

IAR环境如何将数学函数库编排到sLib保护区

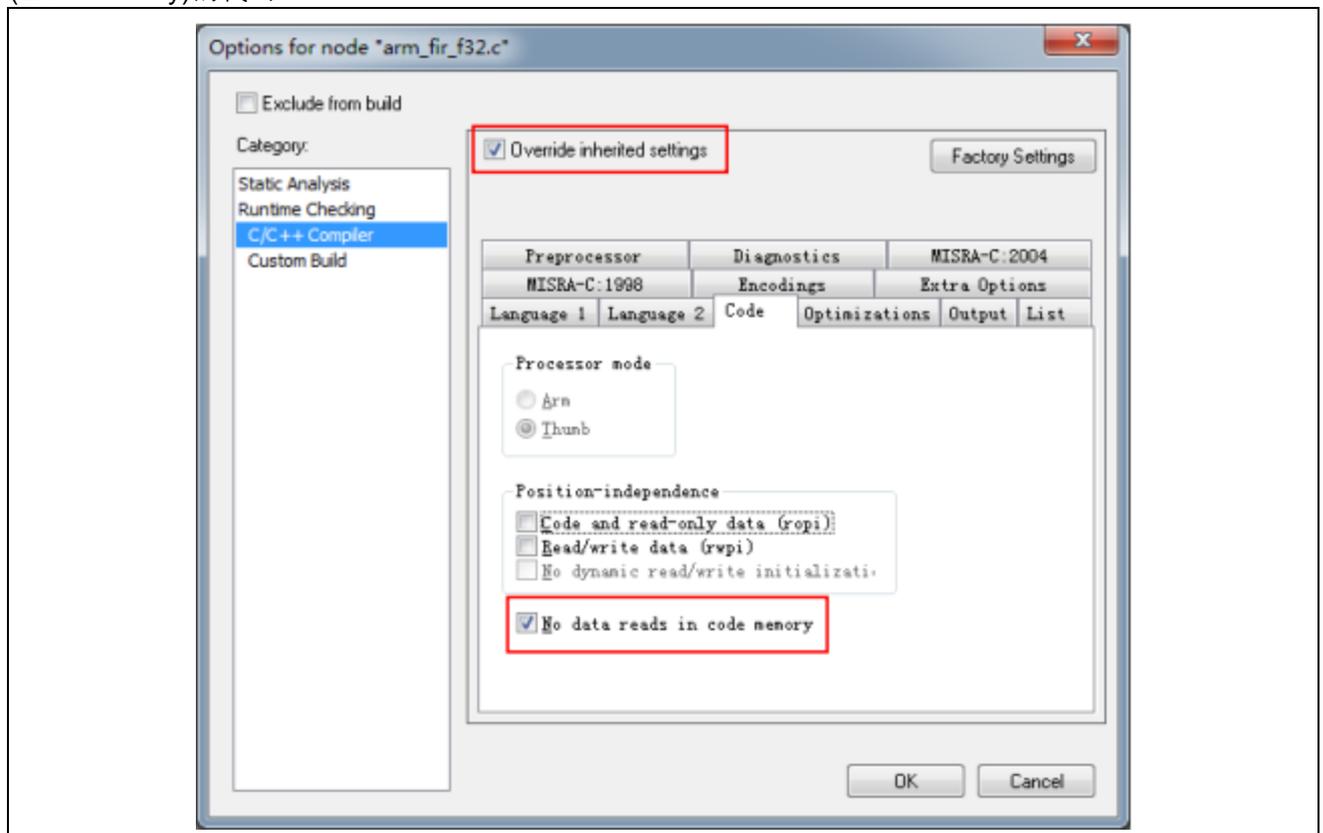
Questions: 在 IAR 环境下，如何将数学函数库编排到 SLIB 保护区？

Answer:

一般会放到 sLib 保护区的函数，通常都是算法的核心部分，用到数学运算的机会很高，相对的也就会用到 ARM 官方或者是 IDE 开发工具提供的数学函数库，这一类的函数库都是事先已经编译过的代码，可能会有文字池(Literal pool)这类格式的的代码，或者是执行数学运算时须用到的常数表(constant table)，这些代码跟常数表就不能放到 sLib-code 保护区，需要额外做编排处理。

处理的方式会因使用的 IDE 开发工具不同而有不同的处理方法，以 Keil MDK 来说，Keil 提供的数学函数库就有文字池(Literal pool)这类格式的的代码，所以不能放进 sLib-code 保护区，处理方式要麻烦许多，在《AN0065_AT32F415_Security_Library_Application_Note》的 3.4.2 章节（其他型号请至雅特力官网查阅对应 AN 的相关章节）里有详细的说明。

在 IAR 的开发工具，处理起来就容易许多，因为 IAR 会根据编译程序的设定条件，选择不同格式的函数库，如 sLib 应用手册里有提到，放在 sLib-code 保护区的代码，编译时需要如下设定去产生只执行 (Execute-only)的代码。



这时候如果 sLib-code 保护区的代码有使用到数学函数库，编译程序在产生代码时，就会引用 Execute-only 格式的数学函数库，因此就可以将此函数库也编排到 sLib-code 保护区，唯一需要注意的是函数库可

能会有运算时要用到的常数表，就必须另外将这些常数表编排到 sLib-data 保护区，以下面这段代码为例：

```
xs_complex Wnr(int lei)
{
    if (lei == 0)
    {
        while (1);
    }
    xs_complex w = { 0 };
    w.real = cos(PI / lei);
    w.imag = -sin(PI / lei);
    return w;
}
```

此处有用到 COS 跟 SIN 这两个三角函数运算式，就会使用到数学函数库。如果这段代码是放在 sLib-code 保护区，且被设定了 Execute-only 选项，在整个项目编译完成后，打开此项目的.map 文件，在 MODULE SUMMARY 段落可以看到如下的讯息：

```
m7Mx_tls.a: [4]
DblAdd.o          376
DblCmpEq.o        32
DblCmpGe.o        46
DblCmpLe.o        46
DblDiv.o          582
DblMul.o          418
DblSub.o          226
DblToFlt.o        104
DblToS32.o        58
FltToDbl.o        72
S32ToDbl.o        34
U32ToDbl.o        26
cos_sin64.o       1 186
fpinit_M.o        34
frexp.o           112
iar_Exp64.o       1 074
ldexp.o           314
logx64.o          884
math_wrappers64.o 58
pow64.o           1 896 1 056
-----
Total:            7 578 1 056
```

此处的 m7Mx_tls.a 就是数学函数库，IAR 开发工具提供的函数库名称都有明确的命名规则，详细信息可以参考《IAR C/C++ Development Guide Compiling and Linking for Advanced RISC Machine sLtd's ARM® Cores DARM -24》这份文件中的 Prebuilt Runtime Libraries 章节。其中开头的 m 代表数学运算功能，第四个字母 x 代表此函数库是 Execute-only 格式，因此就可以放进 sLib-code 保护区。

将函数库编排到 sLib-code 保护区的方法如下：

1. 打开项目的.icf 文件；
2. 如下图红色方框标示的方式，将 m7Mx_tls.a 放进 sLib 保护区；

其中会使用 ro code 这个命令来将函数库的代码部分编排到 sLib-code 保护区，使用 ro data 命令确保函数库里如果有常数表的话，就把它编排到 sLib-data 保护区。

```
/* Place IP Code in page 2 which will be SLIB protected */
place in SLIB_CODE_region { ro code object cfar.o,
                            ro code object FFT.o,
                            ro code object m7Mx_tls.a };

place in SLIB_DATA_region { ro data object cfar.o,
                            ro data object FFT.o,
                            ro data object m7Mx_tls.a,
                            ro data object rfic_spi_code_orthrus_table.o };
```

3. 编译完成后，检查.map 文件中 PLACEMENT SUMMARY 这个段落，如下图，sLib-code 被排到“P2”区块，sLib-data 被排到“P3”这个区块；

```

*****
*** PLACEMENT SUMMARY
***

"A0": place at 0x08004000 { ro section .intvec };
"P1": place in [from 0x08004000 to 0x08005fff] |
      [from 0x08008c00 to 0x0801ffff] { ro };
"P2": place in [from 0x08006000 to 0x080083ff] {
      ro code object cfar.o, ro code object FFT.o,
      ro code object m7Mx_tls.a };
"P3": place in [from 0x08008400 to 0x08008bff] {
      ro data object cfar.o, ro data object FFT.o,
      ro data object m7Mx_tls.a,
      ro data object rfic_spi_code_orthrus_table.o };
define block CSTACK with size = 4K, alignment = 8 { };
define block HEAP with size = 4K, alignment = 8 { };
"P4": place in [from 0x20000000 to 0x2000fbff] {
      rw, block CSTACK, block HEAP };
"P5": place in [from 0x2000fc00 to 0x2000ffff] {
      rw object cfar.o, rw object FFT.o };
initialize by copy { rw };

```

4. 同样在 PLACEMENT SUMMARY 段落中，找到“P2”及“P3”的讯息，如下图，可以看到函数库 m7Mx_tls.a 包含的.o 文件都被放进来了，其中因为 pow64.o 还包含有常数表，常数表的部分就被编排到“P3”区块，也就是 sLib-data 区。

```

"P2":
      0x2366
      .text      ro code 0x08006000 0x768 pow64.o [4]
      .text      ro code 0x08006768 0x20  DblCmpEq.o [4]
      .text      ro code 0x08006788 0x70  frexp.o [4]
      .text      ro code 0x080067f8 0x2e  DblCmpLe.o [4]
      .text      ro code 0x08006828 0x1a2  DblMul.o [4]
      .text      ro code 0x080069cc 0x178  DblAdd.o [4]
      .text      ro code 0x08006b44 0x3a  DblToS32.o [4]
      .text      ro code 0x08006b80 0x22  S32ToDbl.o [4]
      .text      ro code 0x08006ba4 0xe2  DblSub.o [4]
      .text      ro code 0x08006c88 0x246  DblDiv.o [4]
      .text      ro code 0x08006ed0 0x2e  DblCmpGe.o [4]
      .text      ro code 0x08006efe 0x432  iar_Exp64.o [4]
      .text      ro code 0x08007330 0x13a  ldexp.o [4]
      .text      ro code 0x0800746a 0x4a2  cos_sin64.o [4]
      .text      ro code 0x0800790c 0x3ec  FFT.o [1]
      .text      ro code 0x08007cf8 0x3a  math_wrappers64.o [4]
      .text      ro code 0x08007d34 0x68  DblToFlt.o [4]
      .text      ro code 0x08007d9c 0x374  logx64.o [4]
      .text      ro code 0x08008110 0x1d0  cfar.o [1]
      .text      ro code 0x080082e0 0x48  FltToDbl.o [4]
      .text      ro code 0x08008328 0x22  fpinit_M.o [4]
      .text      ro code 0x0800834c 0x1a  U32ToDbl.o [4]
      - 0x08008366 0x2366

"P3":
      0x5f8
      .rodata    const 0x08008400 0x420 pow64.o [4]
      .rodata    const 0x08008820 0x1d8  rfic_spi_code_orthrus_table.o [1]
      - 0x080089f8 0x5f8

```

除了数学函数库之外，项目开发过程中，sLib-code 保护区内的代码可能也会用到其他 IAR 提供的函数库，也可以依照此文件所描述的方法来处理。

类型：MCU 应用

适用型号：AT32F4

主功能：sLib

次功能：无

文档版本历史

日期	版本	变更
2022.2.22	2.0.0	最初版本

重要通知 - 请仔细阅读

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途(及其依据任何司法管辖区的法律的对应情况)，或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：(A) 对安全性有特别要求的应用，如：生命支持、主动植入设备或对产品功能安全有要求的系统；(B) 航空应用；(C) 汽车应用或汽车环境；(D) 航天应用或航天环境，且/或(E) 武器。因雅特力产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，风险由购买者单独承担，并且独力负责在此类相关使用中满足所有法律和法规要求。

经销的雅特力产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。

© 2022 雅特力科技 (重庆) 有限公司 保留所有权利