GEL文件的作用

CCS的GEL语言是一种交互式的命令，它是解释执行的，即不能被编译成可执行文件。它的作用在于扩展了CCS Studio的功能，可以用gel来调用一些菜单命令，对DSP的存储器进行初始化配置等。 对于不同型号的DSP，我们应选择相对应的GEL文件。

    当CCS Studio启动时，GEL文件加载到PC机的内存中，如果定义了StartUp()函数则执行该函数在CCS Studio(V2.3或更低的版本中)，主机和目标板的初始化工作都在Startup()函数中执行。但是对于支持Connect /Disconnect的CCS Studio(V2.4或之后的版本，比如3.1版本，3.3版本)，GEL文件可能并没有正确的执行。因为CCS Studio启动时和目标处理器之间的连接是断开的，当Startup()函数试图访问目标处理器时会出错。因此，V2.4或之后的版本，当CCS启动 时，用一个新的回调函数OnTargetConnect()来执行目标处理器的初始化工作。

    对于支持connect/disconnect的CCS环境，Startup()函数中尽量避免Get\_Reset()、 GEL\_BreakPtAdd()、GEL\_TextOUT()和GET\_OpenWindow()，因为StartUp()执行时CCSStudio的 任何控制窗口还没有打开。

我们对两种不同的gel文件进行对比，让大家更容易理解。

不支持Connect/Disconnect的CCSStudio GEL文件中的StartUp()函数：

             StartUp()

{

setup\_memory\_map();

GEL\_Reset(); /\* Do not call in StartUp() with CCStudio v2.4 or higher \*/

init\_emif(); /\* Do not call in StartUp() with CCStudio v2.4 or higher \*/

}

支持Connect/Disconnect的CCSStudio GEL文件中的StartUp()函数：

StartUp()

{

setup\_memory\_map();

}

    OnTargetConnect()函数在每一次和目标处理器建立连接时都被调用。进行绝对最小的系统初始化处理，保证CCSStudio在目标处 理器上处于一种可信赖的状态例如：禁止看门狗时钟、DSP复位结束。 对于某些平台，调用GEL\_Reset()函数可使CCSStudio处于一种Good状态，可以通过测试来确定是否需要调用。

我们可看出后一种仅建立了存储映射。其他初始化dsp的工作交给了OnTargetConnect()函数去完成。下面我们以6437为例，gel的源码如下：

StartUp( )

{

    Setup\_Memory\_Map( );

}

OnTargetConnect( )

{

    Setup\_Cache( );

    Setup\_Pin\_Mux( );               // Setup Pin Mux

    Setup\_Psc\_All\_On( );            // Setup Power Domains

    Setup\_PLL0\_594\_MHz\_OscIn( );    // Setup Pll1 [DSP @ 594 MHz][Core 1.20V]

    Setup\_PLL1\_DDR\_135\_MHz\_OscIn();

    Setup\_Aemif\_8Bit\_Bus( );        // Setup Async-EMIF [8-bit bus]

  }

注意：GEL文件在CCS启动后常驻内存，在仿真器和目标系统上电的时候起到初始DSP的作用。在上电后再改变GEL文件将不会对DSP产生影响。