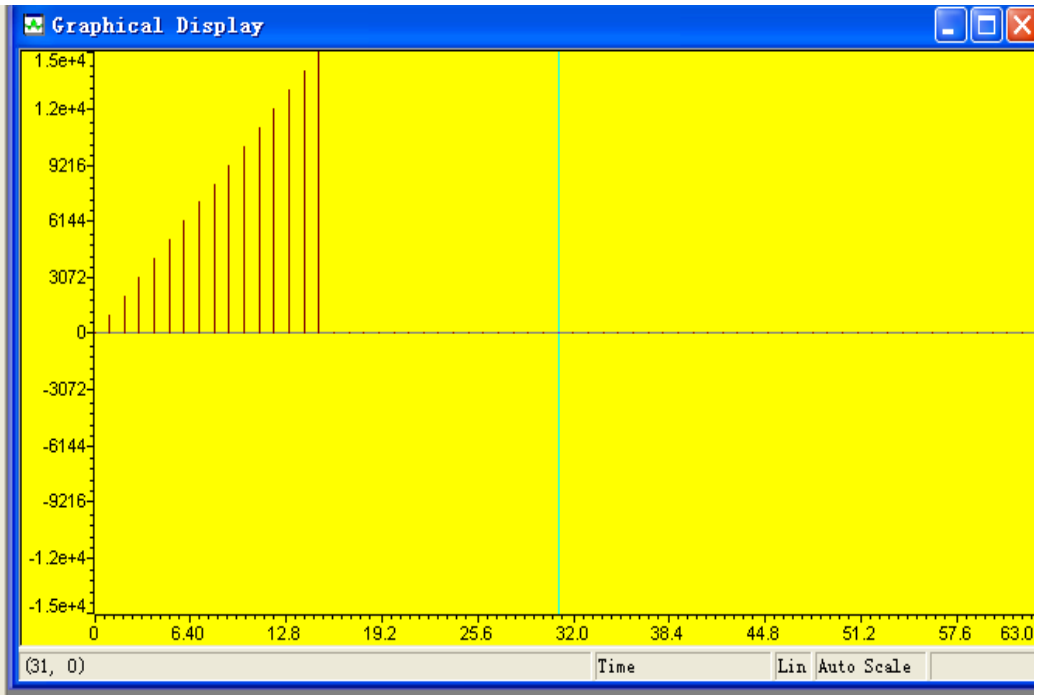
七 、实验框图



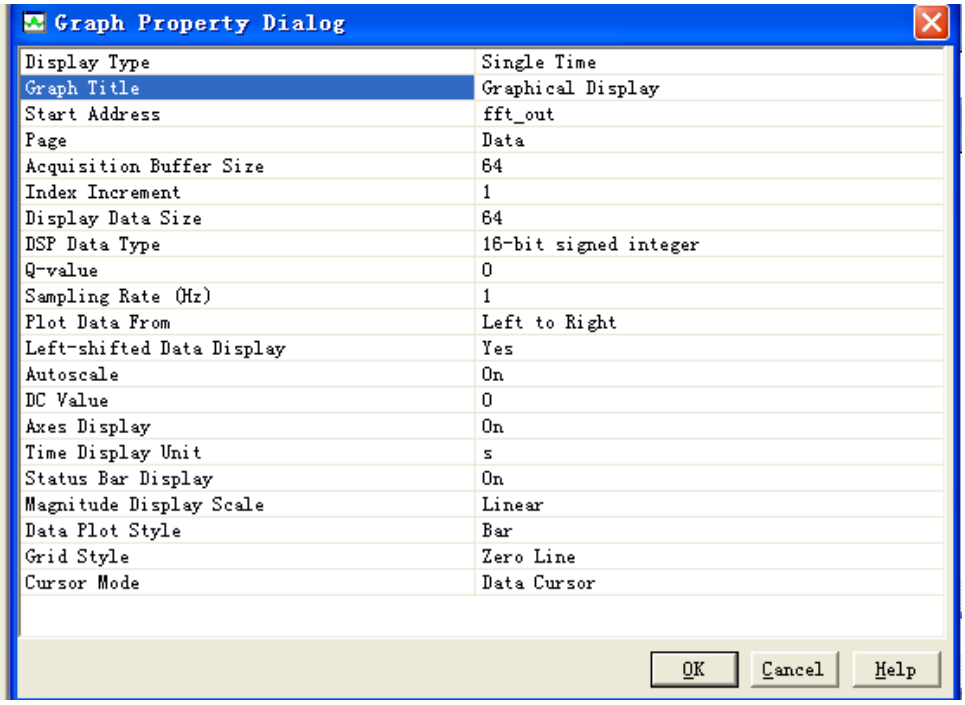
FFT 输入波形 fft_data 参数设置



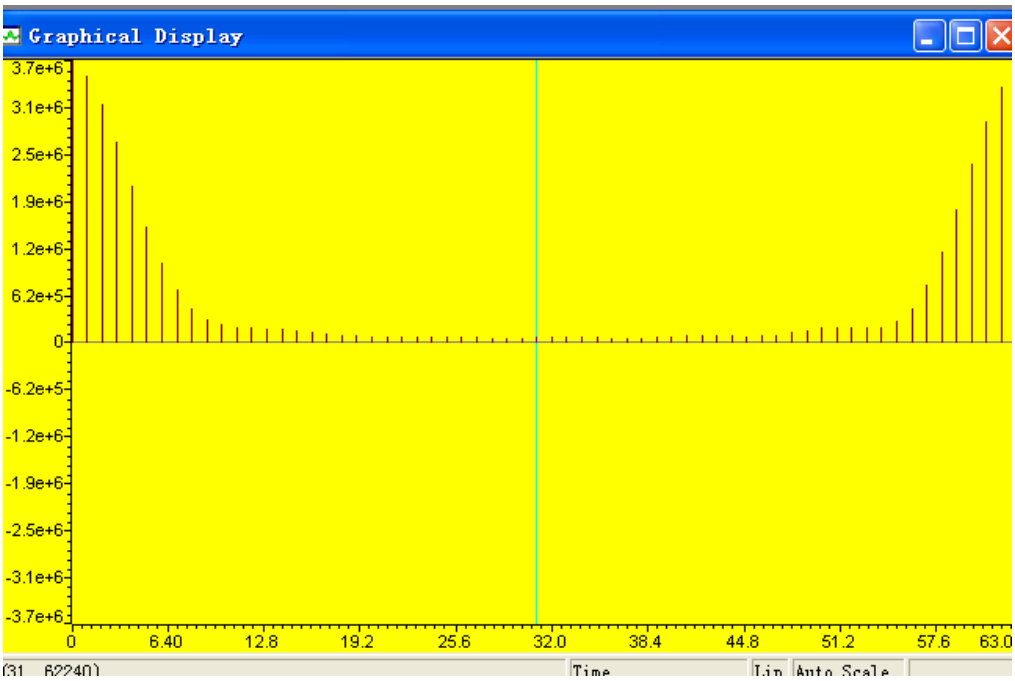
FFT 输入波形 fft_data



FFT 输出波形 fft_out 参数设置



FFT 输出 fft_out 波形



思考题

1. 对于不同的 N，幅频特性会相同嘛？为什么?
2. FFT 谱分析，有何应用？

实验六　数字滤波器的设计

一 、实验目的

1. 熟悉 CCS 开发环境
2. 熟悉 DSP 54X 教学开发实验系统的硬件环境
3. 掌握数字滤波器的设计过程
4. 了解 FIR、IIR 数字滤波器的特性
5. 熟悉设计 FIR、IIR 数字滤波器的原理和方法
6. 了解 DSP 滤波器的特殊寻址方式

二 、实验内容

1. DSP 初始化程序；
2. 确定滤波系数；
3. 滤波器程序；

三 、实验要求

将输入波形通过设计好的数字滤波器，结果用 CCS 的 View ->Graph 显示，分析数字滤波器的性能。

四 、实验环境

计算机、DSP 仿真器，DSP 54X 教学开发实验系统，CCS 软件，EVM54 库文件

五 、实验原理

1. 有限冲击响应数字滤波器的基础理论
2. 无限冲击响应数字滤波器的基础理论
3. 模拟滤波器原理（巴特沃斯滤波器、切比雪夫滤波器、椭圆滤波器、贝赛尔滤波器）
4. 数字滤波器系数的确定方法
5. 双线性变换的设计原理

许多算法如卷积、相关和 FIR 滤波器都需要环形缓冲区。环形缓冲区就是

一个滑动窗口，包含有最新的数据，当新数据到来时，缓冲区覆盖掉最老的数据（如图 1）。实现环形缓冲区的关键就是实现环形寻址。环形缓冲区大小寄存器（BK）用来存储环形缓冲区的大小。尺寸为 R 的环形缓冲区的地址必须对齐在 N 比特边界上（即，环形缓冲区基地址的最低 N 比特必须为 0），其中 N 为满足 2^N>R 的最小整数。如 31-字的环形缓冲区的基地址的最低 5 比特必须为 0（即，×××× ×××× ×××0 0000_2），值 31 必须存入 BK。环形寻址必须满足以下规则：

- 环形缓冲区的最低地址放在 2^N 边界上，2^N 大于缓冲区大小。
- 使用一个小于或等于缓冲区大小的步幅来寻址。
- 当第一次寻址时，辅助寄存器必须指向缓冲区中的一个元素，如 *AR3+%。

环形寻址算法为：

如果 0≤索引+步幅＜BK：索引=索引+步幅；

否则 如果 索引+步幅≥BK：索引=索引+步幅－BK;

否则 如果 索引+步幅＜＝0：索引=索引+步幅+BK。

一般都用寄存器 AR0 来存储步幅值。

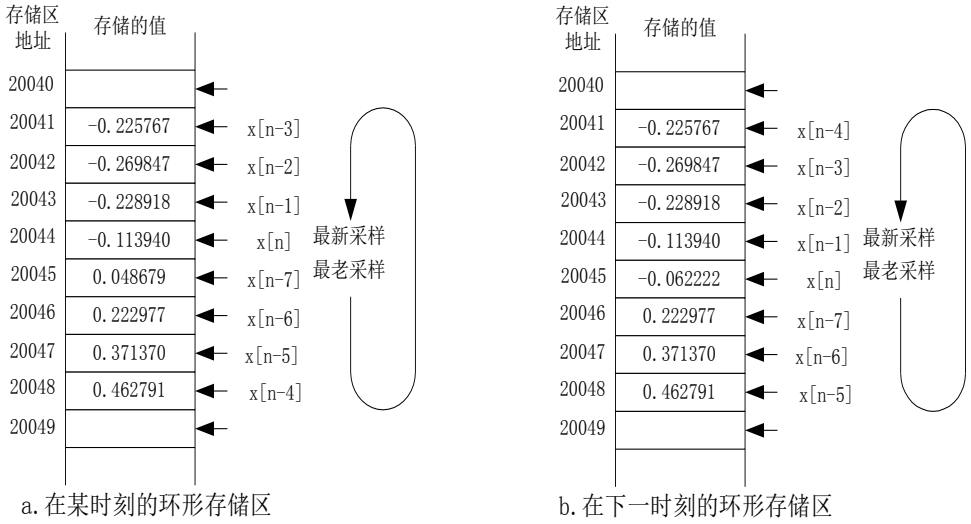


图 1

环形寻址的句法为：

- *ARx-%，表示存取 ARx 所指的地址的值，存取结束后，ARx 的值按照环形寻址模式减 1。
- *ARx-0%，表示存取 ARx 所指的地址的值，存取结束后，ARx 的值按照环

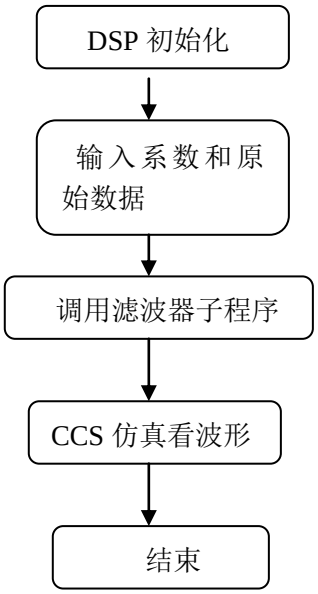
形寻址模式减去 AR0 的值。

- *ARx+%，表示存取 ARx 所指的地址的值，存取结束后，ARx 的值按照环形寻址模式加 1。
- *ARx+0%，表示存取 ARx 所指的地址的值，存取结束后，ARx 的值按照环形寻址模式加上 AR0 的值。

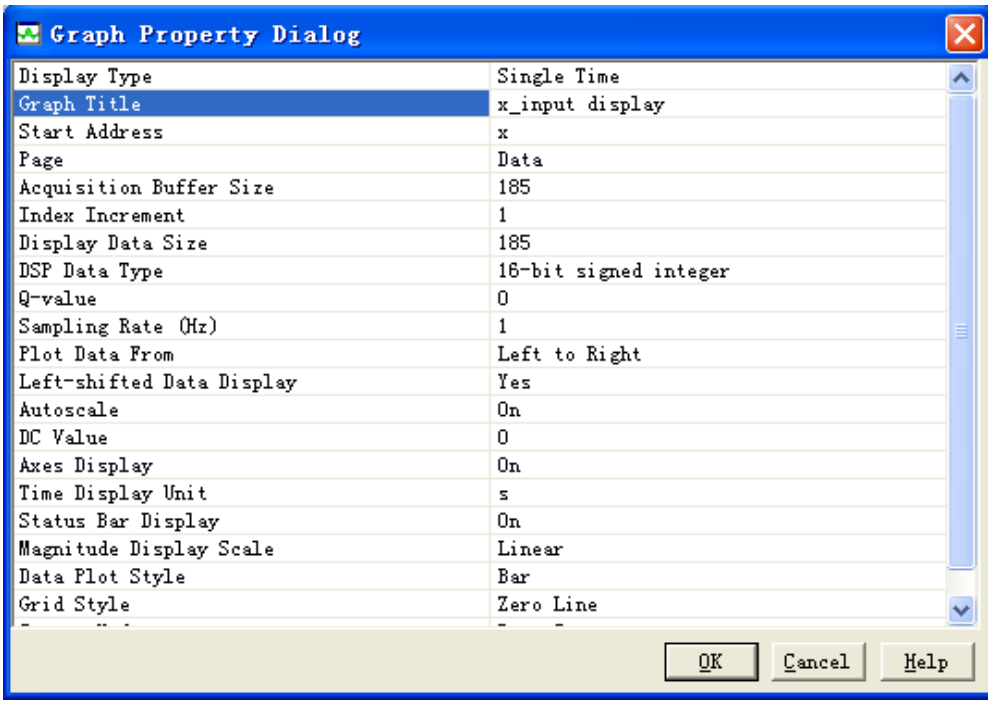
六 、实验步骤

1. 确定滤波器模型，设计滤波器，确定滤波器系数；
2. 初始化 DSP，调用 DSP 初始化程序；
3.编写滤波器子程序；
4.建立工程，编译调试程序；
5.在 Project ->Build Options 中设置添加文件的路径；
6.通过 CCS 的 View→ Graph → Time/Frequency 察看输出波形；
7.比较分析滤波器性能。

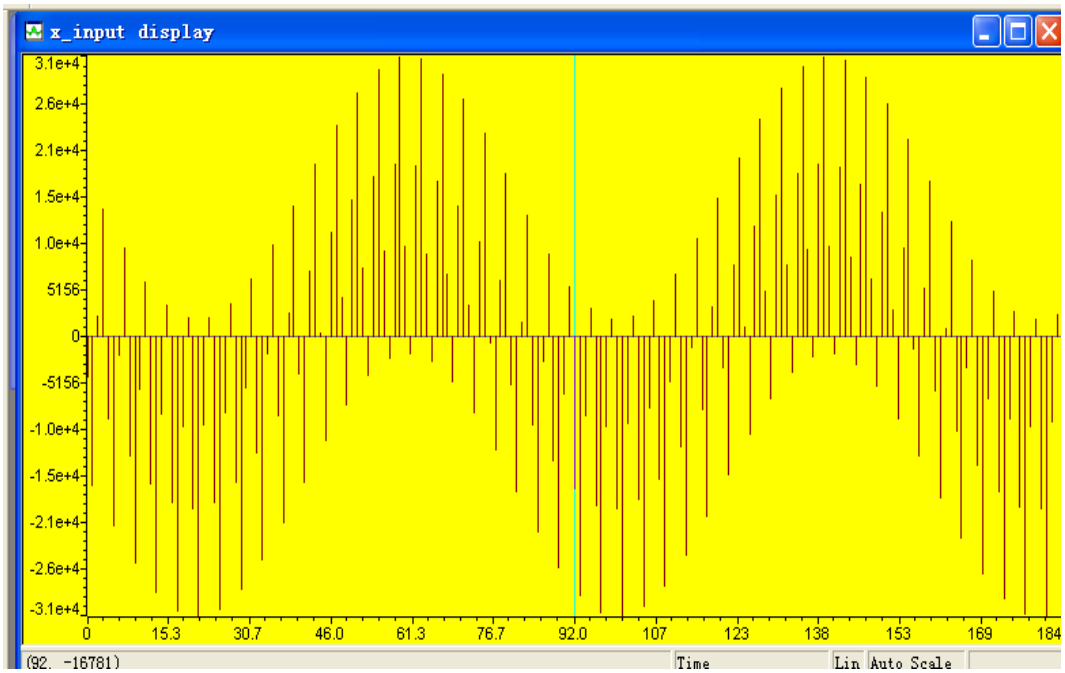
七 、实验框图



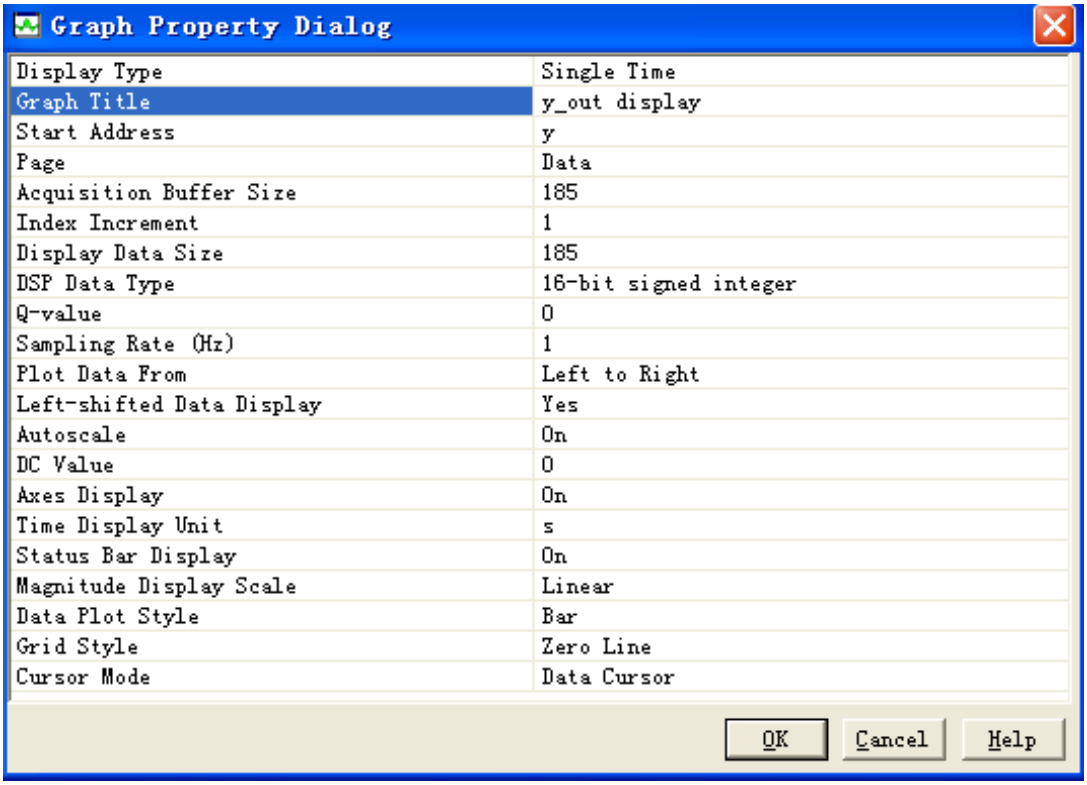
用 CCS 的 View→ Graph → Time/Frequency 察看输入波形 x 设置



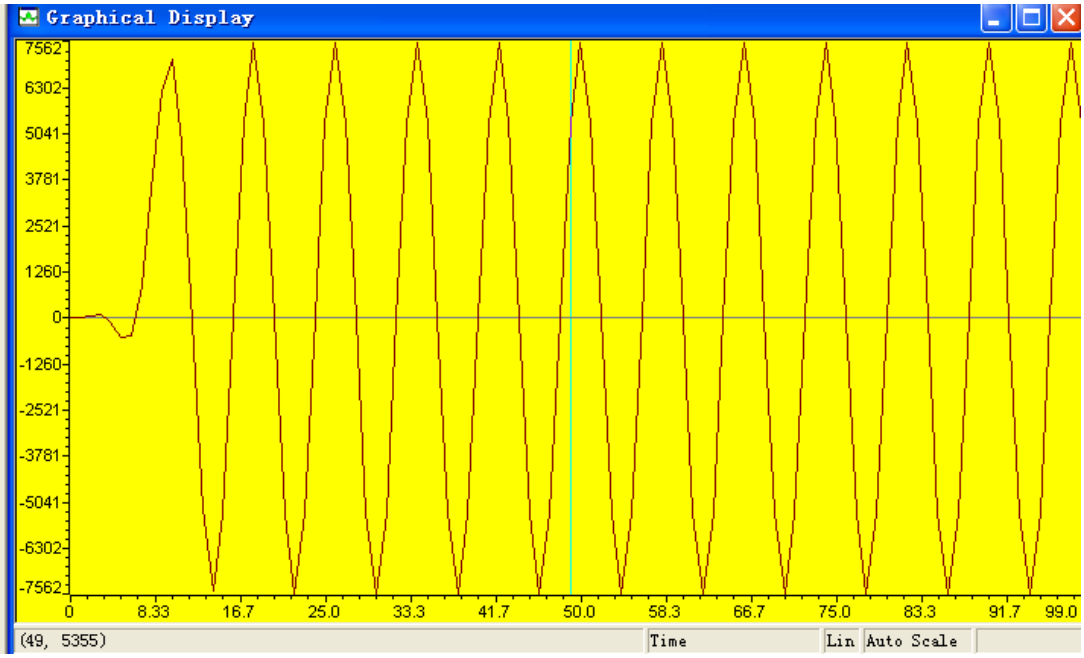
察看输入波形 x



用 CCS 的 View→ Graph → Time/Frequency 察看输出波形 y 设置



察看输出波形 y



思考题

比较 FIR 滤波器与 IIR 滤波器的异同

实验七　定时器及中断实验

一 、实验目的

1. 熟悉 CCS 开发环境
2. 熟悉 DSP 54X 教学开发实验系统的硬件环境
3. 了解 DSP 54X 定时器工作原理
4. 了解 DSP 54X 中断原理及中断相量表的建立
5. 了解数码管 LED 显示原理（动态扫描）
6. 了解 DSP 54X I/O 的寻址方式
7. 掌握键盘扫描原理
8. 掌握长整型数的运算
9. 掌握中断优先级的设置
10. 掌握全局变量、局部变量的概念

二 、实验内容

1. DSP 初始化程序；
2. 定时器 0 初始化；
3. 中断寄存器初始化；；
4. 中断相量表；
5. 定时器中断服务程序（LED 扫描驱动程序）；
6. 键盘扫描驱动程序；
7. 外部中断 1 中断服务程序。

三 、实验要求

通过定时器启动中断，在中断服务程序中扫描数码 LED 显示，由扫描键盘得到不同键盘值，根据不同的键值完成正计数、倒计数、停止、复位等功能。外部中断可终止计数。

资源

键盘 I/O 地址（0x3ffff）；

数码 LED　I/O 地址：段地址＝0x1fff,位地址＝0x2fff;

A#键＝0x01h(正计数)，

B#键＝0x02h(倒计数)，

C#键＝0x03h(复位)，

LED 显示段码：

0＝00H，1＝01H，2＝02H，3＝03H，4＝04H，5＝05H，6＝06H，7＝07H，8＝08H，9＝09H，A＝0AH，B＝0BH，C＝0CH，D＝0DH，-＝0EH，空＝1FH，

四 、实验环境

计算机、DSP 仿真器，DSP 54X 教学开发实验系统，CCS 软件，EVM54 库文件

五 、实验步骤

1.将 DSP 仿真器与计算机并口连接好；

2.将 DSP 仿真器 JTAG 头插入 DSP 5409EVM 板 JTAG 上；

3.打开计算机电源，当计算机启动完毕后，打开电源仿真器和 DSP 54X EVM 板电源，板上 3.3V、1.8V 电源指示灯均亮，若不亮，请立即关闭 EVM 板电源，检查连接和电源电压；

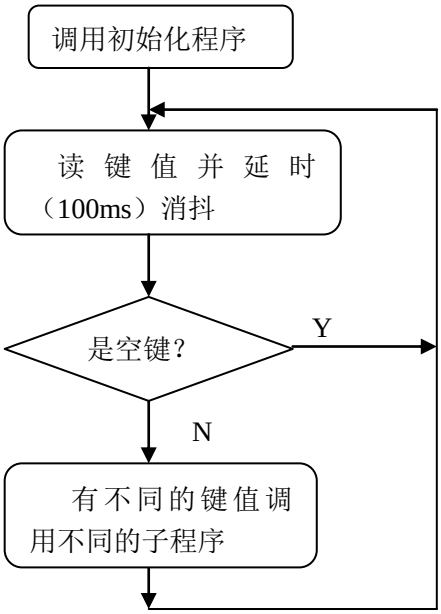
4.双击桌面上 CCS 图标，进入 CCS 操作环境；

5.创建一个工程和源文件，并调试;

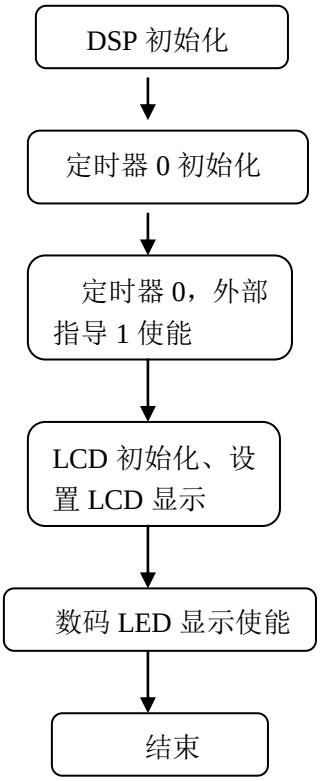
6.在 Project ->Build Options 中设置添加文件的路径。

六 、实验框图

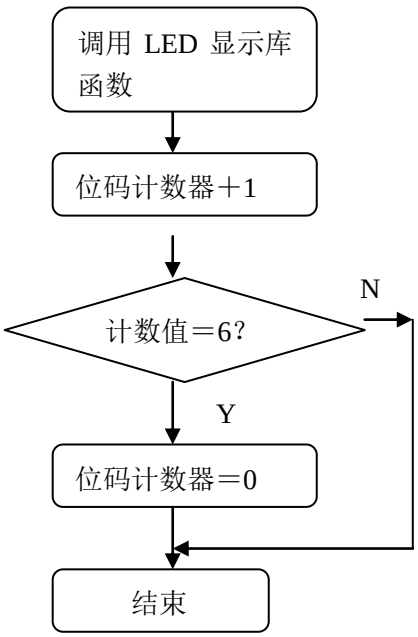
主程序



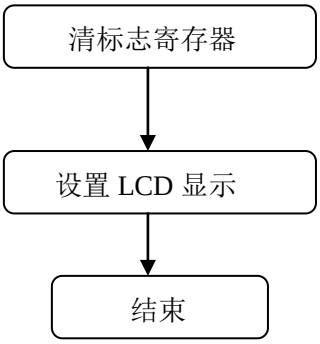
初始化



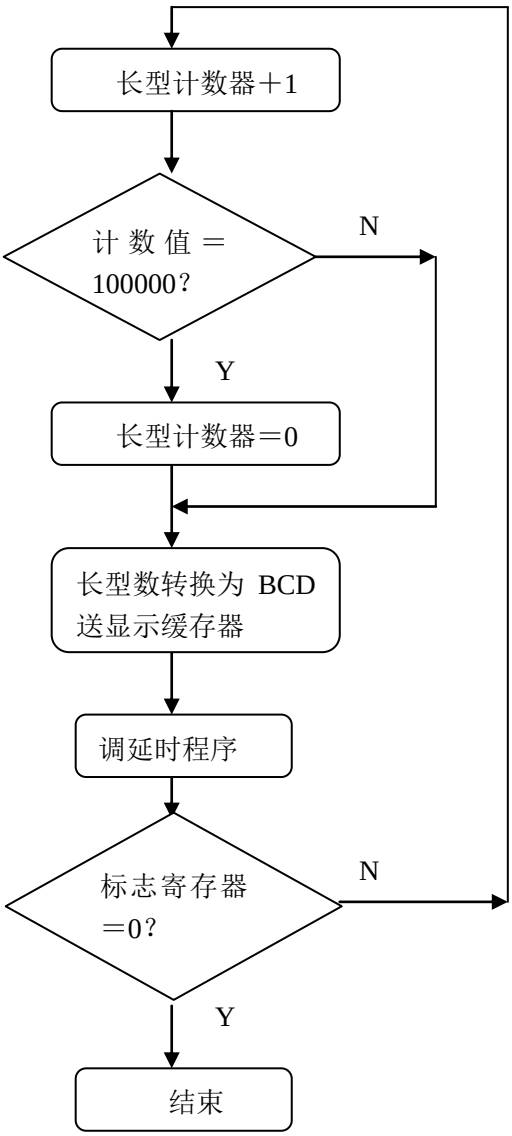
T0 中断程序



EXIT1 中断程序



正计数子程序


长型计数器＋1
计 数 值 ＝
100000?
N
Y
长型计数器＝0
长型数转换为 BCD
送显示缓存器
调延时程序
标志寄存器
＝0?
N
Y
结束

实验八　A/D-D/A 实验

一 、实验目的

1. 熟悉 CCS 开发环境
2. 熟悉 DSP 54X 教学开发实验系统的硬件环境
3. 了解 A/D 芯片 TLV2544 原理
4. 了解 D/A 芯片 TLV5638 原理
5. 了解 DSP McBSP 串口原理
6. 了解 DSP McBSP 串口作 I/O 设置
7. 掌握 DSP 54X I/O 寻址方式
8. 掌握长型浮点数概念

二 、实验内容

1. DSP 初始化程序；
2. LCD 初始化；
3. LCD 显示驱动；
4. A/D 芯片 TLV2544 初始化；
5. D/A 芯片 TLV5638 初始化；
6. DSP McBSP 串口 2 初始化为 I/O 模式；
7. 按 A/D 、D/A 芯片时序，用 DSP 构建 SPI 协议。

资源：

ADC-TLV2544ID 和 DAC-TLV5638CD 共用 McBSP2，一个外部中断 3。具体如下：

TLV2544（A/D） SDO-------BSP_DR2

SDI--------BSP_DX2

SCLK-----BSP_CLK2

CS--------- BSP_FSR2

EOC-------INT3

TLV5638（D/A） DIN----- BSP_DX2

SCLK-----BSP_CLK2

CS--------- BSP_FSR2

三 、实验要求

设置串口 2 为 I/O 模式，用 DSP 构建 SPI 协议：

1. 输入直流信号，启动 A/D 采样，将采样数据显示在 LCD 上，比较采样数据与计算机值应一致；

2. 通过 DSP 编程，产生锯齿波（或三角波），启动 D/A 转换，用示波器观测 D/A 输出应为锯齿波（或三角波）；

3. 输入交流信号（正弦波，频率＝<20KHZ，峰峰值＝<3V），启动 A/D 采样，将采样数据直接输入给 D/A 转换，用示波器观测 D/A 输出波形，比较 D/A 输出波形与输入交流信号应一致；

4. 在 LCD 上能显示相应的提示。能由外部中断终止某任务的执行。

四 、实验环境

计算机、DSP 仿真器，DSP 54X 教学开发实验系统，CCS 软件，EVM54 库文件，A/D 芯片 TLV2544；D/A 芯片 TLV5638，信号发生器，万用表，示波器

五 、实验步骤

1.将 DSP 仿真器与计算机并口连接好；

2.将 DSP 仿真器 JTAG 头插入 DSP 5409EVM 板 JTAG 上；

3.打开计算机电源，当计算机启动完毕后，打开电源仿真器和 DSP 54X EVM 板电源，板上 3.3V、1.8V 电源指示灯均亮，若不亮，请立即关闭 EVM 板电源，检查连接和电源电压；

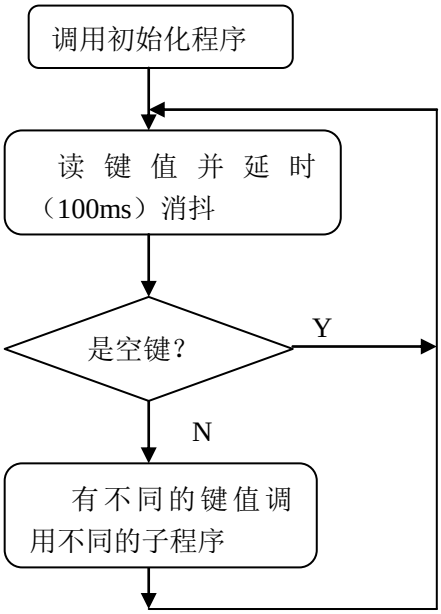
4.双击桌面上 CCS 图标，进入 CCS 操作环境；

5.创建一个工程和源文件，并调试;

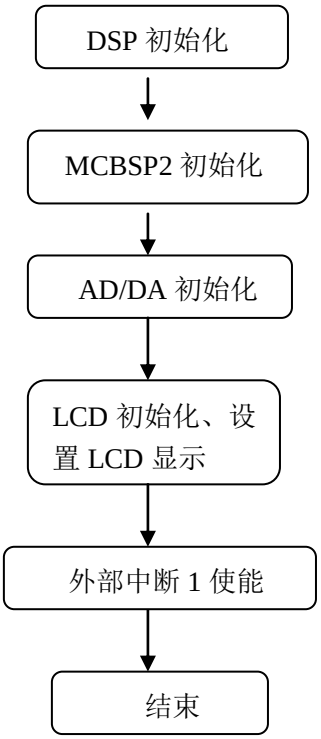
6.在 Project ->Build Options 中设置添加文件的路径。

六 、实验框图

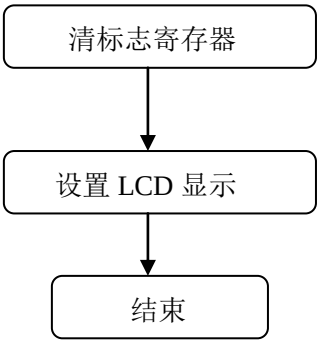
主程序



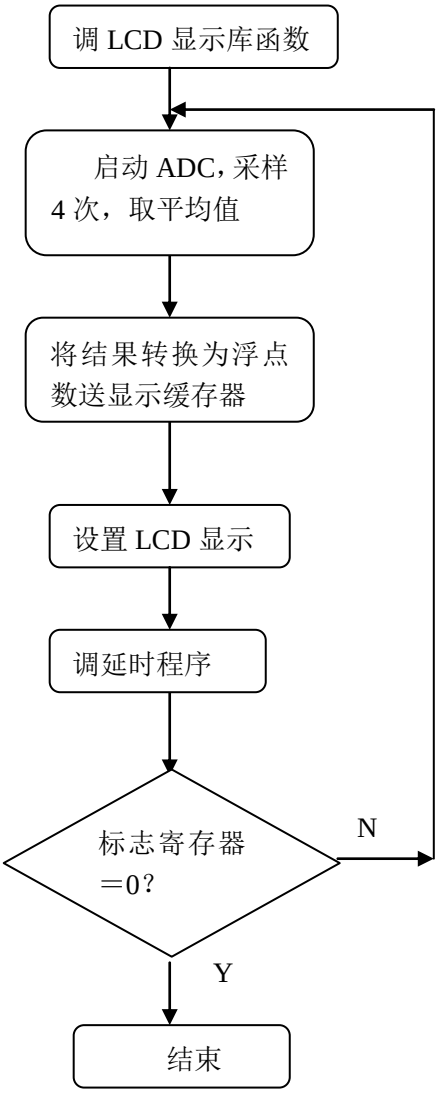
初始化



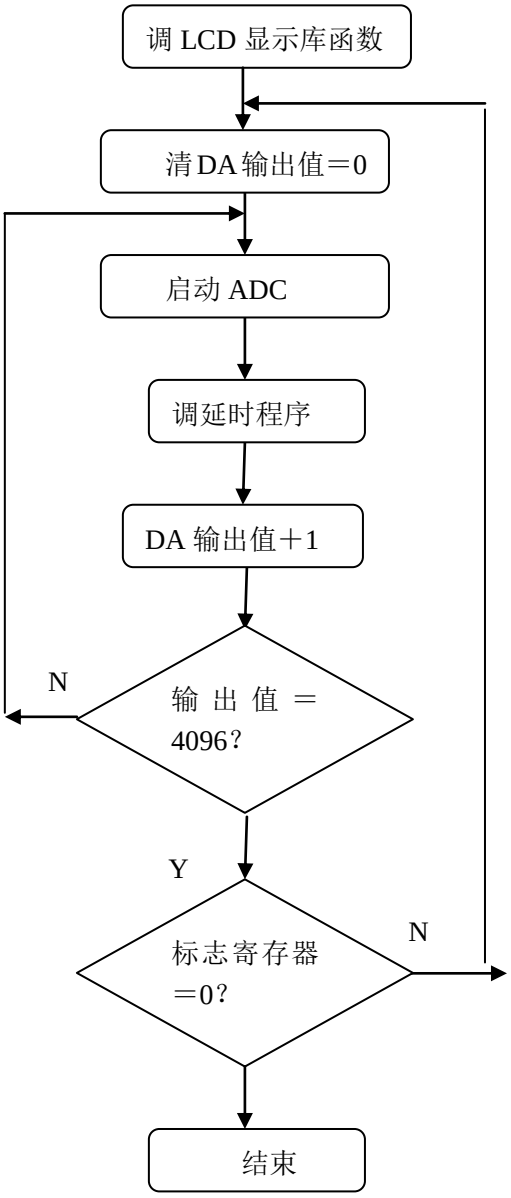
EXIT1 中断程序



ADC 程序


调 LCD 显示库函数
启动 ADC，采样 4 次，取平均值
将结果转换为浮点数送显示缓存器
设置 LCD 显示
调延时程序
标志寄存器 =0?
N
Y
结束

DAC 程序


调 LCD 显示库函数
清DA输出值=0
启动 ADC
调延时程序
DA 输出值+1
输 出 值 = 4096?
N
Y
标志寄存器 =0?
N
结束

附 A： 5416 中断向量表

Vector-relative locations and priorities for all internal and external interrupts are shown in Table 3–18.

Table 3–18. Interrupt Locations and Priorities

NAME	LOCATION DECIMAL	LOCATION HEX	PRIORITY	FUNCTION
$\overline{RS}$, SINTR	0	00	1	Reset (hardware and software reset)
$\overline{NMI}$, SINT16	4	04	2	Nonmaskable interrupt
SINT17	8	08	—	Software interrupt #17
SINT18	12	0C	—	Software interrupt #18
SINT19	16	10	—	Software interrupt #19
SINT20	20	14	—	Software interrupt #20
SINT21	24	18	—	Software interrupt #21
SINT22	28	1C	—	Software interrupt #22
SINT23	32	20	—	Software interrupt #23
SINT24	36	24	—	Software interrupt #24
SINT25	40	28	—	Software interrupt #25
SINT26	44	2C	—	Software interrupt #26
SINT27	48	30	—	Software interrupt #27
SINT28	52	34	—	Software interrupt #28
SINT29	56	38	—	Software interrupt #29
SINT30	60	3C	—	Software interrupt #30
$\overline{INT0}$, SINT0	64	40	3	External user interrupt #0
$\overline{INT1}$, SINT1	68	44	4	External user interrupt #1
$\overline{INT2}$, SINT2	72	48	5	External user interrupt #2
TINT, SINT3	76	4C	6	Timer interrupt
RINT0, SINT4	80	50	7	McBSP #0 receive interrupt (default)
XINT0, SINT5	84	54	8	McBSP #0 transmit interrupt (default)
RINT2, SINT6	88	58	9	McBSP #2 receive interrupt (default)
XINT2, SINT7	92	5C	10	McBSP #2 transmit interrupt (default)
$\overline{INT3}$, SINT8	96	60	11	External user interrupt #3
$\overline{HINT}$, SINT9	100	64	12	HPI interrupt
RINT1, SINT10	104	68	13	McBSP #1 receive interrupt (default)
XINT1, SINT11	108	6C	14	McBSP #1 transmit interrupt (default)
DMAC4,SINT12	112	70	15	DMA channel 4 (default)
DMAC5,SINT13	116	74	16	DMA channel 5 (default)
Reserved	120–127	78–7F	—	Reserved

The bit layout of the interrupt flag register (IFR) and the interrupt mask register (IMR) is shown in Figure 3–22.

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

Figure 3–22. IFR and IMR

附 B：实验箱 EVM 板硬件演示说明

一、 键盘定义：

1	2	3	4	5	6
7	8	9	A	B	C
D	E	F	**Esc**	**Enter**	0

二、 系统进入：

1. 接上电源。
2. 打开箱体开关，＋5VIN 电源指示灯亮
3. 打开系统电源开关 K1(板上顶部中)，3.3V/CVCC/+5V 指示灯亮
4. 打开模拟电源开关 K2(板上顶部右)。
5. 输入系统密码(键 4、3、2、1),LCD 将显示 1234 后将进入系统主菜单。

三、 主菜单说明：

主菜单有 11 个选项：（通过（1---B）键选择，LCD 显示其界面）

test EVM

1=TestLED　　2=TestLCD

3=DCMotor　　4=StepMotor

5=Testkey　　6=TestUART

7=TestADC　　8=TestDAC

9=Codec1　　A=Codec2

B=EDATEST　　C =TCP/IP

四、 LED 演示程序：

按“1”键 LED 将依数码管，双色灯，单色灯次序轮训显示，按 ESC 键退出。

五、 LCD 演示程序：

按“2”键 LCD 依次显示横条，并显示测试结果，按 ESC 键退出。

六、 直流电机演示程序：

按“3”键直流电机将顺时，逆时转动，按 ESC 键退出。

七、 步进电机演示程序：

按“4”键步进电机将顺时，逆时转动，按 ESC 键退出。

八、 键盘测试演示程序：

按“5”键输入所有按键后显示结果退出，按 ESC 键退出。

九、 异步串口演示程序：

将单排针 J41 短路，发送接收跳针短路形成自回路，按“6”键测试串口，程序将发送（0x00—0xFF）的数据，LCD 将显示发送和接收数据的情况，当发送和接受数据一致时，串口正常。按 ESC 键退出。

十、 模数转换演示程序：

按住 S1 或 S2 或 S3 键（板上顶部左），按“7”键，LCD 显示各通道的电压值。其中通道 1 显示 0.8V 或 1.2V 或 1.6V 电压值。

十一、 数模转换演示程序：

按“8”键，用示波器观看锯齿波，按 ESC 键退出。其中输出信号和地在板上单排针有标注（板上顶部左）

十二、 现场录音程序：

按“9”键，将耳机接入到板上的 AUDIO OUT（板上中左），现场声音将通过耳机输出，按 ESC 键退出。

十三、 歌曲播放程序：

按“A”键，通过板上的 LINE IN 接口（板上中左），用 LINE IN 线与计算机声卡连接，将计算机播放的音乐通过耳机输出，按 ESC 键退出。

十四、 EDA 测试程序：

系统断电，将 EDA 板插入扩展总线上（注意方向，并口应在左边，EDA 板上有埋针）。系统上电测试 EDA，按“B”键，在 LCD 屏上显示 4 项测试内容，1.测试 EDA 正计时，2.测试 EDA 倒计时，其间有间隙的蜂鸣声，3.测试 EDA 板上 8 个 LED 灯（38 译码器），4.测试 EDA 拨码开关，拨动开关，在 LCD 屏上显示开关的值；

志寄存器＝0?

十五、 TCP/IP 模块测试程序：

系统断电，将 TCP/IP 模块板插入扩展总线上（注意方向，网口应在左边，板上有埋针）。系统上电测试，按“C”键，在 LCD 屏上显示 IP 地址和网址，1.在 PC 机上点开始－>运行输入 ping -t 192.168.16.2，在 PC 机上能看到通讯的信息；2.在 IE 浏览器地址栏输入 http://192.168.16.2，能浏览网页。