

**Интегрированная среда разработки
и отладки программного
обеспечения
Code Composer Studio**

**Состав, общая характеристика
Настройка (Target and Host Setup)
Интерфейс пользователя**

Проблемы разработки высокотехнологичных устройств

- Стремление к упрощению аппаратных средств приводит к усложнению программного обеспечения (ПО).
- Создание ПО начиная с нулевого цикла значительно увеличивает время разработки.
- Удорожание методов и средств тестирования устройств.

Технология eXpressDSP

Технология **eXpressDSP** – это законченная технология, включающая набор интегрированных программных и аппаратных средств и решений, а также методов разработки, что позволяет существенно ускорить и облегчить процесс разработки высокотехнологичных устройств.

Составные части:

1. интерфейс внутрисхемной эмуляции на базе тестового интерфейса IEEE 1149 JTAG
2. технология RTDX - технология обмена данными в реальном времени (Real Time Data Exchange)
3. интегрированная среда разработчика (IDE) Code Composer Studio (CCS)
4. масштабируемое ядро операционной системы реального времени DSP BIOS
5. DSP Algorithm Standard - набор правил и приемов написания программных модулей
6. интерфейсы приложений (API) для стандартной периферии
7. библиотеки поддержки чипа (Chip Support Library)

Code Composer Studio™ IDE

PLATINUM EDITION



Code Composer Studio Development Tools v3.3

Getting Started Guide

Literature Number: SPRU509H

October 2006

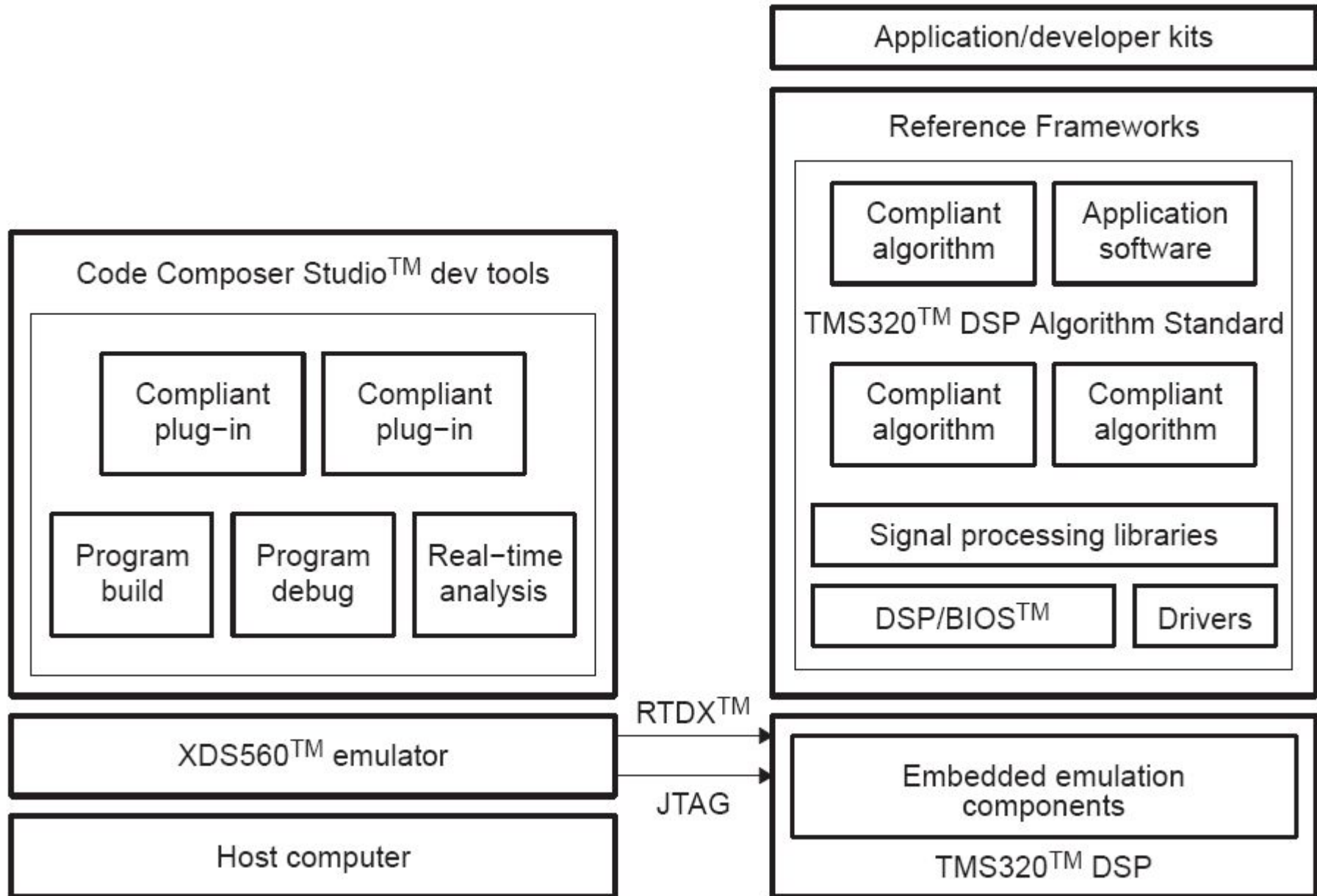
**Интегрированная среда разработки
и отладки программного
обеспечения
Code Composer Studio IDE**

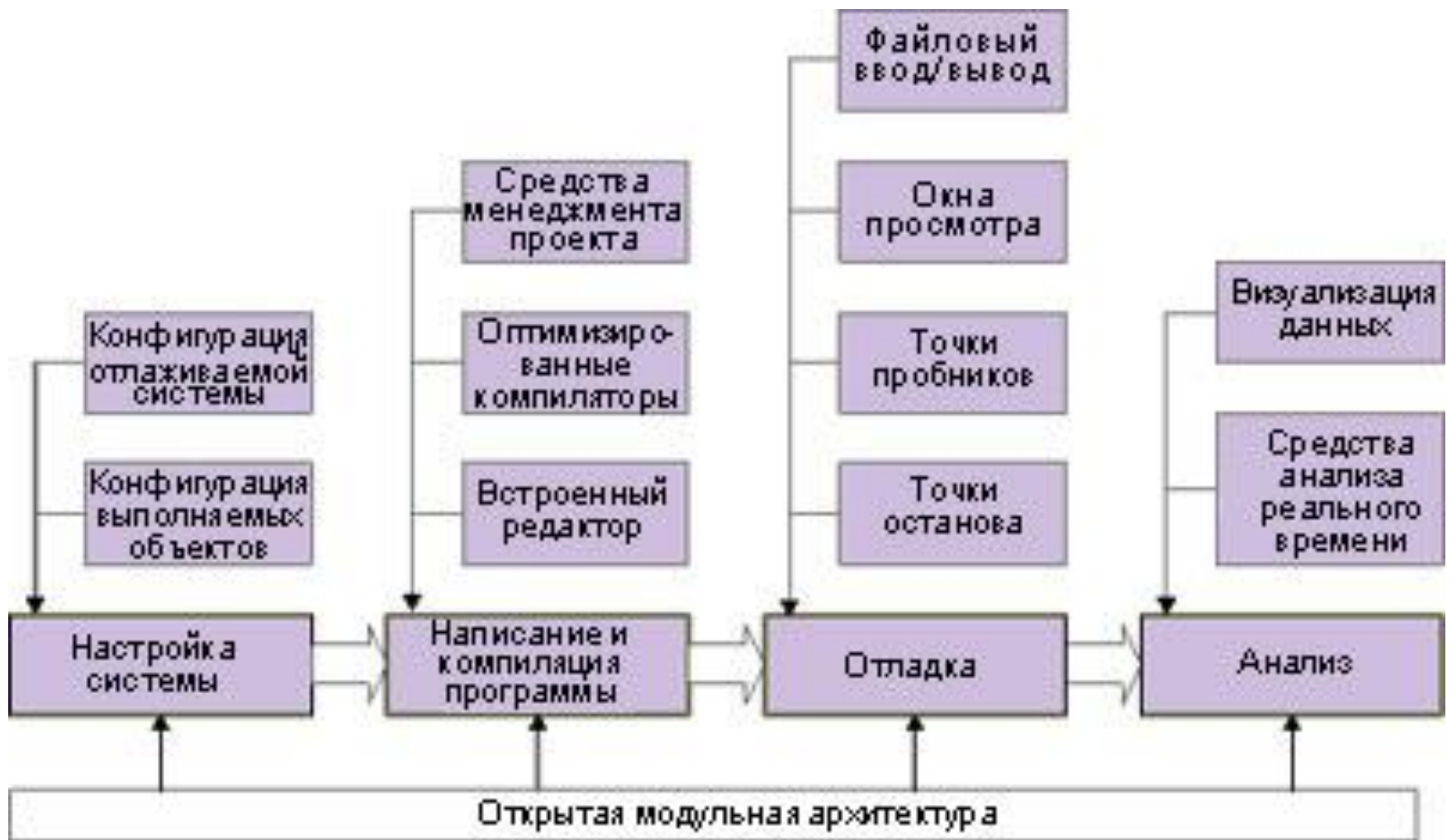
Состав, общая характеристика

Версии Code Composer Studio

- Code Composer Studio v2.x
 1. C6000 Code Composer - ядра C62xx и C64xx
 2. C5000 Code Composer - ядра C54xx и C55xx
 3. C2000 Code Composer - семейства C2xx, C24xx
 4. C3x/C4x Code Composer - семейства C4x, C3x, VC33
- Code Composer Studio v3.x
- Code Composer Studio v4.x
- Code Composer Studio v5.x

Figure 1-1. eXpress DSP™ Software and Development Tools





Цикл разработки с использованием CCS

Отладка программного обеспечения

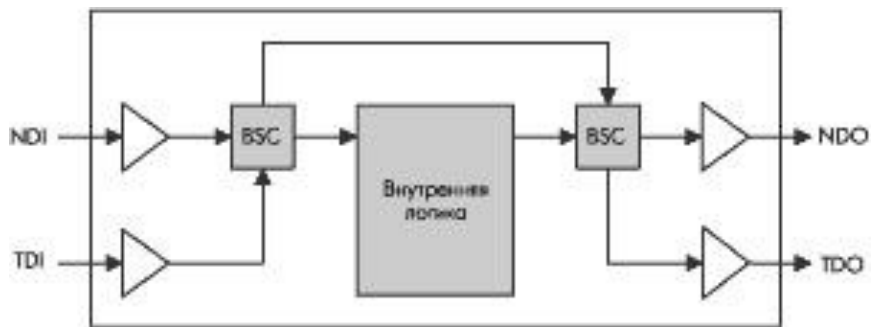
Виды ошибок:

1. синтаксические ошибки
2. семантические ошибки

Отладочные средства:

1. **Симулятор** – программа моделирующая аппаратные средства
2. **Эмулятор** – программно-аппаратный имитатор целевого устройства
3. **Монитор отладчика** – аппаратное средство, резидентно располагающееся внутри целевого устройства
4. **Внутрикристальные аппаратные средства отладки (JTAG)** – обеспечение доступа к внутренним ресурсам системы

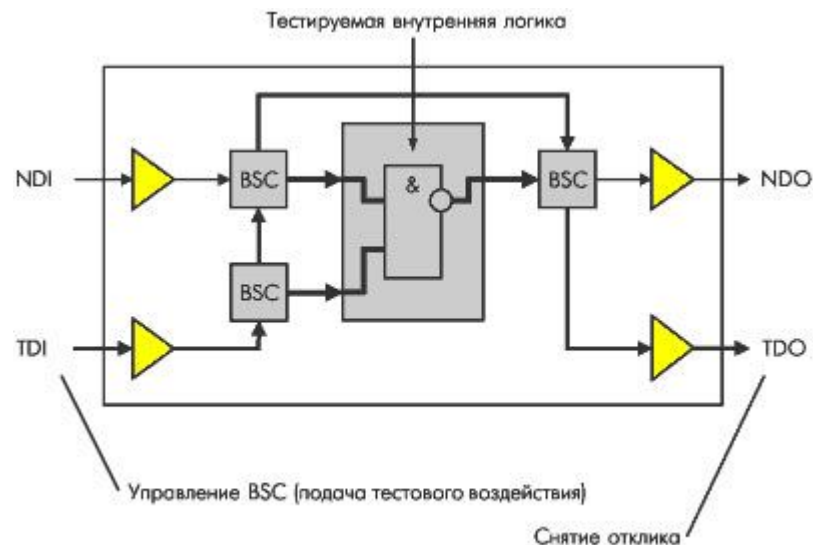
Тестовый интерфейс JTAG



Интеграция архитектуры BSC в устройство



Использование JTAG для тестирования межсоединений



Использование JTAG для тестирования внутренней логики

Тестовый интерфейс JTAG

Выводы интерфейса:

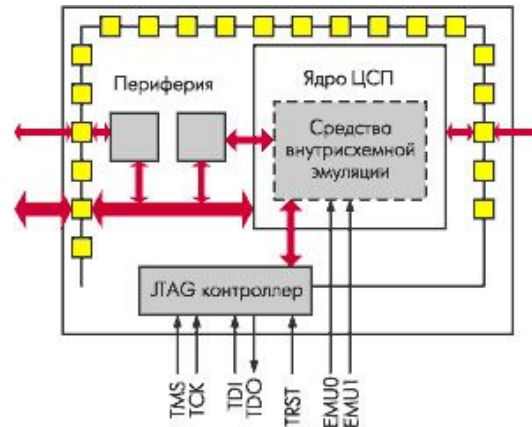
1. TMS - вход управления
2. TCK - тактовый вход
3. TDI - последовательный вход данных
4. TDO - последовательный выход данных.

Состав специального контроллера:

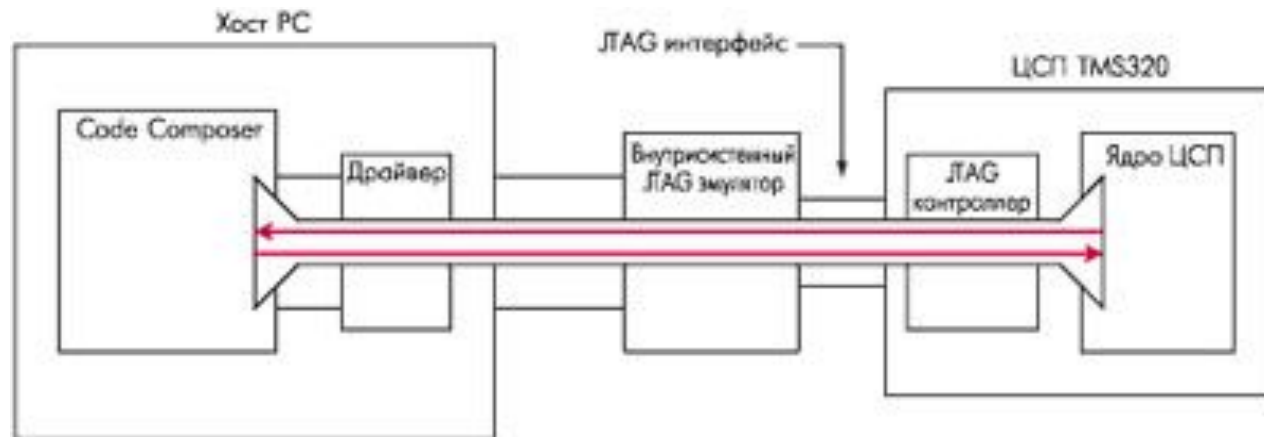
1. TAP (Test Access Port) - блока управления, называемого портом тестового доступа
2. Набор сдвиговых регистров:
 - регистр команд - определяют состояние контроллера и устройства, а также интерпретацию блоками BSC поступающих с TDI данных
 - тестовый регистр - последовательно включенные блоки BSC
 - регистра пропуска (BYPASS) - однобитовый регистр, позволяющий уменьшить длину JTAG-пути и организация прозрачной ретрансляции данных через JTAG-контроллер
 - пользовательские регистры - специфичны для каждого устройства и выполняют дополнительные функции, например, доступ к тестовым блокам процессора или управление конфигурацией ПЛИС.

Все устройства последовательно соединяются входами и выходами данных, при этом все сдвиговые регистры соединяются в цепочку, образуя JTAG-путь

Интеграция средств JTAG в ЦСП Texas Instruments



Обеспечение средствами JTAG канала между ЦСП и отладчиком



Основное назначение внутрисхемного JTAG-эмулятора - обеспечить интерфейс между JTAG-портом ЦСП и компьютером. Совместно с JTAG-контроллером, внутрисхемный эмулятор обеспечивает канал связи между расположенными в ЦСП отладочными узлами и запущенной на PC отладочной средой

Достоинства JTAG-интерфейса

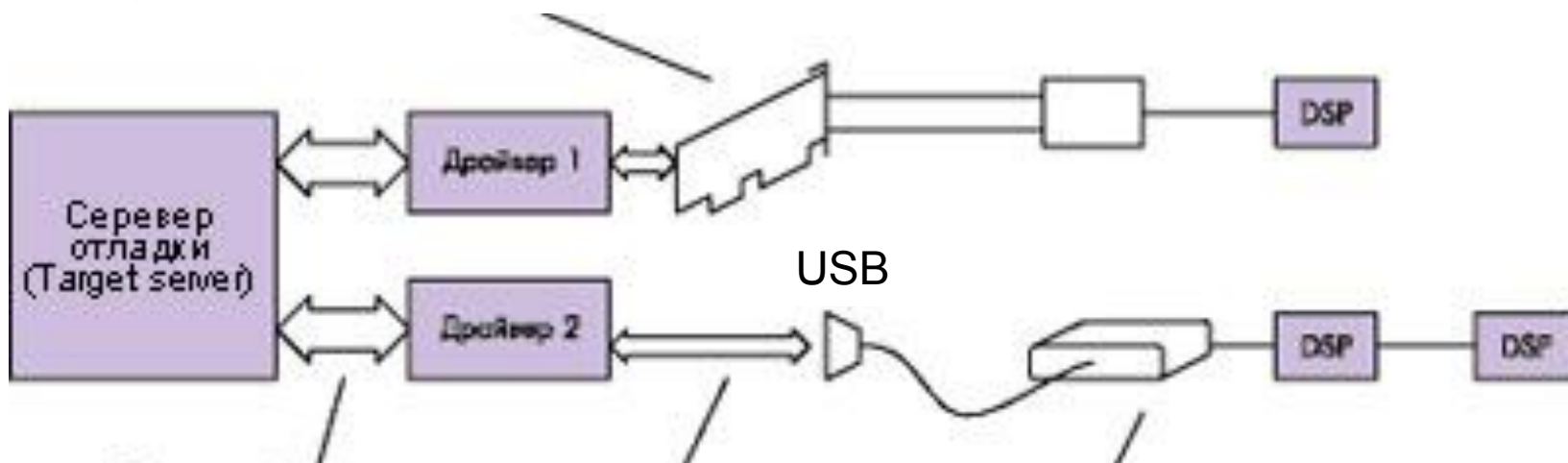
1. Прямой доступ к скрытым ресурсам процессора;
2. Не занимает аппаратных ресурсов системы (портов, памяти) для отладки;
3. Доступ не нарушается при аварийной ситуации в отлаживаемой системе;
4. Не потребляет энергию из отлаживаемой системы;
5. Подключение отладчика через JTAG интерфейс не возмущает систему;
6. Обеспечивает связь с «сырой» системой, в которую еще не загружено никакое ПО;
7. Позволяет соединять последовательно в цепочку несколько устройств (в частности, несколько процессоров в многопроцессорной системе) и производить их совместную отладку.

Интегрированная среда разработки и отладки программного обеспечения **Code Composer Studio IDE**

Настройка (Target and Host Setup)



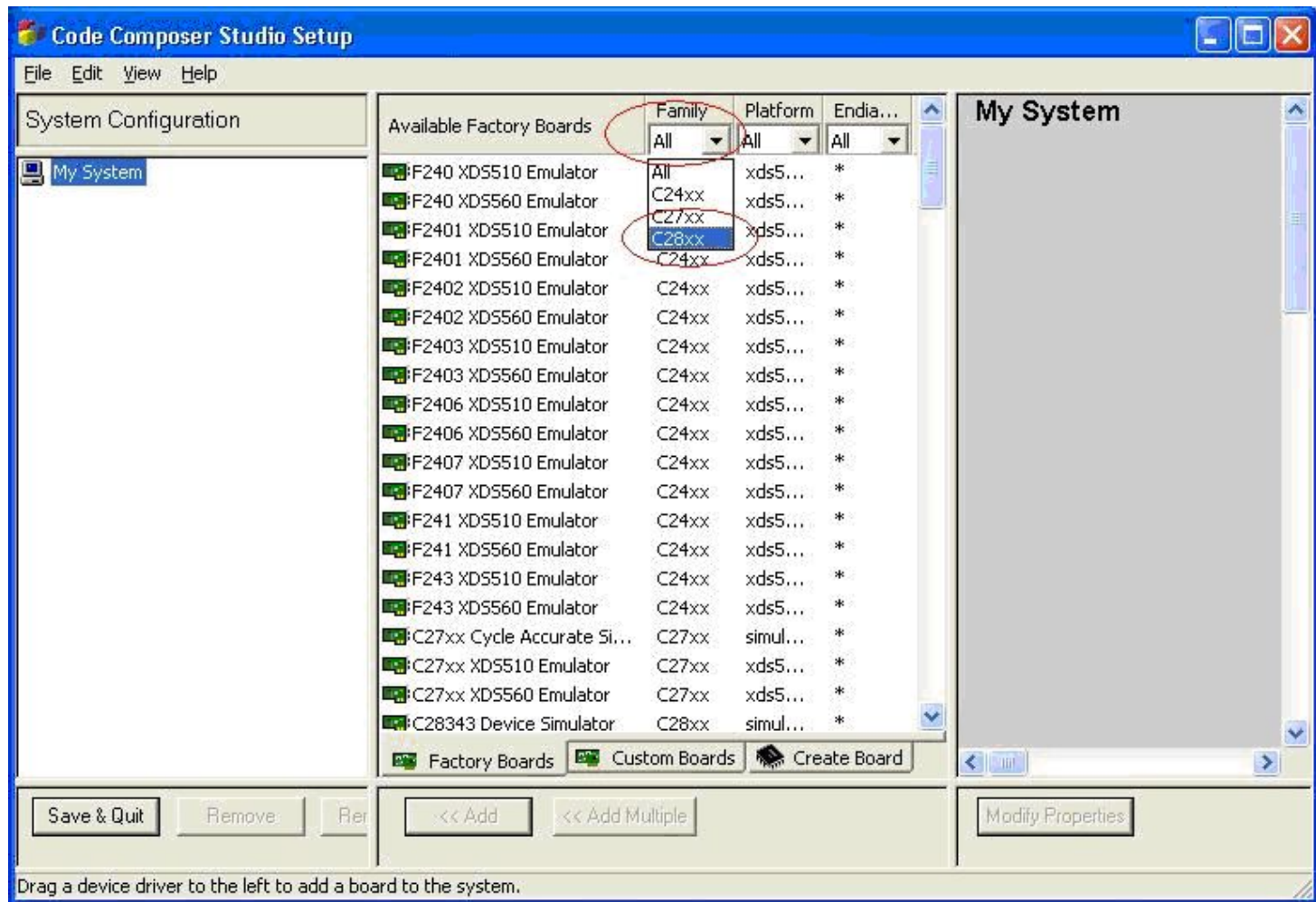
Внутрисхемный эмулятор
для шины PCI



Стандартный
интерфейс между
драйвером
эмулятора и CCS

Интерфейс между
драйвером и
внутрисхемным
эмулятором

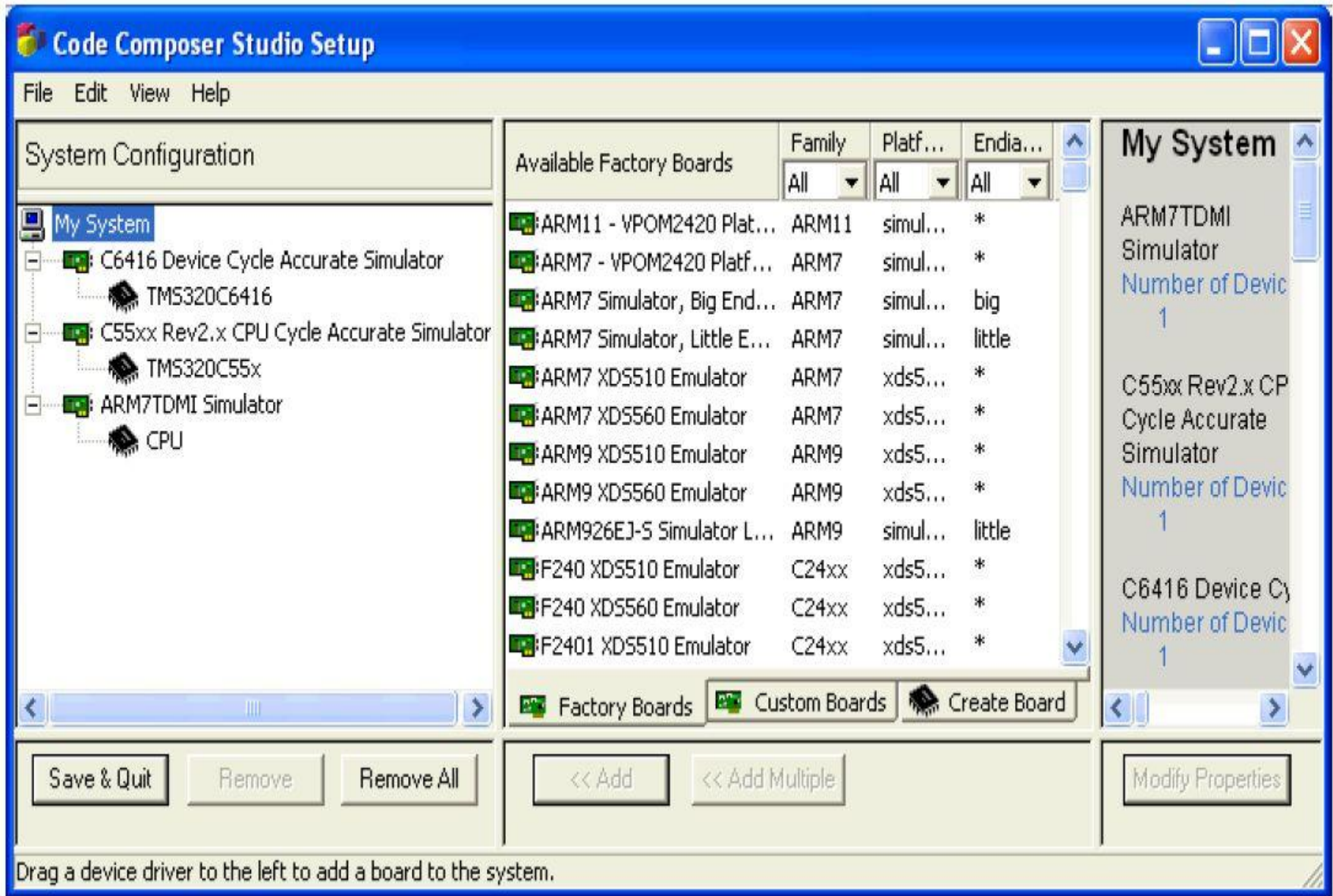
Внутрисхемный
эмулятор



Available Factory Boards	Family	Platform	Endia
	All	All	All
:F240 XDS510 Emulator	C24xx	All	*
:F240 XDS560 Emulator	C24xx	simulator	*
:F2401 XDS510 Emulator	C24xx	xds100usb emulator	*
:F2401 XDS560 Emulator	C24xx	xds510 emulator	*
:F2402 XDS510 Emulator	C24xx	xds560 emulator	*
		xds510 emulator	*

:F28034 XDS100 Emulator	C28xx	xds100...	*
:F28035 XDS100 Emulator	C28xx	xds100...	*
:F28044 XDS100 Emulator	C28xx	xds100...	*
:F2806 XDS100 Emulator	C28xx	xds100	*

Figure 3-1. Standard Setup Configurations





Встроенный язык скриптов GEL (General Extention Language)

1. создание элементов графического пользовательского интерфейса (GUI) для управления отлаживаемой ЦОС-программой;
2. диагностика аппаратной части;
3. начальная установка и конфигурирование;
4. автоматизация часто выполняемых последовательностей команд;
5. добавление пунктов в меню;
6. функции работы с отлаживаемым ЦСП: изменение содержимого памяти и регистров, загрузка программы, добавление и удаление точек останова, сброс ЦСП;
7. возможности работы с создаваемыми в рамках интерфейса CCS окнами ввода/вывода.

Встроенный язык скриптов GEL

Пример

```
OnPreFileLoaded()
{
    XINTF_Enable();
    if (TxtOutCtl==0)
    {
        GEL_TextOut("\nNOTES:\nGel will
enable XINTFx16 during Debug
only.\nEnable XINTF in code prior to use.");
        GEL_TextOut("\nFPU Registers can be
found via GEL->Watch FPU Registers.");
        TxtOutCtl=1;
    }
}
```

Встроенный язык скриптов GEL

Пример

```
hotmenu XINTF_Enable()
{
    /* enable XINTF clock (XTIMCLK) */
    *0x7020 = 0x3700;

    /* GPBMUX1: XA0-XA7, XA16, XZCS0,    */
    /* XZCS7, XREADY, XRNW, XWE0      */
    /* GPAMUX2: XA17-XA19, XZCS6      */
    /* GPCMUX2: XA8-XA15              */
    /* GPCMUX1: XD0-XD15              */
    *(unsigned long *)0x6F96 = 0xFFFFFFFF; /* GPBMUX1 */
    *(unsigned long *)0x6f88 = 0xFF000000; /* GPAMUX2 */
    *(unsigned long *)0x6FA8 = 0x0000AAAA; /* GPCMUX2 */
    *(unsigned long *)0x6FA6 = 0xAAAAAAAA; /* GPCMUX1 */

    /* Uncomment for x32 data bus      */
    /* GPBMUX2: XD16-XD31              */
    // *(unsigned long *)0x6F98 = 0xFFFFFFFF; /* GPBMUX2 */
}
```

Встроенный язык скриптов GEL

Пример

```
/* Zone timing.  
/* Each zone can be configured seperately */  
/* Uncomment the x16 or the x32 timing */  
/* depending on the data bus width for */  
/* the zone */  
  
/* x16 Timing */  
*(unsigned long *)0x0B20 = 0x0043FFFF; /* Zone0 */  
*(unsigned long *)0x0B2C = 0x0043FFFF; /* Zone6 */  
*(unsigned long *)0x0B2E = 0x0043FFFF; /* Zone7 */  
  
/* x32 Timing:  
// *(unsigned long *)0x0B20 = 0x0041FFFF; /* x32 */  
// *(unsigned long *)0x0B2C = 0x0041FFFF; /* x32 */  
// *(unsigned long *)0x0B2E = 0x0041FFFF; /* x32 */  
}
```


**Интегрированная среда разработки
и отладки программного
обеспечения
Code Composer Studio IDE**

Интерфейс пользователя

Code Composer Studio interface showing a C program for signal processing on a TMS320C6416 simulator. The main window displays the source code for `sinewave.c`.

```

Write output data to a graph
dataIO();

/* Apply the gain to the input
processing();
}
}

/* FUNCTION:      Apply signal proces
to generate output
/* PARAMETERS:   BufferContents stru
/* RETURN VALUE: none.

static void processing()
{
    int counter =0;
    int int counter = 0 IIZE;

    while(size--){
        currentBuffer.output[size] = cu
        counter++;
    }
}

```

The right side of the IDE features a graph showing a periodic waveform with a peak amplitude of approximately 1.1×10^9 and a period of about 50 samples. Below the graph is a watch window:

Name	Value	T...	R...
counter	0	int	dec
size	47	int	dec

At the bottom, the console shows the message "SineWave example started." and the register window displays the state of various registers, including the EDMA Registers and Core Registers.

STATUS: HALTED: s/w breakpoint | LE | Ln 33, Col 14

Icons on the Code Composer Studio Toolbar



Launches Code Composer Studio



Rebuilds the project



Builds the project incrementally



Halts execution



Toggles breakpoint



Runs project



Single steps project

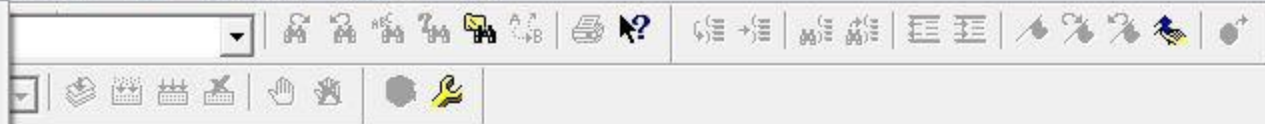


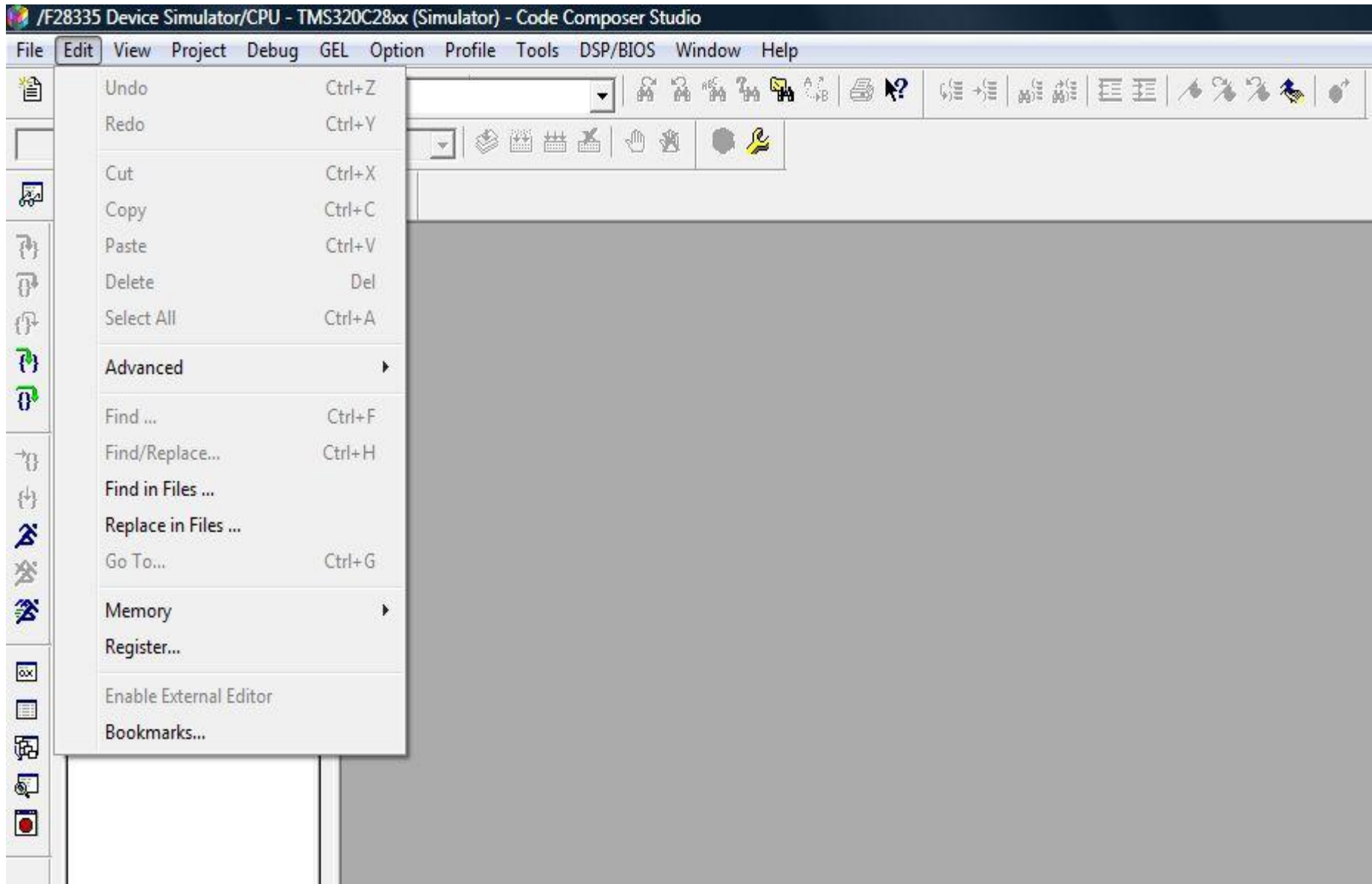
Step out

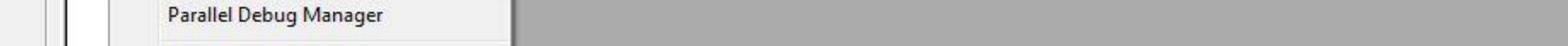
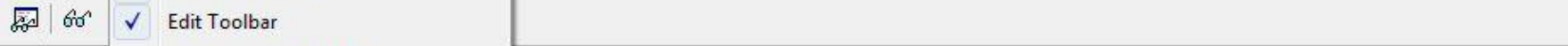


Step over

- New ▶
- Open... Ctrl+O
- Close
- Save Ctrl+S
- Save As...
- Save All
- Load Program... Ctrl+L
- Reload Program Ctrl+Shift+L
- Load Symbols ▶
- Reload Symbols ▶
- Unload Symbols ▶
- Load GEL...
- Data ▶
- Workspace ▶
- Difference between files...
- Merge Files...
- Print... Ctrl+P
- Recent Source Files ▶
- Recent Workspaces ▶
- Recent Program Files ▶
- Recent Symbols ▶
- Recent GEL Files ▶
- Launch Setup
- Exit





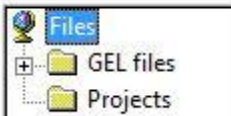
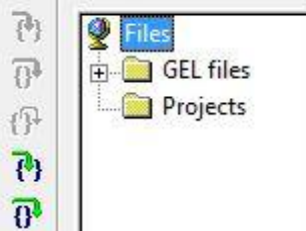
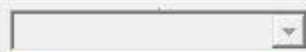


- Standard Toolbar
- GEL Toolbar
- Project Toolbar
- Edit Toolbar
- Advanced Edit Toolbar
- Layout Toolbar
- Status Bar
- Debug Toolbars ▶
- Plug-in Toolbars ▶
- Property Window
- Memory
- Disassembly
- Registers ▶
- Graph ▶
- Threads ▶
- Watch Window
- Quick Watch
- Call Stack
- Expression List
- Output Window
- Symbol Browser
- Project
- Parallel Debug Manager
- Mixed Source/ASM
- Real-time Refresh Options...
- Layout ▶

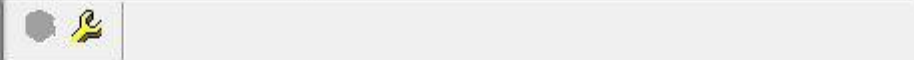


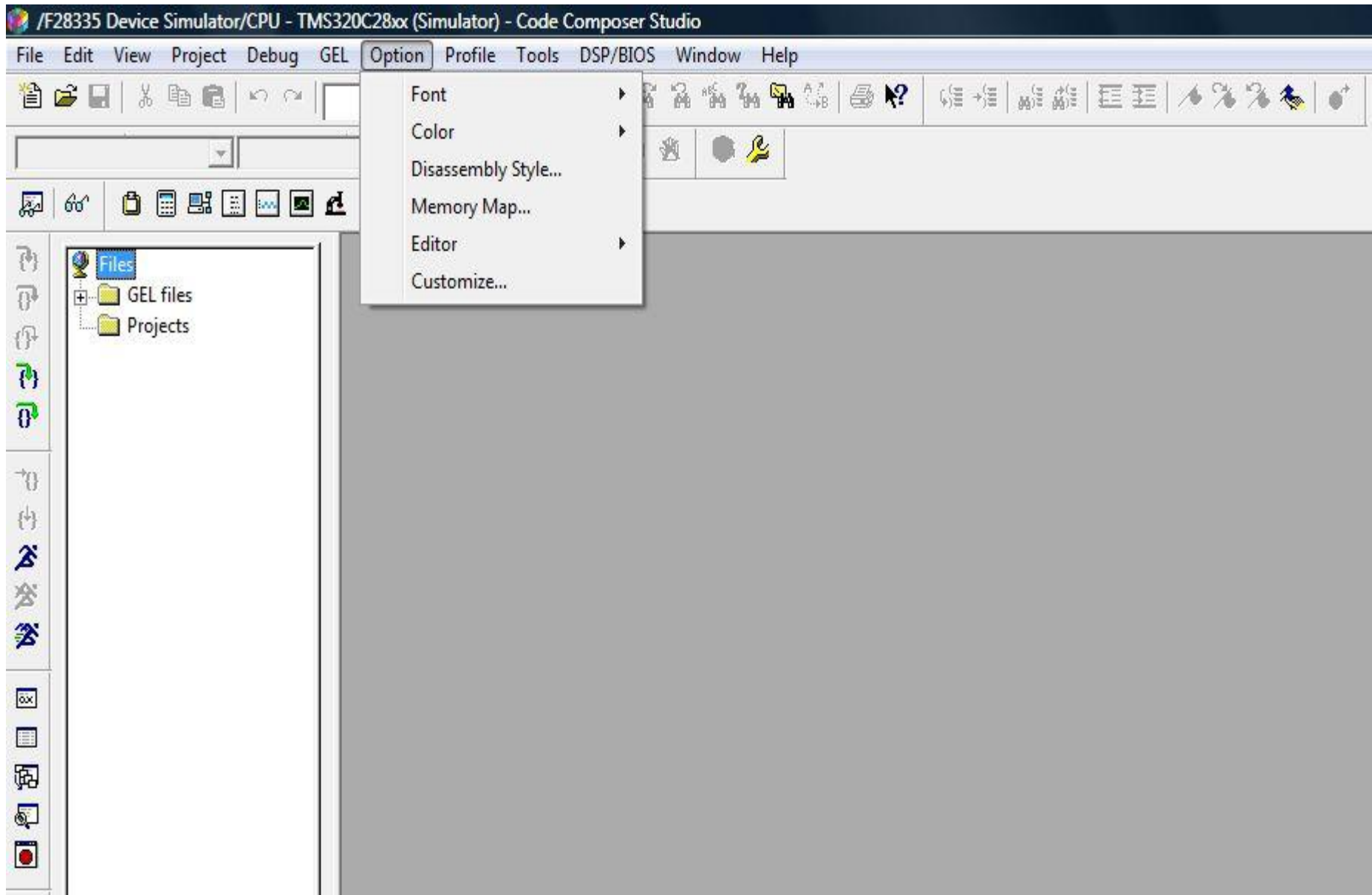
- New...
- Open...
- Use External Makefile...
- Export to Makefile...
- Add Files to Project...
- Save
- Close
- Source Control ▶
- Compile File Ctrl F7
- Build F7
- Rebuild All
- Stop Build
- Build Clean
- Configurations...
- Build Options...
- File Specific Options...
- Project Dependencies...
- Show Project Dependencies
- Show File Dependencies
- Scan All File Dependencies
- Recent Project Files ▶

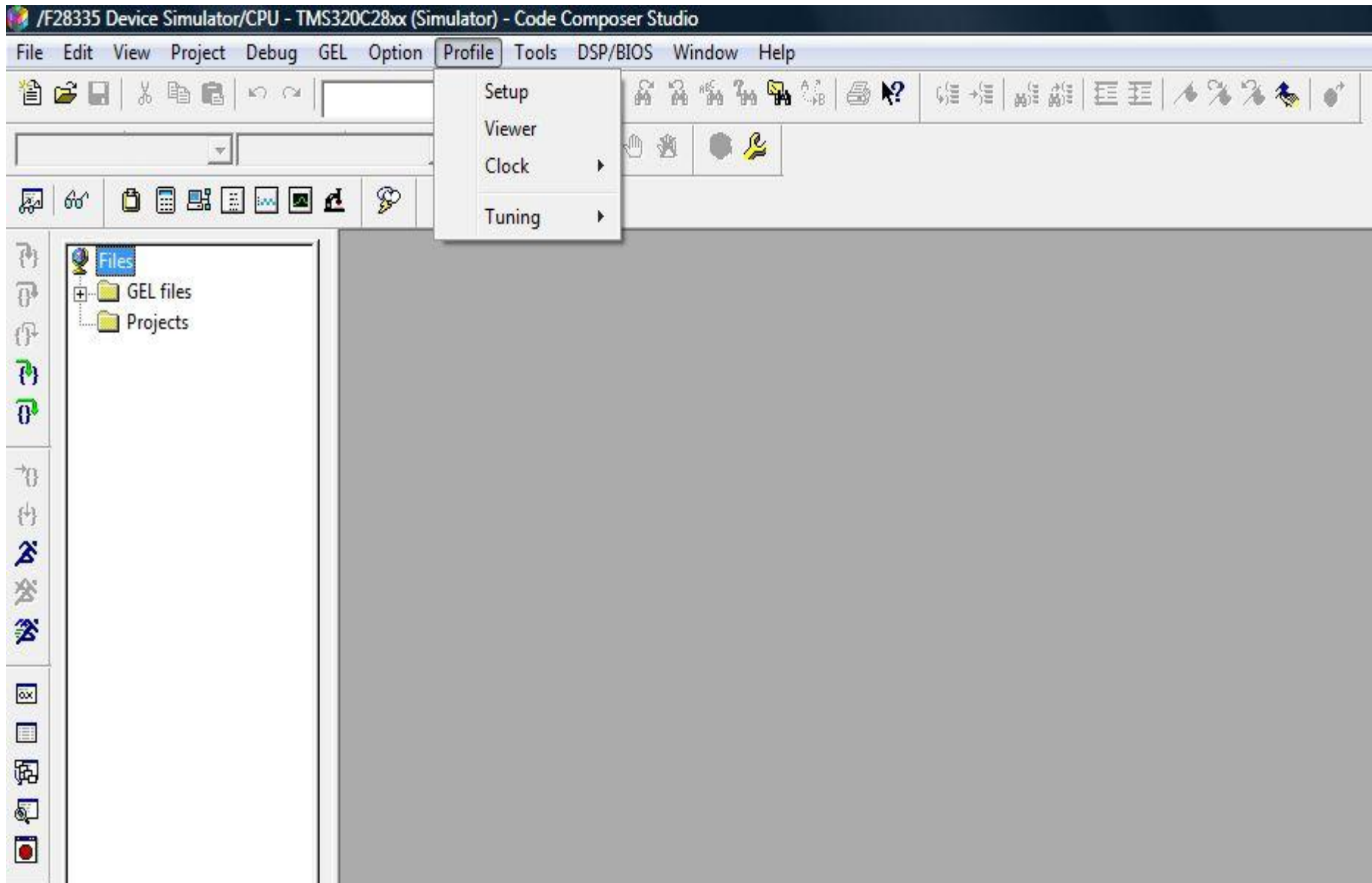


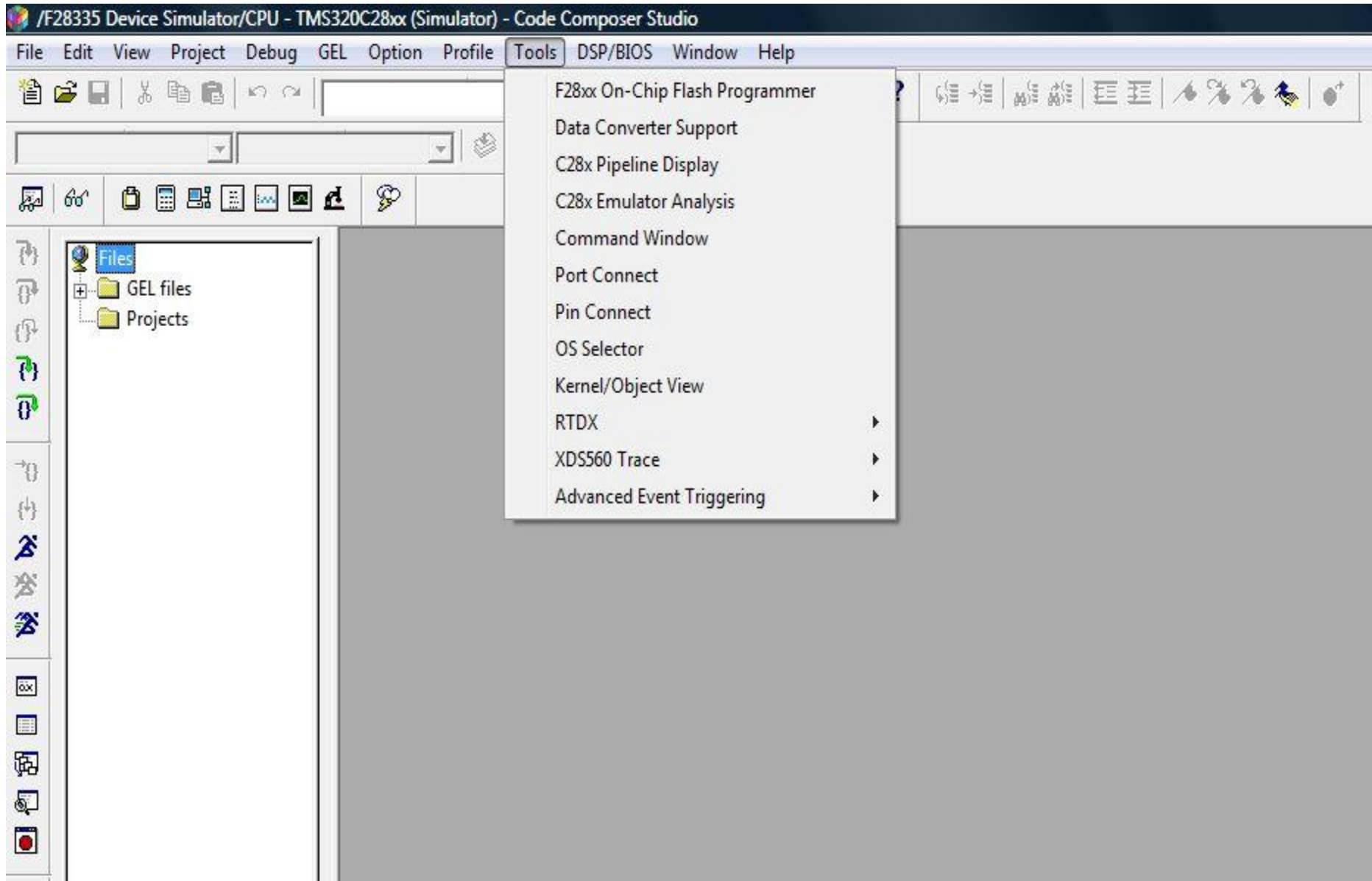


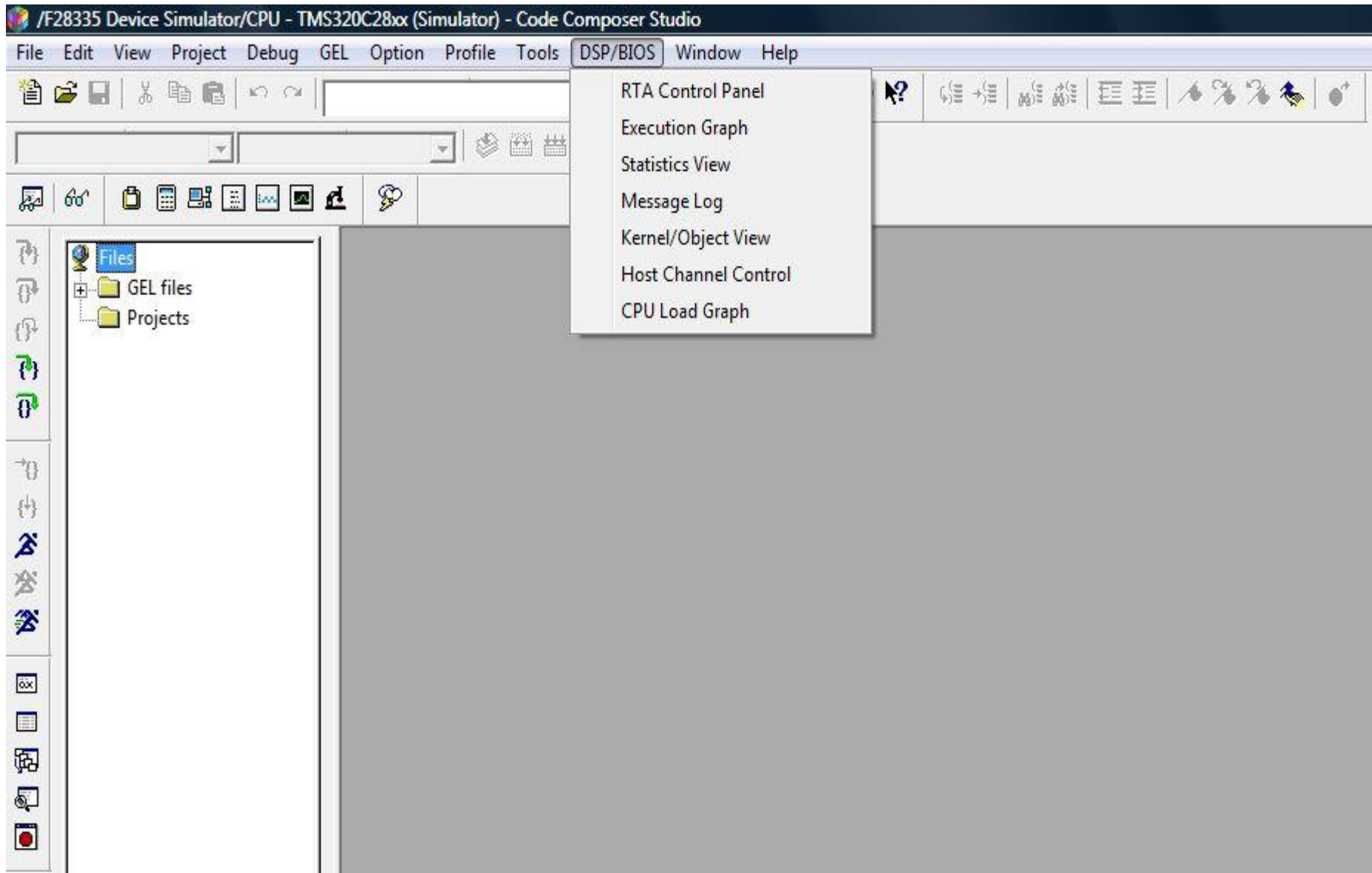
- Breakpoints...
- Assembly/Source Stepping
- Step Into F11
- Step Over F10
- Step Out Shift+F11
- Run F5
- Halt Shift+F5
- Animate Alt+F5
- Run Free Ctrl+F5
- Low Power Run Ctrl+Shift+F5
- Run to Cursor Ctrl+F10
- Set PC to Cursor Ctrl+Shift+F10
- Restart Ctrl+Shift+F5
- Go Main Ctrl+M
- Multiple Operation...
- Advanced Resets
- Reset CPU Ctrl+R
- Reset Emulator Ctrl+Shift+R
- Halt on Reset
- Disconnect Alt+C
- Restore Debug State
- Thread Level Debugging
- Real-time Mode
- Rude Real-time Mode
- Flush Pipeline on Halt

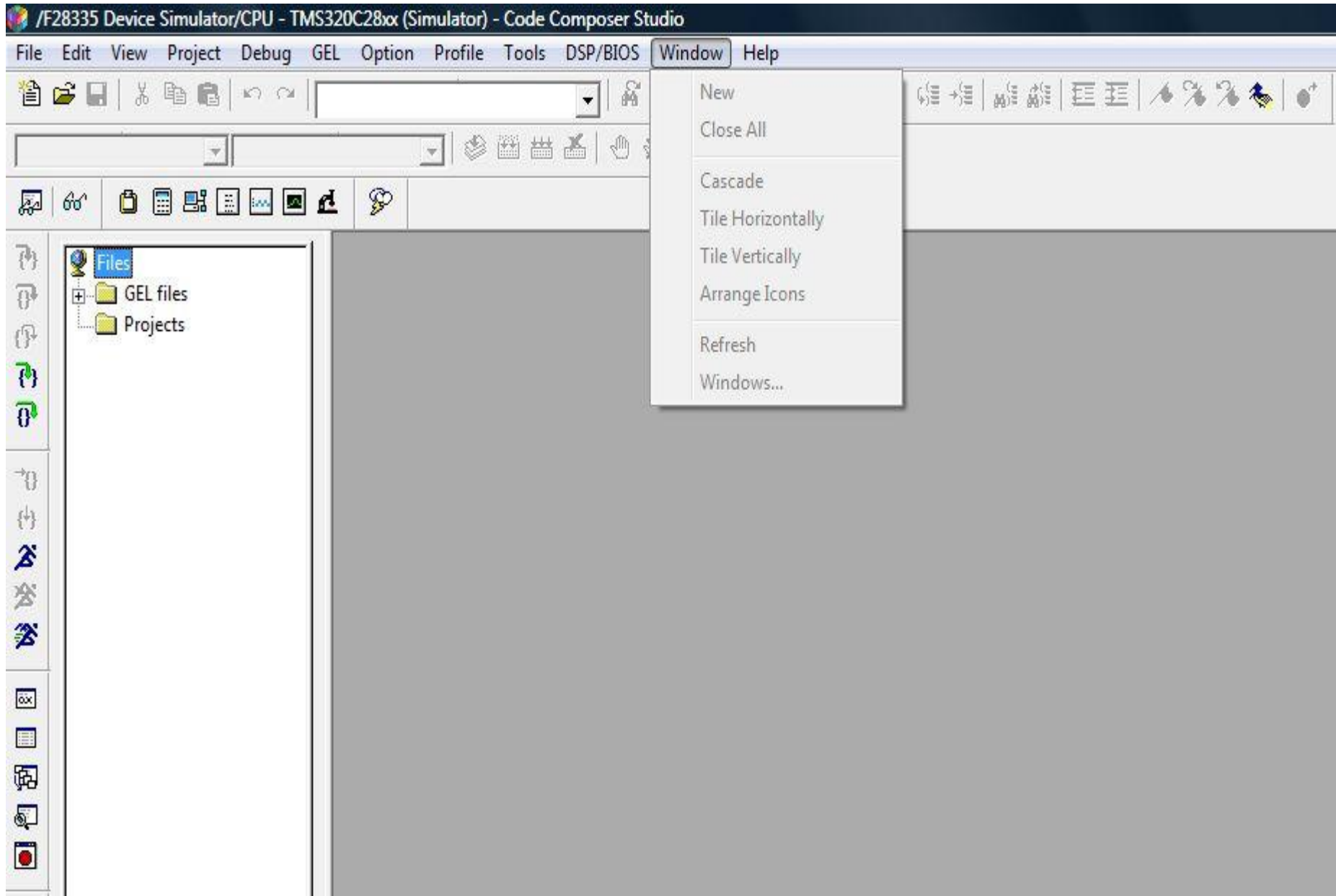


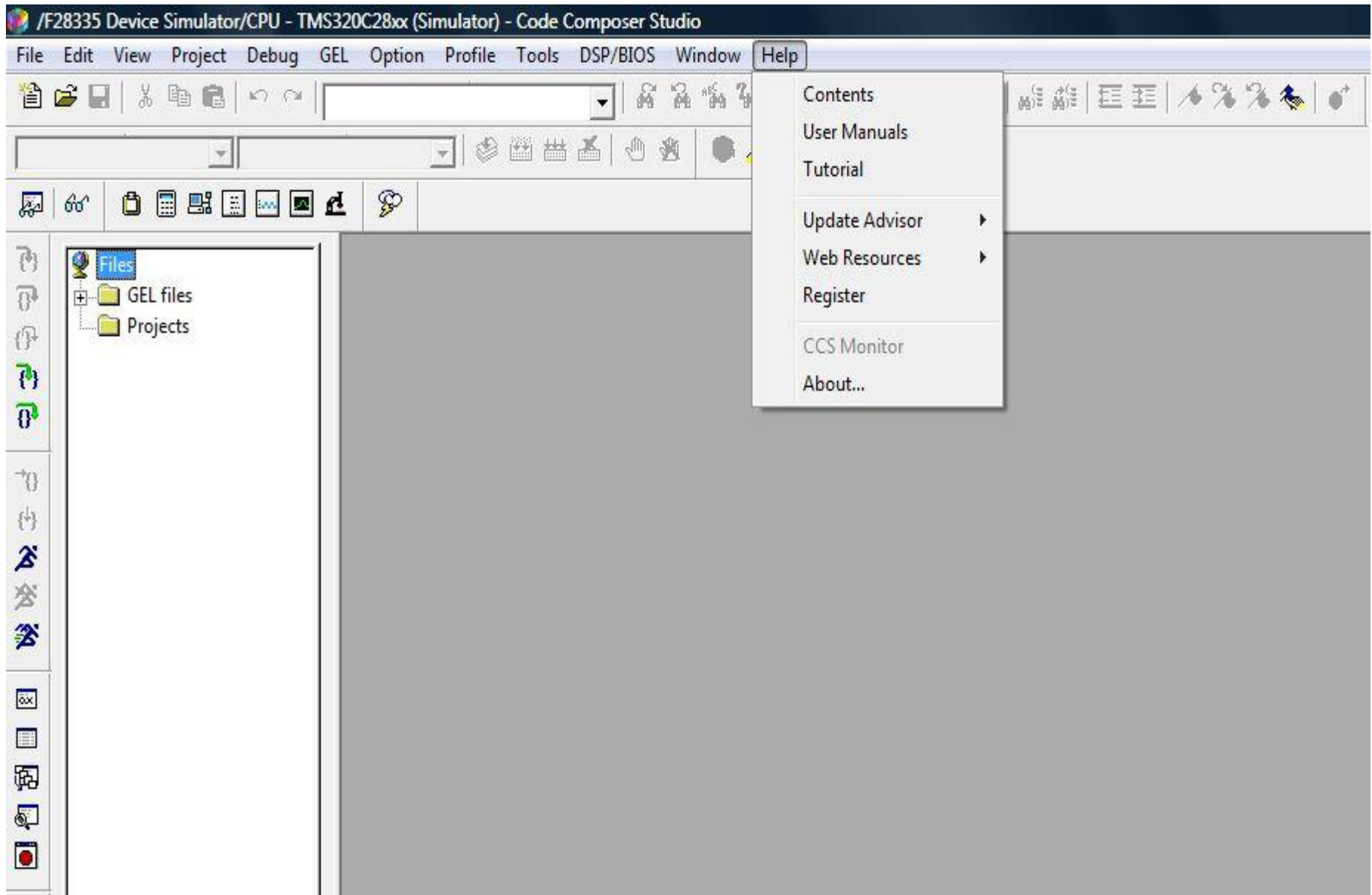












Toolbar with icons for file operations (New, Open, Save, Print, Copy, Paste, Undo, Redo), development (Build, Run, Stop, Refresh), and navigation (Home, Back, Forward).

File Explorer showing a tree view with 'Files', 'GEL files', and 'Projects' folders. A 'Register Window' tooltip is visible over the bottom part of the pane.

Main workspace area, currently empty and greyed out.

CPU Registers

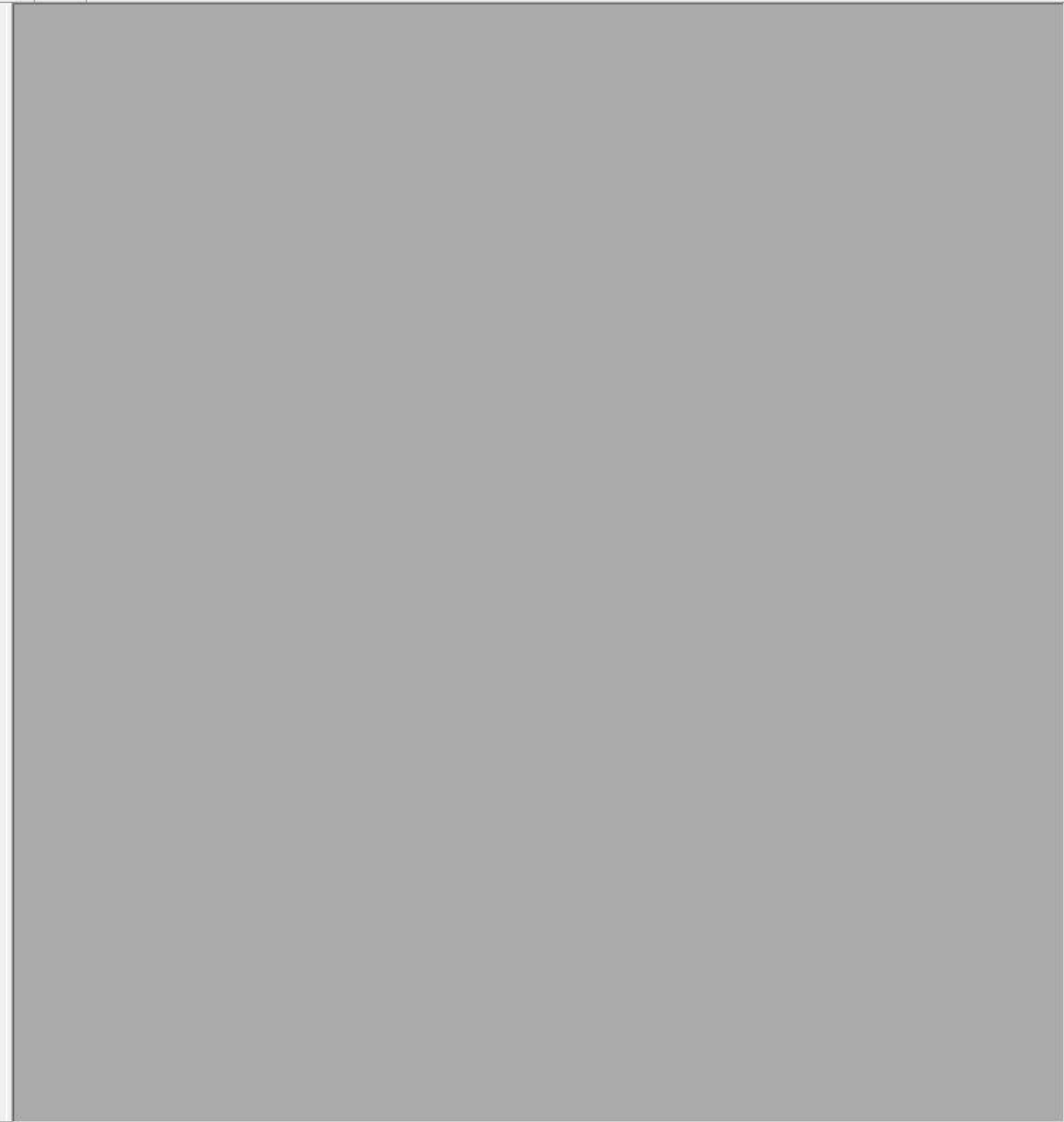
ACC	00000000	XAR4	00000000	ST0	0000
P	00000000	XAR5	00000000	ST1	0A0B
XT	00000000	XAR6	00000000	DP	0000
XAR0	00000000	XAR7	00000000	SP	0400
XAR1	00000000	PC	000000	IER	0000
XAR2	00000000	IC	000000	IFR	0000



Files

- GEL files
- Projects

View Memory



Enter An Address

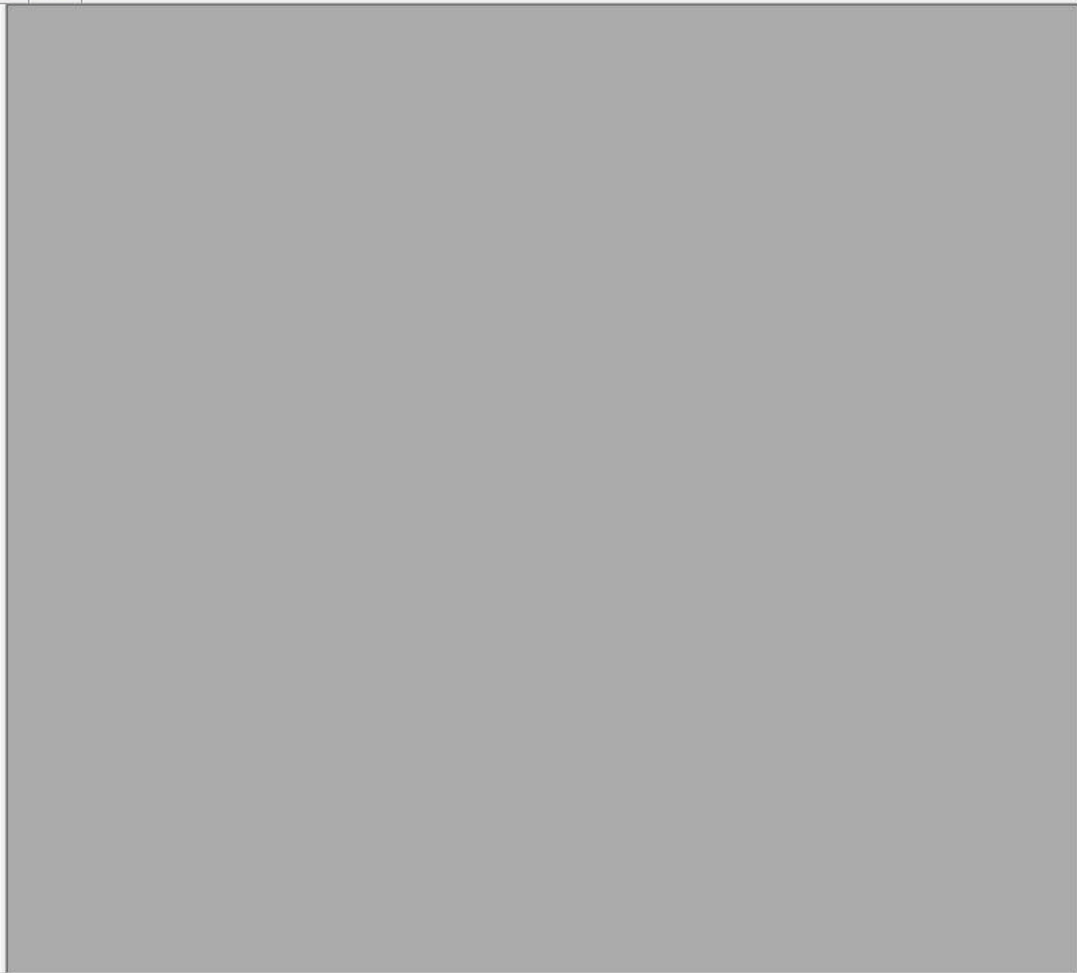
Hex 16 Bit - 1 Data



Files

- GEL files
- Projects

View Stack

The file explorer sidebar shows a tree view with 'Files' at the root, containing 'GEL files' and 'Projects'. Below the tree is a 'View Stack' button and a red stop icon.

Enter An Address

Hex 16 Bit - 1 Data

The memory view sidebar features an 'Enter An Address' input field at the top. Below it are several vertical scroll bars. At the bottom, there are two dropdown menus: 'Hex 16 Bit - 1' and 'Data'.

C-Source Is Not Available



Files

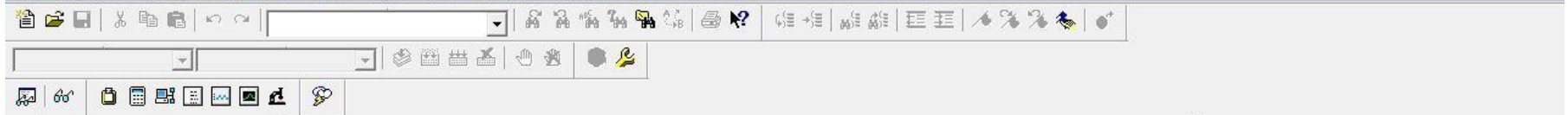
- GEL files
- Projects

View disassembly

000000	0000	ITRAPH
000001	0000	ITRAPH
000002	0000	ITRAPH
000003	0000	ITRAPH
000004	0000	ITRAPH
000005	0000	ITRAPH
000006	0000	ITRAPH
000007	0000	ITRAPH
000008	0000	ITRAPH
000009	0000	ITRAPH

Enter An Address

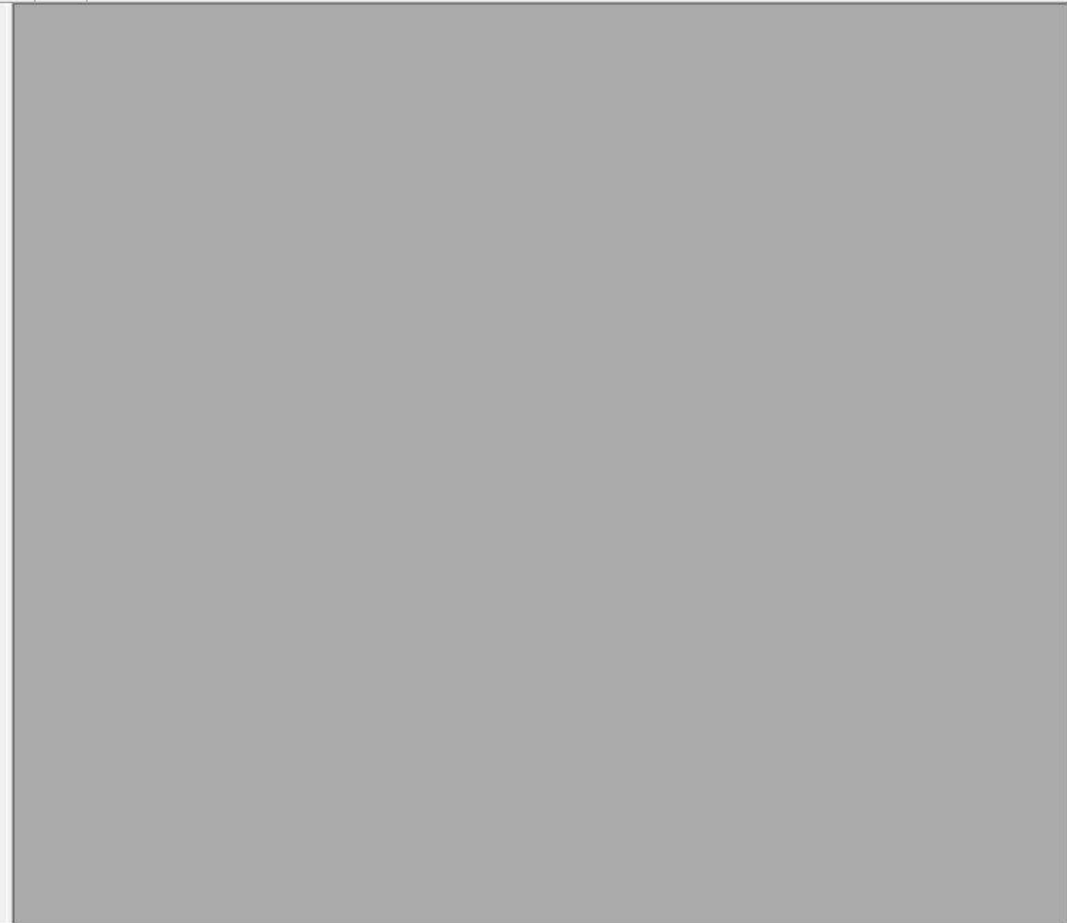
Hex 16 Bit - 1 Data



Files

- GEL files
- Projects

Breakpoint Manager



Enter An Address

Hex 16 Bit -1 Data

Location	Condition	Count	Action	Group	Name