

第12章 DSP开发工具与开发环境

DSP的软件、硬件的开发以及系统的集成，日益关注。如何提高开发速度、降低开发难度，所有开发者共同关心。

除了必须了解DSP本身的结构和技术指标外，大量的时间和精力花费在熟悉和掌握开发工具和环境上。

系统复杂程度的百分之八十取决于软件。所以，设计人员都极为看重先进的、易于使用的开发环境与工具。

DSP的开发环境如何，开发工具的功能是否丰富，使用是否方便，是一件十分重要的事情。

TI的DSP开发环境和工具主要包括以下3个方面：

(1) 代码生成工具（编译器、链接器、优化C编译器、转换工具等）

(2) 系统集成及调试环境与工具

(3) 实时操作系统

一个DSP软件可使用汇编或C语言编写源程序，通过编译、链接、工具产生DSP执行代码。

在调试阶段，可利用软件仿真在计算机上仿真运行；也可利用硬件调试工具将代码下载到DSP中，并通过计算机监控、调试运行该程序。

当调试完成后，可将该程序代码固化到程序存储器中，以便DSP目标系统脱离计算机单独运行。

12.1 系统集成与调试工具

12.1.1 软件模拟模拟器

软件模拟器是一种模拟DSP芯片功能并在非实时条件下进行软件调试的调试工具，它不需要硬件支持，只需

在计算机上运行，是一种**廉价方便的调试工具**。

软件模拟器模拟DSP芯片的I/O口时采用与文件关联的方法来实现，因而**调试中所需的I/O值可从文件中读取，输出的I/O值也可存储在文件中**。

可以按用户定义的时间间隔，周期性的设置中断标志，**仿真中断信号**。

在程序执行之前作初始化，设置断点及跟踪模式。

程序执行一旦终止，就可对内部寄存器、程序和数据存储器作检查和修改，也可显示跟踪寄存器。

整个仿真的记录可以做成一个文件，下次再做仿真的时候，运行该文件就可以恢复同样的机器状态。

软件模拟器的主要功能有：

- (1) 在计算机上执行用户的DSP程序；
- (2) 可修改和查看寄存器；

- (3) 可修改和查看程序和数据存储器，任何时候都可进行存储器的修改，也可在程序装入前进行存储器初始化；
- (4) 可模拟外设，高速缓存（Cache）及流水线、定时功能。
- (5) 可计算指令周期数。
- (6) 可编程的断点设置：可在取指令、读写存储器及错误条件满足时设置断点
- (7) 可进行累加器、程序计数器、辅助寄存器的跟踪
- (8) 指令的单步执行
- (9) 用户设定的中断产生间隔
- (10) 在遇到非法操作码和无效数据访问时提示错误信息
- (11) 从文件中执行命令

早期的软件模拟器与其他的开发工具是分离的。使用起来不方便。现在已经作为CCS (Code Composer Studio) 的一个部件。

12.1.2 集成开发环境CCS

CCS (Code Composer Studio) 是一个完整的集成开发环境，也是目前使用最为广泛的DSP开发软件之一。

最早由GO DSP公司为TI的VC6000系列开发，后来TI收购GO DSP，并将CCS扩展到其它系列。所有的TI DSP都可以用CCS开发。但目前只有VC5000和VC6000的CCS中提供DSP/BIOS功能。因为，VC2000系列的CCS中没有DSP/BIOS功能，也称为CC (Code Composer)，以示区别。

现在所有的TI DSP都可利用该软件工具进行开发。

一、CCS的功能

- (1) 集成可视化的代码编辑界面，可直接编写C、汇编、.H文件、.cmd文件等.
- (2) 集成代码生成工具, 包括汇编器, 优化C编译器, 连接器等.
- (3) 基本调试工具, 如装入执行代码, 查看寄存器, 存储器, 反汇编, 变量窗口等, 支持C源代码级调试.
- (4) 支持多DSP调试
- (5) 可设置硬件断点、数据空间读/写断点、条件断点（使用GEL编写表达式）
- (6)
- (7)
- (8)

- 二、 为CCS安装驱动程序
- 三、 利用CCS开发DSP流程
- 四、 探针工具的使用
- 五、 图形工具的使用
- 六、 分析工具的使用

12.2 仿真器

两种类型的仿真器

- (1) 传统的电路仿真器
- (2) 先进的扫描仿真器

利用传统的电路仿真器进行硬件仿真，仿真器的仿真插头必须插入到硬件电路中DSP相应的位置。仿真插头引脚与DSP芯片引脚一一对应。

扫描仿真器不采用插入仿真的方法，而是通过DSP芯片上提供的几个仿真引脚实现仿真功能。这种方法为TI公司所开发，用来解决高速DSP芯片的仿真。

由于DSP芯片具有高度的并行结构、快速的指令周期、

高密度的封装，采用传统的电路仿真方法很难实现可靠的仿真。

12.2.1 各类仿真器及接口

仿真器品牌：

- (1) 闻亭；
- (2) 合众达 (SEED)；
- (3) 中泰康
- (4) 其他

计算机接口：

PCI 接口/并行口/USB接口

DSP接口：

IEEE1149.1 (JTAG)

扫描仿真消除了传统的电路仿真存在的问题：仿真电缆过长会引起信号失真，仿真插头的可靠性差的问题。

DSP是通过内部移位寄存器扫描链，实现扫描仿真。采用扫描仿真，即使芯片已经焊在电路板上，也可进行仿真调试。带来了极大的方便。

12.2.2 JTAG接口

仿真头采用14根信号线，仿真头扫描仿真信号线采用JTAG标准IEEE1149.1。如图12-1（b）所示。

仿真信号的定义如表12-3

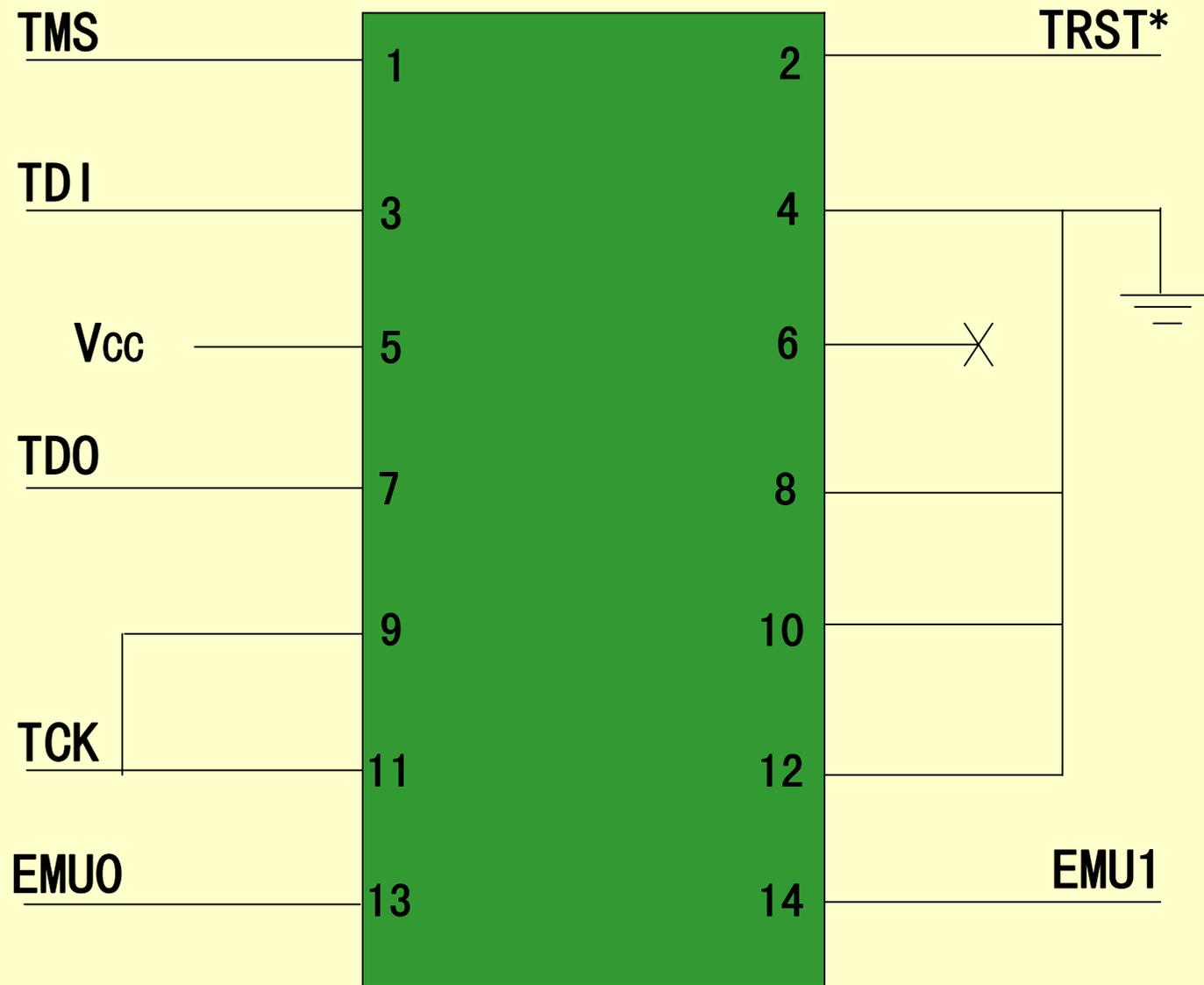
表12-3 JTAG仿真信号的定义

信号	仿真头状态	DSP芯片状态	信号说明
TMS	输出	输入	JTAG测试方式选择
TDI	输出	输入	JTAG测试数据输入
TDO	输入	输出	JTAG测试数据输
TCK	输出	输入	JTAG测试时钟
TRST	输出	输入	JTAG测试复位
EMU0	输入	输出	仿真脚0
EMU1	输入	输出	仿真脚1
PD	输入	输出	指示仿真电缆连接 (在目标系统中应接至+5V)
TCK_RET	输入	输出	JTAG测试时钟返回

TMS	1	2	$\overline{\text{TRST}}$
TDI	3	4	GND
PD (V _{CC})	5	6	no pin (key)†
TDO	7	8	GND
TCK	9	10	GND
TCK	11	12	GND
EMU0	13	14	EMU1

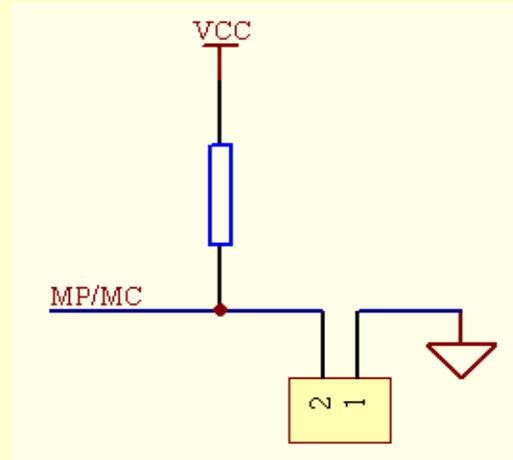
注1: 脚6没有连接，而且应该把双排针连接器该针脚去掉，仿真器的DSP连接器通常用此脚进行定位。

注2: 根据IEEE1149.1标准，TDI、TCK、TMS、TRST*四个引脚最好接上拉电阻。另外EMU0、EMU1两个仿真控制脚也最好接上拉电阻。



➤ DSP仿真环境

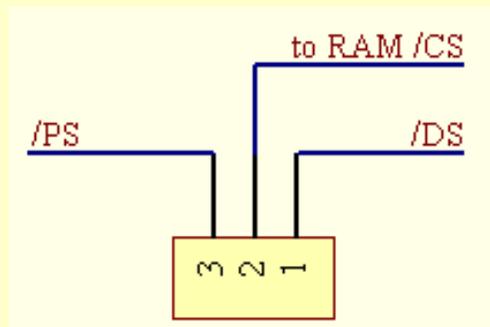
» MP/MC* = 1, 微处理器模式



» VccP (Pin 50) = 5V

» 外部时钟要保证CPUCLK = 40MHz

» 仿真RAM (PS* or DS* 选通)



12.3 DSP开发环境

- Code Composer (v4. 1)
 - » CC For C2000
 - » CC For C3xC4x
- Code Composer Studio (v2. 2)
 - » CCS 2000
 - » CCS 5000
 - » CCS 6000

其他开发工具:

- 软件模拟器 (Software Simulator)
- 初学者开发套件DSK (DSP Starter Kit)
- 软件评估模块EVM (Evaluation Module)

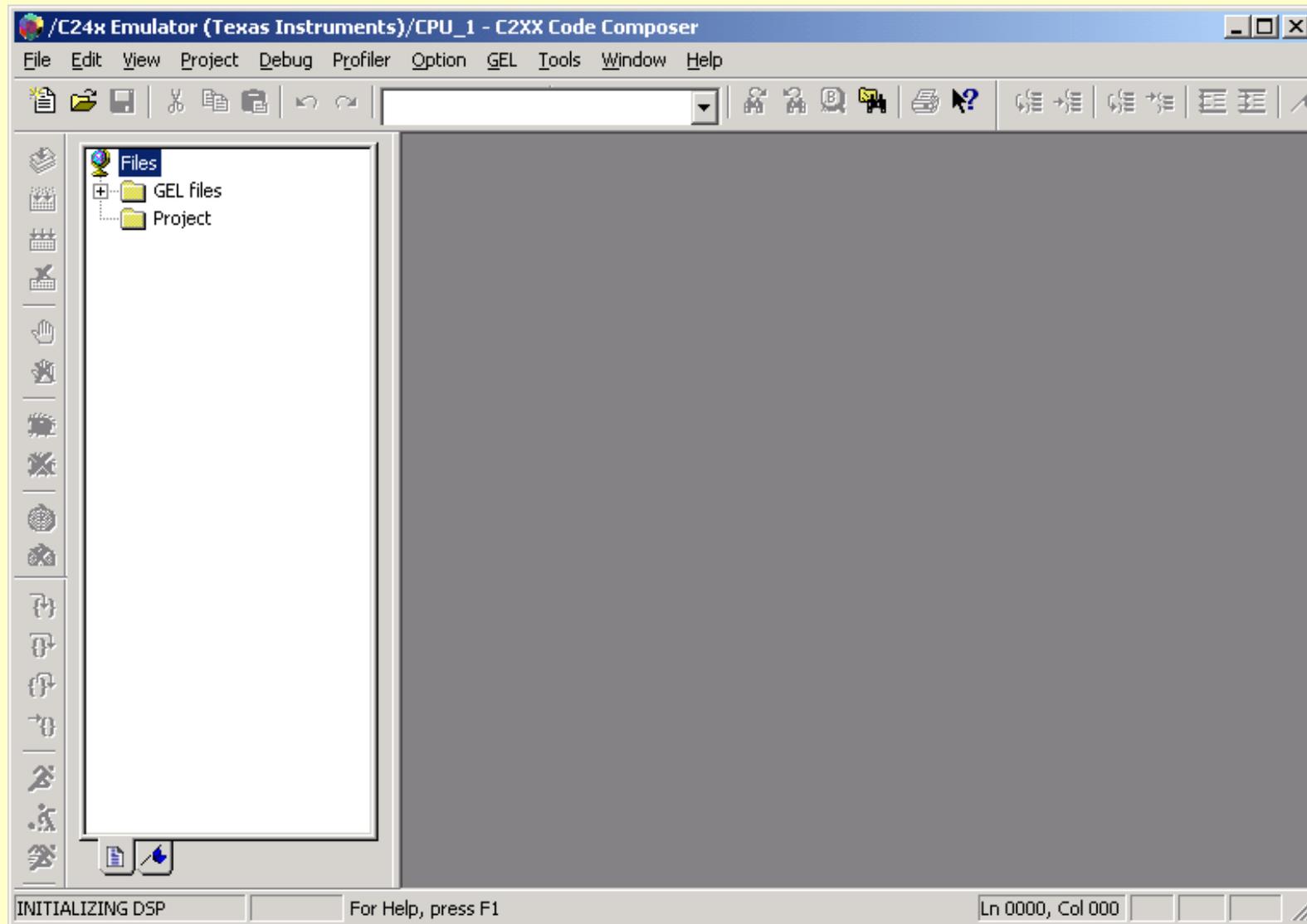
Code Composer特性:

- 集成开发环境
- 对C和DSP汇编文件的目标管理
- 高集成的编辑器调整C和DSP汇编代码
- C代码优化编译功能
- 丰富的调试手段:
 - 断点、单步、
 - 观察窗口、探针、图形分析
 - 寄存器、内存查看
- 目标DSP的帮助

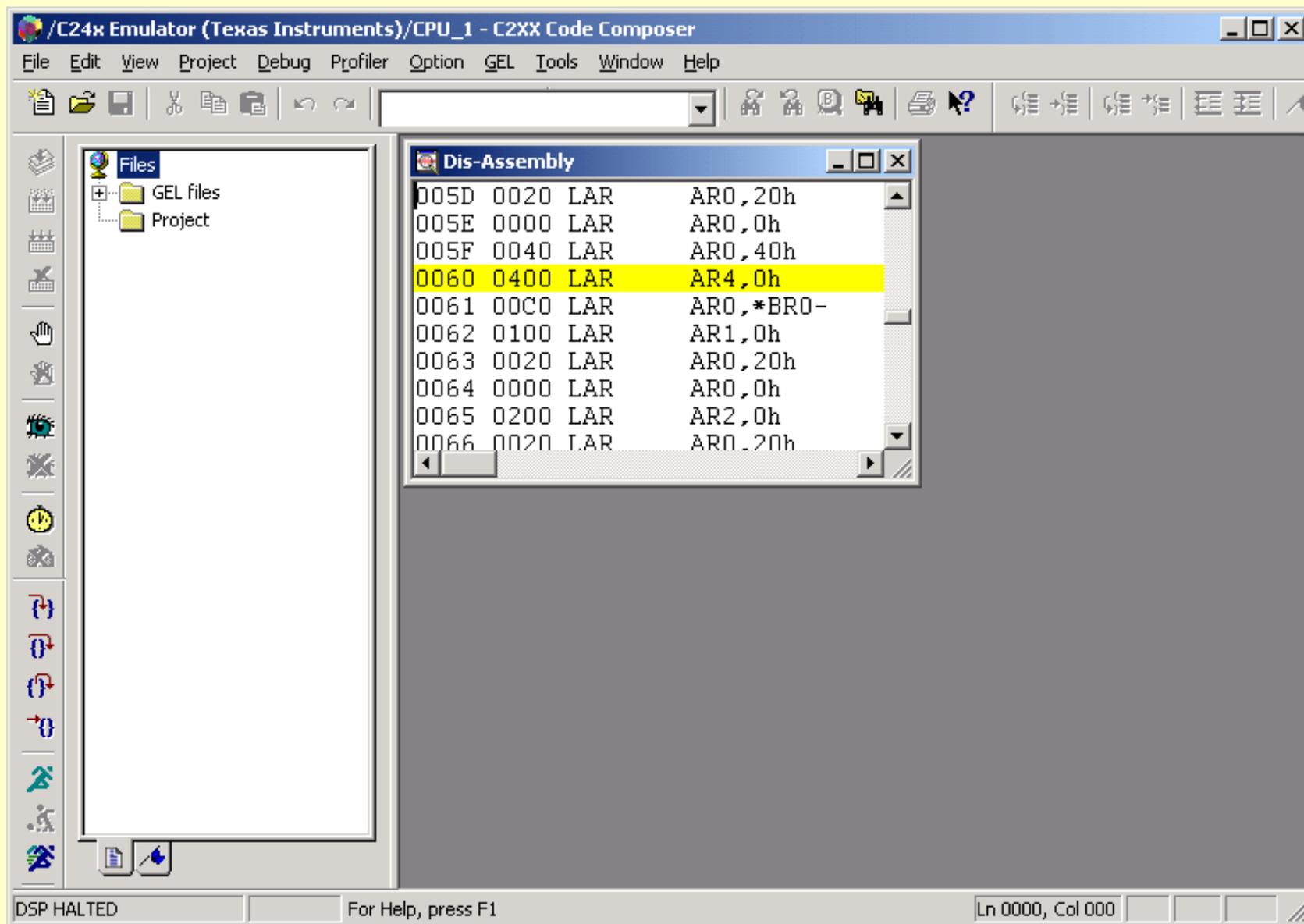
CC初始化



未连接仿真器初始窗口



正常连接仿真器初始窗口



打开工程后窗口

The screenshot displays the C24x Emulator (Texas Instruments) CPU_1 - C2XX Code Composer - MAIN.MAK - [adp.c] interface. The window is divided into several sections:

- File Explorer (Left):** Shows the project structure, including files like `adc.cmd`, `head.h`, `math.h`, `Libraries`, `rts2xx.lib`, `Source`, `adp.c`, and `adv.asm`.
- Code Editor (Center):** Contains the following code:

```
#include <head.h>
#include <math.h>

/*-----*/
/*define i/o port address*/
ioport unsigned int port0000; /*继电器控制1*/
ioport unsigned int port0001; /*继电器控制2*/
ioport unsigned int port0002; /*继电器控制3*/
ioport unsigned int port0003; /*继电器控制4*/
ioport unsigned int port0004; /*继电器控制5*/
ioport unsigned int port0005; /*继电器控制6*/
ioport unsigned int port0006; /*继电器控制7*/
ioport unsigned int port0007; /*继电器控制8*/
ioport unsigned int port0008; /*继电器控制9*/
ioport unsigned int port0009; /*adc通道选择及模拟激光源信号控制端*/
ioport unsigned int port000a; /*差分放大倍数控制*/
ioport unsigned int port000b; /*继电器回读1*/
ioport unsigned int port000c; /*继电器回读2*/
ioport unsigned int port000d; /*继电器回读3*/
ioport unsigned int port000e; /*继电器回读4*/
ioport unsigned int port000f; /*继电器回读5*/
ioport unsigned int port0010; /*继电器回读6*/
ioport unsigned int port0011; /*继电器回读7*/
ioport unsigned int port0012; /*继电器回读8*/
ioport unsigned int port0013; /*继电器回读9*/
ioport unsigned int port0014; /*读adc变换结果*/
ioport unsigned int port0015; /*启动ad变换*/

/*-----*/
unsigned int ad(unsigned int ad_channel); /*use ad to measure dc*/
unsigned int measu_freq(void); /*measure frequency*/
```
- Console Window (Bottom):** Shows the build output:

```
Copyright (c) 1987-1999 Texas Instruments Incorporated
Build Complete,
0 Errors, 0 Warnings.
```
- Error Window (Bottom Right):** Shows a yellow error message: `error = unknown identifier`.

The status bar at the bottom indicates "DSP HALTED" and "For Help, press F1". The current cursor position is "Ln 1, Col 1".

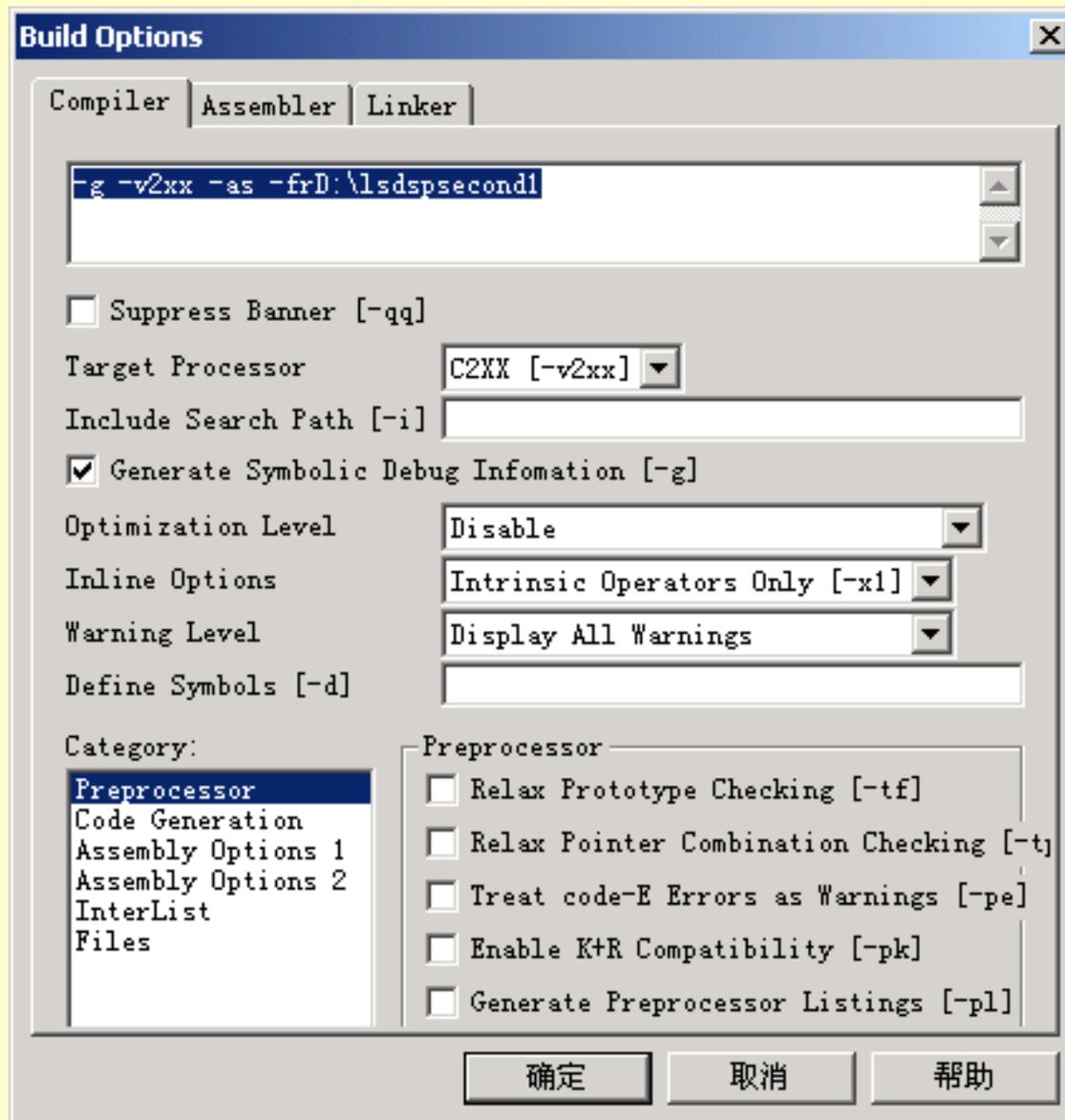
运行过程窗口

The screenshot displays the C24x Emulator interface. The main window shows the following assembly code:

```
*(rticr)=0x0007; /*disable the rti interrupt*/  
  
*(ocra) =0x4000; /*set i/o:IOPA0-3,IOPB0-5 and iopb7*/  
*(ocrb) =0x00C4; /*set CAP3,CAP4,IOPC0,2,4,5*/  
*(pbdatdir)=0xff00;  
*(pcdatdir)=0x1fff; /*set iopc2 output low,others are input*/  
  
*(gptcon) =0xe000; /*disable compare output,don't start ad, count up*/  
*(tlcon) =0x0000;  
*(t2pr) =0xffff; /*set t2cnt period*/  
*(t3pr) =0xfffe; /*set t3cnt period*/  
  
*(evifra) =0xffff;  
*(evifrb) =0xffff;  
*(evifrc) =0xffff;  
*(evimra) =0x0000;  
*(evimrb) =0x0000;  
*(evimrc) =0x0000;  
  
*(xint1cr) =0x8004; /*clear xin1 flag,rising edge,disable xint1*/  
*(xint2cr) =0x0004;  
*(xint3cr) =0x0004;  
  
*(sciccr) =0x0037; /*1 stop bit,odd parity,8 char bits,async mode,idle-line protc*/  
*(scictl1)=0x0013; /*enable tx,rx,internal sciclk,disable rx err,sleep,twake*/  
*(scictl2)=0x0000; /*disable tx&rx int*/  
*(scihbaud)=0x0;  
*(scilbaud)=0x0040; /*baud rate=9600b/s (5MHz sysclk)*/  
*(scipc2)=0x0032; /*enable txd & rxd pins*/  
*(scictl1)=0x0033; /*relinquish sci from reset*/
```

The bottom status bar shows "DSP HALTED" and "For Help, press F1". The bottom right corner shows "Ln 89, Col 1".

工程编译选项 (Project->Options->Compiler):



END

