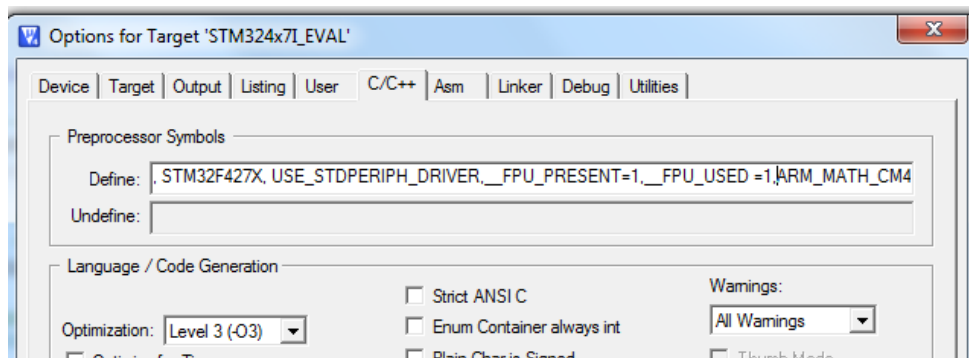


FPU PRESENT=1, FPU USED =1。



- c. 这样编译时就加入了启动 FPU 的代码，CPU 也就能正确高效的使用 FPU 进行简单的加减乘除了。

### 3. 进一步说明使用芯片复杂数学运算使用

对于复杂运算，比如三角函数，开方等运算，需要如下设置：

- 包含 `arm_math.h` 头文件。
- 在工程选项的 C/C++ 选项卡的 `define` 中继续加入语句 `ARM_MATH_CM4`。
- 在工程选项的 C/C++ 选项卡的 `define` 中继续加入语句 `__CC_ARM`。

以使用 `sin`，`cos` 运算举例，需要调用 `arm_sin_f32()` 以及 `arm_cos_f32()`，这两个函数定义在 `arm_sin_f32.c` 和 `arm_cos_f32.c` 中，需要在工程中加入这两个 c 文件。

在 ST 库文件包中的文件目录如下：

```
\stm32f4_dsp_stdperiph_lib\STM32F4xx_DSP_StdPeriph_Lib_V1.1.0\Libraries\CMSIS\DSP_Lib\Source\FastMathFunctions
```

在 keil 安装目录下的文件目录如下：

```
\Keil\ARM\CMSIS\DSP_Lib\Source\FastMathFunctions
```

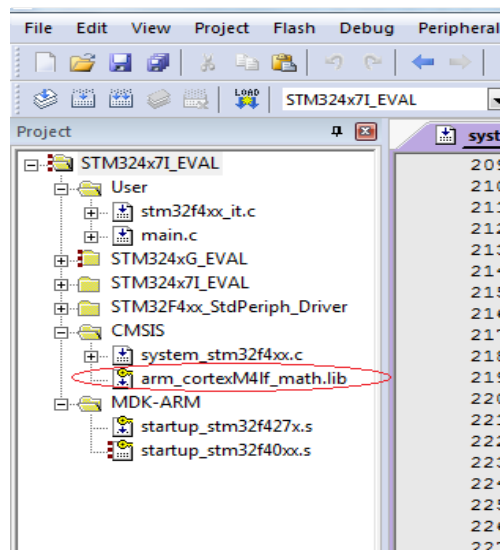
当用到更多数学运算，如开根号，三角运算，求绝对值等等，客户也可以直接在工程中加入 ARM 中的数学运算库 `arm_cortexM4lf_math.lib`，而不需要一个个文件的添加，

在 ST 库文件包中的目录如下：

```
\stm32f4_dsp_stdperiph_lib\STM32F4xx_DSP_StdPeriph_Lib_V1.1.0\Libraries\CMSIS\Lib\ARM
```

在 keil 安装目录下的文件目录如下：

```
Keil\ARM\CMSIS\Lib\ARM
```



#### 四. 结语

从测试效果看当使用了硬件浮点运算单元，数学计算变得简单高效，可以留给系统更多时间处理其他控制程序，有效提升系统效率，节省时间。