

CCS 菜单命令精解

(Version 2.21)

1. 菜单File

1) Workspace Load Workspce : 装载工作空间到CCS ;

2) Workspace Save Workspace : 保存当前工作空间到CCS ;

工作空间 (Workspace) 用来指CCS 的工作环境, 包括程序的断点、打开的文件和各种调试窗口等等。工作空间的主要用途是方便调试, 在调试时保存工作空间, 以后重新载入时, 可直接恢复到先前调试的地方。CCS 正常关闭时, 会自动保存当前工作空间到default.wks 文件 (..\cc\bin)。

2. 菜单View

1) Disassembly : 创建反汇编窗口 ;

2) Memory : 查看memory, 在弹出的对话框Page 选项中, 可选择三种不同的RAM 空间 (Program、Data、I/O), 查错时, 这三种空间都可能会用到 ;

3) Registers : 包括CPU 内核寄存器、状态位、寄存器的低16 位 ;

4) Graph Time/Frequency : 图形化显示数据变量, 在弹出的对话框中经常需要设置的地方有: Graph Title (写入图形窗口的名称标题)、Start Address (变量起始地址, 如果是基本变量, 写入“&变量名”; 如果是数组变量, 则写入“数组名”即可)、Acquisition Buffer Size (获取DSP 内若干个连续数据, 如果显示基本变量, 填入1; 如果是数组变量, 比如显示大小为100 的数组变量int a[100], 填入100)、Display Data Size (图形窗口中显示数据的个数)、DSP Data Type (变量的类型, 可选择16 位还是32 位, 可选择有符号还是无符号等);

5) Watch Window : 变量观测窗口, 填加变量后可以观看变量值 ;

6) Mixed Source/ASM : C 语言和汇编语言混合显示的切换指令。

3. 菜单Project

1) New : 新建工程 ;

2) Open : 打开工程 ;

3) Add Files To Project : 填加文件到工程 : 一个工程由若干个文件构成 ;

4) Build : 编译并连接整个工程——只编译修改过的文件, 但对所有文件进行连接 ;

5) Rebuild All : 编译连接整个工程——编译所有工程中的文件, 并且连接所有的文件, 但文件比较多时, 用此命令比较费时 ;

6) Build Options : 整个工程编译时的选项, 这大概是最复杂的设置选项, 包含许多优化选项 ; 下面介绍几个常用的设置 :

标签页Compiler

Category Preprocessor Include Search Path[I] : 设置C 语言中“#include” 的搜索路径
标签页Linker

Category Basic Autoinit Model : 有三种模式可供选择, 介绍如下 :

Load-time Initialization[-cr] : 调用系统初始化, 不需要从ROM 中执行初始化, 代码下载时就直接对.bss 段进行了初始化。使用C 语言编写程序在调试的时候, 选择此项可以加快调试速度 ;

Run-time Autoinitialization[c] : 调用系统初始化, 就像代码烧写到ROM 中运行一样, 位于RAM 中的变量由ROM 中的执行程序来完成。工程由C 语言编写完成, 烧写到ROM 中时, 一定选择此项 (此项为新建工程的默认选项)。

No Autoinitialization : 不需要进行系统初始化, 即不调用系统库rts2800.lib 中的初始化函数 (入口地址_c_int0)。工程由纯粹的汇编语言编写完成时, 常选择此项。

注 : 上面的设置是对整个工程文件进行设置, 如果只对其中的一个文件进行设置, 可选中文件后单击右键, 选择File Specific Options 进行设置。

4. 菜单Debug

- 1) Breakpoints (F9) : 设置硬件断点 (TMS320F2812 支持2 个硬件断点) 和软件断点;
- 2) Step Into (F8) : 单步调试, 遇见函数调用时, 进入;
- 3) Step Over (F10) : 单步调试, 遇见函数调用时, 不进入函数体内, 仍作单步;
- 4) Step Out (shift+F7) : 当执行Step Into 进入到函数体内时, Step Out 令其执行完此函数;
- 5) Run (F5) : 运行, 代码开始运行, 直到遇到断点时停止DSP 内核运行, 刷新窗口;
- 6) Halt (shift+F5) : 停止运行;
- 7) Animate (F12) : 动画模式运行。与命令Run 的区别是, 遇到断点时先停止DSP 内核, 刷新窗口, 然后在接着继续启动运行, 它常用来连续地刷新变量窗口和生成graph 图形等。切记这不是实时仿真, 因为这里是停止DSP 内核后才刷新窗口的, 其功能相当于遇到断点后连续不停地执行Run 命令;
- 8) Run Free : 自由运行, 遇到断点不停止;
- 9) Run To Cursor : 运行到光标处停止;
- 10) Rest DSP : 复位DSP 中的所有寄存器状态。复位DSP 后, 程序执行地址指向0 (即 PC=0) ;
- 11) Restart : 程序跳至_c_int0 标号, 这是C 语言的入口的地址。注意程序是跳至, 而不是运行至_c_int0 标号;

5. 菜单Profilier

用来测算代码执行时间,剖析的过程实际上是在特定的剖析点插入软件断点,碰到断点后,剖析器读取剖析时钟的值,从而计算两个剖析点之间的时钟之差,即此段代码的执行的周期数。

- 1) Clock Setup : 设置n

Instuction Cycle : 设置一条指令执行的时间,在DSP 中指执行单周期指令的时间,即运行的主时钟,单位为ns。比如:主频为100MHz,则添入10;

- 2) View Clock : 查看代码执行的时间。

6. 菜单Option

- 1) Customize Progma Load Options

Perform Verification During Program Loading

当下载程序到DSP 的RAM 中后,CCS 会执行一个校验程序来检查下载后的程序正确与否。此选项为系统默认,一般不做更改。

Load Program After Building

选中此项,编译工程后,生成的.out 文件会自动下载到RAM 中。

7. 菜单GEL

GEL 是通用扩展语言 (General Extension Language) 的缩写, GEL 是一个大小写敏感但缺少类型检测的解释性语言,在语法上可看作是C 语言的一个子集。GEL 主要用来扩展CCS 的功能,方便用户调试程序,但不是必需的。

8. Tools

- 1) On-chip Flash Programmer : Flash 烧写插件。

切记! 在密码区域不要全写0,在烧写的时候不要掉电或者重启系统,否则会后果严重!