

# IMXUG

## i.MX 配置工具用户指南

第 5 版 — 2023 年 1 月 2 日

用户指南

### 文档信息

信息	内容
关键词	MCUXpresso 配置工具, i.MX
摘要	i.MX 配置工具是 MCUXpresso 配置工具套件的一部分, 该套件提供了一系列的评估和配置工具, 可以为用户从初始评估到量产软件的开发过程提供指导和帮助。i.MX 配置工具可让您轻松地配置 i.MX 处理器的引脚和 DDR 控制器。一般来说, 此配置软件可让您创建、检查、更改和修改处理器引脚的各种配置和复用。它还允许您配置和验证 DDR 设置。本文介绍了 i.MX 配置工具的基本组件, 并列出了使用它们来配置引脚和 DDR 的步骤。



## 1 简介

i.MX 配置工具是 MCUXpresso 配置工具套件的一部分，该套件提供了一系列的评估和配置工具，可以为用户从初始评估到量产软件的开发过程提供指导和帮助。i.MX 配置工具可让您轻松地配置 i.MX 处理器的引脚和 DDR 控制器。一般来说，该配置软件可让您创建、检查、更改和修改处理器引脚的各种配置和复用。它还允许您配置和验证 DDR 设置。本文介绍了 i.MX 配置工具的基本组件，并列出了使用它们来配置引脚和 DDR 的步骤。

**注意：** i.MX 处理器当前仅提供独立桌面版本。

### 1.1 特性

i.MX 配置工具由 Pins（引脚）、TEE 和 DDR 工具组成。

引脚工具可用于：

- 配置引脚的路由/复用
- 管理用于路由初始化的不同功能
- 配置引脚功能/电气特性
- 生成用于路由和功能/电气特性的代码

DDR 工具可用于：

- 配置 DDR 控制器
- 验证 DDR 配置

引脚工具可为目标器件/电路板定义引脚的路由。您可以使用两种方式来共享工具配置：一是使用 MEX 文件中保存的配置，二是使用生成的 C 或 DTSI（可选）代码片段文件（通过导入/导出或直接复制粘贴生成的源文件）。

**注意：** 引脚工具通常生成的代码将引脚路由到外设，但并不用于外设的配置。一些外设可能需要额外配置引脚来分配功能或通道。例如，对于一些 ADC 路由提供引脚和 ADC 外设之间的连接。然后，您可以从 ADC 外设中分配 ADC 通道。

DDR 工具支持您查看和配置基本的 DDR 属性，如内存类型、频率、通道数等，还可以通过工具内置的各种测试用例来测试 DDR 配置。在您指定了连接类型之后，可以选择场景，以及要在这些场景中运行的测试用例，运行测试后，可以查看测试结果、日志和摘要。

### 1.2 版本

对于 i.MX 处理器，此工具称为 i.MX 配置工具，目前仅提供桌面机应用程序版本。此工具连接恩智浦服务器并获取可用处理器的列表。一旦使用，将按需检索处理器数据。要在离线模式下使用该桌面工具，须先在线为指定处理器建立配置。之后，此工具会将指定处理器信息保存在本地用户文件夹中，从而实现快速访问和离线使用。

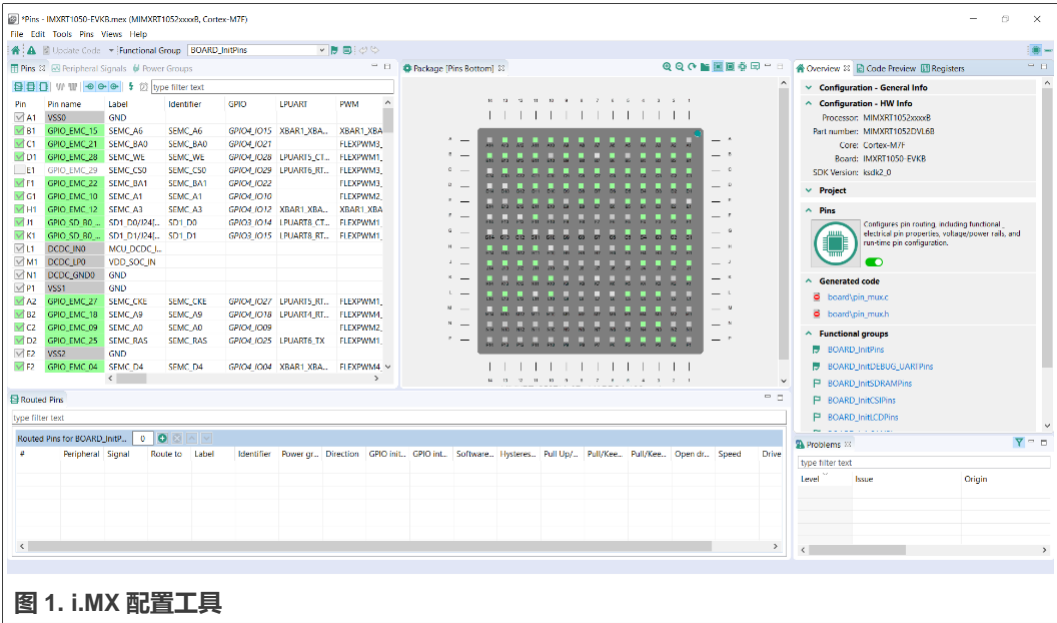


图 1. i.MX 配置工具

1.3 工具本地化

工具仅提供英文和简体中文两种语言版本。  
工具会自动根据电脑的全局设置来调整语言环境。  
如果您想手动更改语言环境，请在命令行中添加以下参数：

```
tools.exe -nl zh
```

您还可以在 tools.ini 文件中添加以下行来设置语言：

```
-Duser.language=zh
```

**注意：**如果您把系统语言环境设为中文，工具会自动以中文显示菜单项、工具提示和帮助。切换语言之后，建议您删除[home\_dir]/.nxp 文件夹，避免菜单项出现缓存问题。

2 用户界面

2.1 启动开发向导

启动配置工具后，会自动弹出启动向导。使用此向导，可以创建配置或打开现有配置。

**注意：**如果您想在以后启动时跳过此向导，请勾选“Open an existing configuration (打开现有配置)”选项下方的“Always open last configuration (始终打开上次配置)”复选框。或者，您也可以在“Preferences (首选项)”中勾选“Automatically open previously used configuration (自动打开以前使用的配置)”复选框来达到同样的效果。

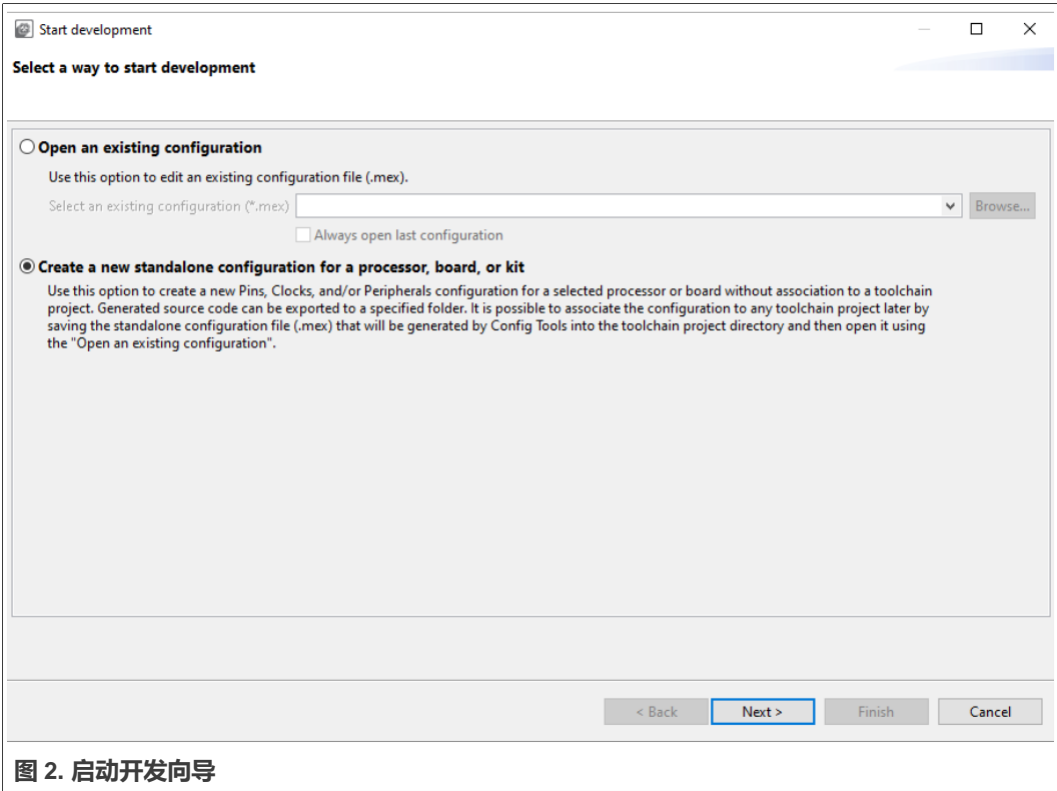


图 2. 启动开发向导

**注意：**这个向导的内容与您在 **Menu bar (菜单栏)** 中选择 **“File > New (文件 > 新建)”** 打开的向导相似。

2.2 创建、保存和打开配置

在本节上下文中，配置指的是存储在 MEX（微控制器导出配置）文件中的通用工具设置。这个文件包含所有可用工具的设置信息。要想解析工具链项目，保存 MEX 文件的文件夹里必须有且仅有一个项目文件。

2.2.1 新建配置

您可以从 **“Start development (启动开发)”** 向导或在 **Menu bar (菜单栏)** 中选择 **File > New (文件 > 新建)** 来新建一个配置。

如果您要使用恩智浦参考电路板或套件进行开发，我们建议您从示例开始，为相应参考电路板或套件新建配置。这种配置包含该参考电路板特有的设置。如果您选择处理器，则新建的配置将为空。

新建配置后，还可以继续从 MEX 文件导入现有配置。如果您已经有可用的配置或者想要重新使用以前的配置，那么它非常有用。要从 MEX 文件导入现有配置，请在 **Menu bar (菜单栏)** 中选择 **File > Import... > Import configuration (\*.mex) (文件 > 导入... > 导入配置 (\*.mex))**。

2.2.1.1 新建独立配置

您可以新建一个不属于任何工具链项目的配置。



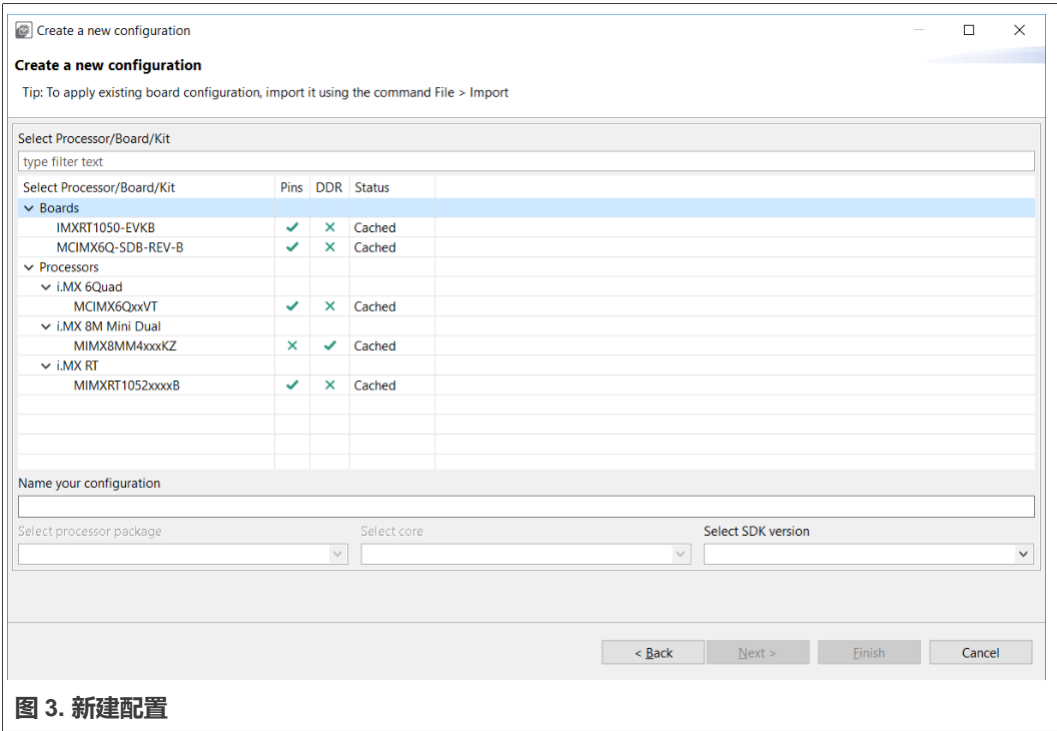


图 3. 新建配置

要新建独立配置，请按照以下步骤操作：

1. 在 “Start development (启动开发)” 向导中，选择 “Create a new standalone configuration for processor, board, or kit (为处理器、电路板或套件创建新的独立配置)”。或者，在 Menu bar (菜单栏) 中，选择 File > New (文件 > 新建)。
2. 单击 Next (下一步)。
3. 从列表中选择处理器、电路板或套件。  
**注意：**如果离线工作，您只能看到本地保存的选项。有关详细信息，请参阅[脱机工作部分](#)。
4. 为您的配置命名。您还可以选择处理器封装、处理器核心类型和 SDK 版本。
5. 单击 Finish (完成)。

2.2.2 保存配置

要保存配置以供将来使用，请从 Menu bar (菜单栏) 中选择 File>Save。

要保存配置的备份，请按照以下步骤操作：

1. 在 Menu bar (菜单栏) 中，选择 File>Save Copy As。
2. 在对话框中，指定配置的名称和目标。
3. 单击 Save (保存)。  
保存含 MEX 文件的文件夹必须正好包含一个项目文件，以便能够解析工具链项目。

2.2.3 打开现有配置

要打开现有配置，请按照以下步骤操作：

1. 在 “Start development (启动开发)” 向导中，选择 Open an existing configuration (打开现有配置)。或者，在 Menu bar 中选择 File > Open。

2. 单击 **“Browse (浏览)”** 导航到您的配置文件。
3. 选择配置文件并单击 **“Open (打开)”** 。
4. 或者，选择 **“Always open last configuration (始终打开最后的配置)”** 跳过 **Start development (启动开发)** 向导并默认加载上次保存的配置。

### 2.2.4 用户模板

您可以把当前配置作为用户模板导出并保存，以后可用作参考配置文件。

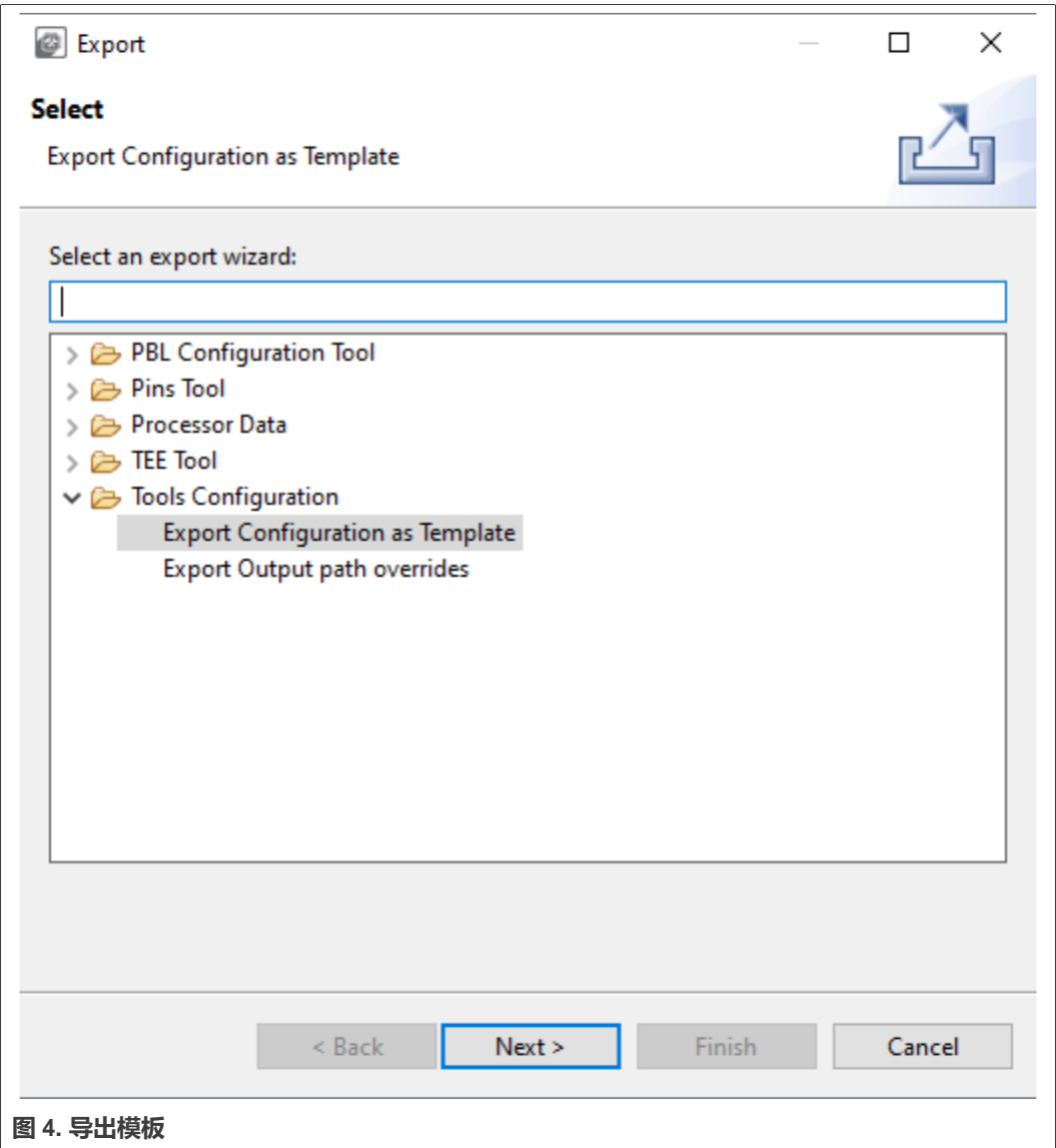


图 4. 导出模板

导出的模板在 **“New Configuration (新建配置)”** 向导中来新建配置。您还可以在生成的代码中为引脚定义自定义标签或为 `#define` 定义标识符前缀。您可以在 **Menu bar (菜单栏)** 中选择 **“File > Export > Tools Configuration > Export Configuration as Template”**，导出配置。

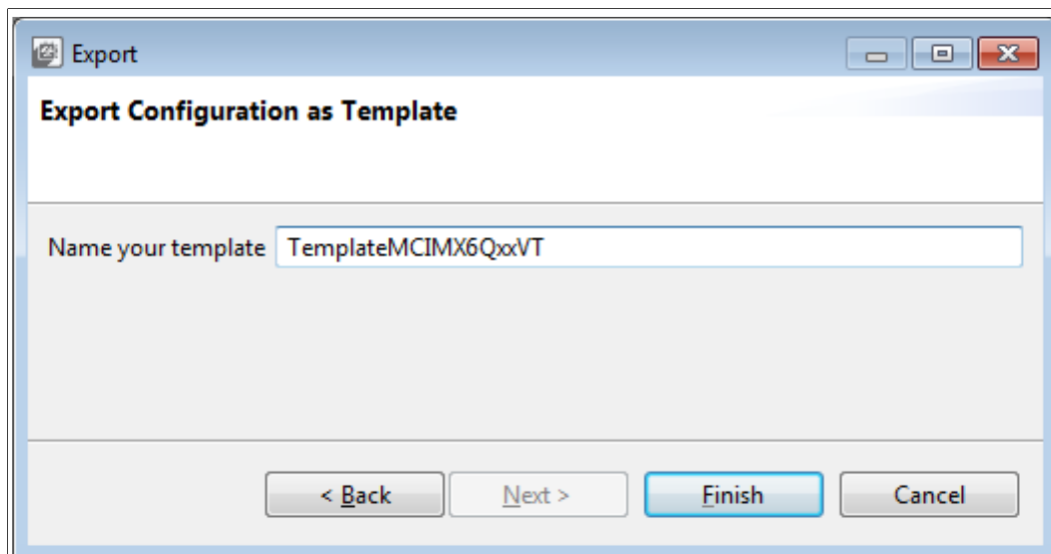


图 5. 从模板新建配置

**注意：**模板存储在本地硬盘上的以下位置：

`{ $user } /.nxp / { tools_folder } / { version } / templates.`

### 2.2.5 导入源代码文件

您可以导入源代码文件，用它们进行更多的配置。

要导入源代码文件，请按照以下步骤操作：

1. 在 **Menu bar (菜单栏)** 中，选择 **File > Import...**。
2. 从列表中，选择 **MCUXpresso Config Tools>Import Source**。

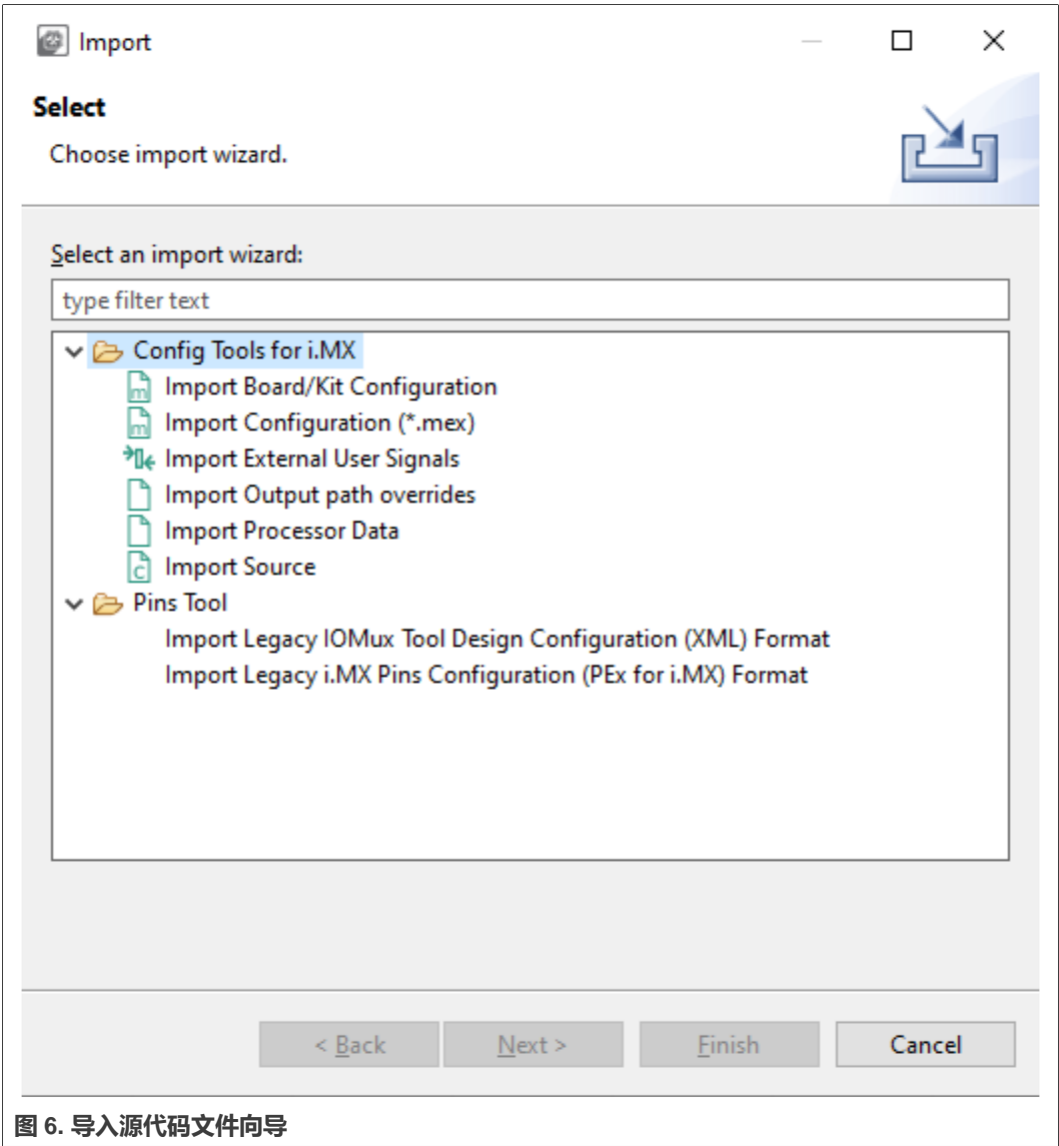


图 6. 导入源代码文件向导

3. 单击 **Next (下一步)** 。
4. 在下一页上，单击 **Browse (浏览)** 指定源文件的位置。
5. 选择要导入的源文件并单击 **Open (打开)** 。
6. 在下一页上，勾选左列中的复选框，选择要导入的函数组（根据工具分类）。
7. 在右列的下拉菜单中选择一种方式来定义如何导入函数组：
  - **Rename (重命名)** – 所有文件合并到当前配置中。它只导入所有的函数。如果导入的函数与现有函数同名，系统会自动给它加上序号。例如，如果配置中存在 `BOARD_InitPins`，则导入的同名函数将重命名为 `BOARD_InitPins1`。
  - **Overwrite (覆盖)** – 将所有文件合并到当前配置中。它只导入所有的函数。如果导入的函数与现有函数同名，系统会用导入的函数替换现有函数。
8. 单击 **Finish (完成)** 。

**注意：**仅 C 文件或符合 YAML 配置的设备树文件才能导入。系统仅导入配置部分，然后根据此配置重新创建整个 C 文件。其余的 C 和 DTSI 文件将被忽略。

### 2.2.5.1 导入配置

要从 MEX 文件导入现有配置，请按照以下步骤操作：

1. 在 Menu bar (菜单栏) 中，选择 File > Import...>。
2. 在 Import (导入) 向导中，选择 MCUXpresso Config Tools > Import configuration (\*.mex)。
3. 单击 Next (下一步)。
4. 在下一页，单击 Browse (浏览) 指定配置文件 (\*.mex) 的位置。
5. 选择要导入的 MEX 文件，然后单击 Open (打开)。
6. 在下一页上，勾选左列中的复选框，可选择要导入的函数组（根据工具分类）。
7. 在右列下拉菜单中选择一种方式来定义如何导入函数组：
  - **Rename (重命名)** – 所有文件合并到当前配置中。它只导入所有的函数。如果导入的函数与现有函数同名，系统会自动给它加上序号。例如，如果配置中存在 BOARD\_InitPins，则导入的同名函数将重命名为 BOARD\_InitPins1。
  - **Overwrite (覆盖)** – 将所有文件合并到当前配置中。它只导入所有的函数。如果导入的函数与现有函数同名，系统会用导入的函数替换现有函数。
8. 单击 Finish (完成)。

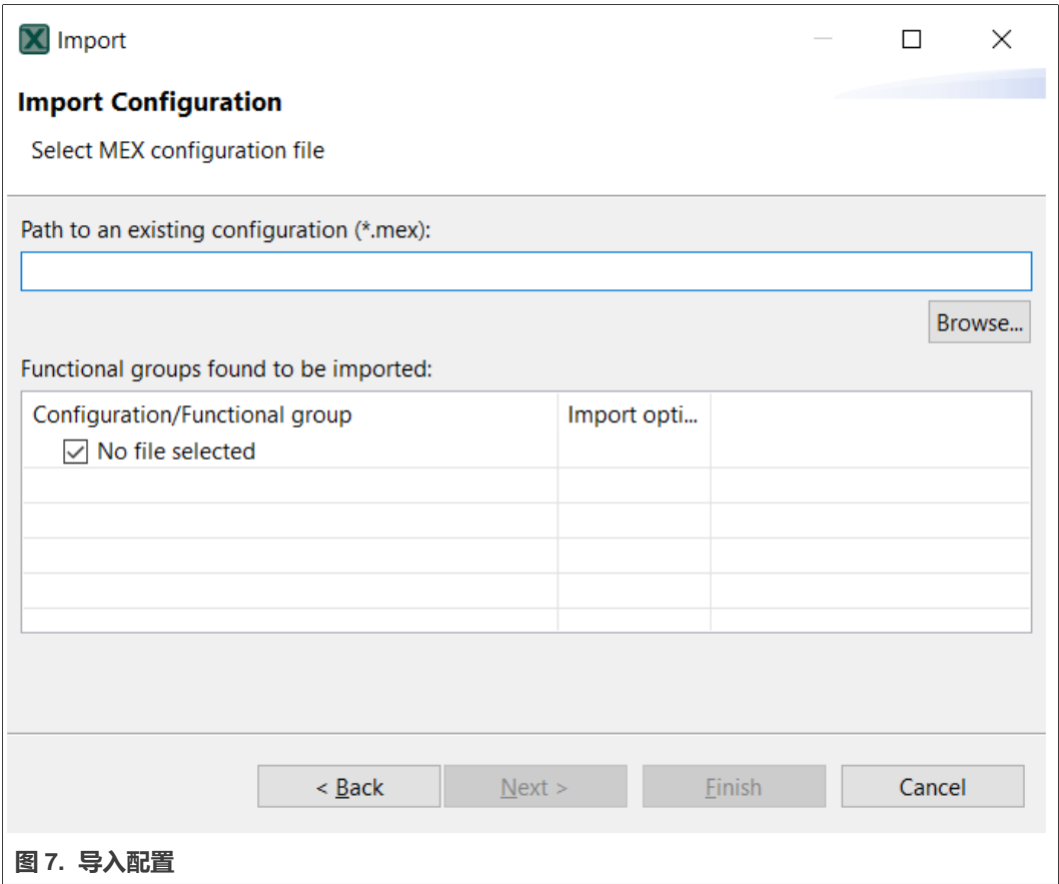


图 7. 导入配置

2.2.5.2 导入电路板/套件配置

使用 CFG 工具数据中提供的默认电路板/套件模板的导入设置进行进一步配置。  
要导入电路板/套件配置，请按照以下步骤操作：

1. 在 **Menu bar (菜单栏)** 中，选择 **File > Import...>**。
2. 在 **Import (导入)** 向导中，选择 **MCUXpresso Config Tools > Import Board/Kit Configuration**。
3. 单击 **Next (下一步)** 。
4. 在下一页，从下拉菜单中选择电路板/套件的版本。
5. 勾选左列中的复选框，选择要导入的函数组（根据工具分类）。
6. 在右列下拉菜单中选择一种方式来定义如何导入函数组：
  - **Rename (重命名)** – 所有文件都合并到当前配置中。它只导入所有的函数。如果导入的函数与现有函数同名，系统会自动给它加上序号。例如，如果配置中存在 BOARD\_InitPins，则导入的函数将重命名为 BOARD\_InitPins1。
  - **Overwrite (覆盖)** – 将所有文件合并到当前配置中。它只导入所有的函数。如果导入的函数与现有函数同名，系统会用导入的函数替换现有函数。
7. 单击 **Finish (完成)** 。

## 2.2.6 导出源文件

您可以使用 Export（导出）向导来导出生成的源文件。

要启动 Export（导出）向导：

1. 从 **Menu bar（菜单栏）** 中选择 **File > Export**。
2. 选择 **Export Source Files（导出源文件）**。

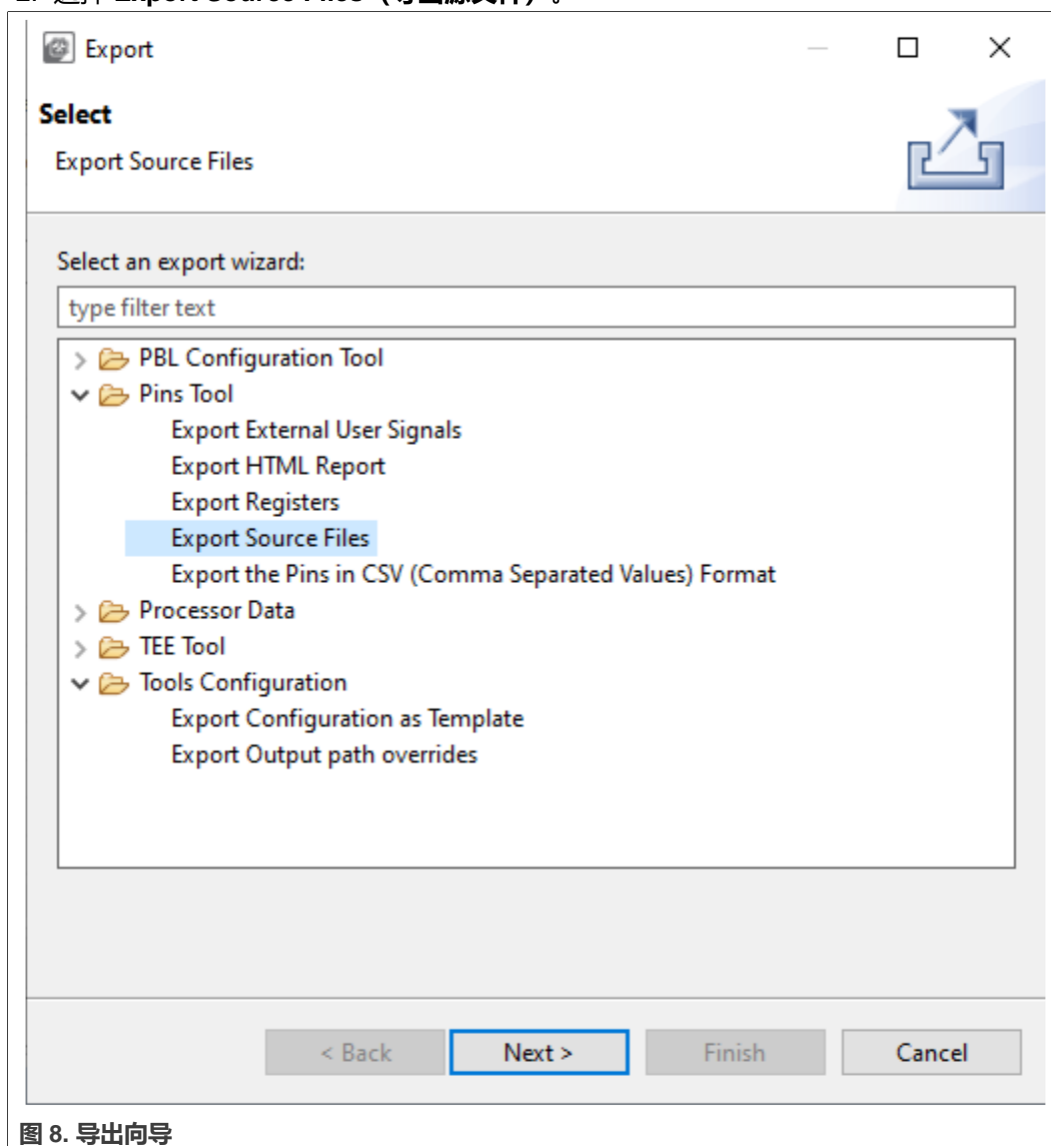


图 8. 导出向导

3. 单击 **Next（下一步）**。
4. 选择要保存生成文件的目标文件夹。

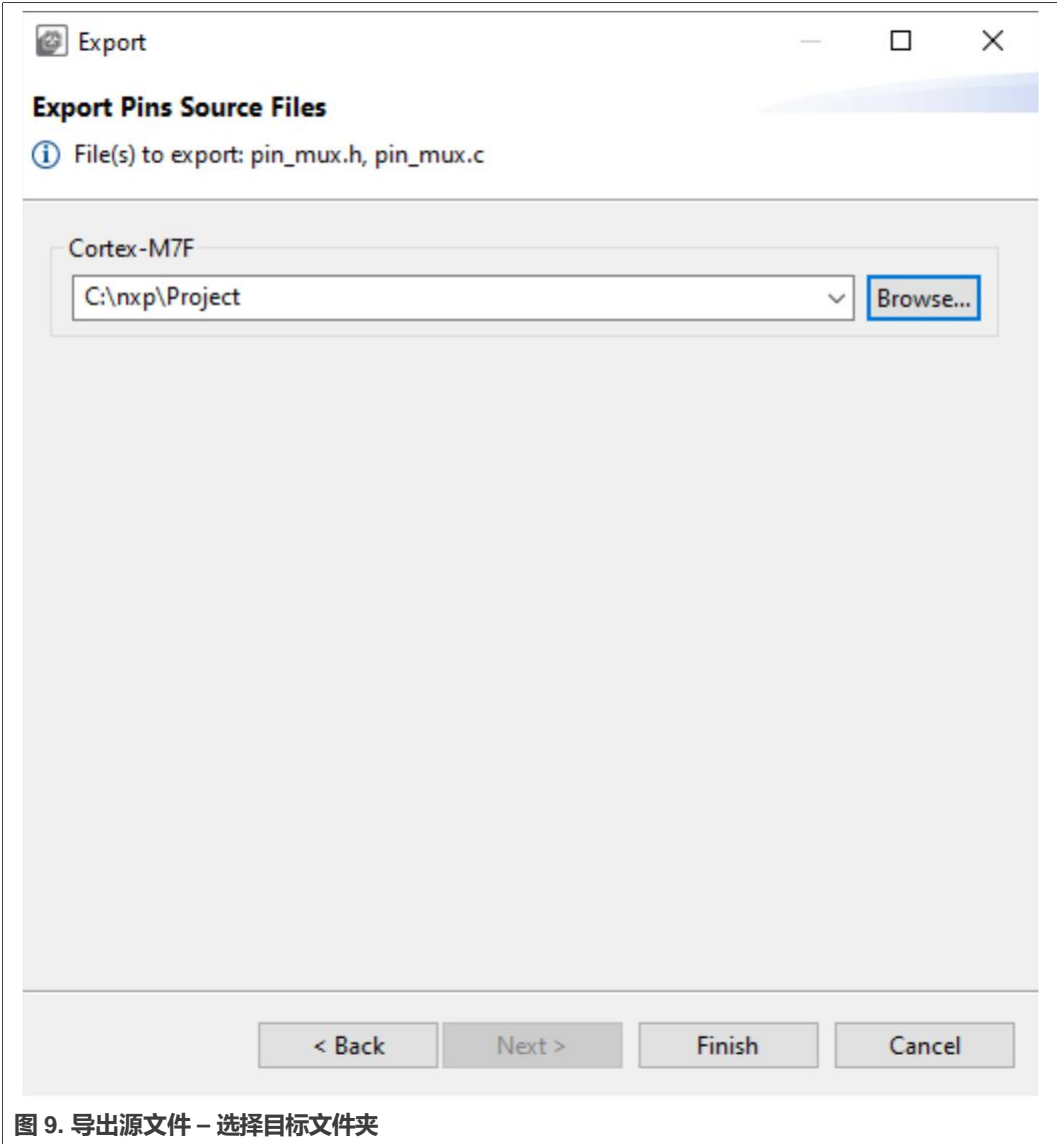


图 9. 导出源文件 – 选择目标文件夹

- 5. 如果是多核处理器，选择要导出的处理器核心。
- 6. 单击 **Finish（完成）**。

2.3 菜单栏

Menu bar (菜单栏) 包含 5 个菜单选项：File (文件)、Edit (编辑)、Tools (工具)、Views (视图)、Help (帮助)，以及一个专用工具菜单。

File (文件) 菜单包含各种文件管理项。

表 1. 文件菜单

菜单项	说明
新建...	创建配置。如需了解更多详细信息，请参阅“ <a href="#">创建、保存和打开配置</a> ”章节。
打开	从 MEX 文件打开配置。
保存	保存当前配置。



表 1. 文件菜单（续）

菜单项	说明
将副本另存为...	创建当前配置的备份副本。
切换处理器	切换到不同的处理器。如需了解更多信息，请参阅“ <a href="#">切换处理器</a> ”章节。
切换封装	切换到不同的处理器封装。如需了解更多信息，请参阅“ <a href="#">切换处理器</a> ”章节。
选择处理器核心	选择处理器核心，完成更多配置。
数据管理器	管理本地数据。如需了解更多信息，请参阅“ <a href="#">管理数据和脱机工作</a> ”章节。
导入...	从源文件导入设置。如需了解更多信息，请参阅“ <a href="#">高级功能</a> ”章节。
导出...	导出源文件和其他工具信息。如需了解更多信息，请参阅“ <a href="#">高级功能</a> ”章节。
退出	退出应用程序。如果有任何未保存的更改，系统会提示您保存更改。

**Edit（编辑）** 菜单包含基本编辑操作以及修改整个框架样式和表现的项目。

表 2. 编辑菜单

菜单项	说明
打开更新代码对话框	配置更改后更新代码。如需了解更多信息，请参阅 <a href="#">第 2.4.2 章节</a> 。
撤销(...)	取消之前的操作。撤回的操作会显示在括号里。
恢复(...)	取消之前的撤销操作。恢复的操作会显示在括号里。
复制	将选定的文本复制到剪贴板。
全选	选择当前字段/视图中的全部文本。
从默认初始化函数调用	设置要从默认初始化函数中调用的当前选定的函数组。
函数组属性	编辑函数组属性。
首选项	编辑首选项。如需了解更多信息，请参阅“ <a href="#">首选项</a> ”章节。
配置首选项	编辑配置首选项。如需了解更多信息，请参阅“ <a href="#">配置首选项</a> ”章节。

**Tools（工具）** 菜单列出了工具框架中可用的所有工具。使用此菜单在不同的工具之间切换。

**Tool-specific（工具特定菜单）** 包含了为各个工具量身定制的项目。仅显示与当前活动工具相关的项目。菜单名称复制当前活动工具的名称。

表 3. Pins 菜单

项目	说明
函数组	编辑函数组属性。
自动路由	尝试解决路由问题。这会弹出 <b>Automatic Routing（自动路由）</b> 对话框，显示已解决和还需要手动处理的路由问题。

表 3. Pins 菜单 (续)

项目	说明
应用扩展板	将扩展板应用于已创建的扩展接头
创建默认路由	打开创建新函数组的对话框，其中包含复位后引脚状态和内部信号。
刷新	刷新生成的代码和整个图形用户界面。
复位为电路板默认设置	复位为电路板/套件的默认配置。
复位为处理器默认设置	复位为处理器的默认配置。

**Views (视图)** 菜单包含工具特定的可用视图列表。从列表中选择一個视图来打开它。或选择一个已打开的视图，高亮显示该视图。选择 **“Reset views (复位视图)”**，可将当前工具视图复位为默认状态。**Help (帮助)** 菜单包含帮助和通用信息的相关项目。

表 4. 帮助菜单

项目	说明
目录	显示用户指南。
快速入门指南	打开快速入门指南的 PDF 文件。
发行说明	显示已安装版本的发行说明。
社区论坛	显示产品相关社区论坛的网页。
处理器信息	显示包含当前所用处理器信息的网页。
套件/电路板信息	显示包含当前所用电路板或套件信息的网页。
打开 SDK API	显示相关 SDK API 的文档。
检查更新	检查产品的更新版本。如果有新版本，会提示您确认并执行更新。
打开备忘录	显示备忘录可帮助使用这些工具。您还可以从文件或 URL 加载备忘录。
关于	显示通用产品信息。

2.4 工具栏

工具栏位于窗口顶部，包括所有工具常用操作的按钮/菜单。如需了解更多信息，请参阅以下章节。

表 5. 工具栏

项目	说明
配置工具概览	打开 <b>Overview (概览)</b> 对话框，显示当前所用工具的信息。
显示问题视图	打开 <b>Problems (问题)</b> 视图。
更新代码	打开更新对话框，您可直接在指定的工具链项目中更新生成的外设初始化代码。
生成代码	当 <b>“Enable Code Preview (启用代码预览)”</b> 首选项被禁用时，重新生成源代码。

表 5. 工具栏 (续)

项目	说明
函数组选择	选择函数组。Peripherals (外设) 工具中的函数组表示初始化为组的一组外设。该工具为含初始化代码的每个函数组生成一个 C 函数。
从默认初始化调用	由默认初始化函数来初始化当前函数组。
函数组属性	打开 <b>Functional group properties (函数组属性)</b> 对话框, 修改函数组的名称和其他属性。
工具选择	显示各个工具的图标。使用它们在工具之间切换。
撤销/恢复	撤销/恢复上次操作。

此外, 根据您选择的不同工具, 工具栏里还会有一些其他选项。如需了解更多信息, 请参阅针对各个工具的专门章节。

2.4.1 配置工具概览

**Config Tools Overview (配置工具概览)** 让您了解当前配置、硬件和项目的基本情况。它还可以让您快速查看哪些工具是供已使用/活跃的, 哪些是未使用/不活跃工具, 以及生成代码和函数组有哪些。**配置工具概览**图标默认位于工具栏左侧。

**配置工具概览**包含以下几个项目。

表 6. 配置工具概览

项目	说明
配置 – 通用信息	显示当前配置的 MEX 文件名称和路径。单击链接打开包含 MEX 文件的文件夹。要导入其他设置, 请单击 <b>Import additional settings into current configuration (将其他设置导入当前配置)</b> 按钮。
配置 – 硬件信息	显示当前配置的处理器、部件号、核心和 SDK 版本信息。
项目	显示工具链项目信息。
引脚/DDR/SERDES/PBL/TEE	显示 <b>引脚、DDR、SERDES、PBL、TEE</b> 工具的基本信息。

要启用/禁用工具, 请单击切换按钮。您可以点击工具图标来切换不同的工具。此外, 您还可以查看以下有关工具的信息:

表 7. 配置工具概览

项目	说明
生成代码	包含源代码文件列表。点击链接可在 <b>Code Preview (代码预览)</b> 视图中打开文件。
函数组	包含当前激活的函数组列表。要在工具栏的 <b>Functional groups (函数组)</b> 选项卡中选择函数组, 请选择相关链接。

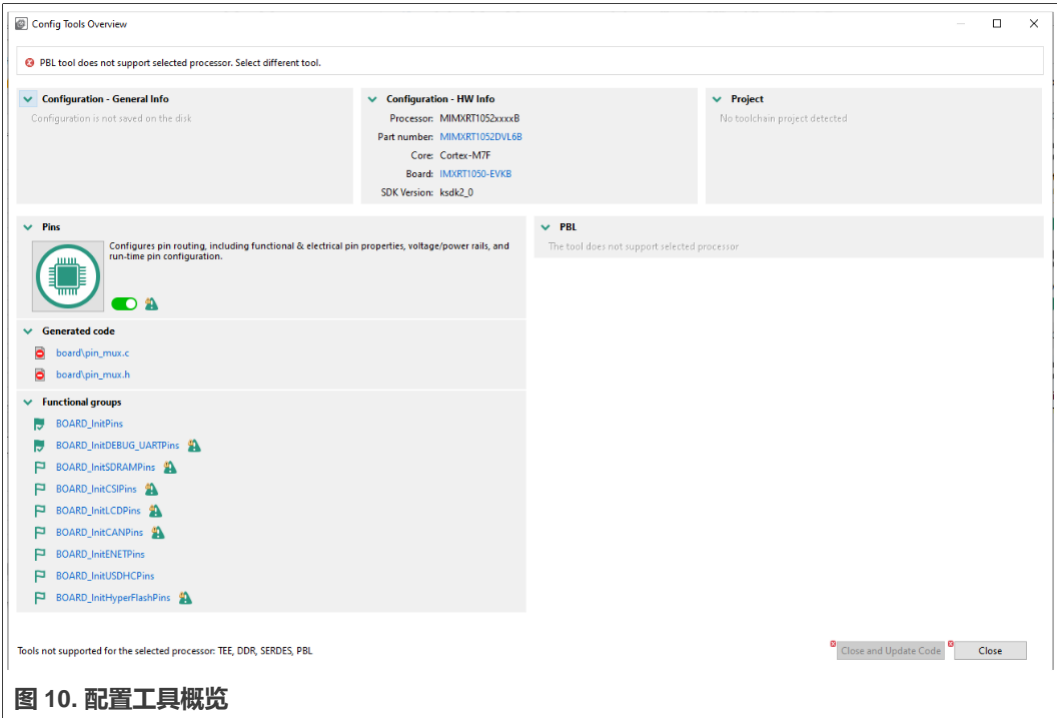


图 10. 配置工具概览

**注意：**不支持的工具不会显示在概览中。

2.4.2 更新代码

如果想更新项目，但不想打开 **Update Files（更新文件）** 对话框，请取消勾选 **“Always show details before Update Code”（更新代码前始终显示详细信息）** 复选框。

要从 **Update Code（更新代码）** 下拉菜单中打开 **Update Code（更新代码）** 对话框，请选择 **“Open Update Code Dialog”（打开更新代码对话框）**。

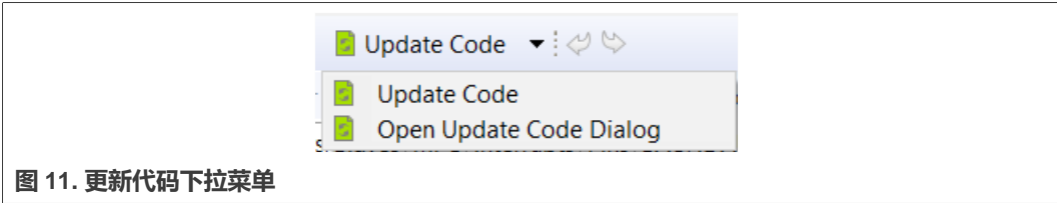


图 11. 更新代码下拉菜单

**注意：**生成的代码会覆盖原来的代码。

只有满足以下条件，才能启用 **Update Code（更新代码）** 操作：

- 如果 MEX 配置保存在工具链项目中，工具中所选的处理器与工具链项目中所选的处理器相匹配
- 选择处理器核心（适用于多核处理器）

2.4.3 函数组

每个**引脚**配置可以包含多个函数组。

这些函数组表示要生成到源代码中的多个函数。使用下拉菜单来切换不同的函数组，并进行相应的配置。

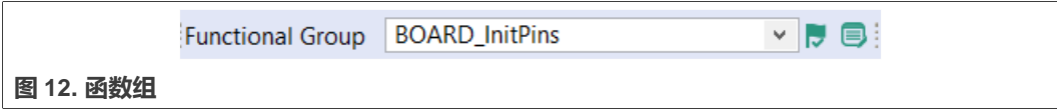


图 12. 函数组

您可以使用另外两个按钮进一步配置函数组：

表 8. 函数组

图标	说明
	在源代码中，启用或禁用“从默认初始化函数调用”的功能
	打开 <b>Functional group properties（函数组属性）</b> 窗口

**注意：**

红色/橙色背景表示配置中的错误/警告。

2.4.3.1 函数组属性

在 **Functional Group Properties（函数组属性）** 窗口中，可以为函数和代码生成配置多个选项。每个设置都适用于所选的函数。您可以指定生成函数的名称，选择运行生成函数的处理器核心（仅适用于多核处理器），或填写函数的说明（该说明在 C 文件中生成）。您还可以根据需要添加、复制和删除函数组。

除了名称和说明外，您还可以为所选的函数组设置参数。

每个配置工具都有特定的函数组属性：

引脚工具：

- **设置自定义#define 前缀** - 如果设置了此属性，在 `pin_mux.h` 中生成的宏将使用特定的自定义前缀。如果未设置此属性，则使用函数组名称作为前缀。
- **前缀** - 自定义前缀字符串。如果为空，就不加前缀。
- **时钟门控启用** - 如果启用该属性，则在生成的代码中启用时钟门控。访问外设需要时钟门控，因此请在其他地方启用时钟门控。
- **内核**（仅适用于多核处理器） - 选择执行此函数的内核。
- **全引脚初始化** - 如果设置此属性，也会在生成的函数中完全初始化引脚的所有特征，即使此属性与处理器复位后的状态匹配。如果未设置，则此值可能为“未指定”或“复位(...)”，这表示不会生成代码，而是期望复位后的状态。
- **去初始化函数** - 如果设置此功能，则会产生一个其他函数，将该函数组中的所有引脚和外设信号设置为其复位后状态。默认情况下，新函数的后缀为 `_deinit`。
- **设置自定义去初始化函数名称** - 允许自定义去初始化函数的名称。

时钟工具：

- **设置自定义#define 前缀** - 如果设置了此属性，在 pin\_mux.h 中定义的宏将使用自定义前缀。如果未设置，则使用函数组名称作为前缀。
- **前缀** - 自定义前缀字符串。如果为空，就不加前缀。
- **其他设置** - 处理器特定的设置因处理器而不同。请查看工具提示，获取详细信息。

外设工具：

- **前缀** - 用于与生成代码中所用函数组相关的标识符、常量和函数。如果未指定，则不使用任何前缀。

TEE 工具：

- **设置自定义 #define 前缀** - 如果勾选了此项，则在生成代码中定义的宏会加上自定义前缀。如果未勾选，就会使用函数组的名称作为前缀。

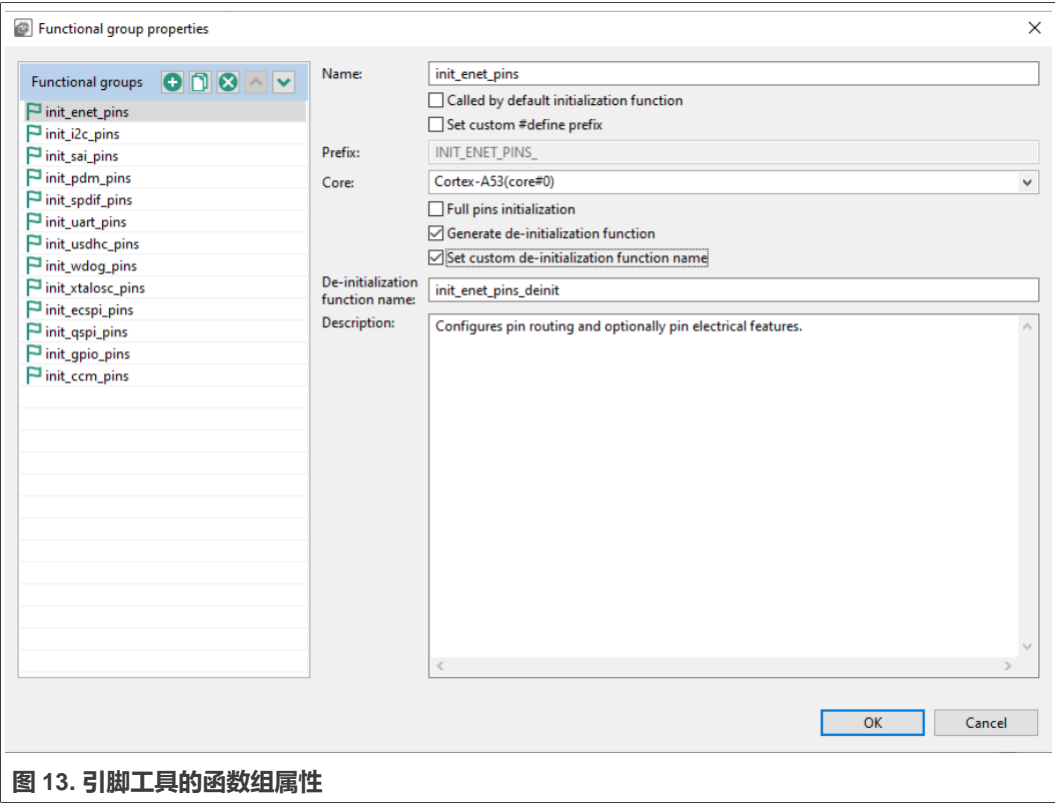


图 13. 引脚工具的函数组属性


2.4.4 撤销/恢复操作

您可以使用 **Toolbar (工具栏)** 上的“撤销/恢复”按钮来撤销操作，也可以通过 **Menu bar (菜单栏)** 中的“Edit (编辑)”菜单执行这些操作。

表 9. 撤销/恢复操作

图标	说明
	取消之前的操作

表 9. 撤销/恢复操作（续）

图标	说明
	取消之前的撤销操作

2.5 首选项

要在 **Preferences（首选项）** 对话框中配置首选项，请从 **Menu bar（菜单栏）** 中选择 **Edit>Preferences**。

**注意：**您可以通过选择对话框右下角的 **Restore Defaults（恢复默认值）** 将设置恢复为默认值。

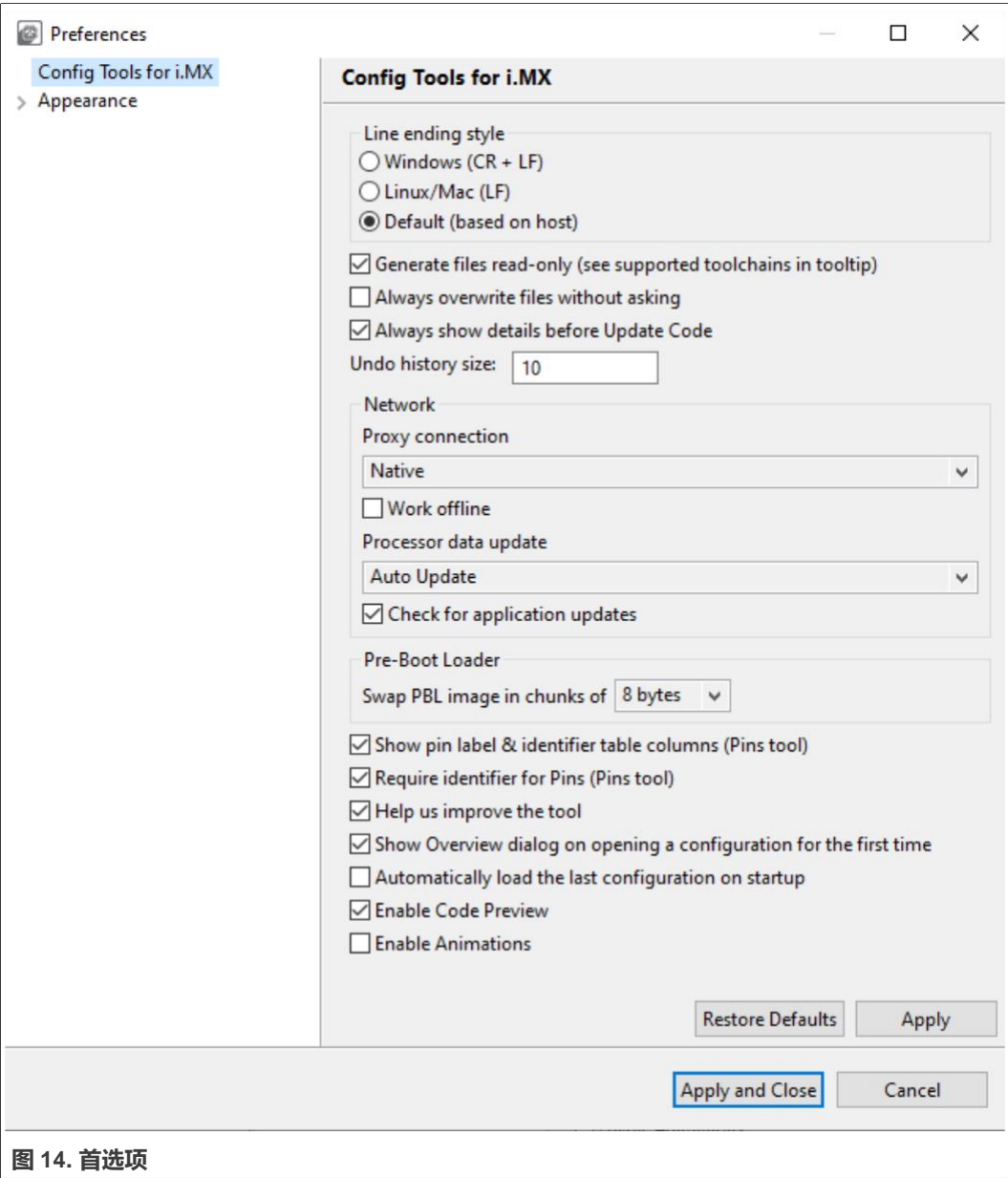


图 14. 首选项

有多种设置可用。

表 10. 首选项

项目	说明
换行符风格	选择 <b>Windows (CR + LF)</b> 、 <b>(LF)</b> 或 <b>默认 (基于主机)</b> 。
生成源文件夹	构建时，自动创建包含源文件的文件夹。
如果没有 yaml，创建空配置	即使没有 yaml，也会生成配置。
总是覆盖文件，无需询问	自动更新现有文件，无需提示。
更新代码前始终显示详细信息	在项目更新前查看更改。
撤销历史大小	输入可撤销的最多步骤数。输入 0 可禁用此功能。
帮助我们改进工具	向恩智浦发送设备配置和工具使用信息。这些信息有助于解决问题和优化工具。
启用代码预览	控制代码生成的方式。启用此首选项后，每次更改配置后都会自动生成代码，并代码预览也会随之更新。禁用该首选项时，代码生成会暂停，代码预览窗口中会出现警告信息，可使用以下任意一种方式手动触发该操作： <ul style="list-style-type: none"><li>单击代码预览窗口中警告信息里高亮显示的“生成代码”链接。</li><li>单击工具栏中的<b>更新代码</b>按钮，在更新代码前先生成代码。</li></ul>
组件模板文件夹	组件模板文件夹的路径。留空则使用默认路径。默认路径是配置工具数据下的 component_templates 文件夹。
允许组件迁移	当打开与工具链项目相关的配置时，外设工具自动检查配置组件是否与项目匹配，如果不匹配则建议迁移。
启用动画	启用户界面中的动画效果，例如更流畅的滚动或展开下拉菜单。

2.6 配置首选项

配置首选项是保存在配置存储文件（MEX）中的通用首选项。

要设置与配置相关的首选项，请从主菜单中选择 **Edit > Configuration Preferences**。

可使用以下首选项：

- **生成 YAML** – 选择在 C 源文件中生成 YAML。
- **在头文件中生成扩展信息** – 选择此选项可在头文件中生成扩展信息。对于在早期 MCUXpresso 版本中创建的项目，默认情况下会勾选此选项。
- **仅为与复位后状态不同的寄存器生成代码** – 选择此项可为与复位后状态不同的寄存器生成代码。



- 自定义源文件版权标头
- 覆盖输出路径

**警告：**源文件不包含 YAML 代码时，就无法导入。

2.7 问题视图

**Problems (问题)** 视图显示各个工具和工具间相互依赖的问题。

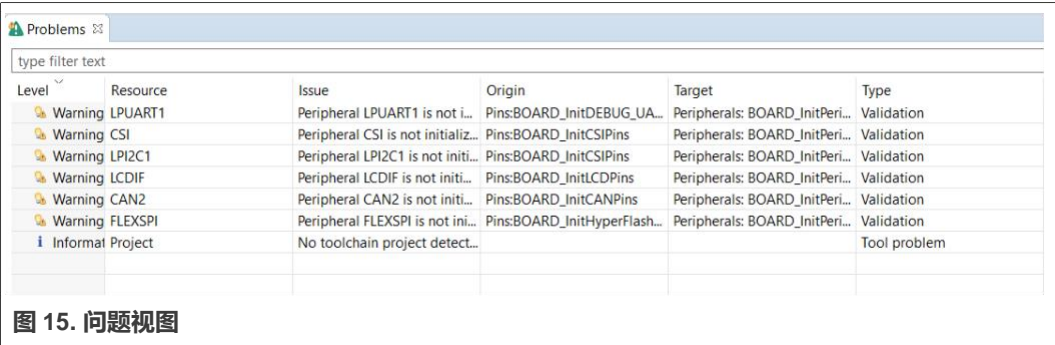


图 15. 问题视图

要打开 **Problems (问题)** 视图，请单击 **Toolbar (工具栏)** 中的 **Show Problems view (显示问题视图)** 按钮，或从 **Menu bar (菜单栏)** 中选择 **Views > Problems**。

问题表包含以下信息：

表 11. 问题视图

项目	说明
级别	问题的严重程度：信息、警告或错误。
资源	与问题相关的资源，如信号名称、时钟信号。
问题	问题描述。
来源	依赖关系的来源信息。
目标	处理依赖关系及解决方案的工具。
类型	问题类型。可以是工具间依赖关系的验证检查，也可以是单个工具问题。

每个问题都由一个右键单击表格行即可访问的弹出菜单。使用该菜单可查看问题的信息或在适用的情况下应用快速修复。您还可以通过右键单击行并选择**复制**，或使用 **Ctrl+C** 快捷键复制行，以供后续使用。您可以使用 **Ctrl+左击** 快捷键来选择多行。

**注意：**只有用“灯泡”图标高亮显示的问题才能进行快速修复。

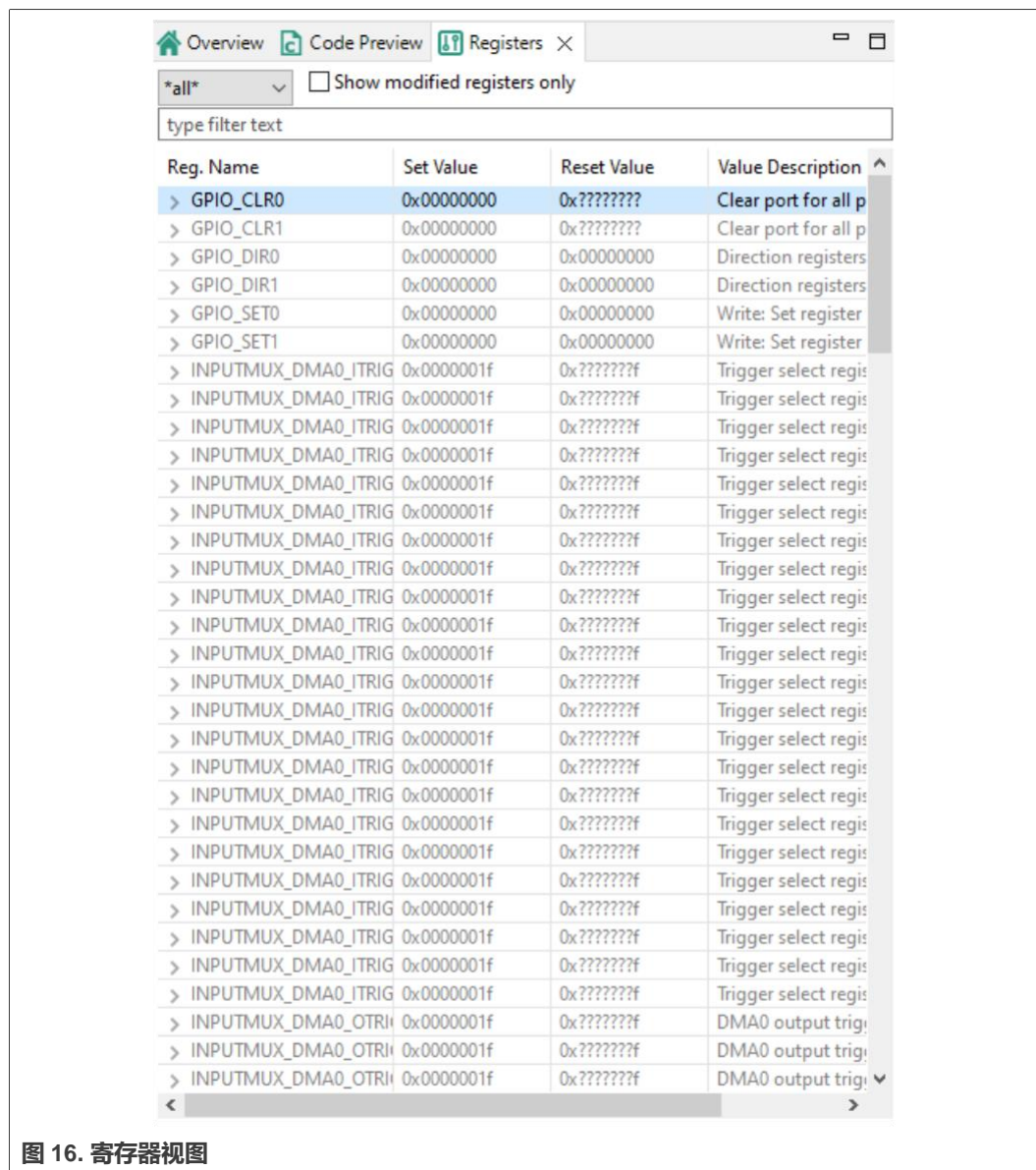
过滤器按钮在**问题视图**视图栏的右侧。

表 12. 过滤器按钮

按钮	说明
	在 <b>问题视图</b> 中过滤信息。如果勾选，仅显示当前工具的问题。如需了解详情，请参阅“ <a href="#">配置首选项</a> ”章节。

## 2.8 寄存器视图

**Registers (寄存器)** 视图列出了工具模型处理的寄存器。您可以查看与当前配置设置相对应的处理器寄存器状态，以及复位后寄存器的默认状态。寄存器的值以十六进制和二进制形式显示。如果寄存器（或位）的值未定义，则显示“?”，而不是数值。



**寄存器视图**包含多个项目。

### 表 13. 寄存器

项目	说明
外设过滤器下拉列表	仅列出所选外设的寄存器。选择 <b>all (全部)</b> ，列出全部外设的寄存器。
仅显示已修改的寄存器复选框	隐藏处于复位后状态或未配置的寄存器。

表 13. 寄存器 (续)

项目	说明
文本过滤器	按文本过滤内容。

下表列出了 **Registers (寄存器)** 视图中使用的颜色高亮显示样式。

表 14. 颜色代码

颜色	说明
黄色背景	表示这个位域已受到工具中最后一次修改的影响。
灰色文字	表示这个位域未被编辑，且值为复位后的值。
黑色文字	表示工具所修改的位域。

2.9 日志视图

**Log (日志)** 视图显示了用户在使用工具时的相关信息。**日志**视图可以按时间顺序显示所有工具多达 100 条记录。

每条日志记录包含时间戳、产生该记录的工具名称、严重性级别和具体消息。如果没有指定工具名称，说明该记录由共享功能产生。

您可以使用组合框过滤**日志**视图的内容，以便仅显示特定工具和/或严重级别信息。不同工具中的过滤器可分别设置。

使用清除按钮可清除缓存的日志记录。这会影响所有工具的**日志**视图。

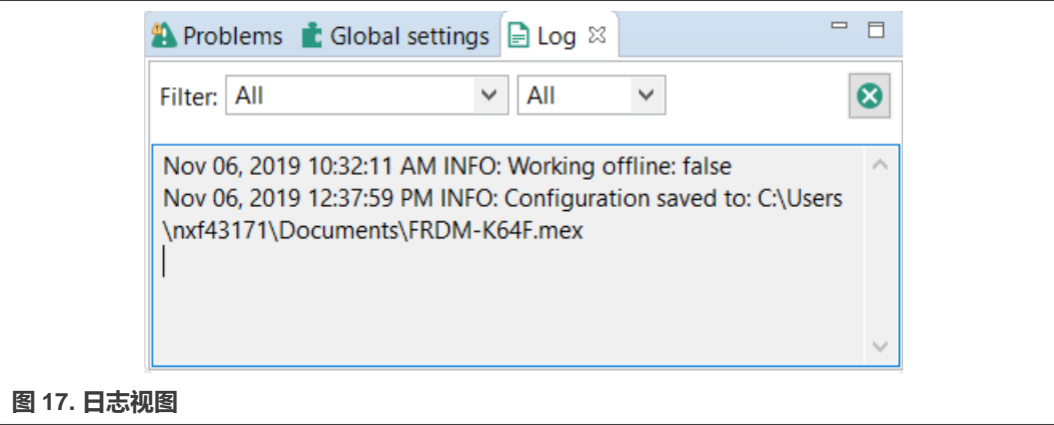


图 17. 日志视图

3 引脚工具

**引脚**工具是配置器件引脚的简便工具。**引脚**工具软件可帮助创建、检查、修改和调整引脚配置和器件复用的任何元素。

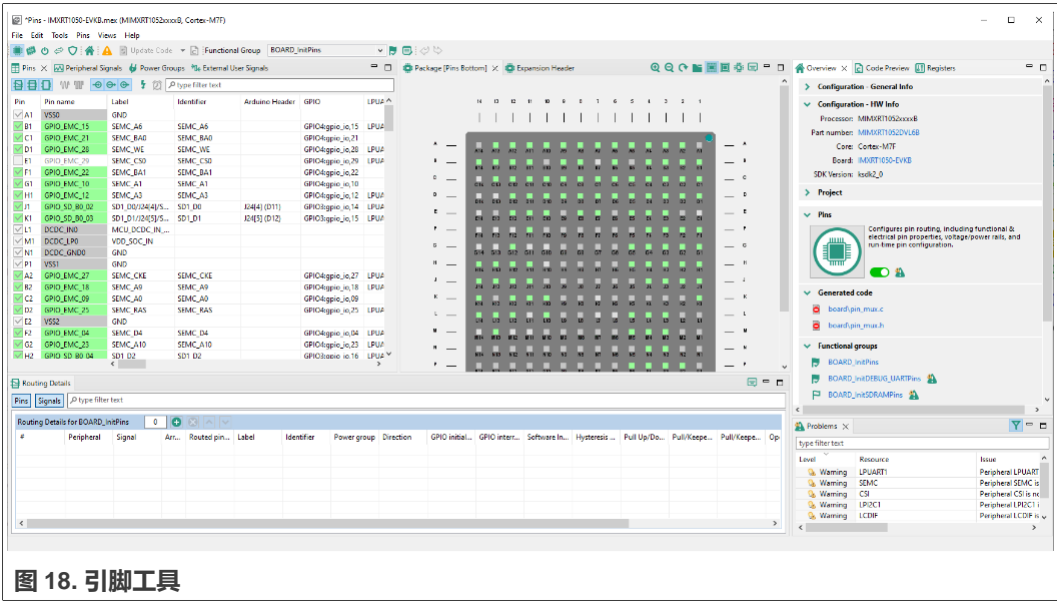


图 18. 引脚工具

### 3.1 引脚路由原理

引脚工具可将外设信号路由到引脚或内部信号。

路由配置可在以下视图中完成：

- Pins (引脚)
- Peripheral Signals (外设信号)
- Package (封装)
- Routing Details (路由详情)

下面两个小节介绍了确定路由路径的两种方法。

#### 3.1.1 从选择引脚/内部信号开始

您可以在路由详情视图中选择引脚或内部信号。

1. 选择引脚/内部信号 (路由的引脚/信号)。
2. 选择一个可用的外设。
3. 为所选外设选择一个可用信号。

路由详情视图中外设列中的项目有以下符号：

- 感叹号和默认颜色文字表示选择该项目可能导致寄存器冲突或该项目不支持所选的信号。
- 感叹号和灰色文字表示该项目无法路由到选定的引脚/内部信号。该项目可用于使用相同信号的不同引脚/内部信号。

**注意：**在引脚视图和封装视图中，只能配置引脚，不能配置内部信号。

3.1.2 外设信号的路由

外设信号表示片上外设输入或输出，可以通过外设间的交叉开关连接到其他片上外设或引脚。您可以在**路由详情**视图中配置这种连接。

有三种外设信号路由可供选择：

- 1. 将信号从一个内部外设（A）的输出端路由至另一个内部外设（B）的输入端

信号从一个内部外设（A）的输出端引向另一个内部外设（B）的输入节点。也就是说，信号从 A 连到 B（A > B）。要以这种方式配置信号，请按照以下步骤操作（以 PWM 触发 ADC（PWM > ADC）为例）：

- a. 在**路由详情**视图中添加一行。
- b. 从**外设**列的下拉列表中选择外设 B。

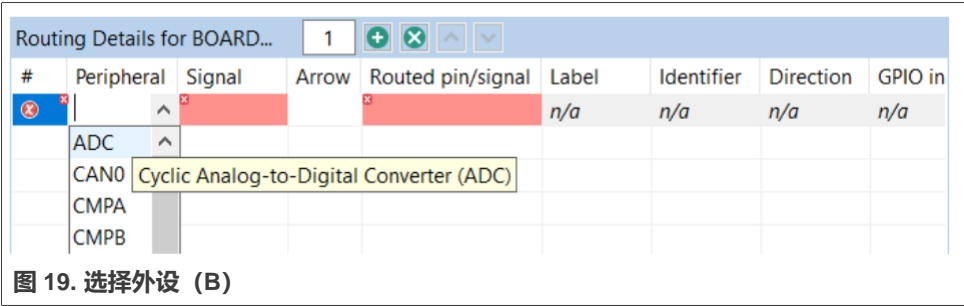


图 19. 选择外设（B）

- c. 从**信号**列的下拉列表中选择外设 B 的输入节点。

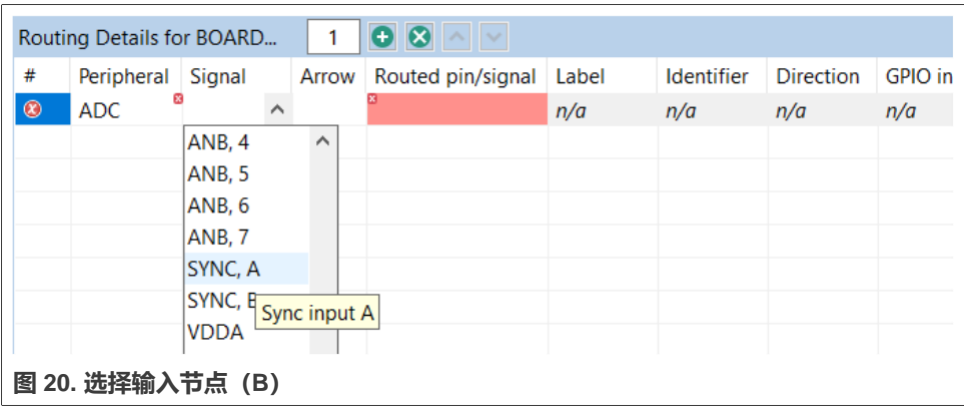
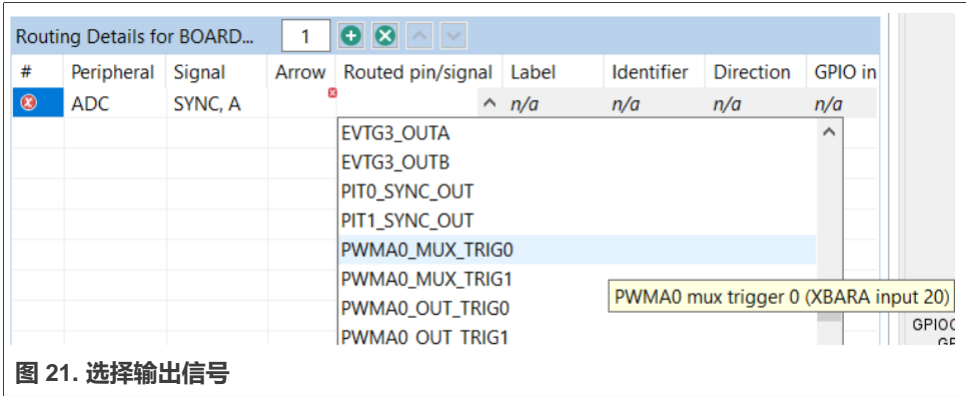
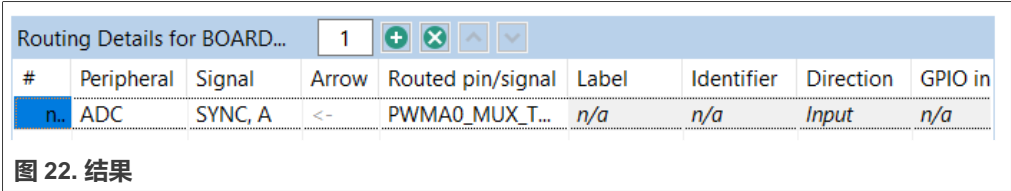


图 20. 选择输入节点（B）

- d. 从**路由引脚/信号**列的下拉列表中选择外设 A 的输出信号。



配置完成后，该行显示如下：



**注意：**必须选择信号连到的 ADC 外设（ADC 的输入端）。由于引脚工具的限制，PWM 外设输出信号不能在路由视图中列出。注意箭头列中的信号方向。

2. 通过外设间的交叉开关将信号从封装上的引脚路由到内部外设输入信号

**注意：**仅当存在交叉开关时才能这样做。

信号从封装上的引脚（XB\_IN）通过外设间交叉开关连接到内部外设（B）输入节点。也就是说，信号从 XB\_IN 连到 B（XB\_IN > B）。要以这种方式配置信号，请按照以下步骤操作（使用 XB\_IN6 将引脚 55 路由至 EVTG0 输入 A（XB\_IN6 > EVTG0）为例）：

- a. 在路由详情视图中添加一行。
- b. 从外设列的下拉列表中选择外设 B。

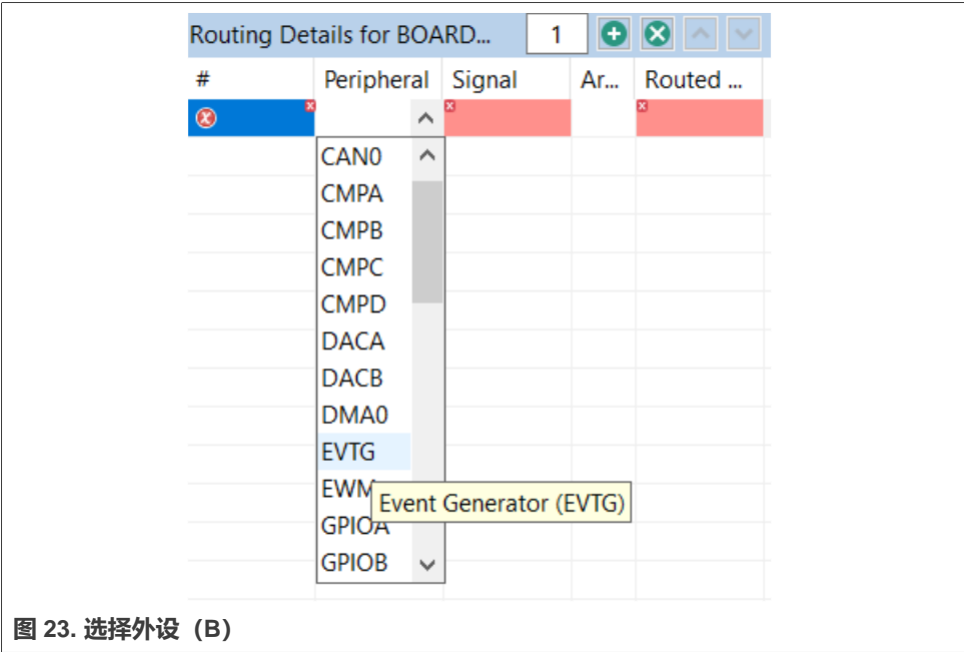


图 23. 选择外设 (B)

c. 从信号列的下拉列表中选择外设 B 的输入节点。

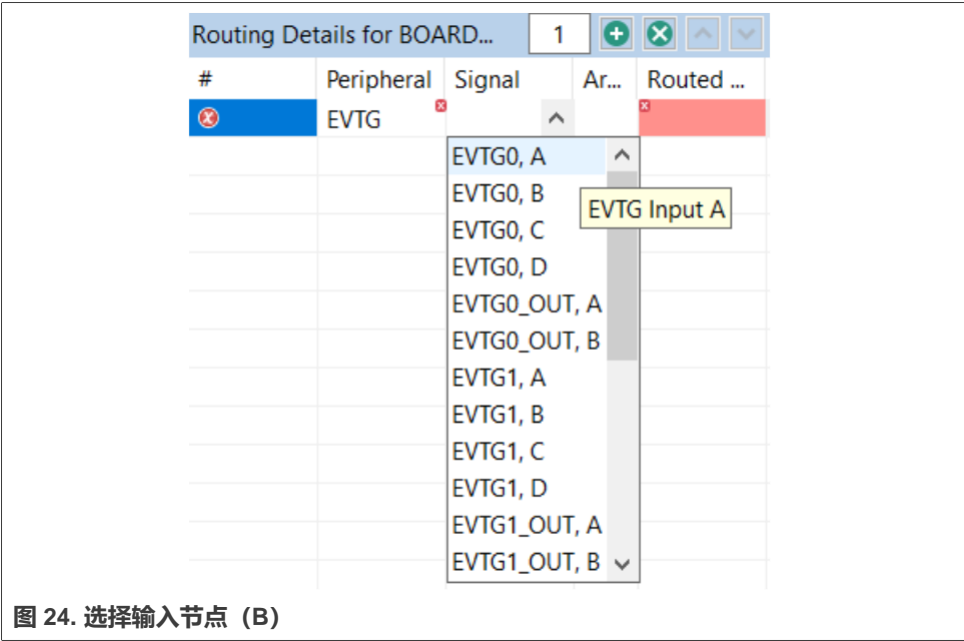


图 24. 选择输入节点 (B)

d. 从路由引脚/信号列的下拉列表中选择 XB\_IN 引脚。



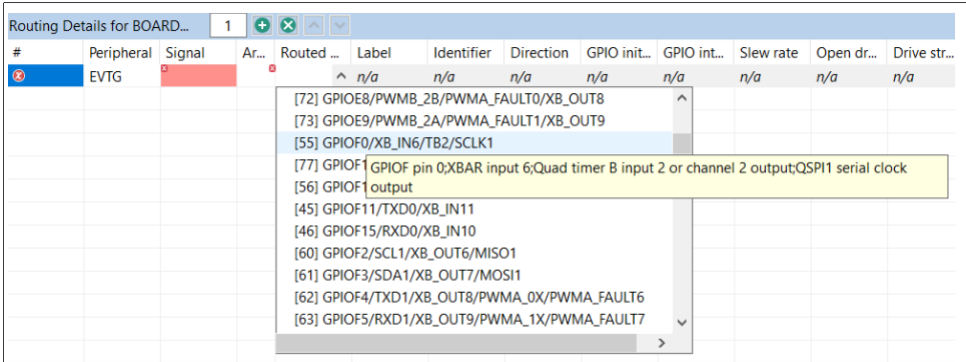


图 25. 选择引脚

配置完成后，该行显示如下：

Routing Details for BOARD... 1				
#	Peripheral	Signal	Ar...	Routed pin/signal
55	EVTG	EVTG0, A	<-	[55] XB_IN6

图 26. 结果

**注意：**在本例中，GPIOF0 与 XB\_IN6、QTimerB 通道 2 输出/输入和 QSPI1 SCLK 信号复用同一个引脚。在这种情况下，因为只有 XB\_IN6 可以路由到 EVTG0 输入 A 上，所以工具会自动选择 XB\_IN6 作为引脚信号。

3. 通过外设间的交叉开关将信号从内部外设（A）输出路由到引脚

**注意：**仅当存在交叉开关时才能这样做。

信号从内部外设（A）输出端出来，通过封装上的外设间交叉开关连到引脚(XB\_OUT)。也就是说，信号从 A 连到 XB\_OUT (A > XB\_OUT)。要以这种方式配置信号，请按照以下步骤操作（以使用 XB\_OUT4 将 EVTG0 输出路由到引脚 87 为例）：

- a. 在路由详情视图添加一行。
- b. 从外设列的下拉列表中选择外设 A。



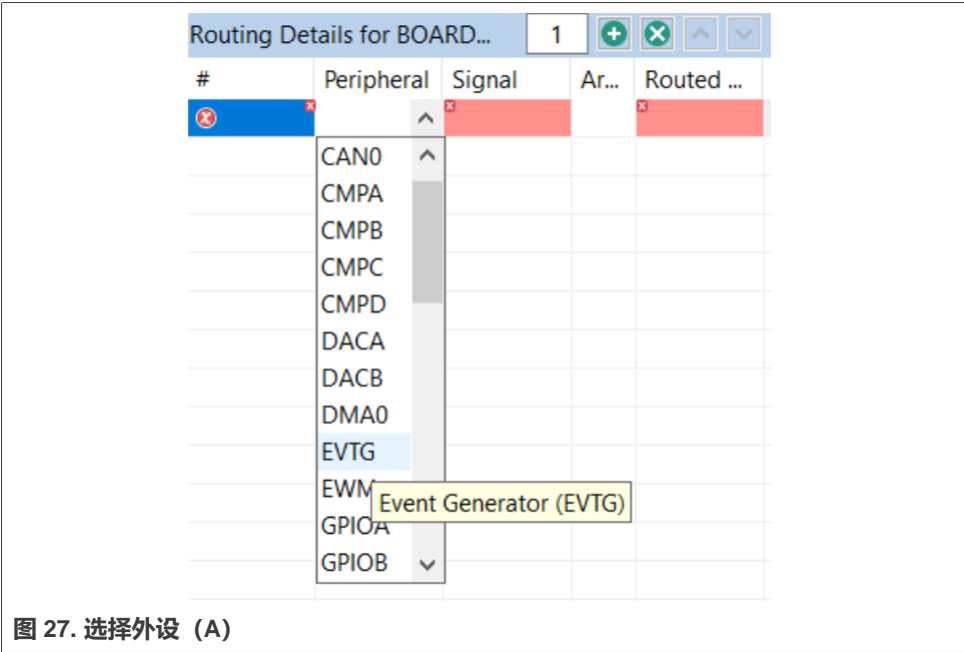


图 27. 选择外设 (A)

c. 从信号列的下拉列表中选择外设 A 的输入节点。

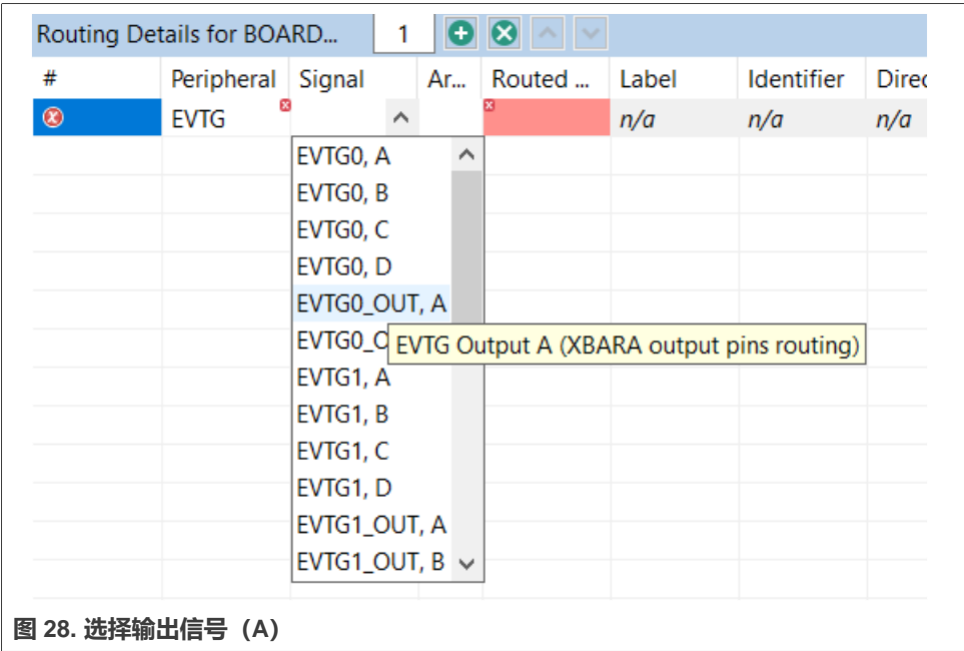


图 28. 选择输出信号 (A)

d. 从路由至列的下拉列表中选择 XB\_OUT 引脚。

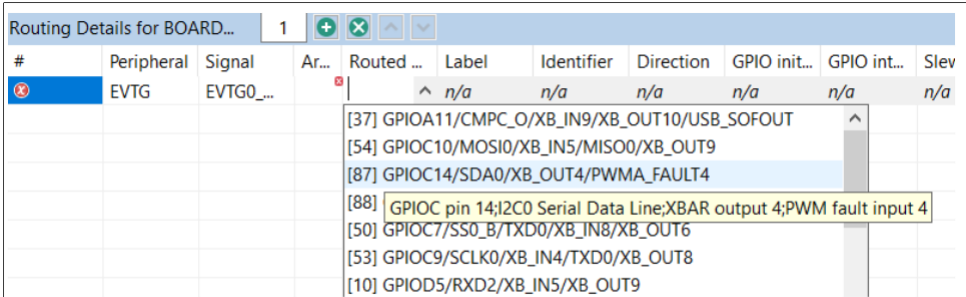


图 29. 选择引脚

配置完成后，该行显示如下：



图 30. 结果

**注意：**在本例中，GPIOC14 与 XB\_OUT4、I2C0 的 SDA 和 eFlexPWMA 的 fault4 复用同一个引脚。在这种情况下，工具会自动将引脚 GPIOC14（引脚 87）配置为 XB\_OUT4 输出信号，因为只有 XB\_OUT4 能连到 EVTGO 输出 A 上。

3.2 使用示例

本节介绍了创建引脚配置示例的步骤，这个配置可在用户项目中应用。  
在本例中，MCIMX6Q-SDB-REV-B 板上 3 个引脚（UART4\_TX、UART4\_RX 和 GPIO\_2）的路由需要重新配置，以匹配电路板设计更改（例如客户修改），该设计更改使用了 UART4 替代 UART3，因此这些引脚必须重新路由并且/或者还需要调整红色 LED 引脚的电气属性，因此该工具生成的文件都是使用了上述修改过的默认电路板配置。

- 1. 为 MCIMX6Q-SDB-REV-B 板新建配置。

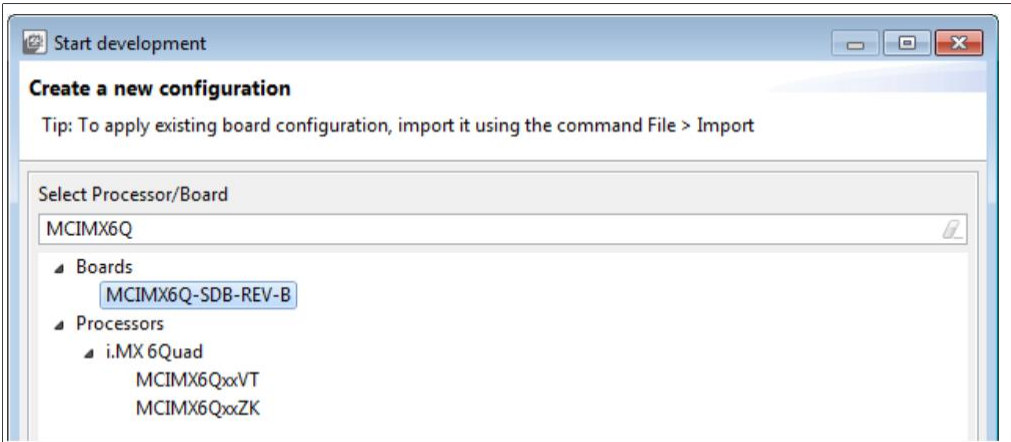
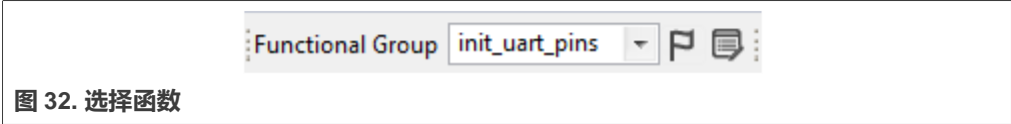
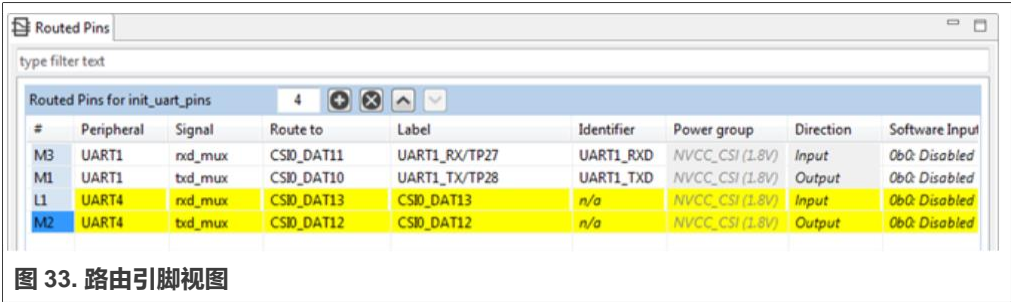


图 31. 新建配置

2. 打开工具后，在工具栏下拉菜单中选择函数组 “init\_uart\_pins”，显示其路由引脚。



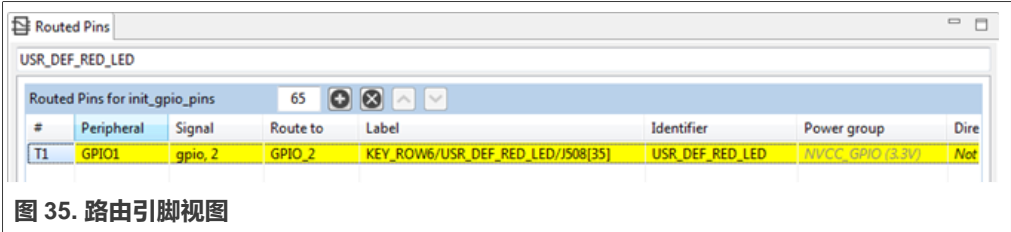
3. 要将原来的 RX/TX 引脚路由从 UART3 更改为 UART4，你必须在路由引脚视图中的外设和路由至列中更改 init\_uart\_pins 选项的配置，并更改引脚配置，将 RX/TX 引脚重新路由到 UART4 的不同引脚，以符合修改后的电路板设计要求。



4. 您还可以在表格右侧的特定属性列中调整这些引脚的电气属性配置。
5. 要改变原来路由到 GPIO\_2 引脚的红色 LED 引脚的配置，必须选择函数组 ‘init\_gpio\_pins’ 。



6. 在路由引脚视图使用过滤功能搜索 “USR\_DEF\_RED\_LED” 的配置，就能看到简化的表格内容，从而轻松修改当前 GPIO\_2 引脚的路由选择。



7. 您可以根据修改后的设计需要，调整电气属性或把引脚重新连接到另一个不同引脚上。
8. 在右侧的代码预览视图中，引脚工具会自动生成 imx6q-board.dtsi、pin\_mux.c 和 pin\_mux.h 源代码。

Code Preview

imx6q-board.dtsi@Cortex-A9 (Core #0)pin\_mux.c@Cortex-A9 (Core #0)pin\_mux.h@Cortex-A9 (Core #0)

```
/*
 * TEXT BELOW IS USED AS SETTING FOR TOOLS *****
 !!GlobalInfo
 product: Pins v4.1
 processor: MCIMX6QxxVT
 package_id: MCIMX6Q7CVT08AD
 mcu_data: i_mx_1_0
 processor_version: 3.0.0
 board: MCIMX6Q-SDB-REV-B
 power_domains: {PCIE_VPH: '2.5', SATA_VPH: '2.5', NVCC_SD3: '3.3', NVCC_NANDF: '3.3'
 NVCC_EIM0: '3.3', NVCC_MIPi: '2.5', NVCC_EIM1: '3.3', HDMI_VPH: '2.5', NVCC_EIM2:
 NVCC_ENET: '3.3', NVCC_DRAM: '1.5'}
 * BE CAREFUL MODIFYING THIS COMMENT - IT IS YAML SETTINGS FOR TOOLS *****
 */

/dts-v1/;

#include "skeleton.dtsi"
#include "imx6q-pinfunc.h"

/ {
    model = "Freescale i.MX 6Quad User Board";
    compatible = "fsl,imx6q-board", "fsl,imx6q";

    soc {
        #address-cells = <1>;
        #size-cells = <1>;

        iomuxc: iomuxc@020e0000 {
            compatible = "fsl,imx6q-iomuxc";
            reg = <0x020e0000 0x4000>;
        };
    };

    &iomuxc {
        pinctrl-names = "default";
    };
};
```

图 36. 生成的源代码

9. 现在，可以将源代码内容复制粘贴到您的应用或 IDE 中。或者，也可以导出生成的文件。要导出文件，请单击 **Code Preview（代码预览）** 视图右上角的 **Export（导出）** 按钮，或选择菜单 **File > Export**，在 **Export** 对话框中展开引脚工具的树形图控件，并选择 **Export Source Files（导出源文件）** 选项。

**注意：**工具生成的面向电路板的设备树（DTS）DTSI 文件只是一个片段，并不是完整的设备树文件内容。只有基本的设备树元素、初始骨架和处理器特定的"pinfunc.h"，以及 fsl,pins = <...>内容定义的函数组，这些定义根据工具 UI 定义的引脚路由和功能配置提供初始 IOMUXC 模块配置。为了应用工具生成的引脚配置，必须手动把这些内容与现有的 Linux BSP 设备树文件合并。该工具也不会生成或导出处理器特定的"pinfunc.h"文件，该文件包含所有支持的 DTS 引脚功能配置宏的定义。这个文件没有被特意集成在工具输出里，因为它是 Linux BSP 支持包的一部分。

3.3 用户界面

引脚工具由多个视图组成。

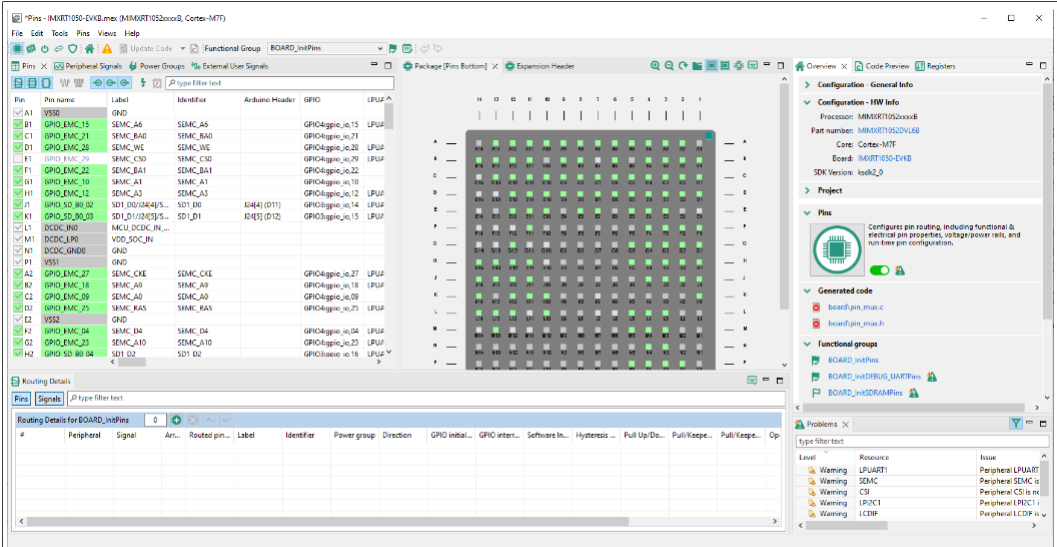


图 37. 引脚工具用户界面

3.3.1 引脚视图

引脚视图以表格形式显示所有引脚。

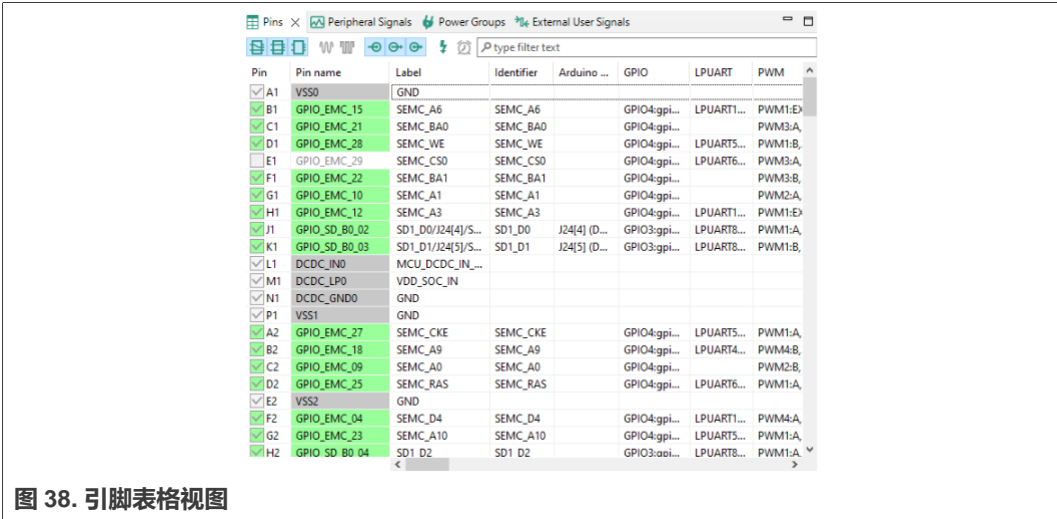


图 38. 引脚表格视图

该视图显示给定设备上所有可用引脚的列表。**Pin name (引脚名称)** 列显示引脚的默认名称，或引脚是否路由。下面的列是可选的。它们是 **Label (标签)**、**Identifier (标识符)** 和 **Expansion header connections (扩展头连接)** (每个扩展头一列)。如果引脚被路由到某一外设，引脚名称也会随之变更为所选外设的对应功能名称。表格的下一列显示了给定外设上的外设和信号以及引脚名称。项目较少的外设累积在最后一列中。

要将引脚路由/取消路由到某个外设，请在 **Pin (引脚)** 列中选择相应单元格。路由引脚以绿色高亮显示。如果存在路由冲突，引脚会以红色高亮显示。

每个路由引脚都会出现在 **Routed pins (路由引脚)** 表中。

当在配置中指定多个功能时，**Pins (引脚)** 视图主要显示所选功能的引脚。不同功能的引脚以浅色透明显示，只有切换到此功能，才能对它们进行配置。

选择一行打开一个下拉列表，它提供以下选项：

- 路由/取消路由引脚。
- 在 **Package (封装)** 视图中高亮显示此引脚。
- 设置引脚的标签和标识符。
- 为引脚添加注释。可以在 **Code Preview (代码预览)** 视图中查看注释。

**提示：** 如果一个引脚可以路由到多个信号，那么它会用省略号 (...) 来表示。选择该单元格打开对话框，从多个可用信号中进行选择。该对话框还显示默认路由的信号。

3.3.2 封装

**Package (封装)** 视图显示处理器的封装信息。处理器封装提供包括资源分配在内的封装概况。

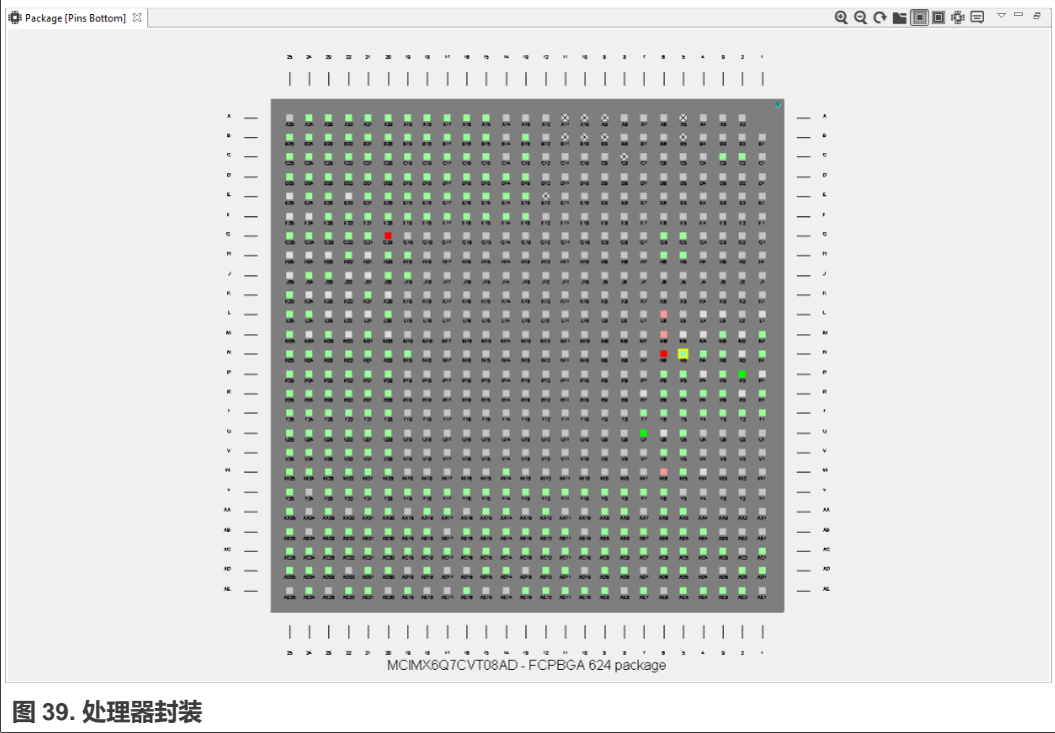


图 39. 处理器封装

这个视图显示了封装概况和引脚位置。中间是外设。

要在 **Pins (引脚)** 和 **Routing Details (路由详情)** 视图中高亮显示引脚/外设配置，请右键单击引脚或外设并选择 “**Highlight (高亮显示)**”。

对于 BGA 封装，请使用 **Resources (资源)** 图标来查看它们。

- 绿色表示路由引脚/外设。
- 灰色表示没有路由的引脚/外设。
- 深灰色表示专用的引脚/外设。它是默认路由的，对生成的代码没有影响。

该视图还显示封装类型和说明（引脚类型和数量）。

工具栏中提供了以下图标：

表 15. 工具栏选项

图标	说明
	放大封装图像。
	缩小封装图像。
	旋转封装图像。
	显示从底部可以看到的引脚。此选项仅适用于 BGA 封装。
	显示从顶部可以看到的引脚。此选项仅适用于 BGA 封装。
	显示资源。此选项仅适用于 BGA 封装。
	切换封装。
	封装图例。
	选择显示为引脚标签的信息。此选项不适用于 BGA 封装。

**注意：**根据所选的处理器封装，并非所有视图都可用。

**Switch package（切换封装）** 图标启动 “Switch package for the Processor（切换处理器封装）” 。



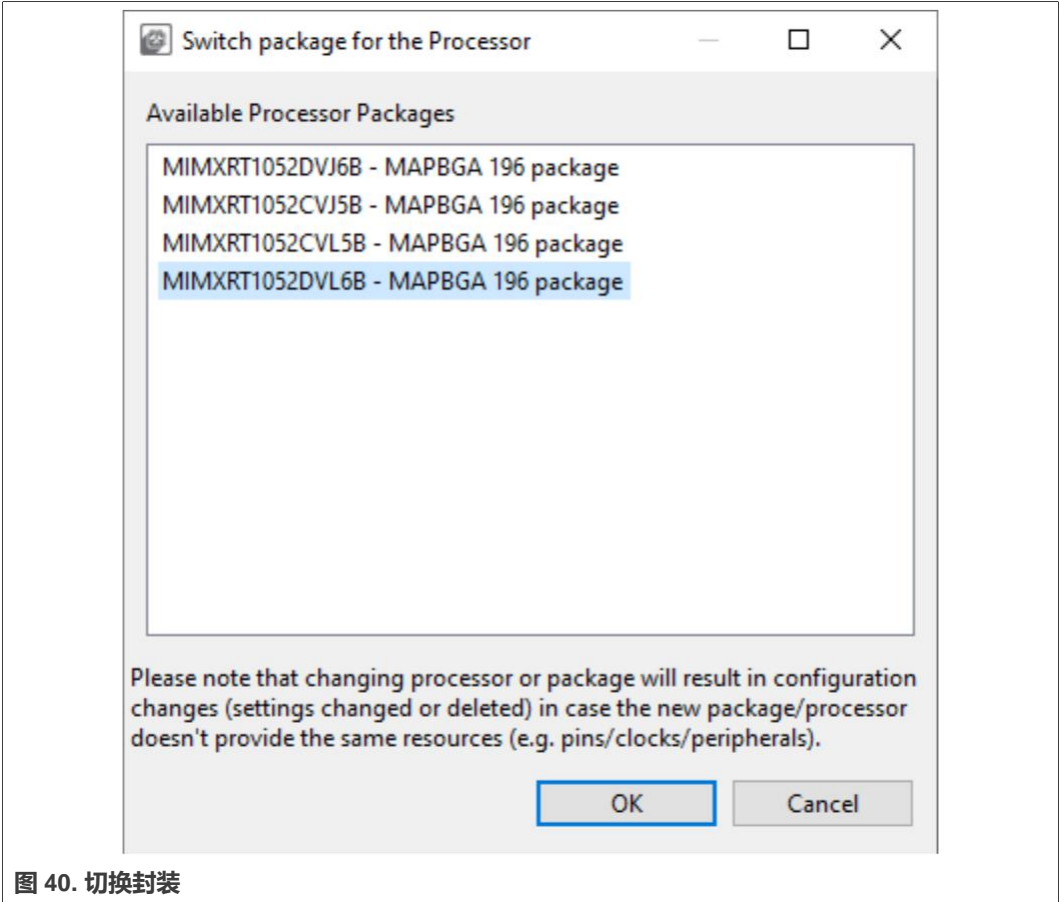


图 40. 切换封装

**Switch package for the Processor（切换处理器封装）** 窗口显示可用处理器封装的列表，显示封装类型和引脚数量。

3.3.3 外设信号视图

**Peripheral Signals（外设信号）** 视图显示了外设及其信号的列表。只有 **Peripheral Signals（外设信号）** 和 **Pins（引脚）** 视图显示带状态的复选框（已分配）。

表 16. 状态代码

颜色代码	状态
	错误
	已配置
	未配置
	警告
	专用：设备是默认路由的，对生成的代码没有影响。



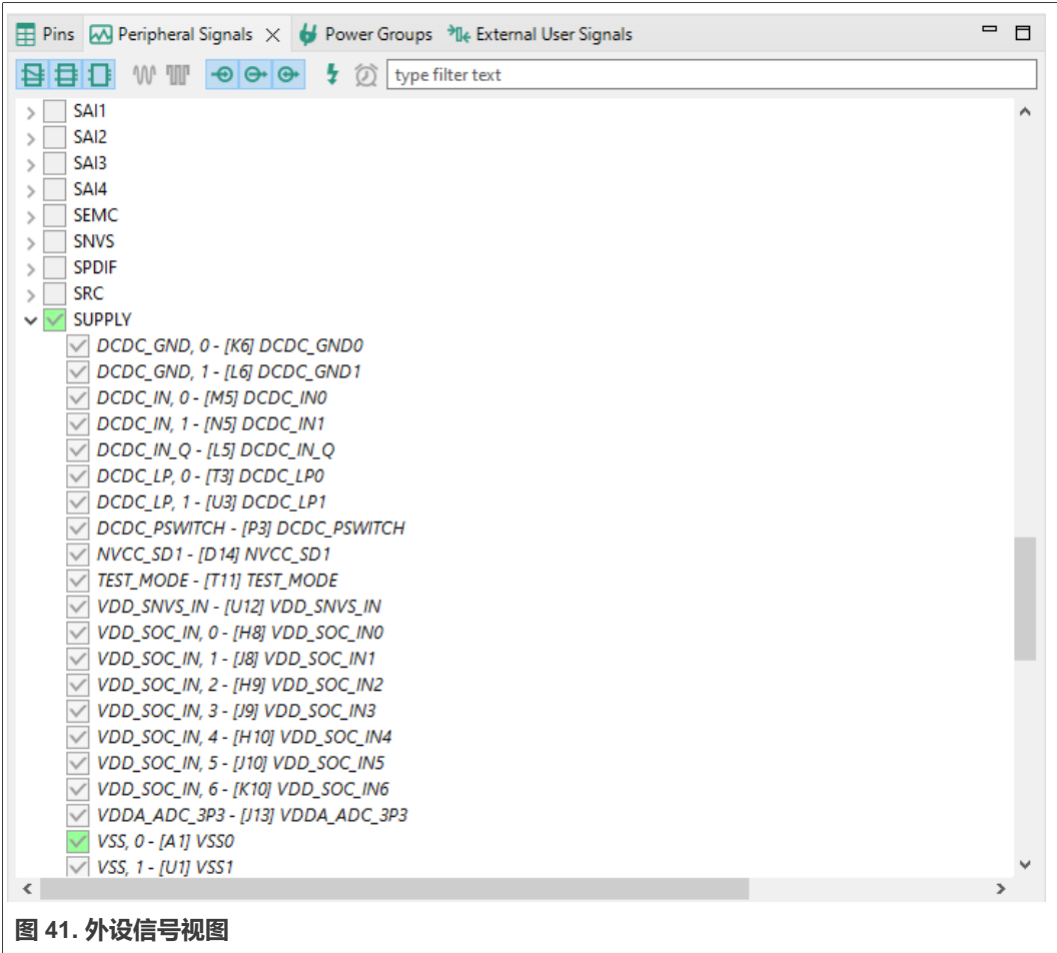


图 41. 外设信号视图

要连接或断开引脚，请勾选或取消勾选相应的复选框。

要在 **Package（封装）** 和 **Routing Details（路由详情）** 视图中高亮显示外设的引脚和路由配置，请右键单击信号并选择 **Highlight（高亮显示）**。

要批量连接或断开引脚，请单击外设并在弹出的 **Select signals（选择信号）** 对话框中勾选或取消勾选相应的选项。

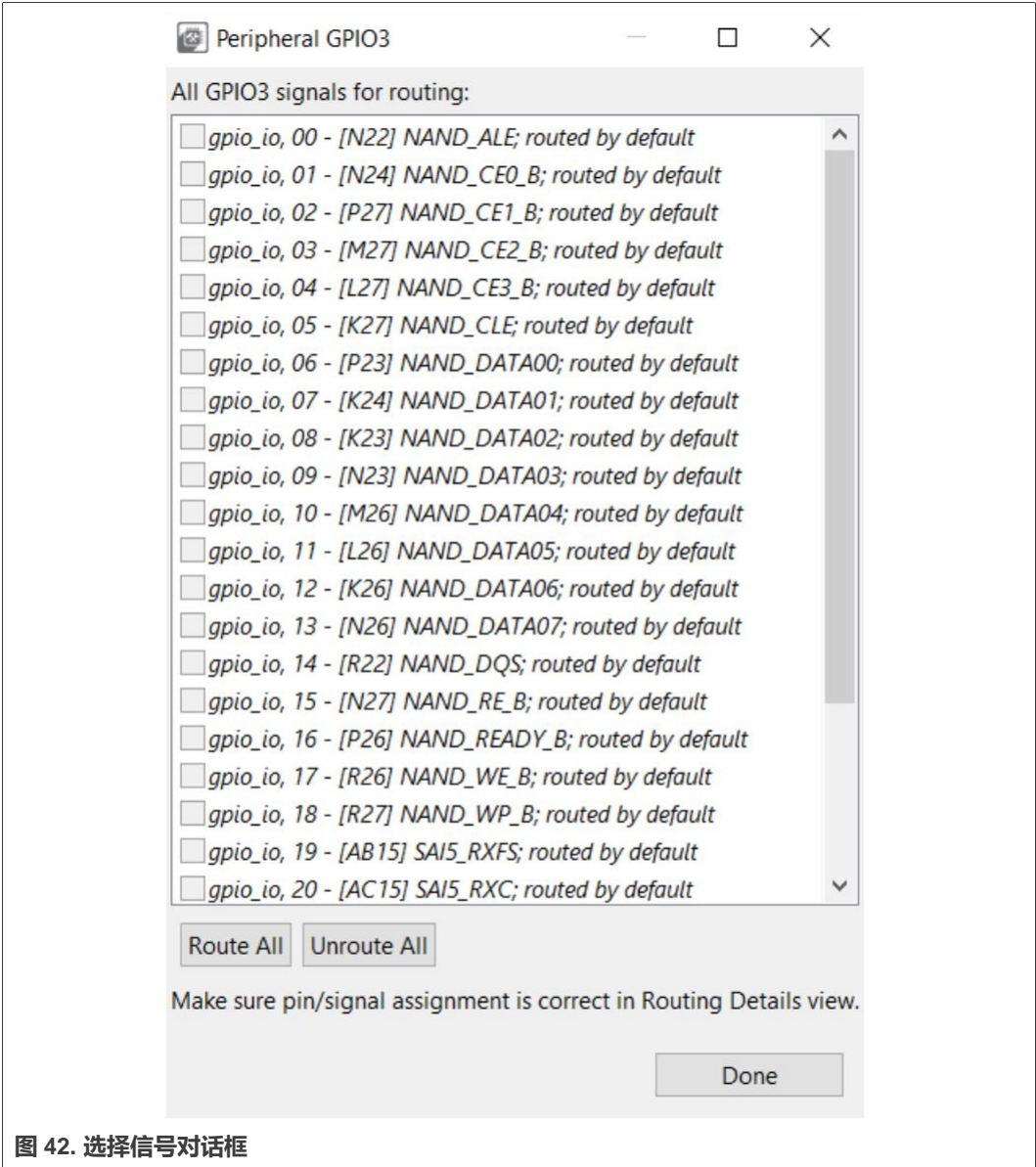


图 42. 选择信号对话框

3.3.3.1 引脚和外设信号视图中的过滤控件

下图展示了 Pins（引脚）和 Peripheral Signals（外设信号）视图中的过滤控件。



图 43. 过滤控件

在表格/树形图中输入任何文本进行搜索。它会搜索包含指定文本的引脚/外设信号。您也可以使用通配符 “\*” 和 “?” 来帮助您过滤所需的结果。使用 “空格” 可以同时搜索多个字符串。

3.3.4 路由详情视图

在 **Routing Details (路由详情)** 视图中，您可以查看和配置路由引脚和内部信号。您还可以配置和查看引脚的电气属性。它显示了配置中的引脚配置情况，其中每个引脚都有对应的信号名称和函数。

**注意：**电气特性仅针对表格中的引脚进行配置。例如，路由引脚

新建配置时，该表格是空的，表示没有配置任何引脚。每一行表示一个引脚的配置情况，如果没有冲突，代码会立即更新。对于电路板/套件，引脚已经路由了。

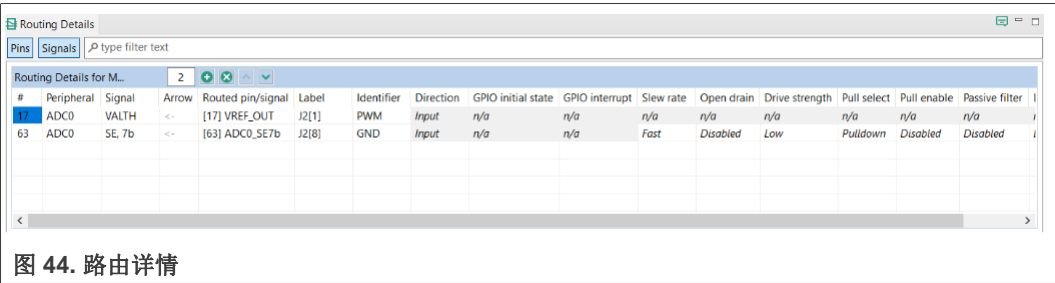


图 44. 路由详情

单击视图工具栏中的 **Add new row (添加新行)** 按钮，在表格末尾添加一行。

依次选择 **Peripheral (外设)**、**Signal (信号)** 以及 **Route to (路由到)** 的引脚来配置引脚/信号。

使用表格右侧的列来设置电气特性。

您也可以使用 **Pins (引脚)** 和 **Peripheral Signals (外设信号)** 视图来路由引脚和外设信号，并在 **Routing Details (路由详情)** 视图中查看/修改配置情况。如果不支持该功能，则显示 n/a。

要高亮显示 **Package (封装)** 和 **Pins (引脚)** 视图中的外设/引脚信息，请右键单击该行并选择 **Highlight (高亮显示)**。

要过滤行，请在视图工具栏的过滤区域中键入文本或搜索短语。

**注意：**输入搜索文本时，系统也会在引脚全名中搜索文本，并显示包含搜索文本的行。

要只显示引脚或信号，请使用视图工具栏中的 **Pins (引脚)** 和 **Signals (信号)** 按钮。

要在表格末尾添加一行，请单击 **Add new row (添加新行)** 按钮。

要删除选中的行，请单击 **Delete the selected row (删除选中的行)** 按钮。

要删除特定的行或在指定位置插入新行，请右键单击并使用下拉列表命令。

要添加指定数量的行，请在该字段中输入行数。

要清空表格，请键入 0。

要更改行的顺序，请使用箭头图标向上或向下移动一行。

要按文本过滤表格条目，请在 **type filter text (键入过滤文本)** 字段中输入文本字符串。

要复制该行，请右键单击该行中的任意单元格，然后选择 **Copy (复制)**。稍后可以通过右键单击该表格并选择 **Paste (粘贴)**，将复制的行粘贴到另一个函数组或配置的 **Routing Details (路由详情)** 视图中。

灰色背景表示只读项目。

斜体字表示该值未配置，它显示了复位后的值，并且不会生成代码，因此配置依赖于复位后的值或从不同功能中设置的值。

提示:

- 单击视图右上角的 **Routing Details Legend (路由详情图例)** 按钮，可显示解释各个字段含义的对话框。

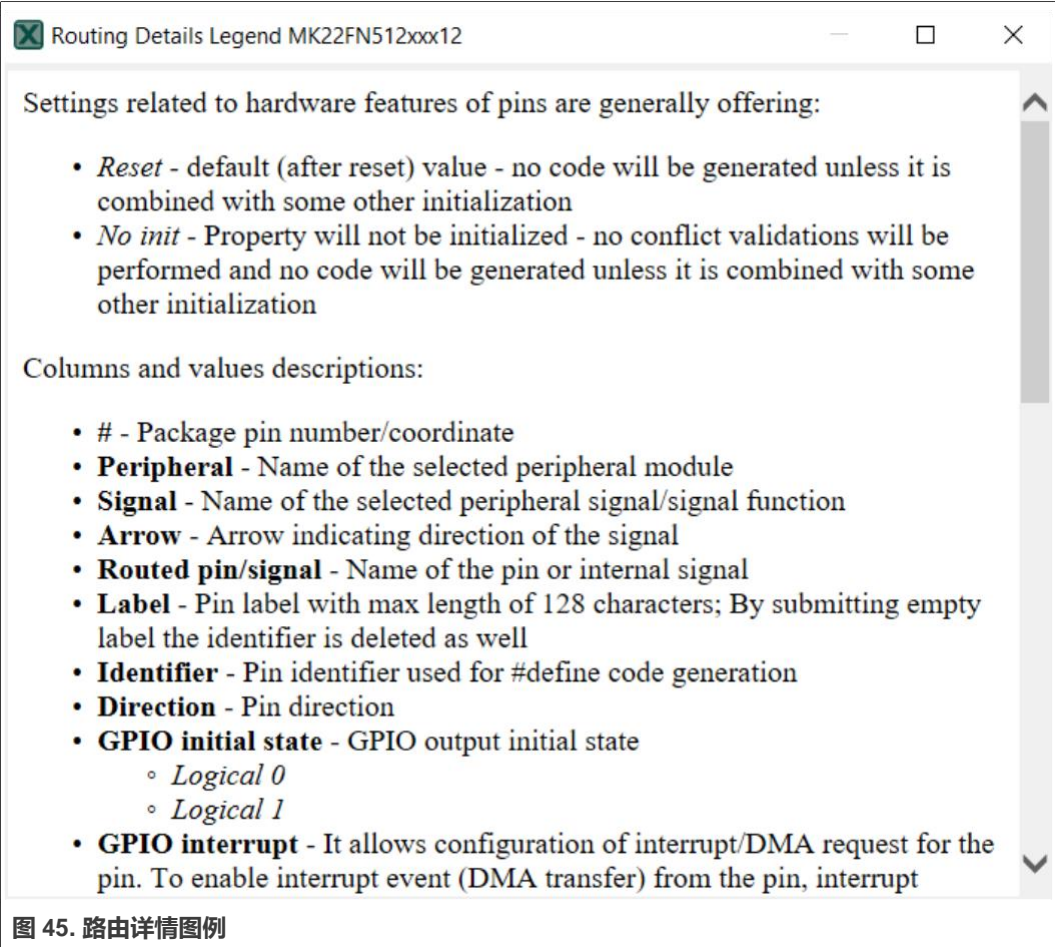


图 45. 路由详情图例

- 斜体字显示的值表示复位后的值。如果在其他功能中配置此值，实际值可能与复位后的值不同。  
使用下拉菜单选择所需的值。
- 如果选择与复位后值相同的值，此工具将始终生成代码来设置这个功能。  
使用下拉菜单中的“复位”值将此值恢复到复位后状态。
- 如果一个项目不支持恢复到复位后的值，则 **Reset (复位)** 菜单不可用。
- 第一行显示 BGA 封装上的引脚编号或坐标。

3.3.4.1 标签和标识符

您可以为任何引脚定义标签，以便在用户界面中更容易识别它们。

电路板和套件已有预设的标签。然而，您也可以可以在 **Pins（引脚）** 和 **Routing Details（路由详情）** 视图中定义所列的引脚标签。

要设置/更新 **Labels（标签）** 和 **Identifier（标识符）** 列的可见性，请从 **Menu bar（菜单栏）** 中选择 **Edit > Preferences（编辑 > 首选项）**，然后勾选 **Show pin label & identifier table columns（Pins tool）（显示引脚标签和标识符表列（引脚工具））** 复选框。

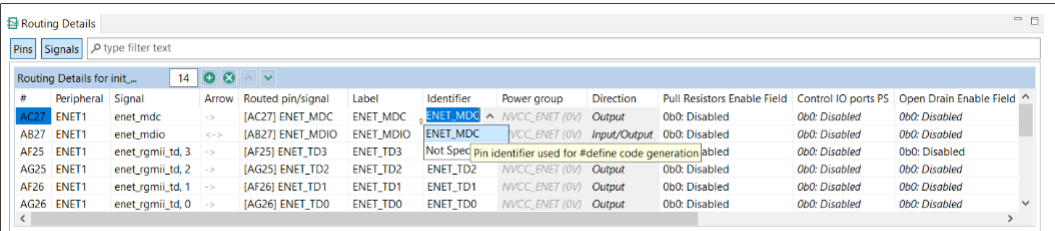


图 46. 标签和标识符

引脚标识符用于在 pin\_mux.h 文件中生成 #define。不过，这是一个可选参数。如果没有定义该参数，则不会生成 #define 代码。此外，您还可以使用 “;” 字符作为分隔符，定义多个标识符。您也可以直接在 **Routing Details（路由详情）** 视图中的 **Identifier（标识符）** 列表单元格中输入标识符来设置标识符。

P1	CSIO_P1A1CK	CSIO_P1A1CK	
R1	GPIO_17	SPDIF_OUT/PCIE_RST_B/J504[40]	PCIE_RST_B
T1	GPIO_2	KEY_ROW6/USR_DEF_RED_LED/J508[35]	USR_DEF_RED_LED;KEY_ROW
U1	LVDS0_TX0_P	LVDS0_TX0_P	
V1	LVDS0_TX2_P	LVDS0_TX2_P	

图 47. 引脚标识符

在这种情况下，如果引脚已经路由，则可以从下拉列表选择一个值作为标识符。请参阅 **Routing Details（路由详情）** 视图中的 **Identifier（标识符）** 列。

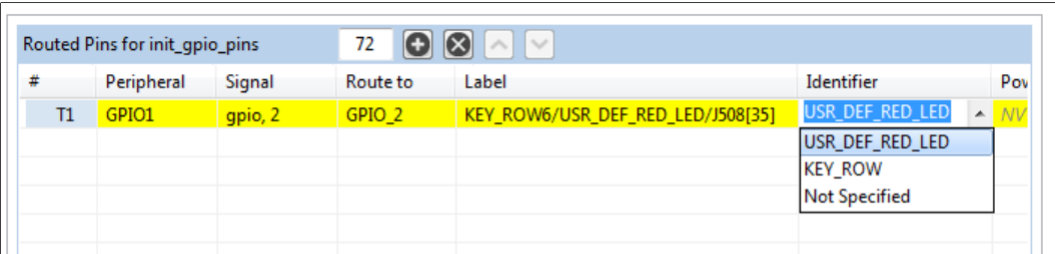


图 48. 路由详情表格中的标识符

系统会检查在 pin\_mux.h 文件中生成的定义是否有重复。重复的定义会在 **标识符** 列中显示为错误。请参见 [标识符错误](#)。



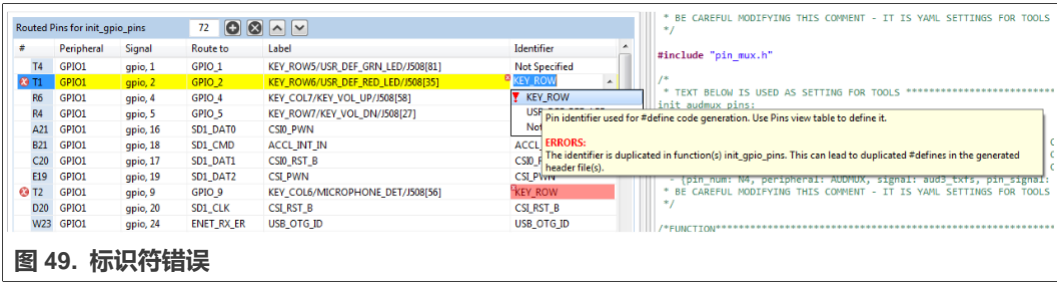


图 49. 标识符错误

您也可以从 **Routing Details（路由详情）** 视图选择要在指定路由中使用的引脚。不过，标识符必须是有效的 C 语言标识符，且必须在源代码中使用。

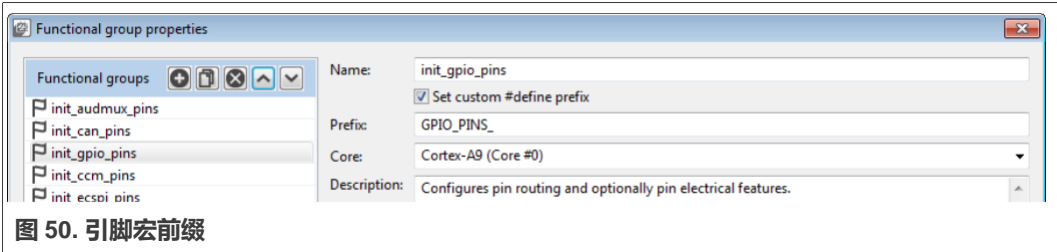


图 50. 引脚宏前缀

如果使用了多个函数，则每个单独的函数都可以包含一个特殊前缀。勾选 **Pins > Functional Group Properties > Set custom #define prefix（引脚 > 函数组属性 > 设置自定义#define 前缀）** 复选框，可输入特定函数中的宏前缀，此前缀用于 **pin\_mux.h** 文件中生成的代码。输入的前缀文本必须是 C 语言标识符。如果未勾选，则使用 **Function name（函数名称）** 作为默认前缀。

3.3.5 扩展头

在 **Expansion Header（扩展头）** 视图中，您可以添加和修改扩展头配置、映射连接器并路由引脚信号。您还可以导入扩展板并将其应用于扩展头。

某些板，例如 LPCXpresso55S69，带有预设的扩展头。

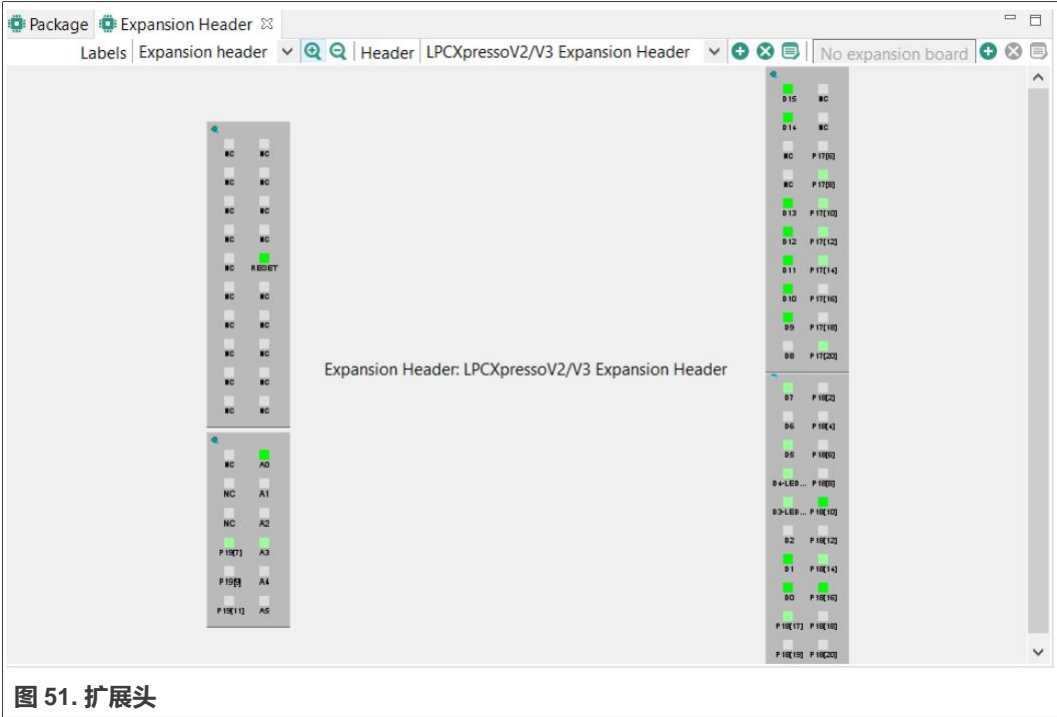


图 51. 扩展头

并非每个支持的设备都会自动预设扩展头。如果没有预配置扩展头，请按照以下步骤创建和修改扩展头配置：

1. 从 **Toolbar (工具栏)** 中选择 **Views > Expansion Header (视图 > 扩展头)**，打开视图。
2. 视图工具栏中选择 **Add (添加)** 按钮，添加扩展头。
3. 在 **Add New Expansion Header (添加新扩展头)** 窗口中，从下拉列表中选择 **Header type (扩展头类型)**。

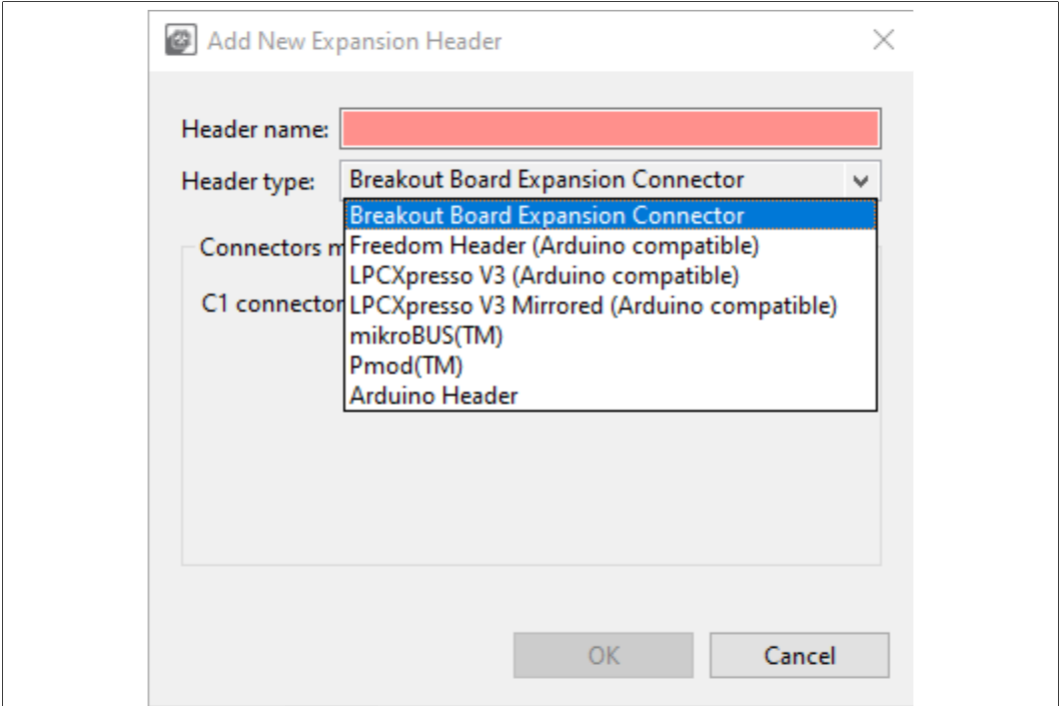


图 52. 添加新扩展头

4. 命名标头并映射连接器。

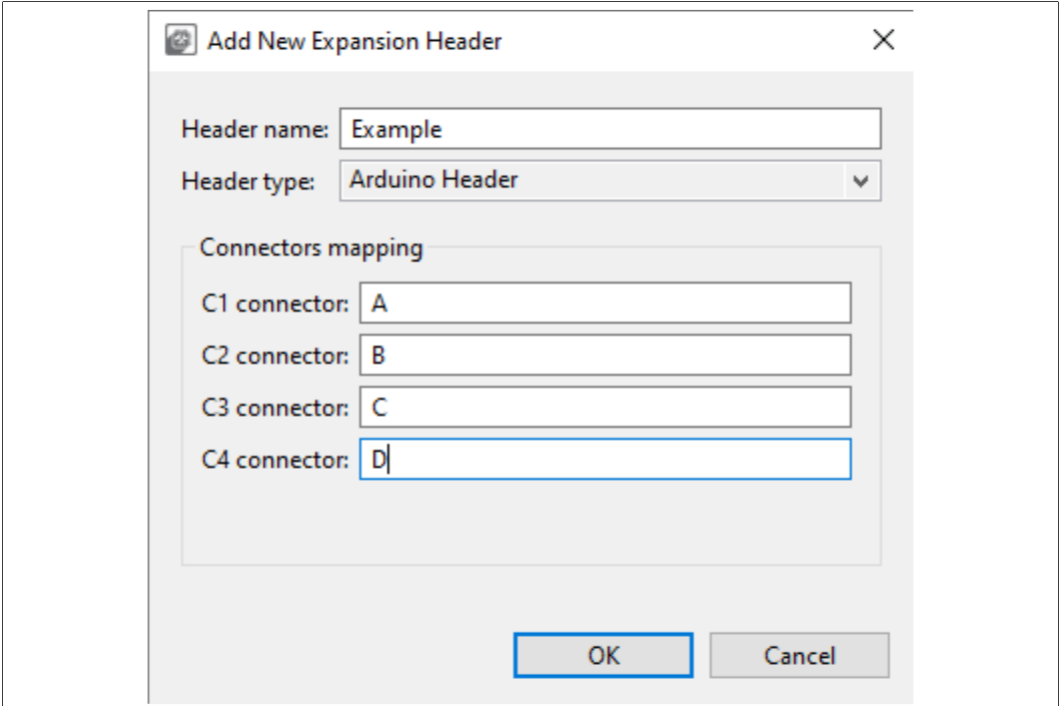


图 53. 添加新扩展头

5. 选择 **OK (确定)**。

现在，**Expansion Header (扩展头)** 视图会显示连接器的布局。您可以将鼠标指针移动到引脚上，可显示更多信息。右键单击引脚可显示其他选项的快捷菜单。



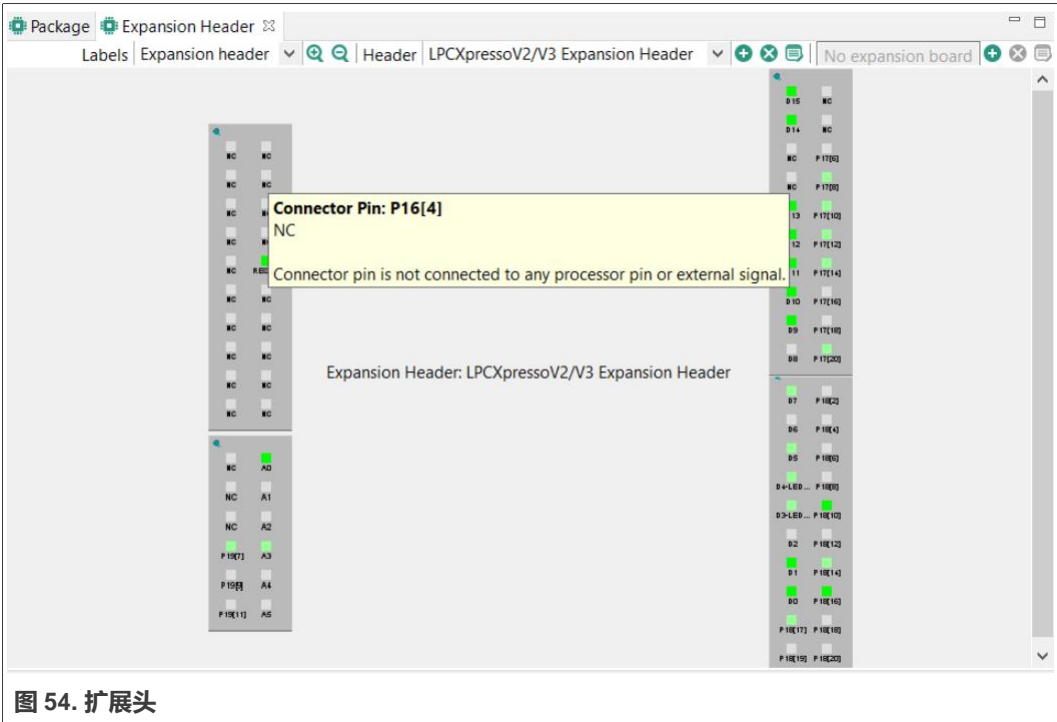


图 54. 扩展头

- 6. 要将扩展头引脚映射到处理器引脚，请右键单击扩展头引脚并选择 **Connect（连接）**。
- 7. 在 **Connector Pin（连接器引脚）** 对话框中，从列表中选择处理器引脚/外部信号，然后单击 **OK（确定）**。

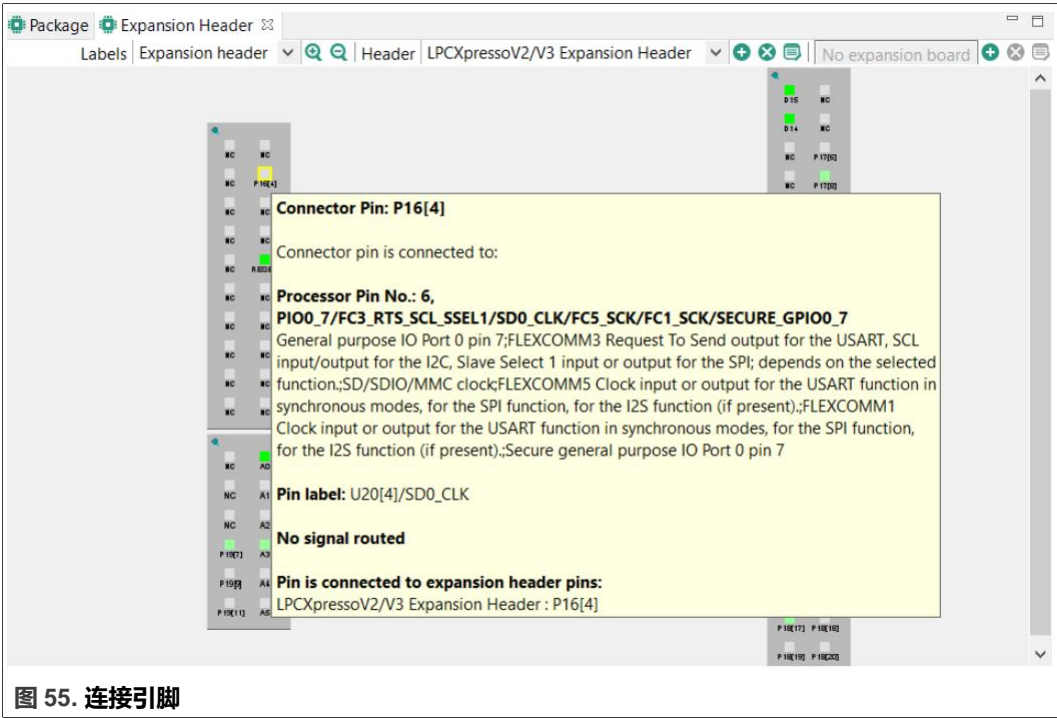


图 55. 连接引脚

- 8. 要路由引脚，右键单击头引脚并选择 **Route（路由）**。
- 9. 在 **Pin（引脚）** 对话框中，从列表中选择信号并单击 **OK（确定）**。  
这样，连接器引脚就成功路由了。

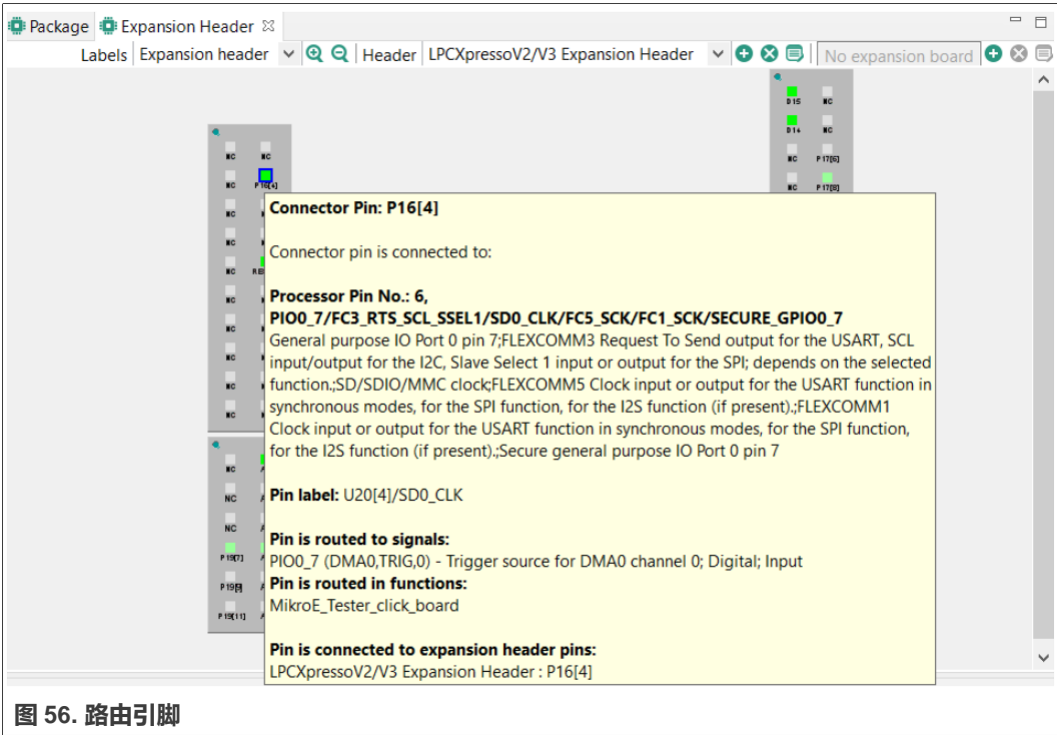


图 56. 路由引脚

您可以创建多个扩展头配置。在此视图的下拉列表中切换不同的配置。

要在 **Pins (引脚)** 和 **Routing Details (路由详情)** 视图中高亮显示引脚/路由配置，请右键单击连接器引脚并选择 **Highlight (高亮显示)**。

选择 **Edit (编辑)** 按钮可随时修改配置参数。**Pins (引脚)** 视图中的信息会自动更新。

选择 **Remove (删除)** 按钮删除配置。

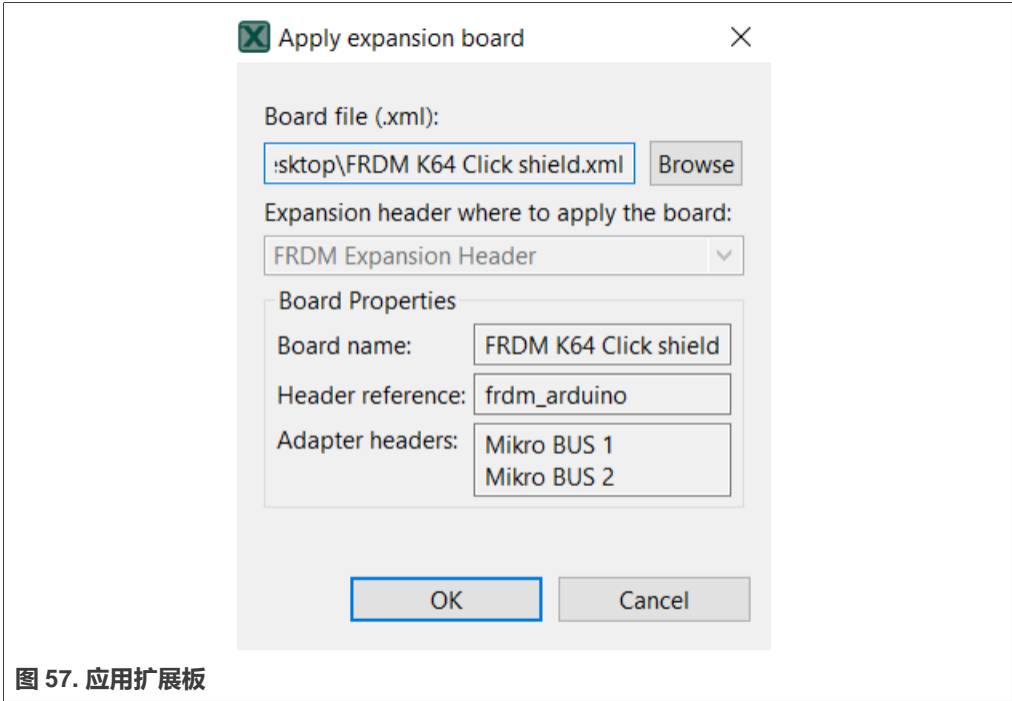
使用 **Label (标签)** 下拉列表来切换扩展头、电路板和路由的显示信息。

3.3.5.1 扩展板

在 **Expansion Header (扩展头)** 视图中，您还可以将扩展板应用到已创建的扩展头上。扩展板配置可以 XML 文件的形式导入到引脚工具。根据所选的处理器，此工具会推荐适当的路由方式。

**注意：** 每个扩展头只能配置一块扩展板。

- 1. 在 **Expansion Header (扩展头)** 视图中，单击 **Apply expansion board to the selected header (将扩展板应用到所选标头)**。或者，从 **Menu bar (菜单栏)** 中选择 **Pins>Apply expansion board (引脚 > 应用扩展板)**。
- 2. 在 **Apply expansion board (应用扩展板)** 对话框中，单击 **Browse (浏览)** 找到含扩展板信息的 XML 文件，然后单击 **OK (确定)**。



- 3. 单击 **OK (确定)**，应用扩展板。
- 4. 在下一页中，选择是为扩展板创建一个新的函数组，还是修改现有的函数组。如果是后者，使用下拉列表从可用的函数组中选择一个。
- 5. 在 **Expansion Board Routing (扩展板路由)** 表中，查看扩展板引脚的建议路由。如果要更改引脚的路由，请单击 **Route (路由)** 列中的引脚单元，在 **Connector pin (连接器引脚)** 对话框中选择信号，然后单击 **Done (完成)**。

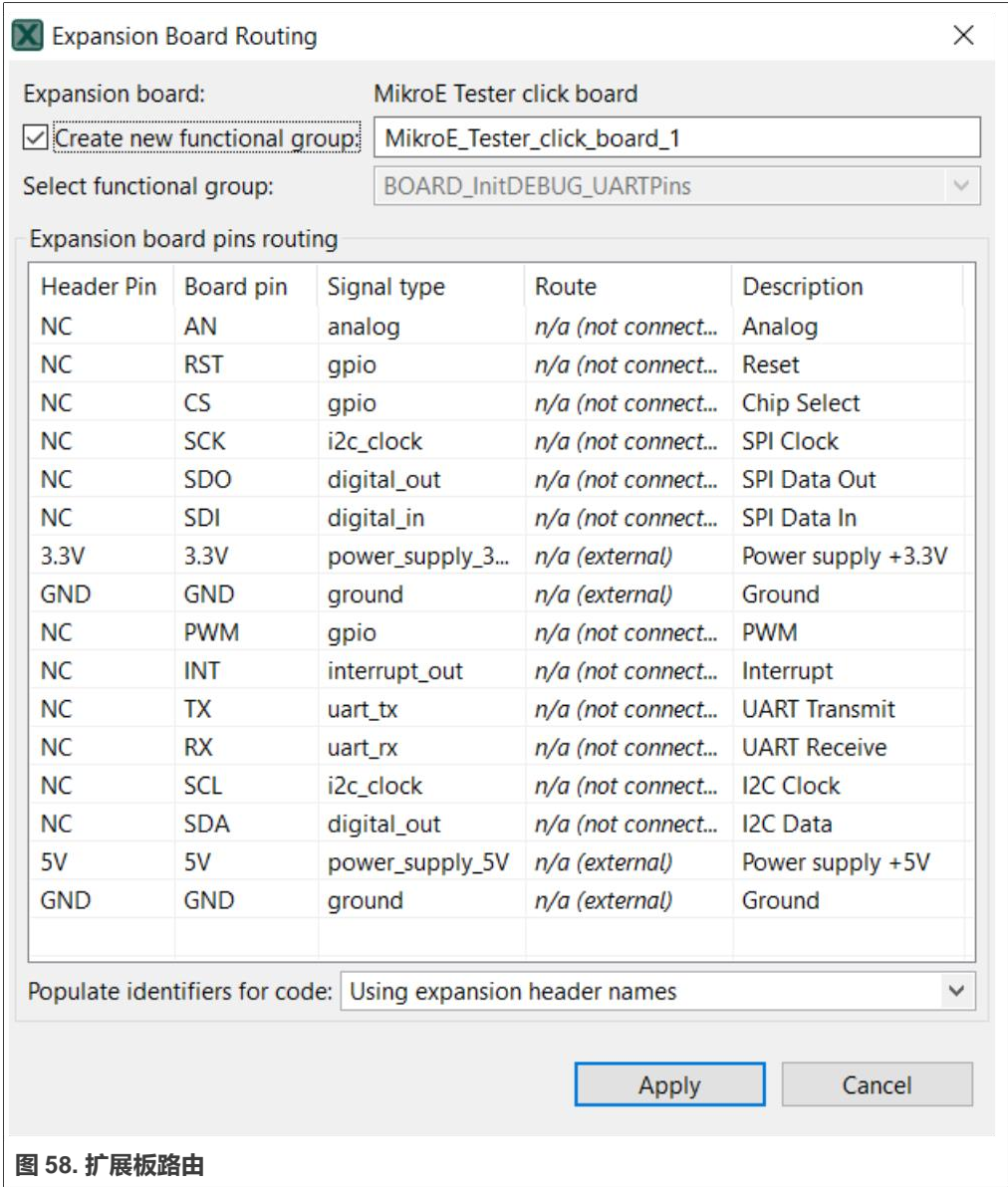


图 58. 扩展板路由

6. 选择要填写的代码标识符的方式。有以下选项可用：
- 扩展头名称

• 扩展板名称

• 无
7. 单击 **Apply (应用)** 来应用设置。
- 随时单击 **Expansion Header (扩展头)** 视图中的 **Configure routing for expansion board (配置扩展板路由)** 按钮，更改扩展板信号的路由。

3.3.6 电源组

如果处理器支持电源组，则在 **Pins (引脚)** 和 **Peripheral Signals (外设信号)** 选项卡旁边，会出现一个额外的选项卡。

**注意：**并不是所有器件都支持该功能。

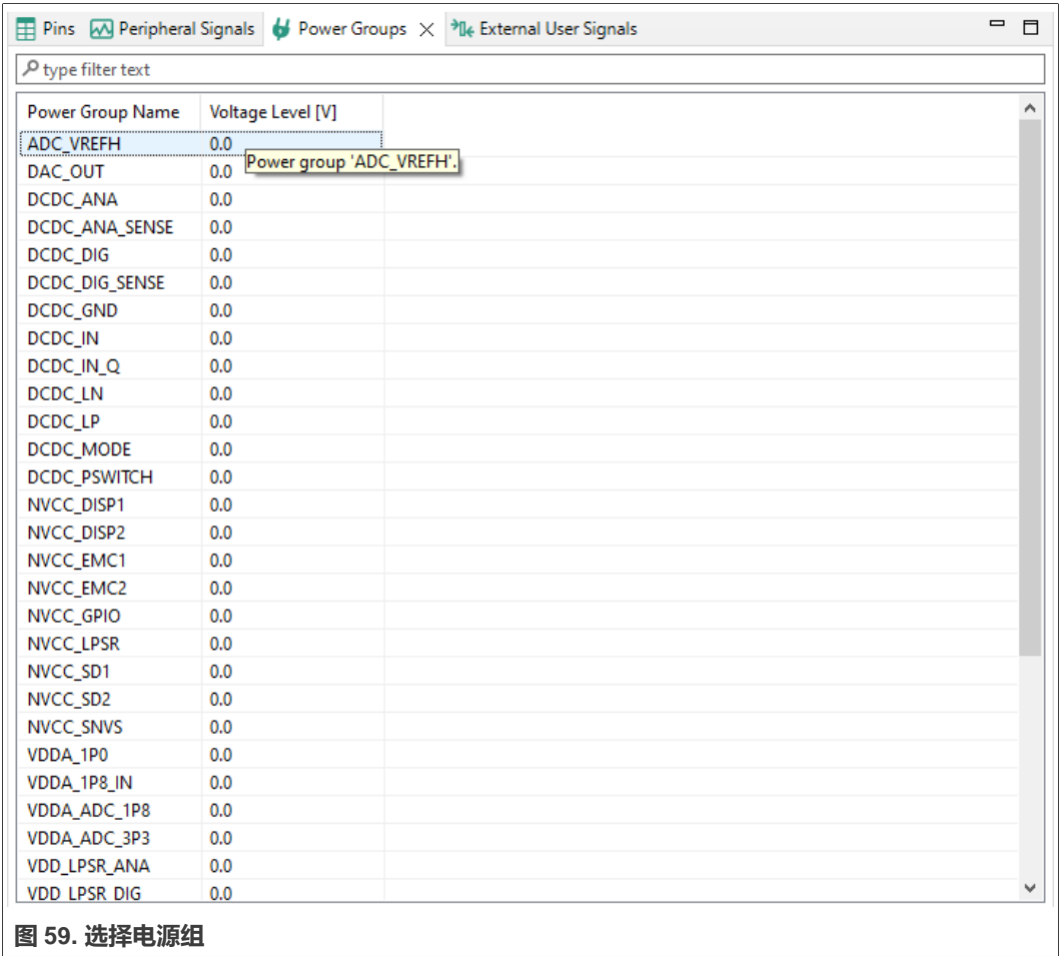


图 59. 选择电源组

3.3.7 外部用户信号视图

这个视图默认是禁用的，必须从菜单 **Views -> External Custom Signals**（视图 -> 外部自定义信号）中打开。用户可以通过该视图对信号进行自定义描述。

外部用户信号在表格中有一个定义的唯一 ID、所连接的引脚以及任意数量的附加文本信息。所有这些都可以自定义。

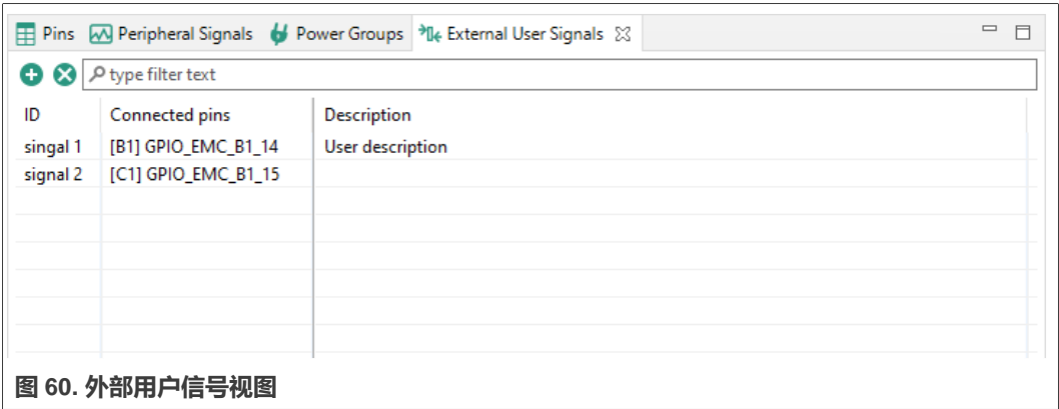


图 60. 外部用户信号视图

您可以从所选信号的弹出菜单连接引脚。一个信号可连接多个引脚，一个引脚也可以连接多个信号。定义某些信号后，**Pins (引脚)** 视图会增加一个 External User Signals（外部用户信号）列。引脚和信号之间的连接也可在此进行。

可使用表头的弹出菜单指定其他列。

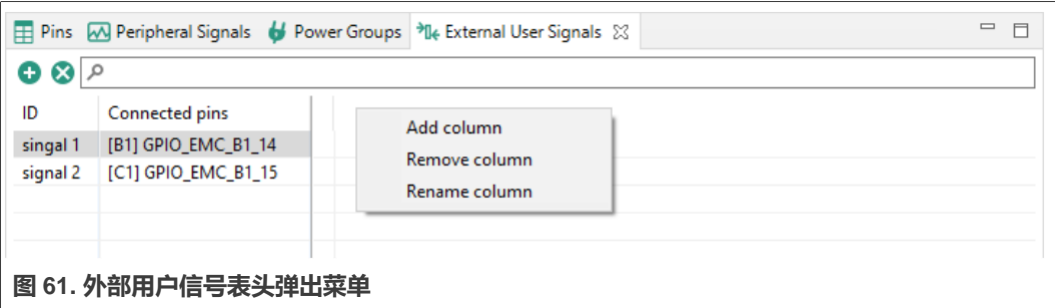


图 61. 外部用户信号表头弹出菜单

如果需要，外部用户信号也可以导出为 CSV 文件，然后导入到另一个配置中。不支持信号的合并，因此当一个配置中已经定义一些信号时，它们会被导入的信号替换。

3.3.8 函数

函数用于将一组路由的引脚分组，并在函数中创建配置代码，然后可以由应用程序调用函数。该工具允许创建多个用于配置引脚复用的函数。

在 **Pins (引脚)**、**Peripheral Signals (外设信号)** 和 **Package (封装)** 视图中，引脚的使用情况会以 50% 的不透明度显示。每个函数可定义一组已路由的引脚或重新配置路由的引脚。

在配置中指定多个函数时，封装视图主要显示所选函数的引脚和外设。其它函数的引脚和外设以浅色透明度显示，在切换到该函数之前无法进行配置。

3.3.9 高亮显示和颜色编码

使用高亮显示可以轻松识别封装中的路由引脚/外设。默认情况下，在 **Package (封装)** 视图中高亮显示当前选择（引脚/外设）。

- 在 **Package (封装)** 视图中，引脚/外设以黄色边框高亮显示。如果选中了高亮显示的引脚/外设，它周围会有蓝色边框。
- 红色表示引脚有错误。
- 绿色表示该引脚已复用或使用。
- 浅灰色表示该引脚可用于复用，但未复用或使用。
- 深灰色表示该引脚/外设是专用的。它是默认路由的，并且对生成的代码没有影响。

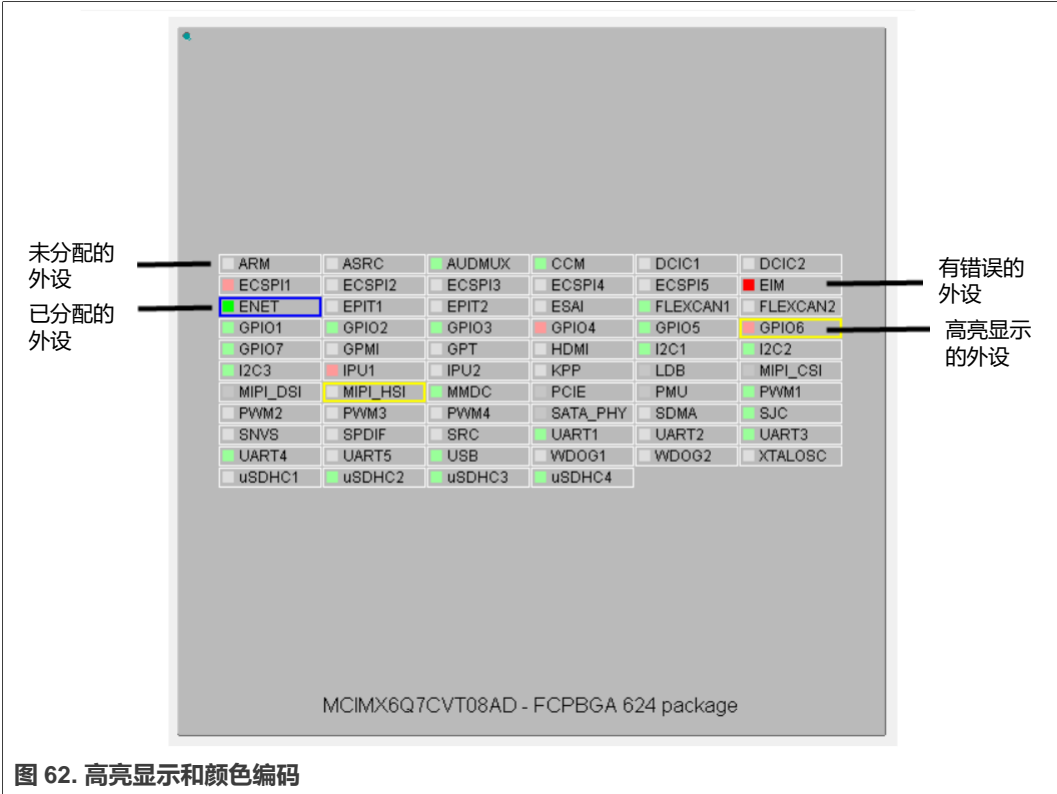


图 62. 高亮显示和颜色编码

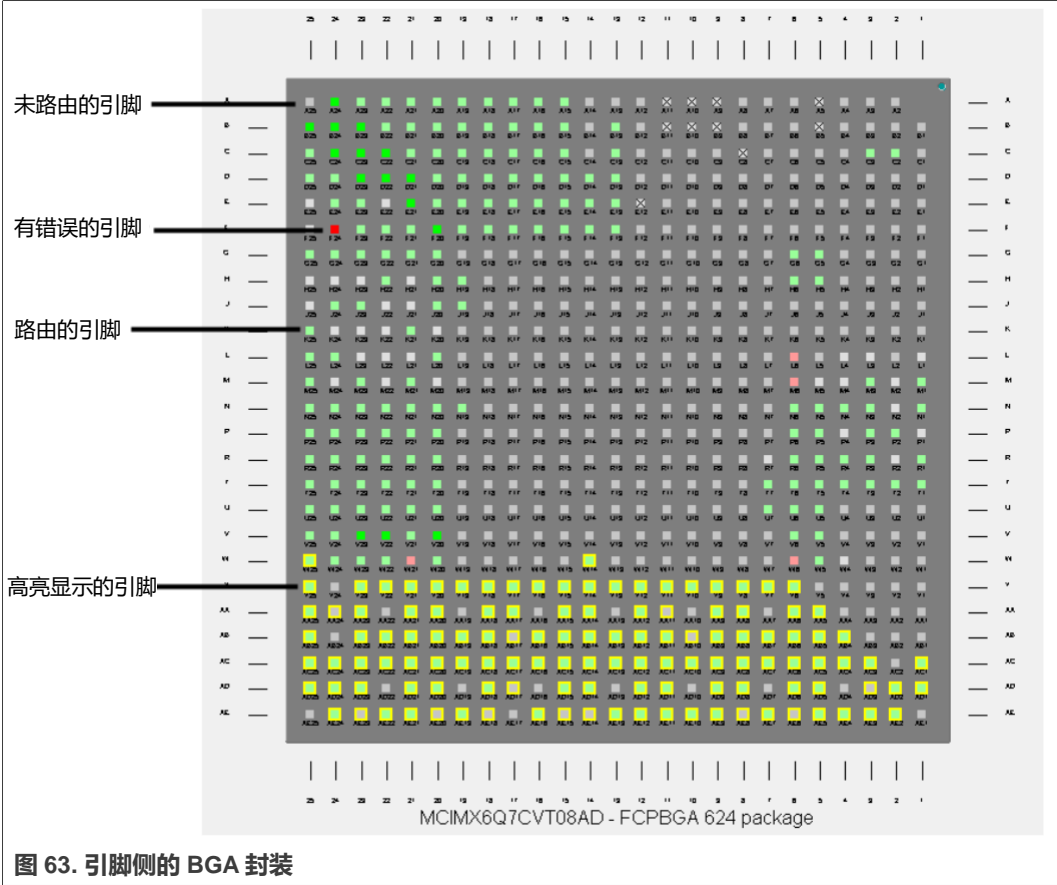


图 63. 引脚侧的 BGA 封装



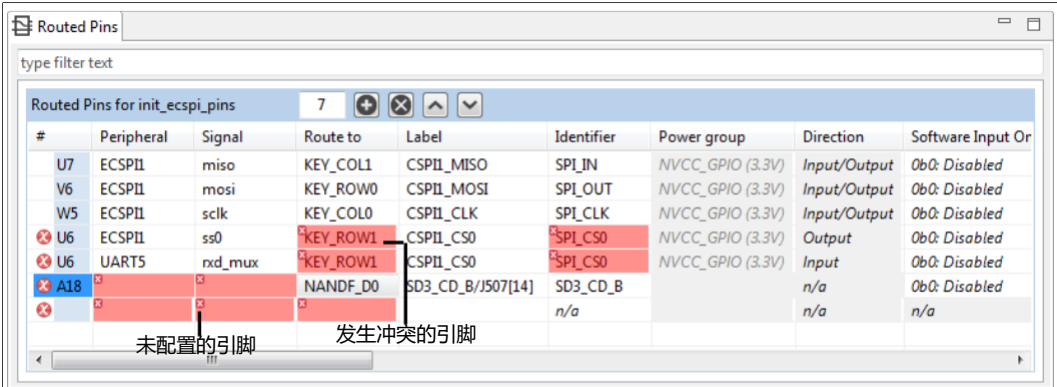


图 64. 引脚冲突

#	Peripheral	Signal	Route to	Label	Identifier	Direction	Slew rate
33	GPIOE	GPIO, 26	PTE26	J2[1]/D12[4]/LED RGB_GREEN	Not Specified	Input	Slow
71	FTM0	CH, 0	FTM0_CH0	J1[5]	n/a	Output	Fast

图 65. 警告

- **Package (封装) 视图**
  - 单击外设或使用弹出菜单高亮显示外设：
    - 以及所有分配给该外设的引脚。
    - 或者如果该外设还没有分配任何引脚，就高亮显示所有可用的引脚。
  - 单击引脚或使用弹出菜单来高亮显示引脚和相关的外设。
  - 单击封装外部的空白处，可取消高亮显示。
- **Peripherals/Pins (外设/引脚) 视图**
  - 外设和引脚的高亮显示方式与上述相同。

3.4 错误和警告

引脚工具会检查路由中是否有冲突，以及配置是否有错误。路由冲突是在所有的 INIT 函数（默认的初始化函数）之间进行检查的。您可以在不同的函数（非 INIT 函数）中对同一个引脚配置不同的路由，以实现动态的引脚路由重新配置。

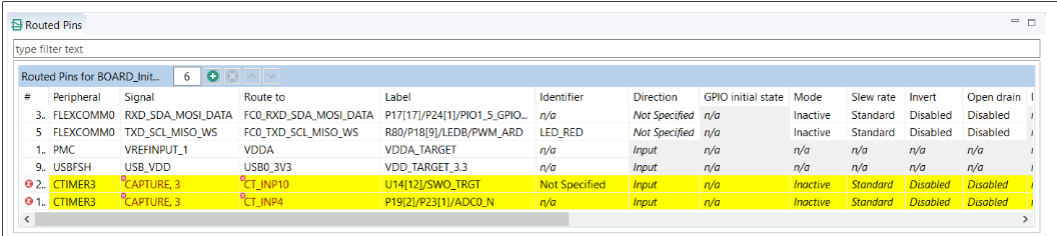


图 66. 错误和警告

如果遇到错误或警告，**Routing Details (路由详情)** 视图中的冲突会在该行的第一列中用红色标记，错误/警告会在产生冲突的单元格中用红色字体显示。上图的最后两行显示了发生错误配置的外设/信号。详细的错误/警告消息会以工具提示的形式出现。



有关错误和警告颜色的详细信息，请参阅“[高亮显示和颜色编码](#)”章节。

3.4.1 不完整路由

不完整路由的单元格以红色背景显示。要生成合适的引脚路由，请单击下拉箭头并选择适当的值。单元格上的红色 X 符号表示出错。

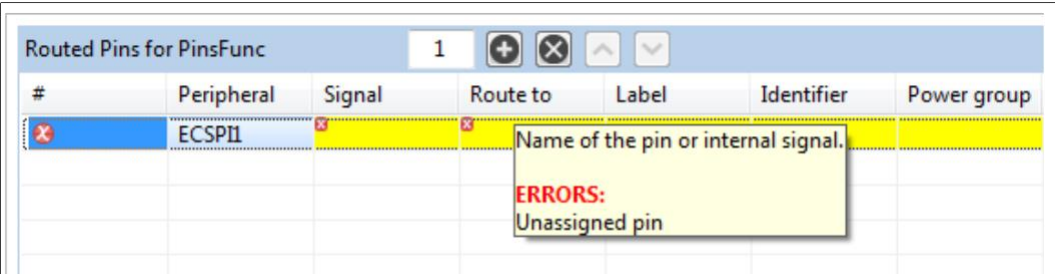


图 67. 不完整路由

将鼠标悬停在单元格上，可以查看有关冲突或错误的更多详细信息，通常会列出发生冲突的行。

您也可以从主菜单中选择 **Pins > Automatic Routing (引脚 > 自动路由)** 来解决任何路由问题。

**注意：**并非所有路由问题都能自动解决。在某些情况下需要您手动调整。

3.4.2 电源组电压电平冲突

当通过不同的电源组配置外设信号路由引脚时，由于在不同的 **Power groups (电源组)** 视图中有可能会设置了不同的电压电平值，因此引脚工具会提供可能的电压电平冲突信息。如果出现潜在的电压电平冲突，则会显示一个警告 - 这对硬件电路板设计人员非常有用。

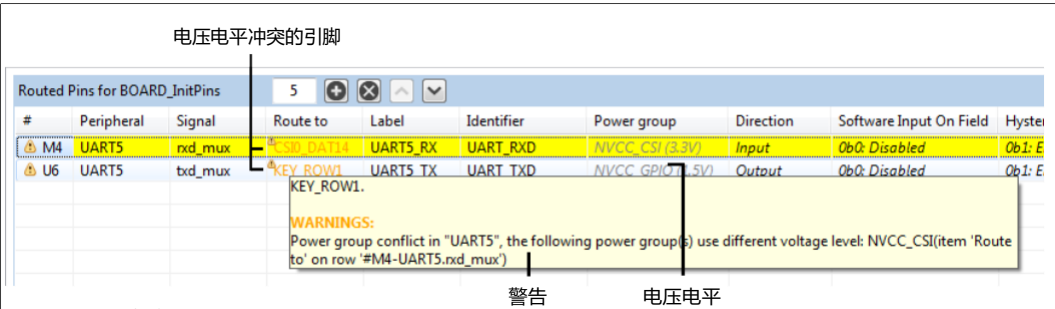


图 68. 电源组视图

### 3.5 代码生成

如果设置正确且没有报告错误，代码生成引擎会立即重新生成源代码。您可以在 **Pins (引脚)** 工具的 **Code Preview (代码预览)** 视图中查看生成的代码。

**Code Preview (代码预览)** 会自动高亮显示当前与前一次迭代代码之间的差异。您可以单击 **Set viewing style for source differences (设置源代码差异查看样式)**，在两种高亮显示模式之间进行选择。您页可以从同一下拉菜单中完全禁用高亮显示。

对于多核，为每个核生成源代码。相应的文件显示为 @Core #{number} 标签。

**注意：** 标签名称可能因所选的多核处理器系列/类型而不同。

您也可以将生成的代码复制并粘贴到源文件中。此视图会为每个函数生成代码。除函数注释外，还以 YAML 格式存储工具配置。这个注释不是用于直接编辑的，而是用于稍后恢复引脚配置。

**注意：** **Code Preview (代码预览)** 视图包含 **Export (导出)** 按钮和导出源文件的功能 - 链接到导出向导-- **Pins Tool > Export Source Files (引脚工具 > 导出源文件)** 选项 - 允许在多核启用配置中按所用内核导出源文件。

YAML 配置包含每个引脚的配置。它只存储非默认值。

**提示：** 对于多核处理器，它将为每个内核生成源文件。如果处理器受 SDK 支持，默认情况下会从 main 生成 BOARD\_InitBootPins 函数调用。您可以为每个函数指定 Call from BOARD\_InitBootPins (从 BOARD\_InitBootPins 调用)，以便生成适当的函数调用。

### 3.6 在代码中使用引脚定义

引脚工具会生成命名常量的定义，您可以在应用程序代码中使用它们。这些常量是根据用户指定的标识符来命名的，可以让您编写与配置的路由无关的代码。如果您更改了路由信号的引脚，此应用程序也能正确地引用到相应的引脚。

例如，当 LED\_RED 被指定为路由至 PTB22 引脚的标识符时，会在 pin\_mux.h 中生成以下定义：

```
#define BOARD_LED_RED_GPIO GPIOB /*!<@brief GPIO device name:
GPIOB */
#define BOARD_LED_RED_PORT PORTB /*!<@brief PORT device name:
PORTB */
#define BOARD_LED_RED_PIN 22U /*!<@brief PORTB pin index: 22 */
```

定义名称由函数组前缀和引脚标识符组成。更多详情，请参见[第 2.4.3 节](#)和[第 3.3.4.1 节](#)。

例如，要在使用 SDK 驱动程序 (fsl\_gpio.h) 的应用程序中写入这个 GPIO 引脚，您可以使用以下代码来引用标识符为 LED\_RED 的引脚生成的定义：

```
GPIO_PinWrite(BOARD_LED_RED_GPIO, BOARD_LED_RED_PIN, true);
```

3.7 引脚完全初始化

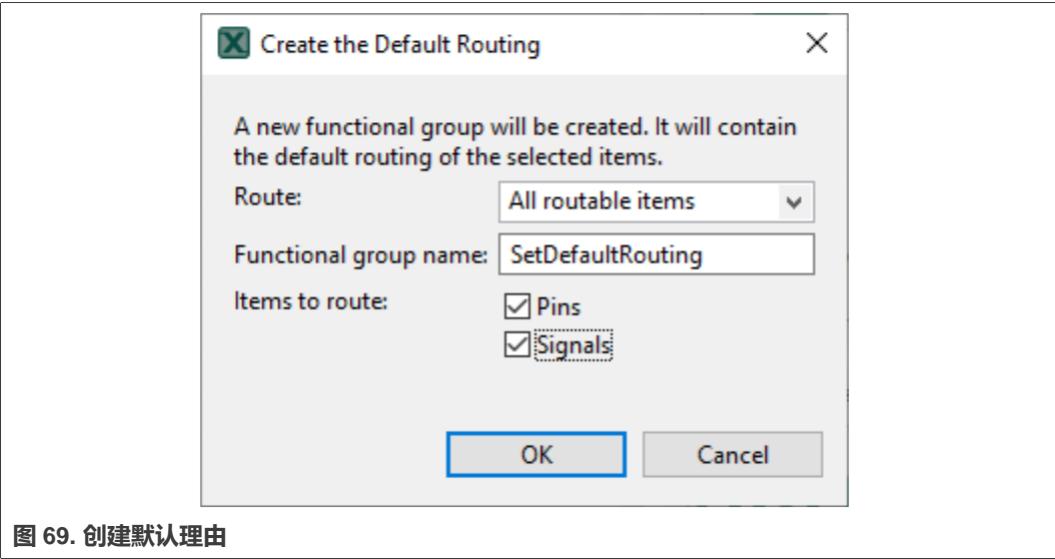
在某些情况下，默认值并不可靠，因为在应用程序之前可能有运行的代码会修改引脚配置，例如 bootloader（引导加载程序）。Full initialization of pins（引脚完全初始化）选项可确保即使是使用复位后状态的项目也能完全初始化。该选项是针对每个 Functional group（函数组）的，可以强制完全初始化路由。默认情况下不启用 Full initialization of pins（引脚完全初始化）。启用后，现有路由的电气属性会发生变化。“复位”值将更改为与之对应的显式值。禁用该选项后，引脚工具会将与复位后状态相匹配的值更改为“复位”值。

3.8 创建默认路由

如有需要，可以创建一个新的函数组，将默认信号路由至引脚和内部信号。该功能可在 Pins -> Create Default Routing（引脚 -> 创建默认路由）中使用。用户可以在此选择：

- 是路由所有引脚和信号，还是只路由其他函数组中没有路由的引脚和信号。
- 新函数组的名称。
- 是否为引脚和/或内部信号创建路由。

在创建的函数组中，将设置引脚的完全初始化功能。引脚的电气属性将设置为复位后的状态。



4 DDR 工具

本章节介绍 DDR 配置和验证工具，它是 i.MX 配置工具的嵌入式组件。

DDR 工具有两个主要功能：配置和验证。

新项目配置页面中会标记支持的设备。

**注意：**DDR 工具是“按原样”提供的，可帮助客户评估、调试和优化设计。该工具提供的结果或其中的任何部分，在任何情况下都不能取代传统的验证和合规方法，仍然需要执行这些方法来声明设计符合相应的 JEDEC 标准。

4.1 新建 DDR 工具项目

要使用 DDR 工具，首先需要创建一个新项目。

要新建 DDR 工具项目，请按照以下步骤操作：

- 1. 打开 **Config tools for i.MX (i.MX 配置工具)** 。
- 2. 选择 **Create a new standalone configuration for processor, board, or kit (为处理器、电路板或套件新建独立配置)**，然后单击 **Next (下一步)**。
- 3. 从 **Processors (处理器)** 中，选择一个支持 DDR 工具的器件，然后单击 **Finish (完成)**。
- 4. 要打开 DDR 工具视图，请单击 **DDR 工具图标**。

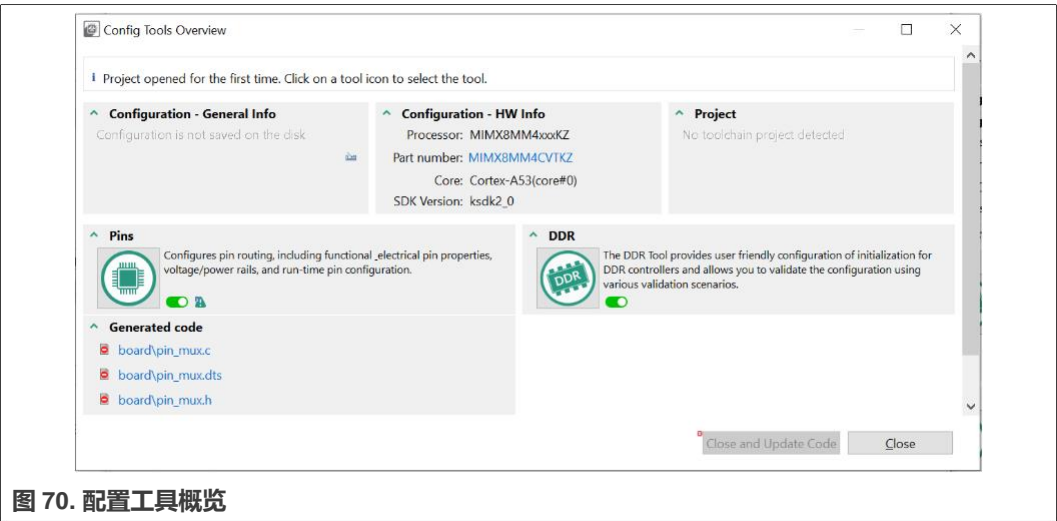


图 70. 配置工具概览

- 5. 要使用 DDR 工具，需要接受免责声明。

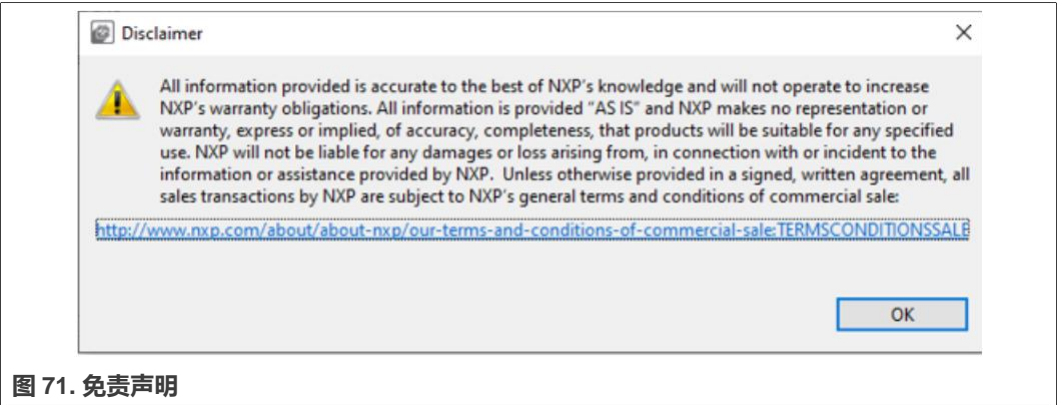


图 71. 免责声明

4.2 DDR 配置

DDR 配置提供了一个用户友好的图形界面来配置 DDR 接口和其他相关子系统。当使用不同的内存模块进行配置时，可以使用它来更改 DDR 控制器和 PHY 配置，并优化与信号完整性相关的参数。

4.2.1 导入初始化脚本

Import initialization script（导入初始化脚本）可以加载 Register Programming Aid (RPA)（寄存器编程辅助(RPA)）工具提供的初始化脚本，并绕过用户界面（UI）配置。要获取最新的 RPA，请参考[恩智浦网络社区](#)上的以下链接。

- 1. 要导入 RPA 初始化脚本，请使用 Import initialization script（导入初始化脚本）按钮并浏览所需的\*.ds 文件。

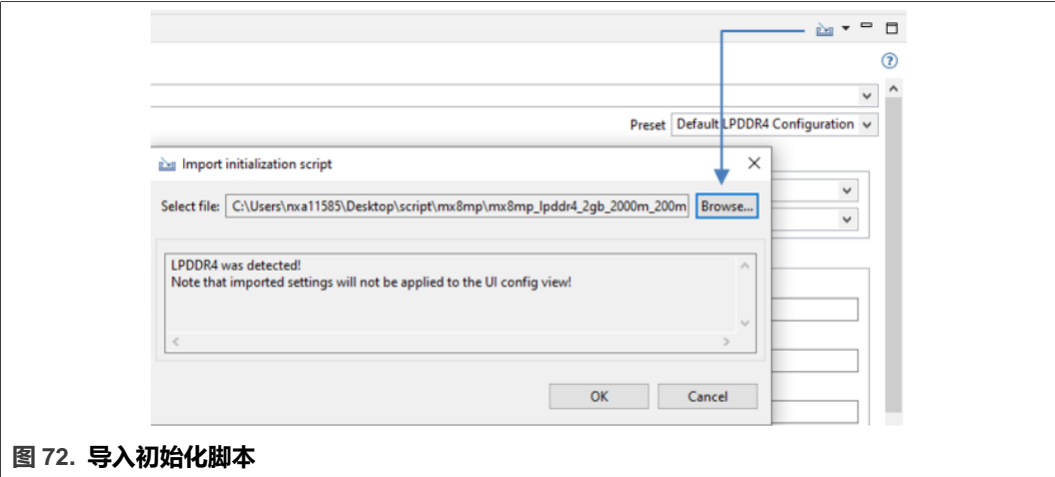


图 72. 导入初始化脚本

- 2. 要加载\*.ds 文件并禁用 UI 配置界面，请单击 OK（确定）。

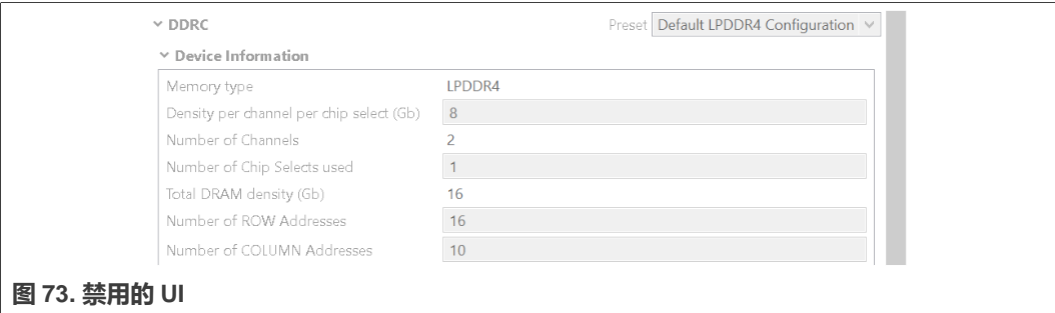


图 73. 禁用的 UI

- 3. 导入的 \*.ds 文件内容显示在 Code Preview（代码预览）中，即 ddr\_config.ds

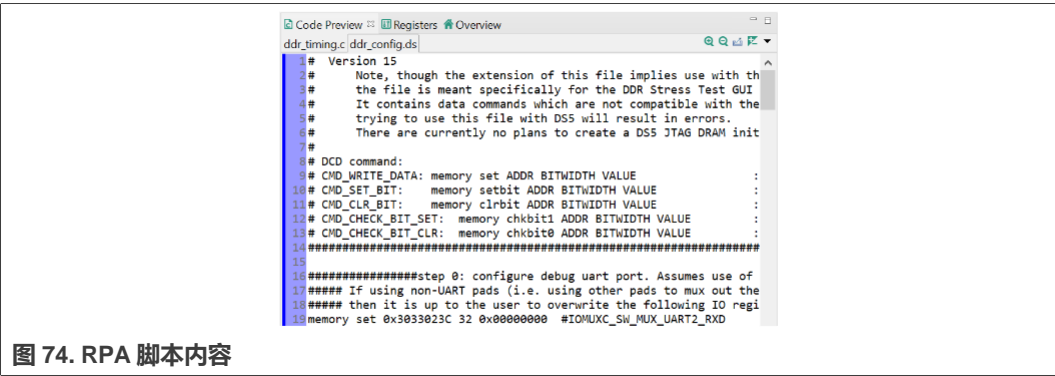


图 74. RPA 脚本内容

4.2.2 从目标导入

Import from target（从目标导入）可以加载已配置的工作目标的 DDR 初始化，并绕过用户界面配置。该选项仅适用于有可用 JTAG 连接的设备。

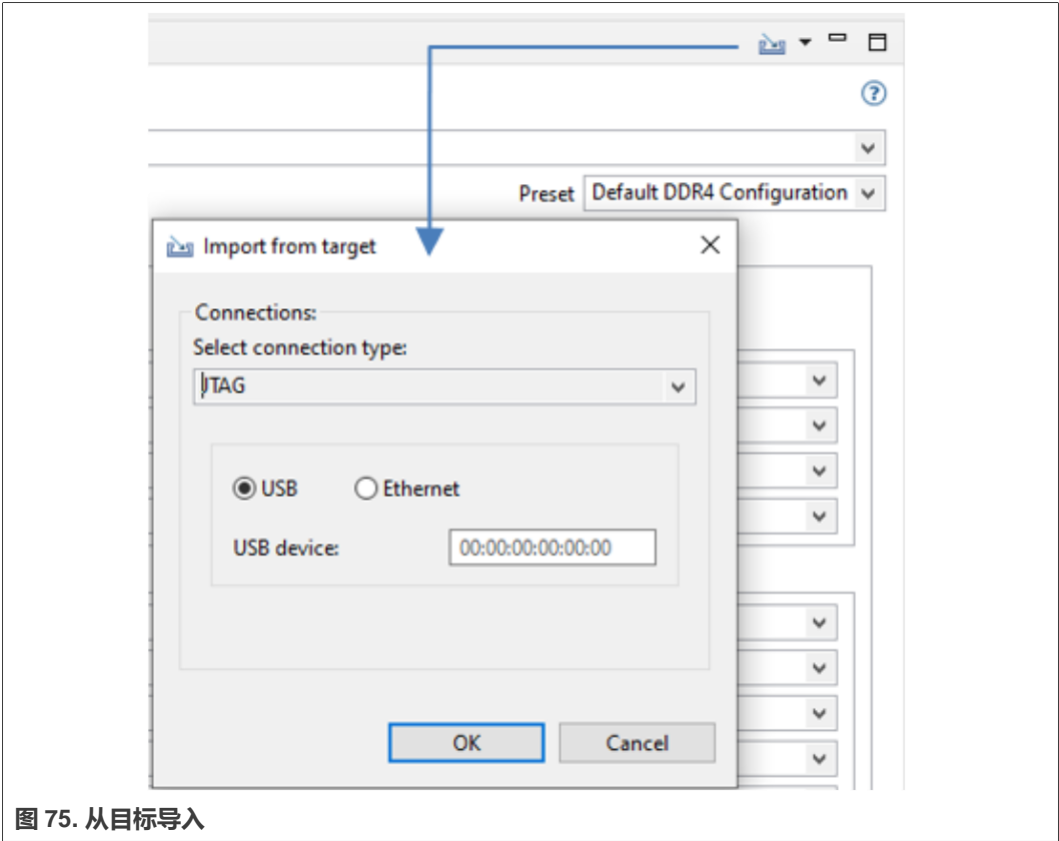


图 75. 从目标导入

4.2.3 启用手动配置

要切换回用户界面配置，请按 “Enable manual config (启用手动配置)” 按钮。

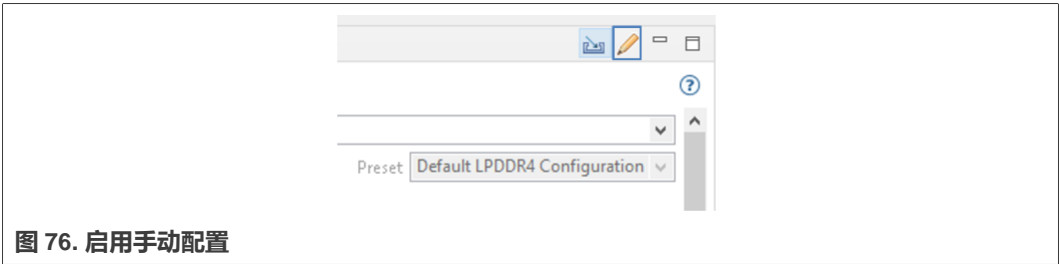


图 76. 启用手动配置

4.2.4 用户界面配置

用户界面配置允许您手动更改设备信息、PHY 选项或特定设计配置。有两种模式可供选择：**Basic (基本模式)** 和 **Advanced (高级模式)**。

**注意：**仅建议有经验的用户使用 **Advanced (高级)** 模式。

**Basic (基本)** 模式允许您配置与设计相关的参数。

1. 器件信息

▼ Device Information

Memory type

LPDDR4

Density per channel per chip select (Gb)

8

Number of Channels

2

Number of Chip Selects used

1

Total DRAM density (Gb)

16

Number of ROW Addresses

16

Number of COLUMN Addresses

10

Number of BANK addresses

3

Number of BANKS

8

Bus Width

32

▼

Number of frequency setpoints

1

▼

Clock Cycle Freq (MHz)

1500 MHz

▼

Clock Cycle Time

666.667 ps

图 77. 器件信息用户界面

2. LPDDR4 电路板数据总线配置

▼ Board data bus config

▼ Channels

#	<								B							>
DRAM data bus	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRAM data bus (User Input ->)	7	6	5	4	3	2	1	0	14	15	10	13	12	11	9	8
Byte lane	<			0				>	<			1				>
Data bus bits within byte lane	7	6	5	4	3	2	1	0	6	7	2	5	4	3	1	0
<																>

图 78. 电路板数据总线配置用户界面

3. UART 端口选择

UART Port

UART2

▼

图 79. UART 选择用户界面

4. DBI 选择 (适用于 LPDDR4)

DBI

Disable

▼

图 80. DBI 选择

5. 内联 ECC 配置 (适用于支持内联 ECC 的器件) 可选择以下参数:

- a. 启用/禁用内联 ECC

Inline ECC

Enabled

▼

图 81. 内联 ECC

- b. ECC 区域粒度



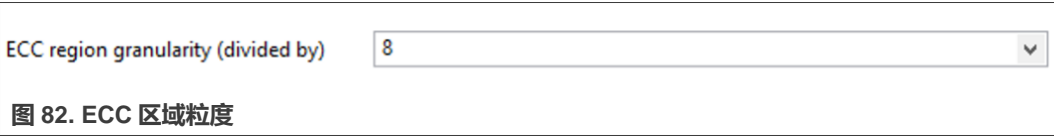


图 82. ECC 区域粒度

- c. 每个主内存区域的 ECC 保护配置
- ECC config binary/non-binary aligned (ECC 配置二进制/非二进制对齐)** 部分概述了 ECC 配置。

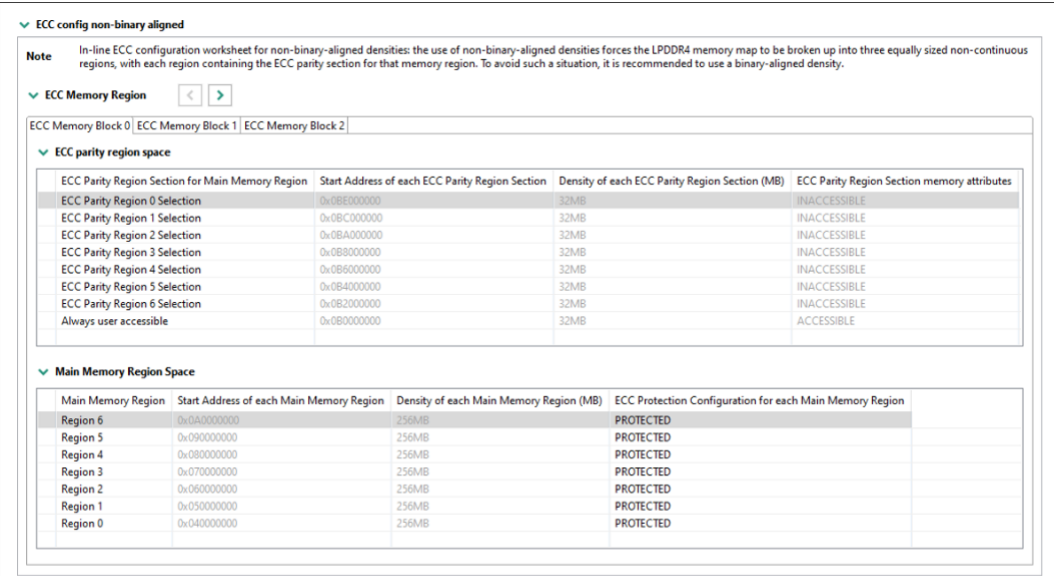


图 83. ECC 配置二进制/非二进制对齐部分

**Advanced (高级)** 模式支持您配置其他参数。

1. 固件版本是 BSP 官方支持的版本，但 **DDR tool (DDR 工具)** 提供了在多个版本之间选择的可能性。



图 84. 固件版本选择用户界面

**注意：**只能使用特定 SoC 和 BSP GA 版本的固件版本。并非每个 SoC 都支持所有固件版本。

2. PHY 日志级别选择

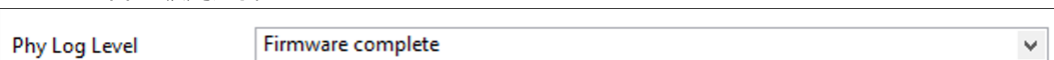


图 85. PHY 日志级别选择

3. IOMUX 配置

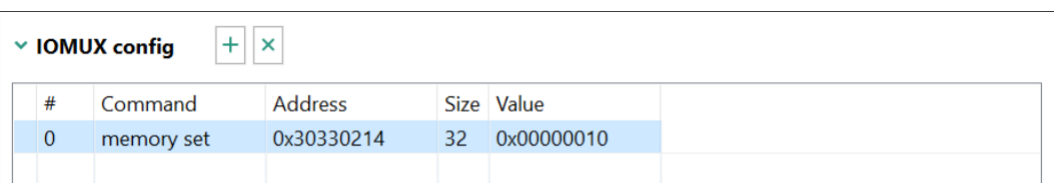


图 86. IOMUX 配置用户界面

4. PMIC 配置顺序



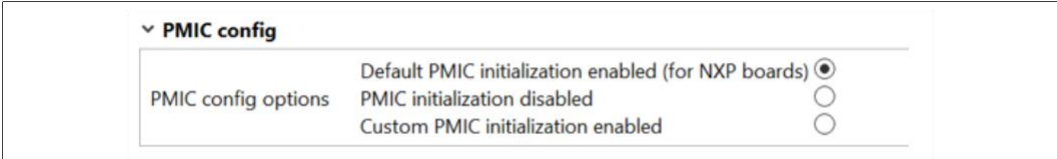


图 87. PMIC 配置用户界面

5. 自定义配置

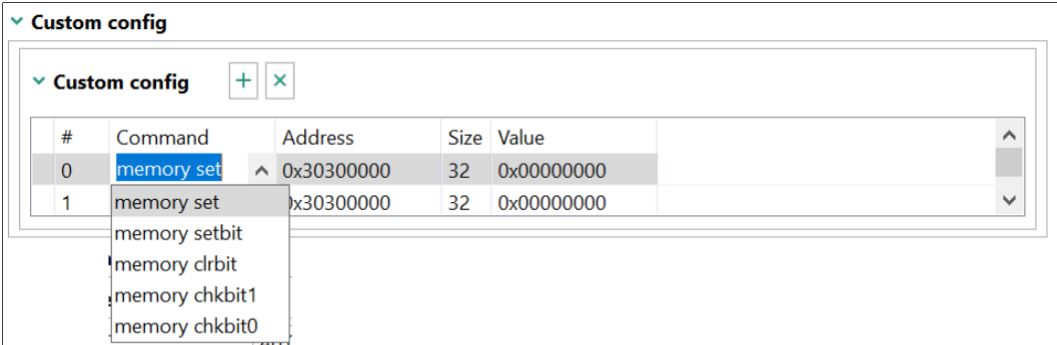


图 88. 自定义配置用户界面

**注意：** 写入不正确的地址可能会导致意外行为。

6. DQ ODT 和 DS 配置

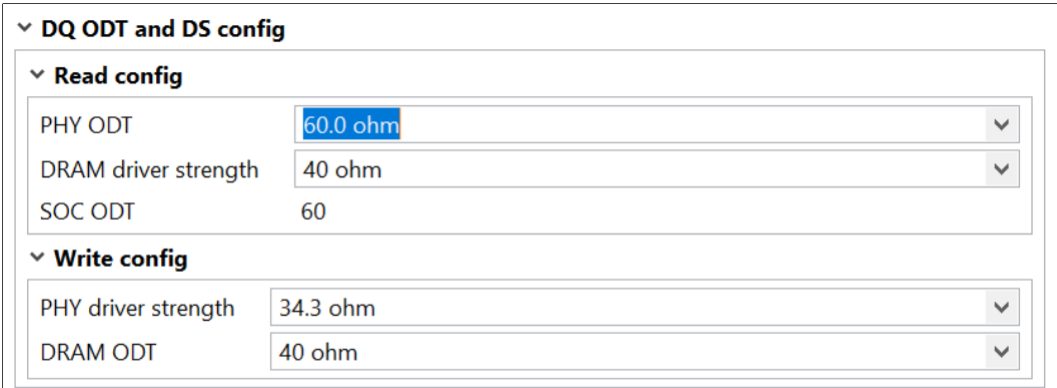


图 89. DQ ODT 和 DS 配置用户界面

7. CA ODT 和 DS 配置 – 仅供参考



图 90. CA ODT 和 DS 配置用户界面

4.2.5 代码生成

您可以在 **Code Preview (代码预览)** 视图中把配置转换为 C 语言代码，供 U-Boot SPL 驱动程序使用。

只需在 GUI 中做出任何修改，就可以触发代码生成，生成的代码会在 **Code Preview (代码预览)** 视图中高亮显示。

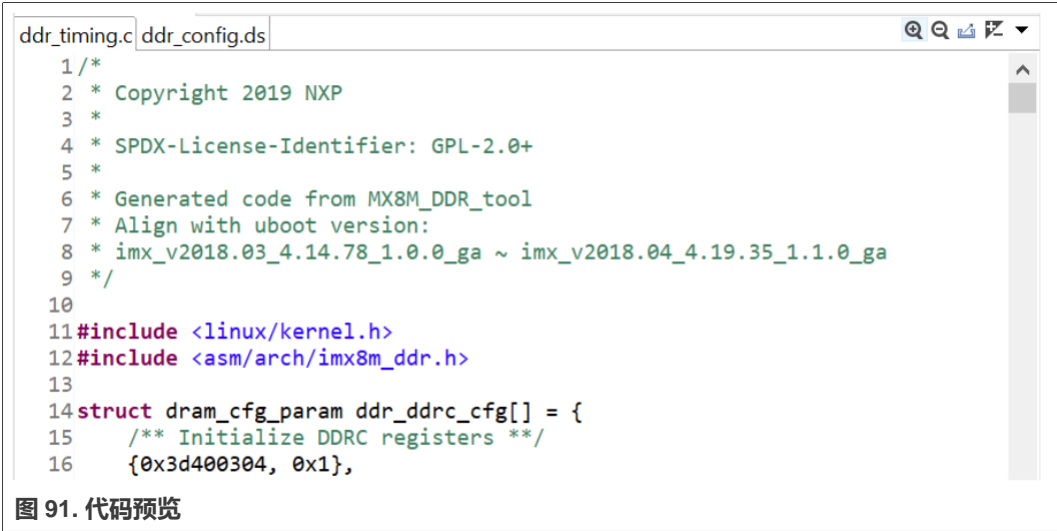


图 91. 代码预览

您可以使用 **DDR tool Export Wizard (DDR 工具导出向导)** 将 **Code Preview (代码预览)** 中的文件保存到磁盘上。

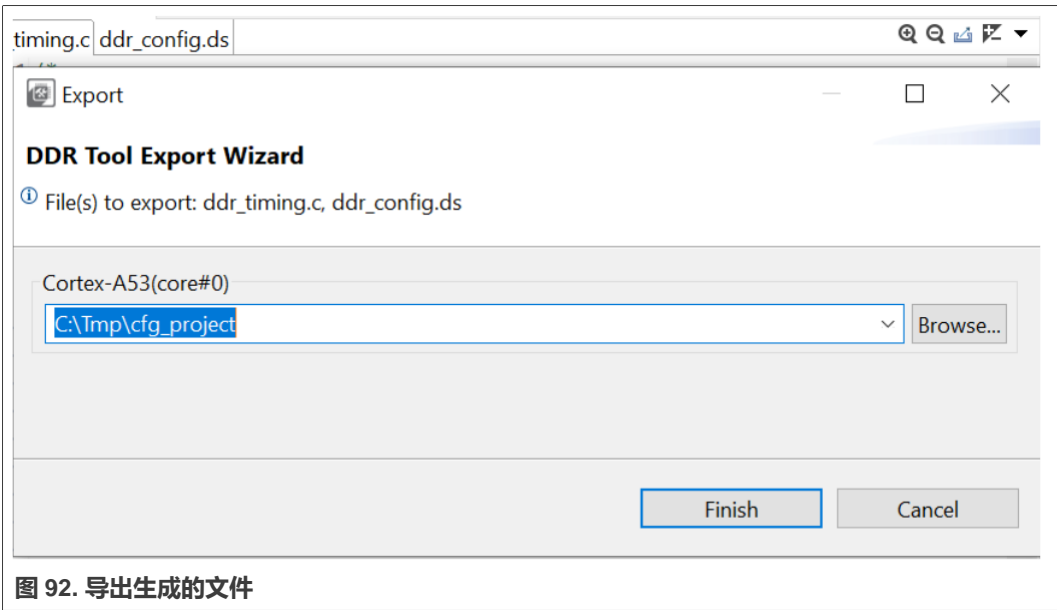
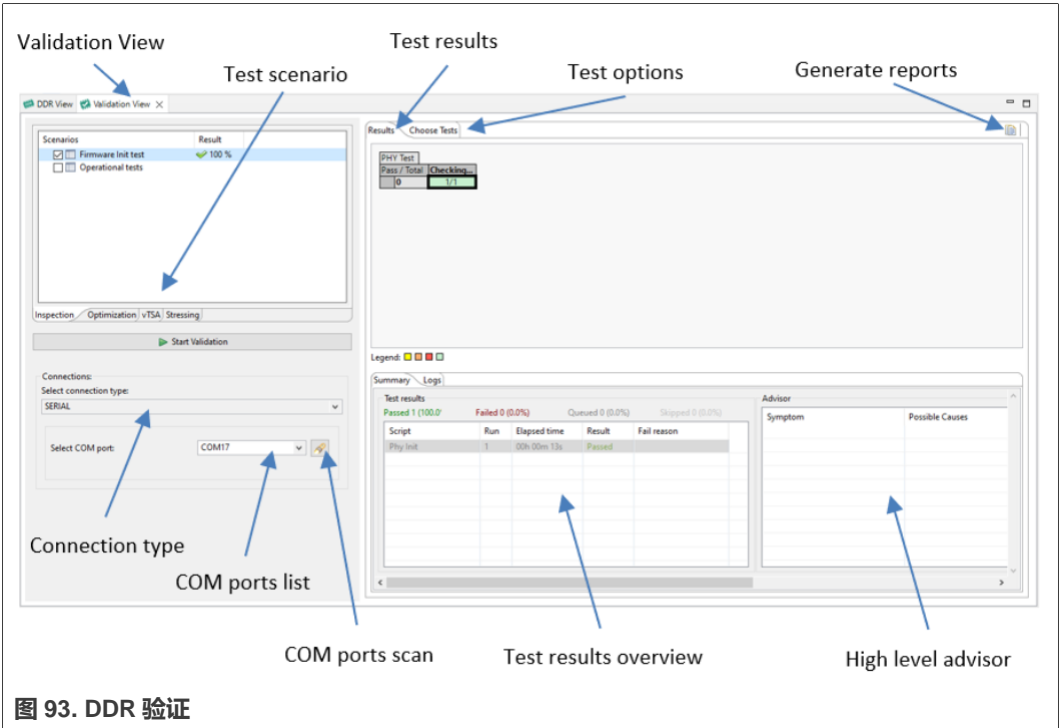


图 92. 导出生成的文件

### 4.3 DDR 验证

DDR 验证将测试映像下载到处理器的内部 RAM 中，这样可以使用不同的场景来评估 DDR 性能。

DDR 验证在无操作系统的环境下，评估电路板上 DDR 接口的稳定性。



4.3.1 连接

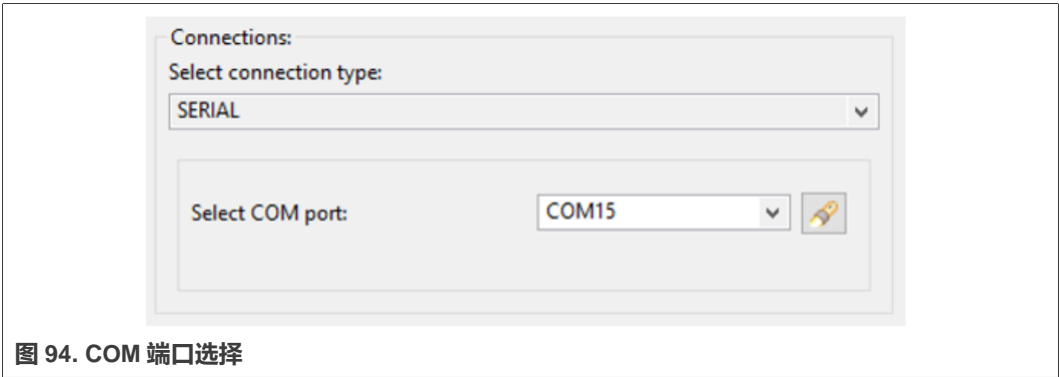
本节介绍与各种类型电路板的连接。

4.3.1.1 支持串行下载模式/制造模式的电路板

要连接到电路板，请按照以下步骤操作：

1. 将电路板引导模式（boot mode）配置为串行下载模式/生产模式，并为电路板加电启动。
2. 将主机的 UART 电缆连接到电路板上 A 核的 UART 端口。
3. 将主机 USB 电缆连接到串行下载模式下所用电路板的 USB 端口。在 Windows 设备管理器中，会显示“HID 兼容设备”或“USB 输入设备”。

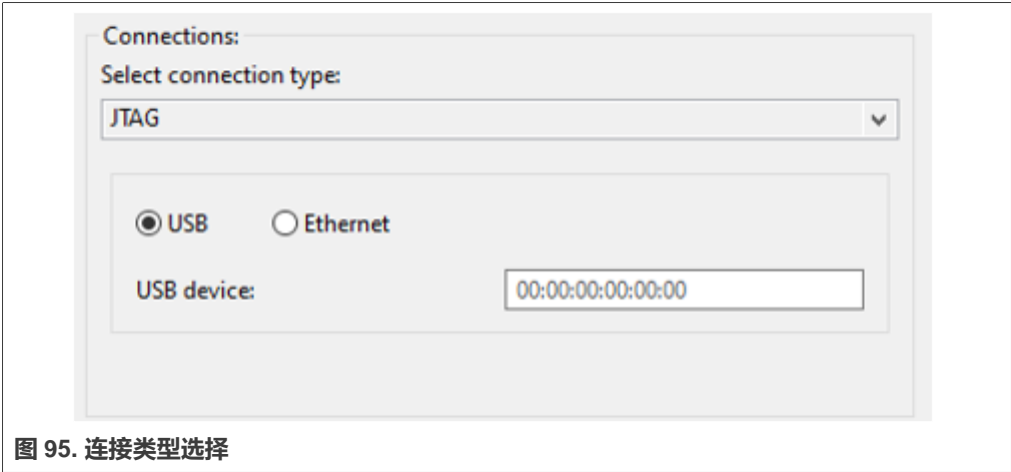
将电路板连接到主机后，使用 **COM port scan (COM 端口扫描)** 搜索 UART 端口。COM 端口下拉列表中会显示所有可用的 UART 端口。



选择正确 UART，可用作 A 核调试 UART 端口。

4.3.1.2 支持 JTAG 连接的电路板

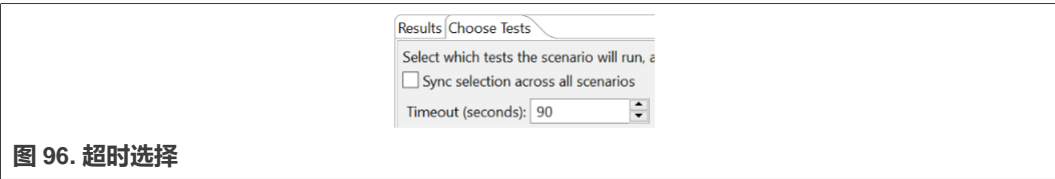
- 1. 将 JTAG 调试器（仅支持 CWTAP 调试器）连接到电路板。
- 2. 选择 USB 或以太网连接。



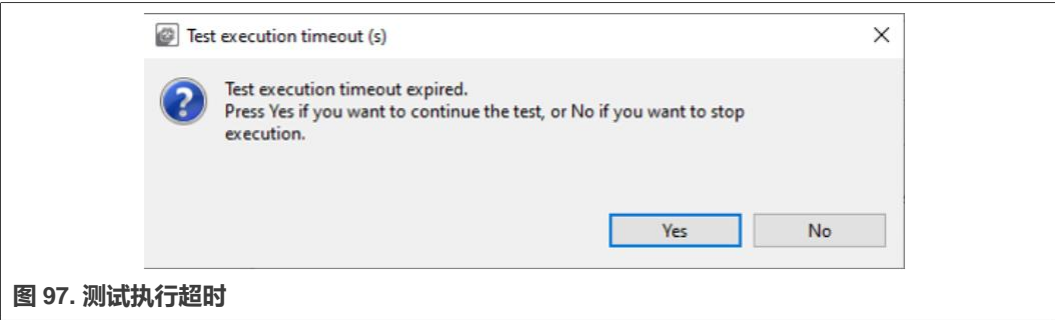
4.3.2 测试场景

设置 DDR 配置和电路板连接后，即可执行不同的 **Test scenarios（测试场景）**。您可以设置 **Test options（测试选项）** 中的参数来自定义每个测试。

根据所选的测试和选项，执行时间可能会有所不同。默认设置为 90 秒超时，以确保在出现问题时可通过超时退出测试。要更改默认值，请编辑 **Timeout (seconds)（超时(秒)）** 选项：



初始超时到期后，请输入是否继续执行测试。



要开始执行测试，请按下 “**Start Validation (开始验证)**” 按钮。您可以从 **Logs (日志)** 控制台查看运行测试的状态。默认情况下，日志级别设置为 **ERROR**。您还可以使用其他日志级别选项，它们会在控制台中显示不同的输出信息：

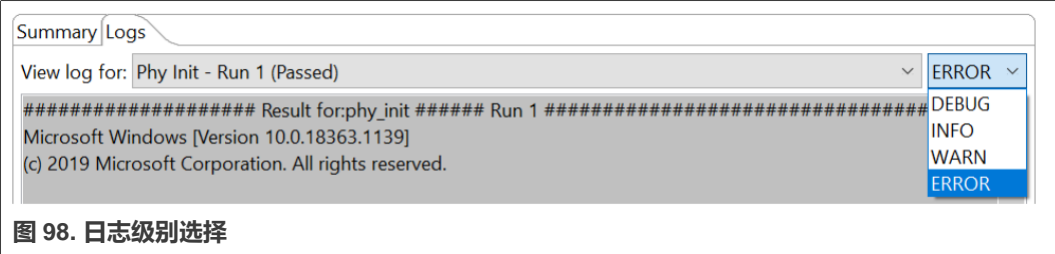


图 98. 日志级别选择

测试结束时，**Results (结果)** 中会显示 PASS/FAIL (通过/失败) 状态。**Summary (摘要)** 栏会显示测试摘要。

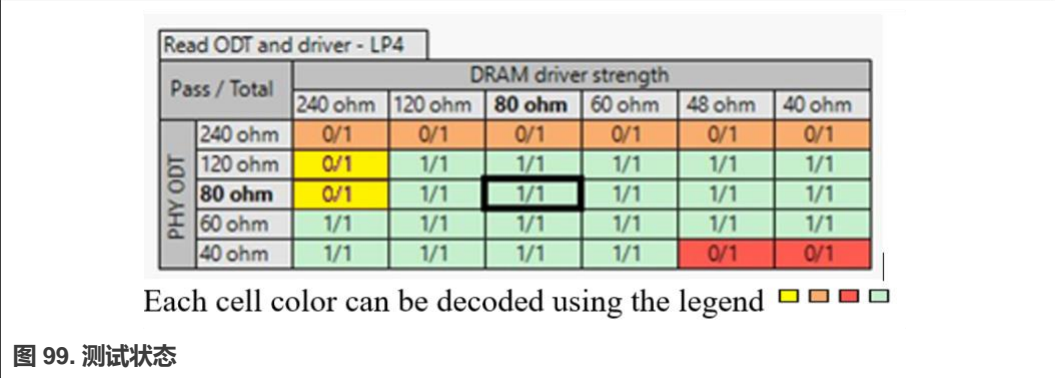


图 99. 测试状态

- 黄色表示测试失败
- 橙色表示配置错误
- 红色表示目标连接错误或脚本异常
- 绿色表示测试通过

如果测试失败，**Advisor (顾问)** 部分的 **Summary (摘要)** 视图会显示 **Symptom (症状)** 和 **Possible Causes (可能的原因)**，以便在查找 DDR 子系统问题时提供调试过程的高级指导。

Advisor	
Symptom	Possible Causes
Generic failure	1. Use the latest version of the DDR tool to generate the proper setting for the customer's configuration 2. Ensure the bit swapping or byte swapping properly follow the limitation that may apply and the DDR tool has been provide the correct ... 3. Review and verify schematics is correct 4. Verify if the used memory device is compatible with the SoC 5. Verify the proper input frequency and correct PLL configuration are applied 6. Verify the DDR layout guideline document from NXP has been followed 7. Verify the voltages on the board 8. Verify the board has been properly reset and the power up sequence 9. Check HW for manufacturing issues

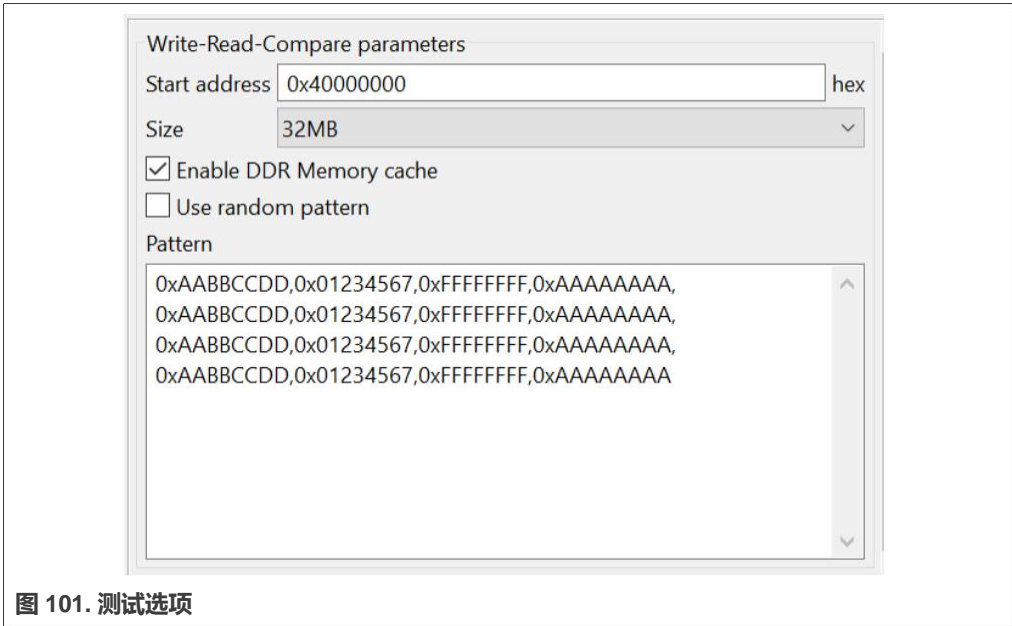
图 100. 顾问部分

DDR 工具提供了几种测试场景，可分为检查、优化、vTSA 和压力测试。

4.3.2.1 检查

通过执行以下测试，检查显示 DDR 控制器和 DDR PHY 配置的状态：

- 1. *Firmware Init (固件初始化)* – 执行 DDR PHY 培训以检查 DDR PHY 配置。
- 2. *Operational (运行)* – 运行 **Write-Read-Compare/Walking Ones/Walking Zeros (写入-读取-比较/遍历 1/遍历 0)** 测试，执行基本内存访问测试。每个测试都有起始地址、大小、启用 DDR 内存缓存、访问模式/模式选项等选项。



- 3. *ECC Regions test (ECC 区域测试)* – 用 1 位错误注入测试每个 ECC 区域，以验证区域的 ECC 能力（受保护或不受保护）。  
**注意：**该测试仅适用于支持内联 ECC 的设备。

4.3.2.2 优化

DQ ODT 和驱动强度测试扫描 DQ IO 配置，为读取和写入创建电路板特定的驱动强度与 ODT 通过/失败的映射图。

**注意：**通过 RPA 初始化脚本导入绕过用户界面配置时，无法进行优化。

Read ODT and driver							
Pass / Total		DRAM driver strength					
		240 ohm	120 ohm	80 ohm	60 ohm	48 ohm	40 ohm
PHY ODT	240 ohm	0/1	0/1	0/1	0/1	0/1	0/1
	120 ohm	0/1	1/1	1/1	1/1	1/1	1/1
	80 ohm	0/1	1/1	1/1	1/1	1/1	1/1
	60 ohm	1/1	1/1	1/1	1/1	1/1	1/1
	40 ohm	1/1	1/1	1/1	1/1	1/1	1/1

图 102. DQ ODT 和 DS 映射图

对于通过的单元格（绿色单元格），启用“在 DDR 配置中应用当前选择”选项（右击单元格）。它将各自的驱动强度和 ODT 值设置到配置中，供其他场景使用。

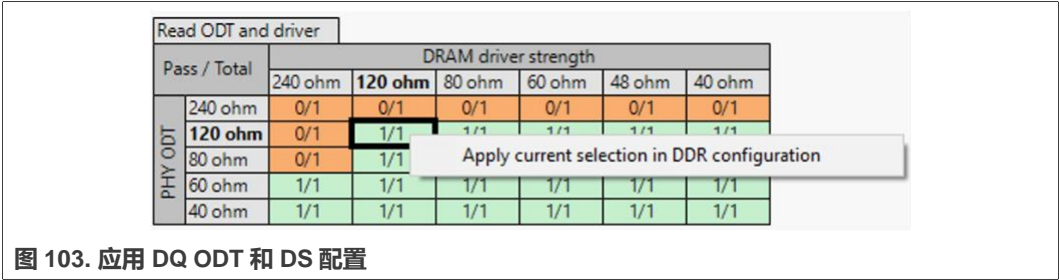


图 103. 应用 DQ ODT 和 DS 配置

**注意：**在推导最佳 ODT/驱动强度值时，可以使用驱动强度与 ODT 映射图作为一个标准。该图不能作为做出这一决定的唯一依据。

**注意：**恩智浦强烈建议使用经过我们 GA BSP 测试和验证的默认 ODT 和驱动强度值。为确保您的设计符合电路板布局要求，请参阅《i.MX 8M 硬件开发人员指南》。

1D 优化测试的 Vref（参考电压）会扫描 PHY Vref 和/或 DRAM Vref，以确定 PHY 训练失败时能够通过 PHY 训练的值，以及 2D 训练后得到的训练值。

当测试通过时，可以启用“在 DDR 配置中应用当前选择”选项。这样就可以把相应的 PHY Vref 和 DRAM Vref 值保存到配置中。

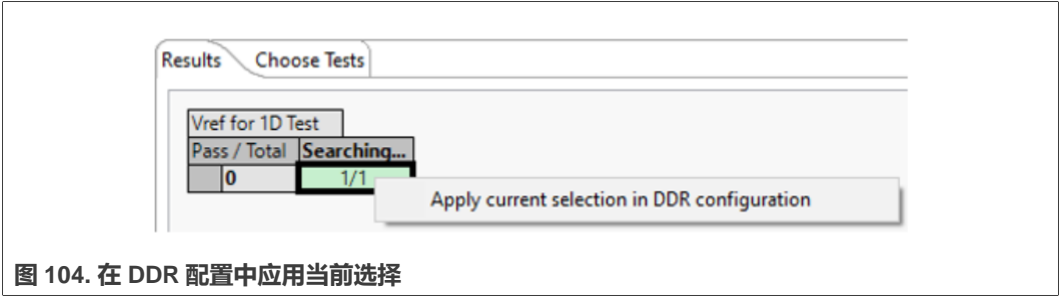


图 104. 在 DDR 配置中应用当前选择

CA 优化的 Vref 测试只进行 1D 训练，并在训练结束后读取 VrefCA 的值。

当测试通过时，可以启用“Apply the current selection in the DDR configuration (在 DDR 配置中应用当前选择)”选项，把 VrefCA 的值设置到配置中。

4.3.2.3 vTSA

vTSA 通过运行测试来确定 DDR 子系统的裕量，从而进行虚拟时序信号分析。

诊断写入裕量 (Diag Write Margin) / 诊断读取裕量 (Diag Read Margin) 测试为每个 DQ 通道绘制虚拟数据眼图。



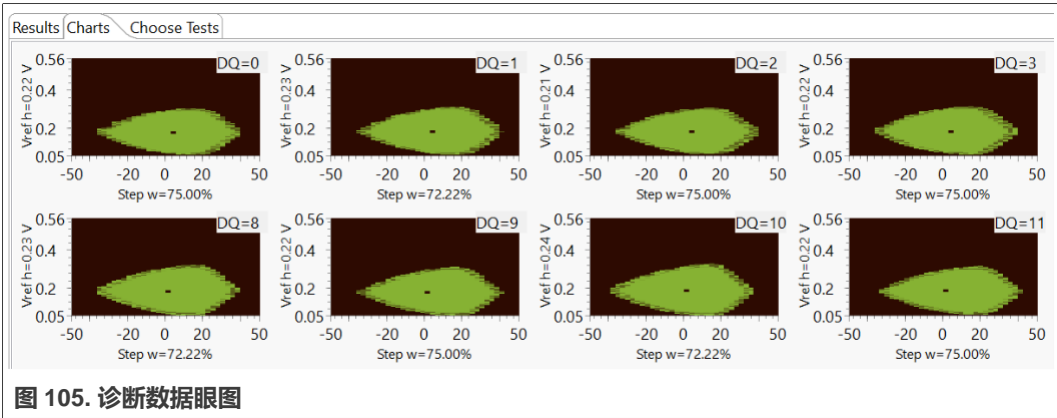


图 105. 诊断数据眼图

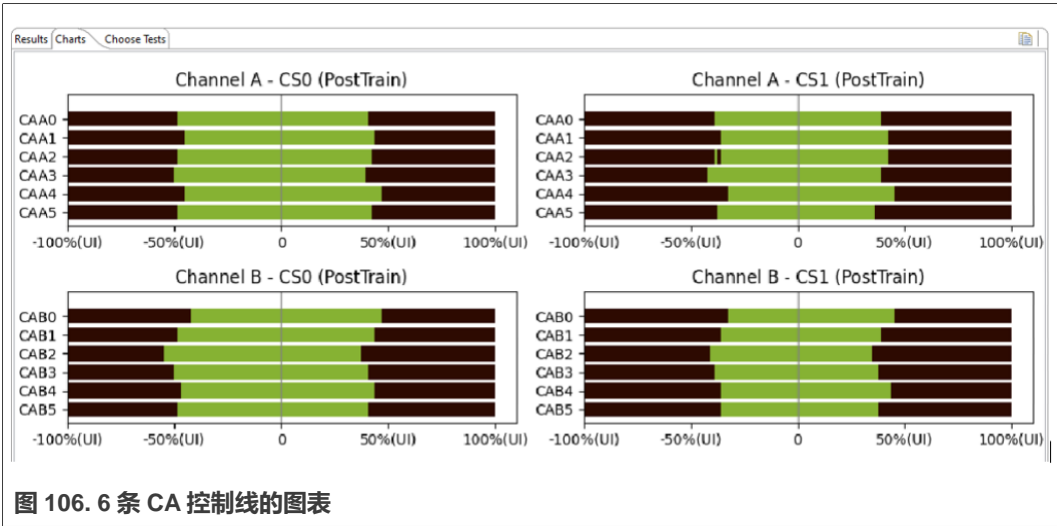
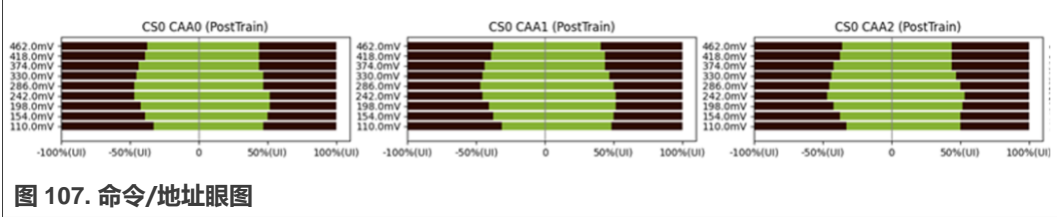


图 106. 6 条 CA 控制线的图表



**注意：**有关 vTSA 的详细信息，请参阅常见问题解答部分。

4.3.2.4 压力测试

要更深入地测试 DDR 配置的稳定性，可以使用压力测试场景，它的测试套件涵盖了不同的情况。

有两种方式可以运行压力测试：

- 1. 单次运行：选择的不同选项（大小、启用 DDR 内存缓存、失败时停止）来运行一遍测试套件。如果测试失败，可在 **Logs（日志）** 控制台中查看套件中每个测试的状态，把日志级别设为 **INFO**



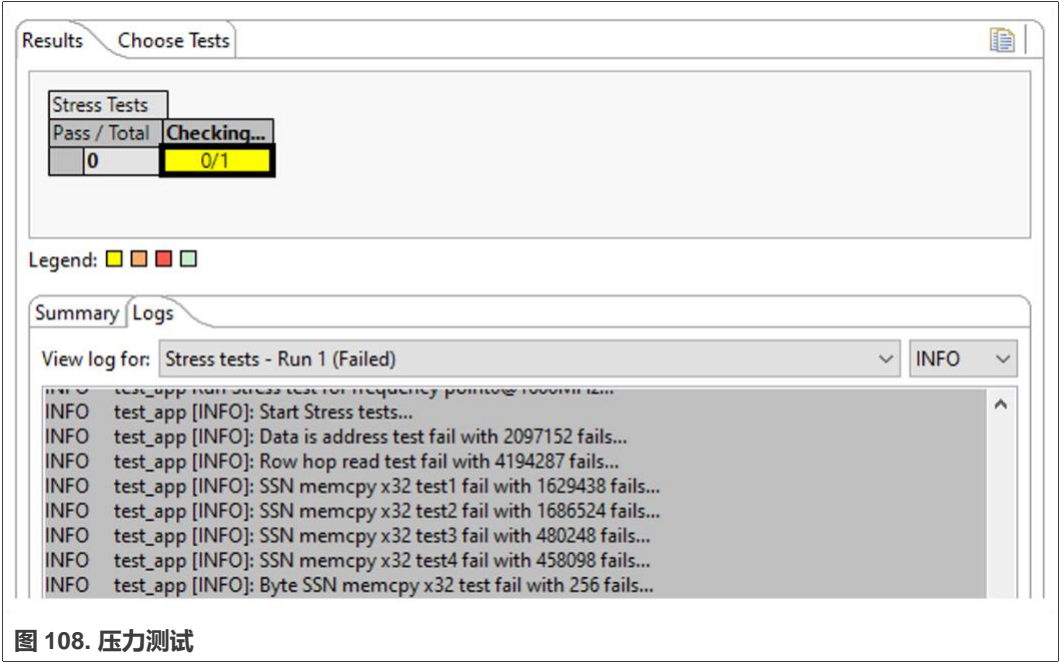


图 108. 压力测试

2. 测试持续时间在选定的时间内运行测试套件。这适合过夜测试。

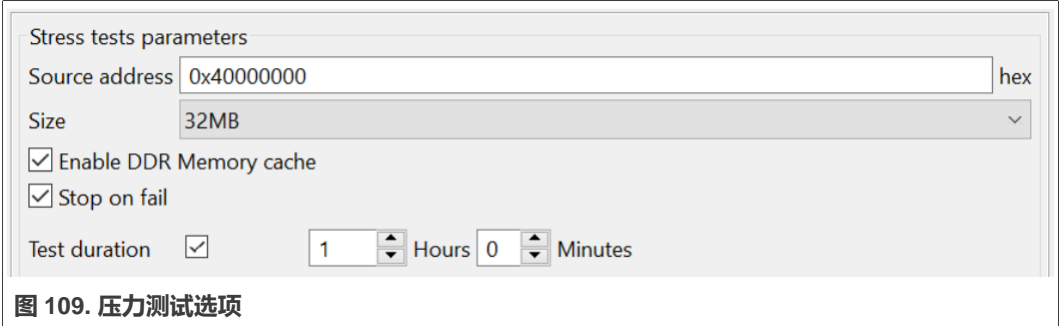


图 109. 压力测试选项

在 **Logs (日志)** 控制台中，您可以监控测试执行情况，查看迭代次数和持续时间。



图 110. 压力测试结果

**注意：** 确保 **Timeout (seconds) (超时(秒))** 设置高于 **Test duration (测试持续时间)** 设置，否则测试将超时结束。

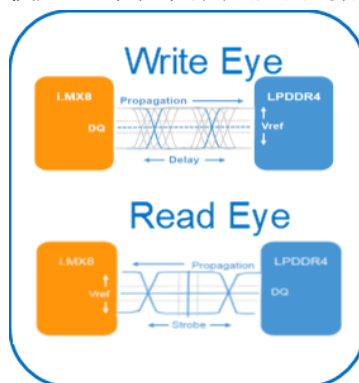
## 4.4 常见问题解答

### 1. vTSA 是什么意思？

- vTSA 是虚拟时序信号分析 (Virtual Timing Signal Analysis) 的缩写。
- 它是使用内存控制器本身来测试裕量的方法，而不需要外部的测试设备。“虚拟”并不意味着模拟！
- 内存控制器可以调整时序、电压基准、终端阻抗等参数，优化输入和输出信号。
- “训练”是一个过程，内存控制器可扫描这些参数并找到具有最大裕量的配置。
- vTSA 只需记录并输出这些信息，这可以让我们深入了解系统的信号裕量情况，而无需使用测试设备。
- 初始化和校准设置也可转储到文件中进行分析。

### 2. vTSA 输出是什么？

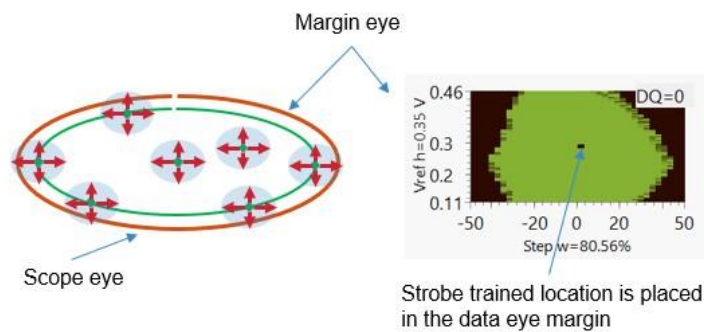
- 虚拟时序信号分析 (vTSA) 通过运行一系列写入/读取操作来生成写入和读取数据眼图，而不是采用传统的硬件方法，即使用高速示波器来进行手动物理 TSA (pTSA) 测量。
- 因此，vTSA 输出本身近似于实际的写入/读取眼图。
- 您可能会发现延迟线和 VREF 的训练值与 vTSA 报告的这些值相比会有一些差异。
- vTSA 只报告它在报告眼图中“最宽”部分检测到的值，这些值可能会因每次运行的不同而有所变化。
- 使用此工具的关键是展示虚拟写入/读取眼图，让用户相信其设计的稳健性。



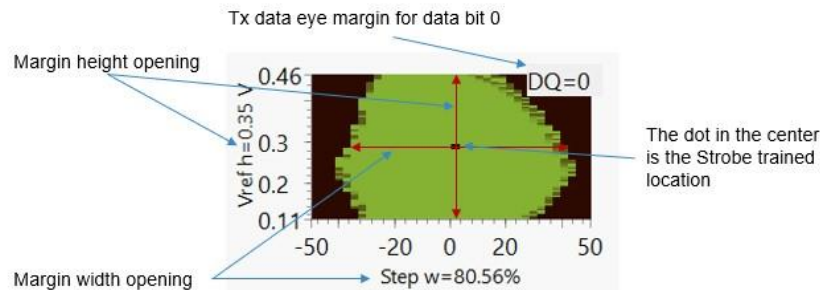
### 3. 需要有关 vTSA 的更多信息吗？

- vTSA 工具测出的是实际 pTSA 的近似值。因此，您可能会注意到训练的延迟线值与 vTSA “中” 值之间存在一些偏差。据观察和预计，这种偏差可能高达 ~20 ps。从生成的眼图中关注的重点是验证数据眼图中训练的延迟线值是否有充足裕量。
- 对于训练的 VREF 值，LPDDR4 器件模式寄存器 14 (MR14，保存了训练的 VREF 值) 适用于所有字节通道。也就是说，根据 JEDEC 规定，每个字节通道没有 MR14，而是 MR14 适用于所有字节通道。

- c. 对于符合恩智浦 DDR 布局指南的电路板，VREF 训练值周围应留有充足的裕量。  
您可以在相应的《恩智浦硬件开发指南》中找到这些指南。
4. 数据眼图裕量是如何生成的？
- a. 我们可以使用多个延迟步长来调整每个 DQ 和 DQS 的位置。
  - b. 当 DQ 位置以零单位时间间隔 (UI) 步长延迟到 1 单位时间间隔 (UI) 步长延迟跨越 DQS 时，我们会对每种步长进行写入-读取-比较测试来判断是否通过测试。
  - c. 测试会在所有可用的 VREF 步长中重复 DQ 以单位时间间隔穿越的过程 (b)。
  - d. 测试 b 可以生成一条线，在每个 VREF 步长下重复生成这些线就能生成数据眼图裕量。
  - e. DQS 信号与训练的 VREF 和延迟步长的交点放在生成的数据眼图裕量中。
  - f. 裕量眼图中每个**通过**点都已满足设置、保持和电压要求。



5. 数据眼图旁边的信息是什么？
- a. 单位时间间隔 = 1/数据速率；例如，在 3200MT/s 数据速率下，单位时间间隔 = 312.5 ps
  - b. x 轴表示时间。它是以百分比表示的单位时间间隔，范围为-50%至+50%
  - c. x 轴数据眼图裕量宽度开口以单位时间间隔的百分比表示。例如：步骤 w=80.56% 的 UI
  - d. y 轴表示电压。
  - e. y 轴开放数据眼图裕量高度/幅度开口以电压显示。例如：Vref h=0.35 V



6. 多大的裕量才算是好的？
- a. 要确定所需的裕量掩码，您必须执行以下操作：
    - 优化 DDR 接口

- 优化设置后，在最恶劣的情况下（温度、电压、频率、模式），生成客户电路板 DDR 接口的最恶劣情况的数据眼图裕量。
  - 使用 DDR 训练优化/居中选通信号/眼图裕量
- b. 数据眼图裕量中的绿色像素表示通过的单元格。这意味着该绿色像素的设置和保持时间以及 VIH<sub>Lac</sub>/dc 都达标了。
  - c. 数据眼图裕量中选通信号位置周围的任何其他绿色像素都是 DDR 为此 DQ 提供的额外裕量。
7. 为什么训练的 Vref 有时不在数据眼图正中心？
- a. VREF 训练必须选择一个值，该值对应通过 VREF 范围的所有字节通道，并将该值写入 LPDDR4 MR14 寄存器中。这意味着对于单字节通道，虽然训练的 VREF 值可能看起来并不在数据眼图正中心，但它是为了给这个字节通道和其他字节通道提供最佳的裕量而选择的。
8. 如何检查是否选择了错误的 UART？

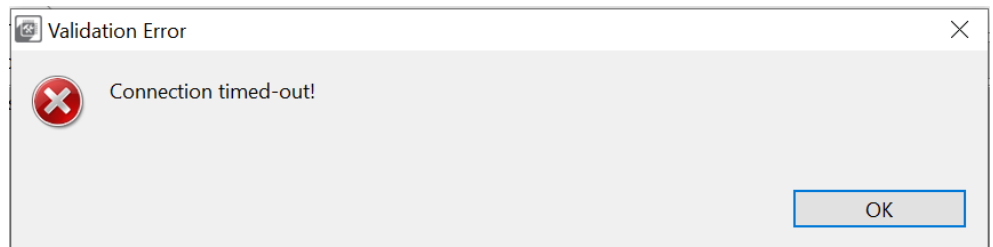
将日志级别设置为 DEBUG，然后查看控制台中的信息

```
File "C:\nxp\i.MX_CFG_v9\bin\python38\serial\serialwin32.py", line 62, in open
    raise SerialException("could not open port {!r}: {!r}".format(self.portstr, ctypes.WinError()))
serial.serialutil.SerialException: could not open port 'COM4': PermissionError(13, 'Access is denied.', None, 5)
```

或者

```
Traceback (most recent call last):
  File "C:\ProgramData\NXP\mcu_data_v9\processors\MIMX8MM4xxxKZ\ksdk2_0\mem_validation\ddrc\scripts\common\base_test.py", line 224,
    assert self.is_waiting_for_input()
AssertionError
```

9. 如果测试超时怎么办？
- a. 如果测试超时，会弹出以下窗口



- b. 增加超时（秒）选项并重新运行测试。如果出现以下错误，请关闭电路板电源，拔下 UART 电缆，然后打开电路板电源，插入 UART 电缆

```
Traceback (most recent call last):
  File "C:\ProgramData\NXP\mcu_data_v9\processors\MIMX8MM4xxxKZ\ksdk2_0\mem_validation\ddrc\scripts\common\base_test.py", line 224,
    assert self.is_waiting_for_input()
AssertionError
```

## 5 可信执行环境工具

采用 **Trusted Execution Environment（可信执行环境，TEE）** 工具，您可以配置内存区域、总线主控和外设的安全策略，以隔离和保护该应用程序的敏感区域。

您可以在 **Security Access Configuration（安全访问配置）** 及其子视图中设置应用程序不同部分的安全策略，并在 **Memory Attribution Map（内存属性映射）**，**Access Overview（访问概览）** 和 **Domains Overview（域概览）** 视图中查看这些策略。

使用 **User Memory Regions（用户内存区域）** 视图可以方便地概览内存区域及其安全级别。

您还可以在 **Registers (寄存器)** 视图中查看 TEE 工具处理的寄存器，并在 **Code Preview (代码预览)** 工具中检查代码。

**注意:** 为使您的配置生效, 请确保在 **Security Access Configuration (安全访问配置)** 视图的 **Miscellaneous (杂项)** 子视图中勾选相应的“启用安全检查”选项。

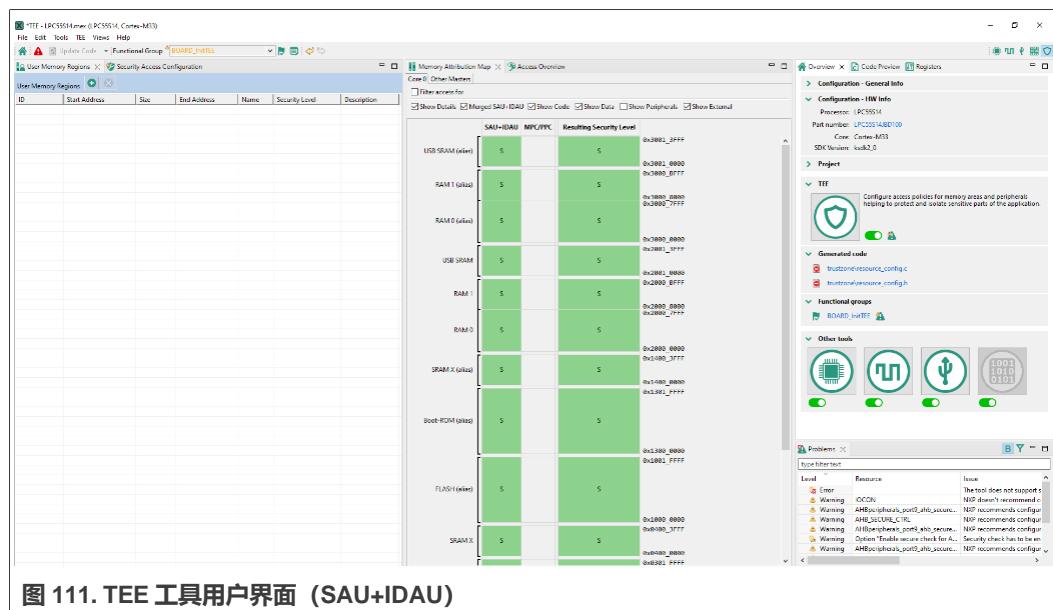


图 111. TEE 工具用户界面 (SAU+IDAU)

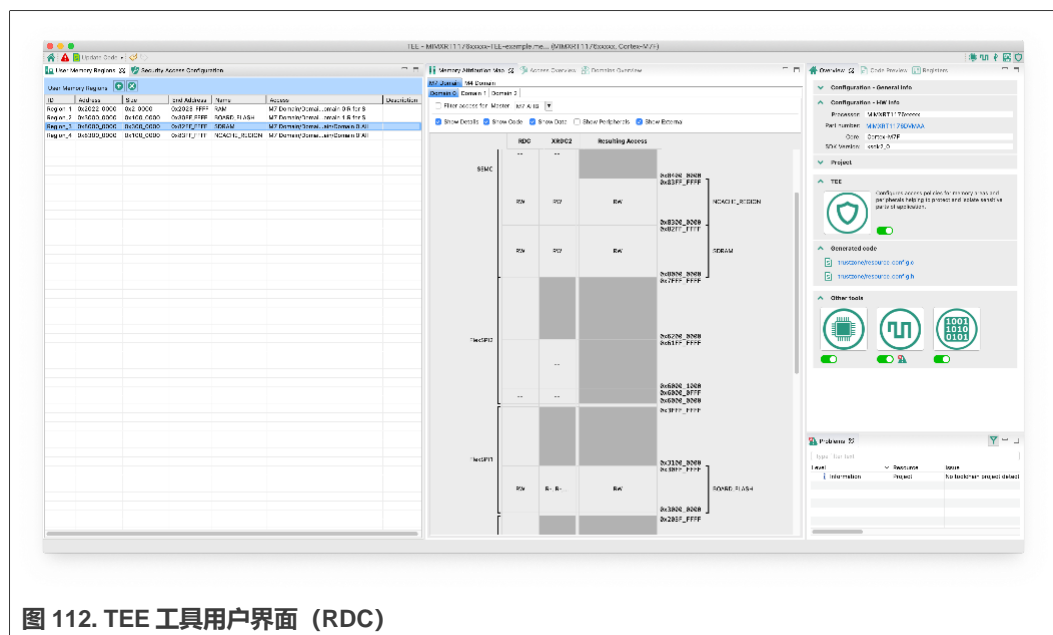


图 112. TEE 工具用户界面 (RDC)

## 5.1 AHBSC 支持安全扩展功能的器件

TEE 工具的功能和外观基于加载器件的安全模型。

本节介绍该工具的功能和外观，它适用于通过 AHBSC (AHB 安全控制器) 支持 TrustZone-M 安全扩展的器件。

它目前支持以下类型的器件：

- LPC55Sxx
    - LPC55S69, LPC55S66
    - LPC55S16, LPC55S14
    - LPC55S06, LPC55S04
  - RT6xx、RT5xx
    - MIMXRT685S, MIMXRT633S
    - MIMXRT595S, MIMXRT555S, MIMXRT533S1q
- 注意：仅限预生产。*

### 5.1.1 用户内存区域视图

在 **User Memory Regions (用户内存区域)** 视图中，可以创建和维护内存区域及其安全级别的高级配置。您可以创建区域、为其命名、指定其地址、大小和安全级别，并为其添加说明。然后，您可以借助 **Problems (问题)** 视图修复设置中的任何错误。

单击视图标题栏中 **Add new memory region button (添加新内存区域按钮)**，创建新内存区域。

单击该行的单元格，输入/更改内存区域的参数。在 **Security Level (安全级别)** 列中，有以下选项可供选择：

- **NS-User** - 非安全用户
- **NS-Priv** - 非安全特权
- **S-User** - 安全用户
- **S-Priv** - 安全特权
- **NSC-User** - 非安全可调用用户
- **NSC-Priv** - 非安全可调用特权
- **任何级别**

配置错误会在相关单元格中以红色图标高亮显示。如果问题可以轻松修复，可以右击该单元格，显示所提供解决方案的下拉列表。

选择表格行并单击视图标题栏中的 **Remove selected memory region(s) (删除所选内存区域)** 按钮，即可删除内存区域。

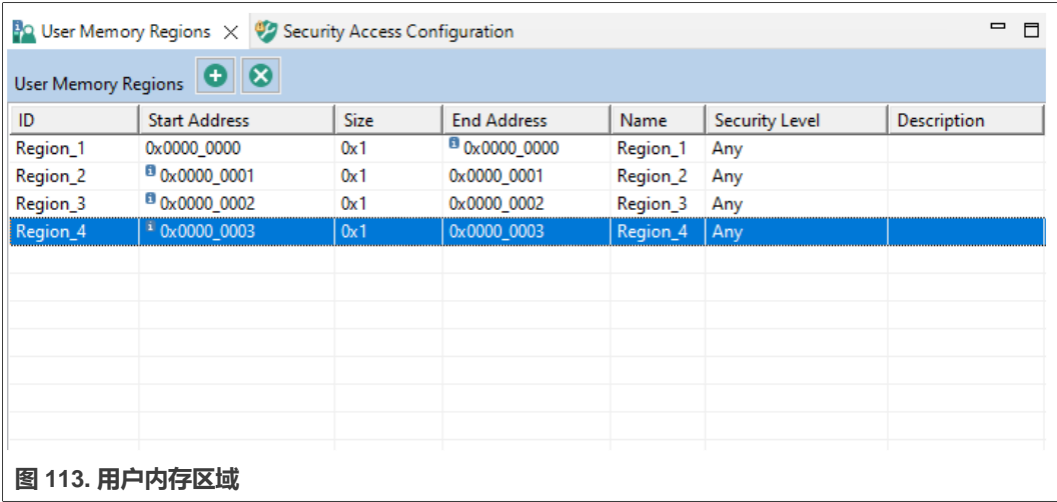


图 113. 用户内存区域

5.1.2 安全访问配置视图

在 **Security Access Configuration (安全访问配置)** 视图中，您可以通过多种方式配置应用程序的安全策略。如需了解有关详情，请参阅以下章节。

5.1.2.1 SAU

在 SAU 子视图中，您可以启用并配置 SAU（安全属性单元）。

启用后，您可以设置 SAU 内存区域，指定其起始地址、大小或结束地址，并指定其访问级别。禁用时，SAU 会自动将整个内存空间设置为安全访问级别。启用后，SAU 会将每个未覆盖（即未配置）的内存区域视为安全区域，因此只能为已覆盖（已配置）的内存区域选择 NS 或 NSC。

您可以在两个访问级别之间选择：

- **NS** - 非安全
- **NSC** - 非安全可调用

或者，您也可以通过选择 **All Non-Secure (全部非安全)** 选项，将所有 SAU 内存区域设置为非安全访问级别。

**注意：**此选项仅在禁用 SAU 时才可用。

您还可以勾选 **Generate sources for disabled regions (为禁用区域生成源代码)** 选项，决定是否禁用的内存区域生成代码。



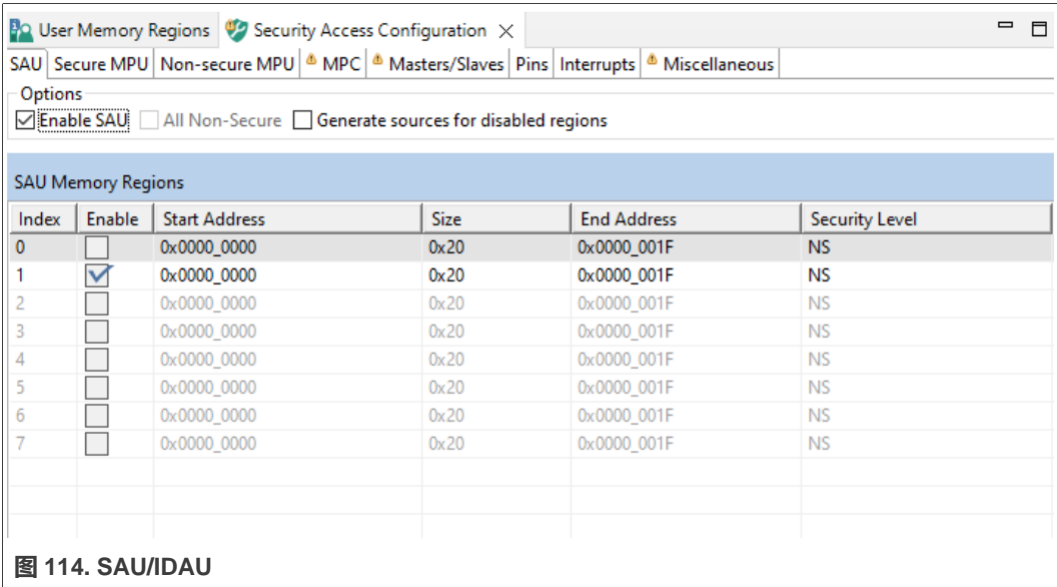


图 114. SAU/IDAU

5.1.2.2 中断

在 **Interrupts (中断)** 子视图中，可以为设备外设中断设置安全属性。如果处理器包含多个内核或处理单元，则可能会出现额外的 **Handling by Core (按内核处理)** 表格。在这些表格中，您可以指定来自外设的中断是否可以由内核或处理单元处理。

默认情况下，所有中断都被设置为 **Secure (安全)**。如果要更改中断源的安全属性，请左键单击中断的 **Secure (安全)** 单元格，然后从下拉菜单中进行选择。或者，右键单击中断的 **Name (名称)** 单元格，从弹出菜单中选择安全属性。要选择多个条目，请按住 **“Ctrl+左键单击”** 快捷键，然后右键单击所选区域来显示弹出菜单。或者，在选择一行后按住 **Shift+Up/Down** 键来扩大选择范围。



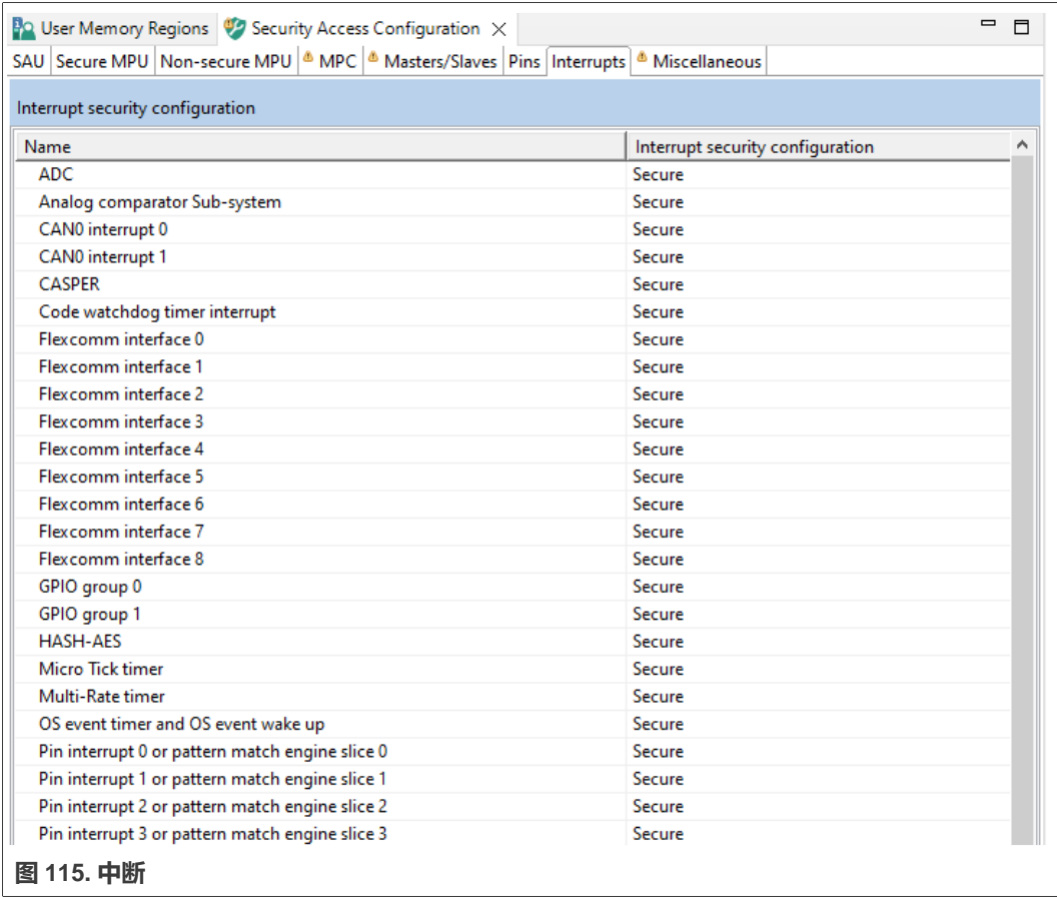
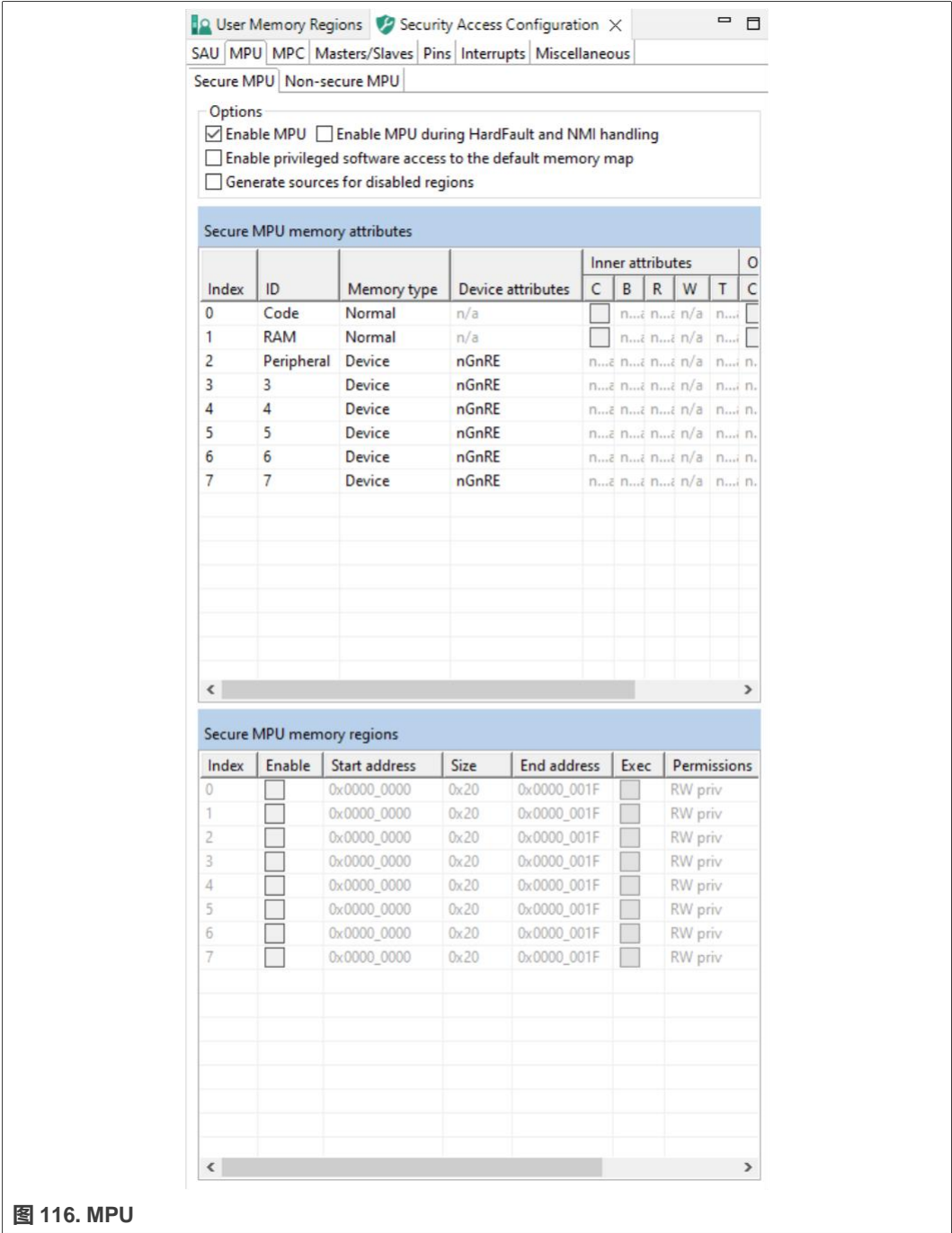


图 115. 中断

5.1.2.3 安全/非安全 MPU

在 **Secure MPU (安全 MPU)** 和 **Non-secure MPU (非安全 MPU)** 子视图中，可以启用和配置 MPU（内存保护单元）。您可以创建区域，指定其地址、大小和其他参数。使用 **安全 MPU** 子视图配置安全级别，使用**非安全 MPU** 配置非安全安全级别。



MPU 默认是禁用的，必须勾选 **Enable MPU（启用 MPU）** 选项来启用。

**注意：**并不是每个设备都支持 MPU。

使用 **MPU Memory Attributes（MPU 内存属性）** 表格来命名和配置 MPU 内存属性集。单击 **Memory Type（内存类型）** 和 **Device Attributes（设备属性）** 列的单元格，显示可用选项。

使用 **MPU Memory Regions（MPU 内存区域）** 表格启用和配置 MPU 内存区域。

1. **Enable（启用）** 区域。
2. 指定 **Address（地址）** 。

3. 指定 **Size (大小)** 或 **End Address (结束地址)**。
4. 如果希望区域能够运行代码，请设置 **Exec (执行)** 选项。
5. 设置 **Permissions (权限)** (只读或读/写)。
6. 设置 **Privileges (特权)**。  
*注意：可通过勾选 **Enable privileged software access to the default memory map (启用特权软件访问默认内存映射)** 选项，为所有非 MPU 处理的内存区域设置默认的特权访问。*
7. 设置 **Shareability (共享性)** 或缓存选项。
8. 从 **Mem.Attr.** 中分配 **MPU Memory Attributes (MPU 内存属性)** 表格中的一个内存集。  
一个内存集可分配给多个区域。

#### 5.1.2.4 MPC

在 **MPC** (内存保护检测工具) 子视图中，您可以根据物理地址定义为整个内存扇区设置安全策略。

左键单击 **Security level (安全级别)** 列中的相关单元格并从下拉列表中选择，即可设置内存扇区的安全级别。或者，也可以右键单击 **Sector (扇区)** 列中的相关单元格，然后从弹出菜单中选择安全级别。要选择多个条目，请使用 **Ctrl+左键单击** 快捷键，然后右键单击所选区域以查看弹出菜单。

有四种安全级别可供选择，按照安全级别从低到高排序：

- **NS-User** - 非安全用户
- **NS-Priv** - 非安全特权
- **S-User** - 安全用户
- **S-Priv** - 安全特权

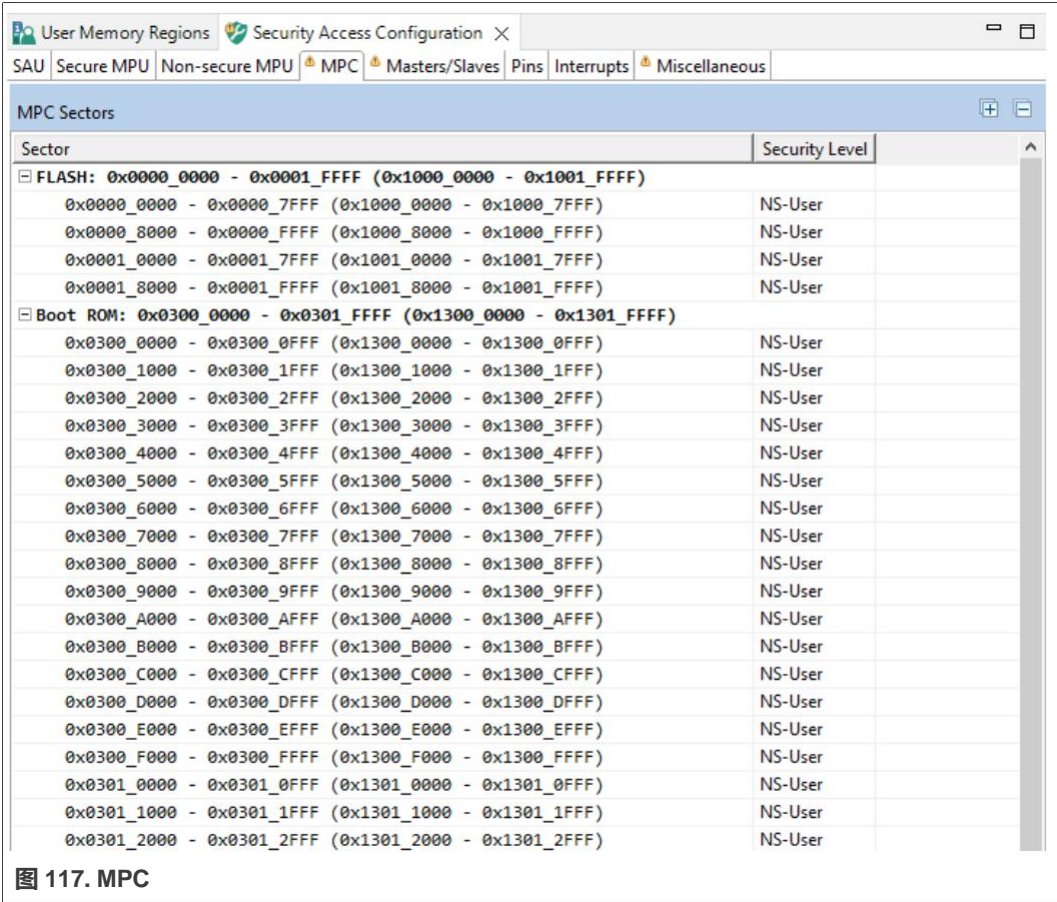


图 117. MPC

5.1.2.5 主设备/从设备

在 **Masters/Slaves (主设备/从设备)** 子视图中，您可以为总线主设备和从设备配置安全级别。

左键单击 **Security level (安全级别)** 列中的相关单元格并从下拉列表中选择，即可设置总线主设备/从设备的安全级别。或者，右击 **Master (主设备)** 和 **Slave (从设备)** 列中的相关单元格，从弹出菜单中选择安全级别。要选择多个条目，请按住 **Ctrl+左键单击** 快捷键，然后右键单击所选区域以查看弹出菜单。

有四种安全级别可供选择，按照安全级别从低到高排序：

- **NS-User** - 非安全用户
- **NS-Priv** - 非安全特权
- **S-User** - 安全用户
- **S-Priv** - 安全特权

您可以通过选择以下选项进一步指定主设备和从设备安全级别之间的相互关系：

- **Simple Master in Strict Mode (严格模式下的简单主设备)** - 选择此选项，只允许简单总线主设备在同一级别上读写。取消选择后，允许在同一级别和较低级别上读写。
- **Smart Master in Strict Mode (严格模式下的智能主设备)** - 选择该选项，允许智能总线主设备仅在同一级别执行、读取和写入存储器。取消选择后，仅允许在同一级别上执行，在同一级别和较低级别上读写。

**注意：**指令类型总线主设备安全级别必须与总线从设备安全级别相同。数据和其他安全级别必须不低于总线从设备安全级别。

User Memory RegionsSecurity Access Configuration X

SAUSecure MPUNon-secure MPUMPCMasters/SlavesPinsInterruptsMiscellaneous

Options

☒ Simple Master in Strict Mode☒ Smart Master in Strict Mode

Master	Security Level	Slave	Security Lev ^
Simple master			
CANFD	NS-User	ADC0	NS-User
DMA0	NS-User	AHB_SECURE_CTRL	NS-User
DMA1	NS-User	ANACTRL	NS-User
HASHCRYPT	NS-User	CAN0	NS-User
USBFS	NS-User	CASPER	NS-User
USBFSH	NS-User	CRC_ENGINE	NS-User
		CTIMER0	NS-User
		CTIMER1	NS-User
		CTIMER2	NS-User
		CTIMER3	NS-User
		CTIMER4	NS-User
		DBGMAILBOX	NS-User
		DMA0	NS-User
		DMA1	NS-User
		FLASH	NS-User
		FLEXCOMM0	NS-User
		FLEXCOMM1	NS-User
		FLEXCOMM2	NS-User
		FLEXCOMM3	NS-User
		FLEXCOMM4	NS-User
		FLEXCOMM5	NS-User
		FLEXCOMM6	NS-User
		FLEXCOMM7	NS-User
		GINTE	NS-User

图 118. 主设备/从设备

5.1.2.6 引脚

在 **Pins (引脚)** 子视图中，您可以指定是否允许读取 GPIO 状态。

所有引脚的 GPIO 读取状态默认设置为 **Allow (允许)**。如果要更改引脚读取 GPIO 状态，请左键单击引脚的 **Reading GPIO state (读取 GPIO 状态)** 单元格，然后从下拉菜单中选择。或者，右键单击引脚的 **Name (名称)** 单元格，从弹出菜单中选择读取 GPIO 状态。要选择多个条目，请按住快捷键 **Ctrl+左键单击**，然后右击所选区域查看弹出菜单。或者，也可以在选择一行后按住 **Shift+Up/Down** 来扩大选择范围。

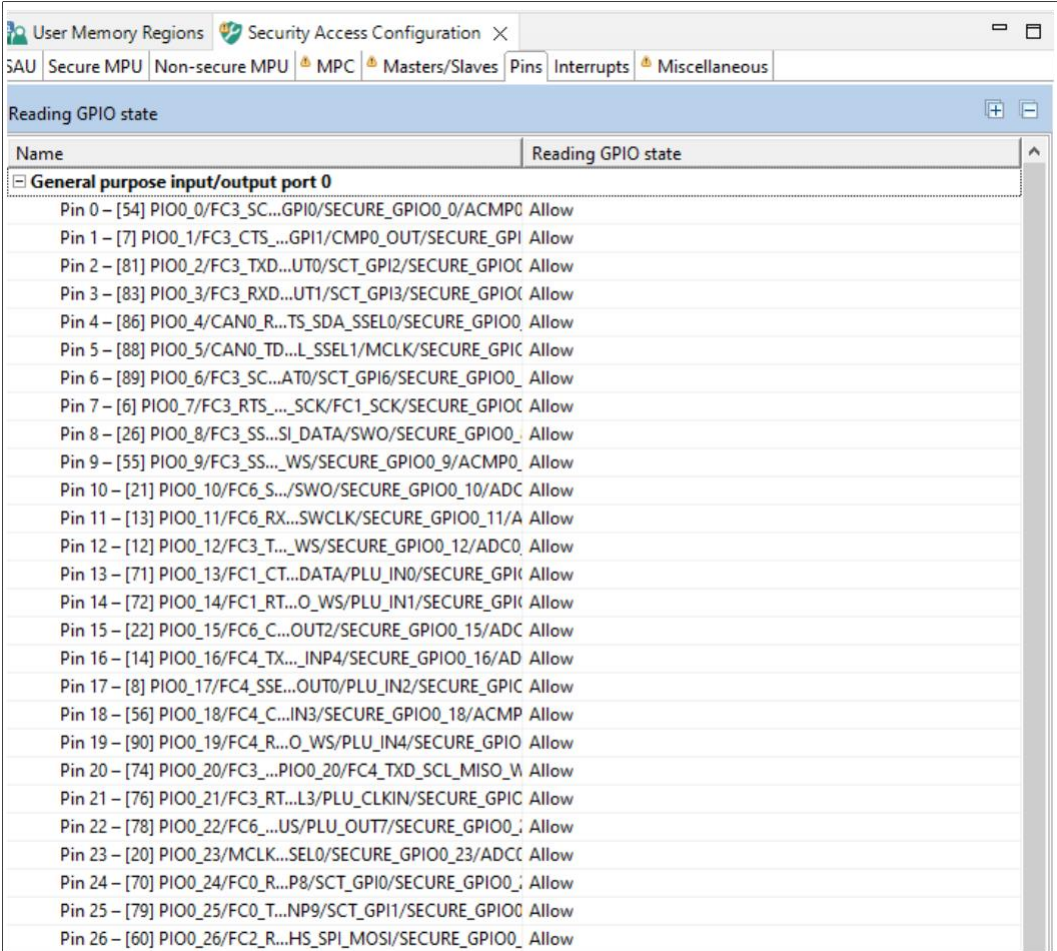


图 119. LPC55S69 上的引脚选项卡

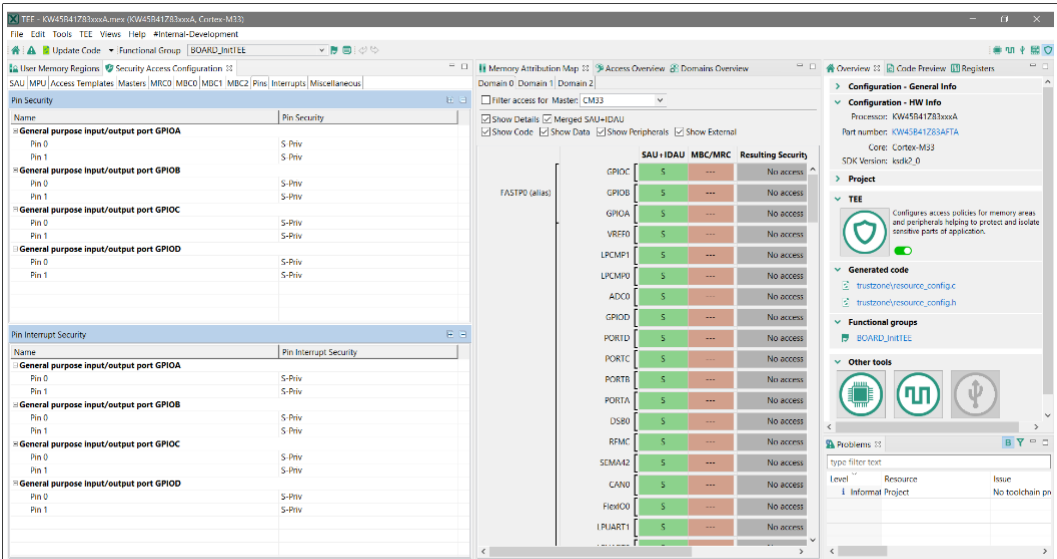


图 120. KW45 上的引脚选项卡



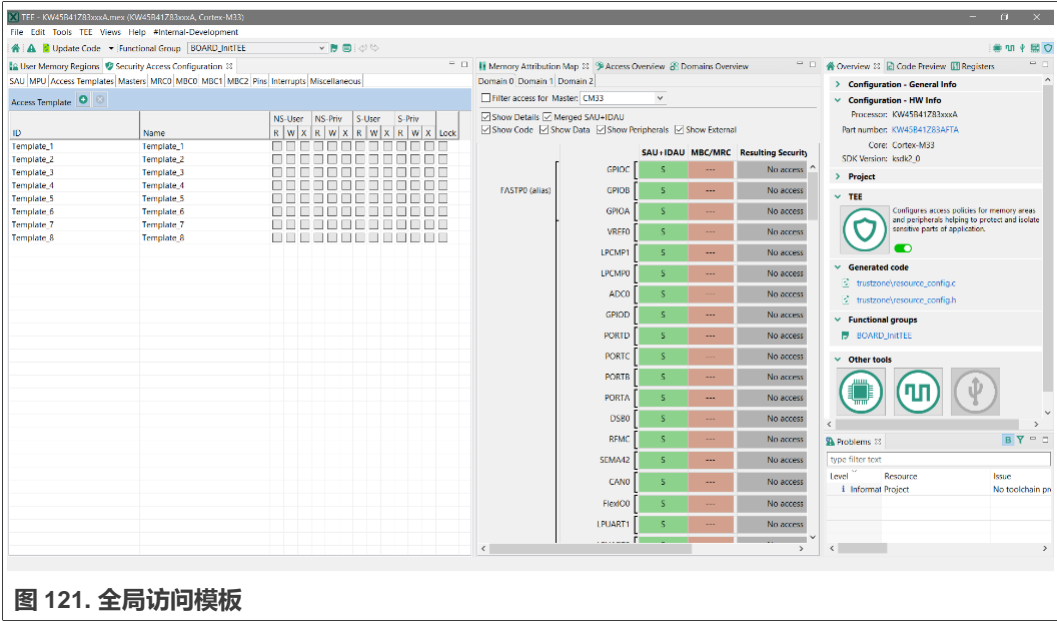


图 121. 全局访问模板

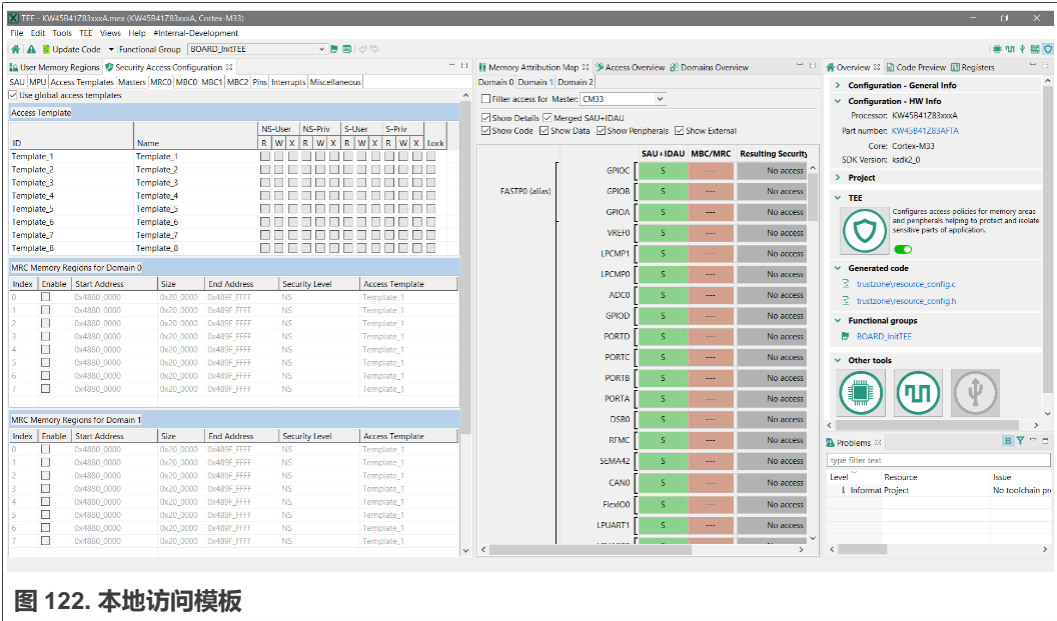


图 122. 本地访问模板

5.1.2.7 杂项

在 **Miscellaneous (杂项)** 子视图中，您可以设置各种配置选项。这些选项的列表取决于处理器数据，并且差异很大。所有选项都会影响寄存器设置，您可以在 **Register (寄存器)** 视图中查看它们。只有部分选项会直接影响您在 **Security Access Configuration (安全访问配置)** 视图中所做的配置。将光标指向各个选项可显示工具提示，解释每个选项的功能。

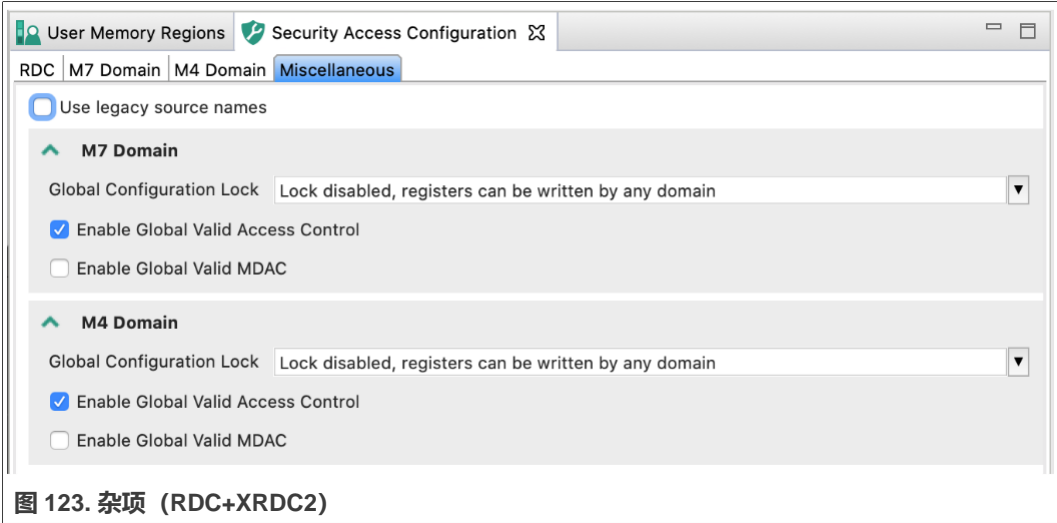


图 123. 杂项 (RDC+XRDC2)

5.1.3 内存属性映射

在 **Memory attribution map (内存属性映射)** 中，您可以查看为内存区域设置的安全级别。此视图为只读视图。

5.1.3.1 内核 0

在 **Core 0** 子视图中，您可以查看为代码、数据和外设内存区域设置的 Core 0 安全级别。该表格为只读。

**Access by Master (按主设备访问)** 表格显示 **MSW** 或 **SAU+IDAU**、**MPC** (内存保护检测工具) 安全级别，以及所列代码、数据和外设内存区域的 **Resulting access level (结果访问级别)** 状态，以及它们的物理地址。

要设置显示选项，请按照以下步骤操作：

- 1. 单击 **Filter access for (过滤访问)** 复选框启用过滤选项。
- 2. 从 **Master (主设备)** 下拉菜单中选择要查看的主设备安全访问。
- 3. (可选) 也可从 **Security mode (安全模式)** 下拉菜单中选择，设置所选主设备的安全级别。这个设置不会影响配置。
- 4. (可选) 取消选择 **Show details (显示详情)** 和 **Merged SAU+IDAU (合并 SAU+IDAU)** 选项，可自定义输出。
- 5. (可选) 在 **Filter (过滤器)** 区域过滤显示的内存区域。

将光标指向彩色编码单元格，可显示一个工具提示，说明安全级别组合的信息。

双击单元格，打开 **Security Access Configuration (安全访问配置)** 中的相关设置。



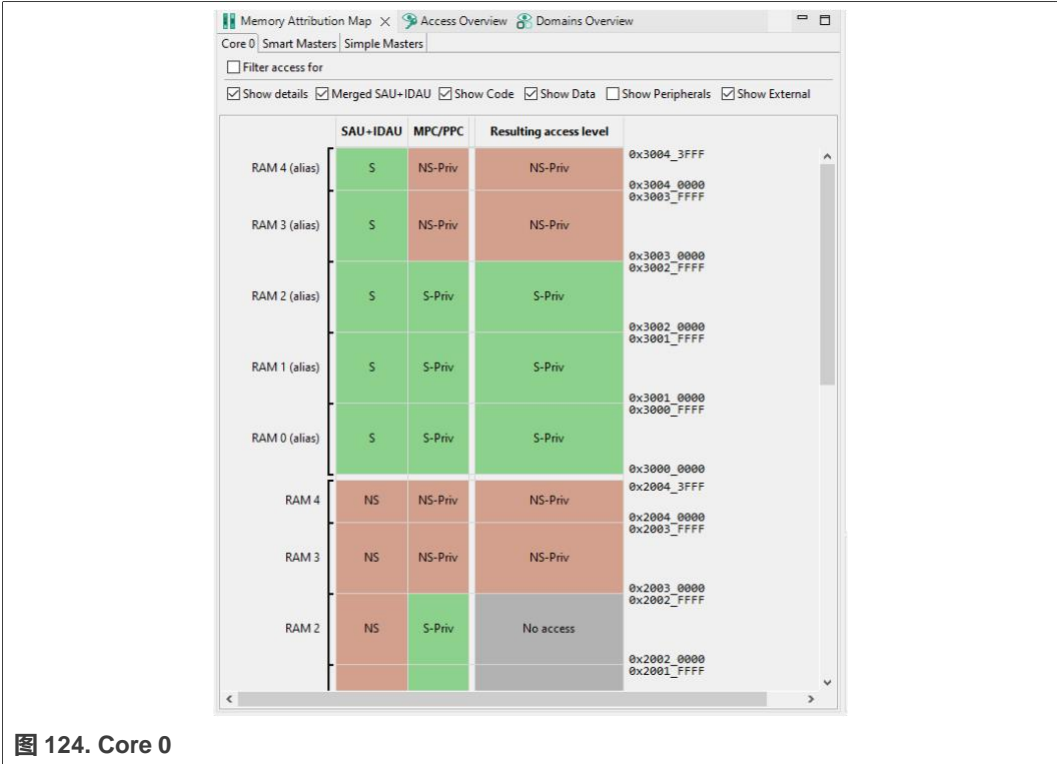


图 124. Core 0

5.1.3.2 简单主设备和智能主设备

在 **Simple Masters (简单主设备)** 和 **Smart Masters subviews (智能主设备)** 子视图中，可以查看存储器与简单/智能主设备访问权限相关的安全属性。该表格为只读表格。要设置显示选项，请按照以下步骤操作：

- 1. 单击 **Filter access for (过滤访问)** 复选框启用过滤选项。
- 2. 从 **Master (主设备)** 下拉菜单中选择要查看的主设备类型安全访问。
- 3. (可选) 取消选择 **Show Details (显示详细信息)**、**Show Code (显示代码)**、**Show Data (显示数据)**、**Show Peripherals (显示外设)** 和 **This Domain Only (仅限此域)** 选项来自定义输出。
- 4. (可选) 在 **Filter (过滤器)** 区域过滤显示的内存区域。

将光标指向彩色编码字段，可显示一个工具提示，说明安全级别组合的信息。

双击单元格，打开 **Security Access Configuration (安全访问配置)** 中的相关设置。

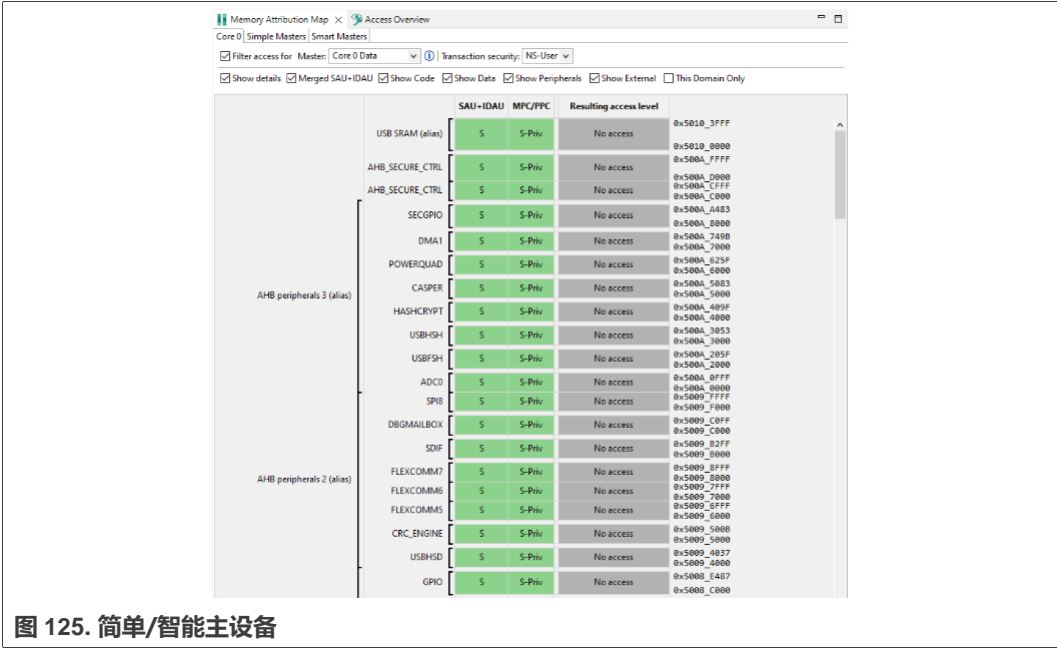


图 125. 简单/智能主设备

5.1.4 访问概览

在 **Access Overview（访问概览）** 中，可以查看在 **Security Access Configuration（安全访问配置）** 视图中设置的安全策略。

纵轴显示所有主设备，并按其安全设置分为彩色编码组。

横轴显示内存范围和从设备总线/外设。

将光标指向某个条目，就会显示有关该条目信息的工具提示。

您可以使用工具栏右侧的按钮，按安全性或主设备对显示的信息进行分组。

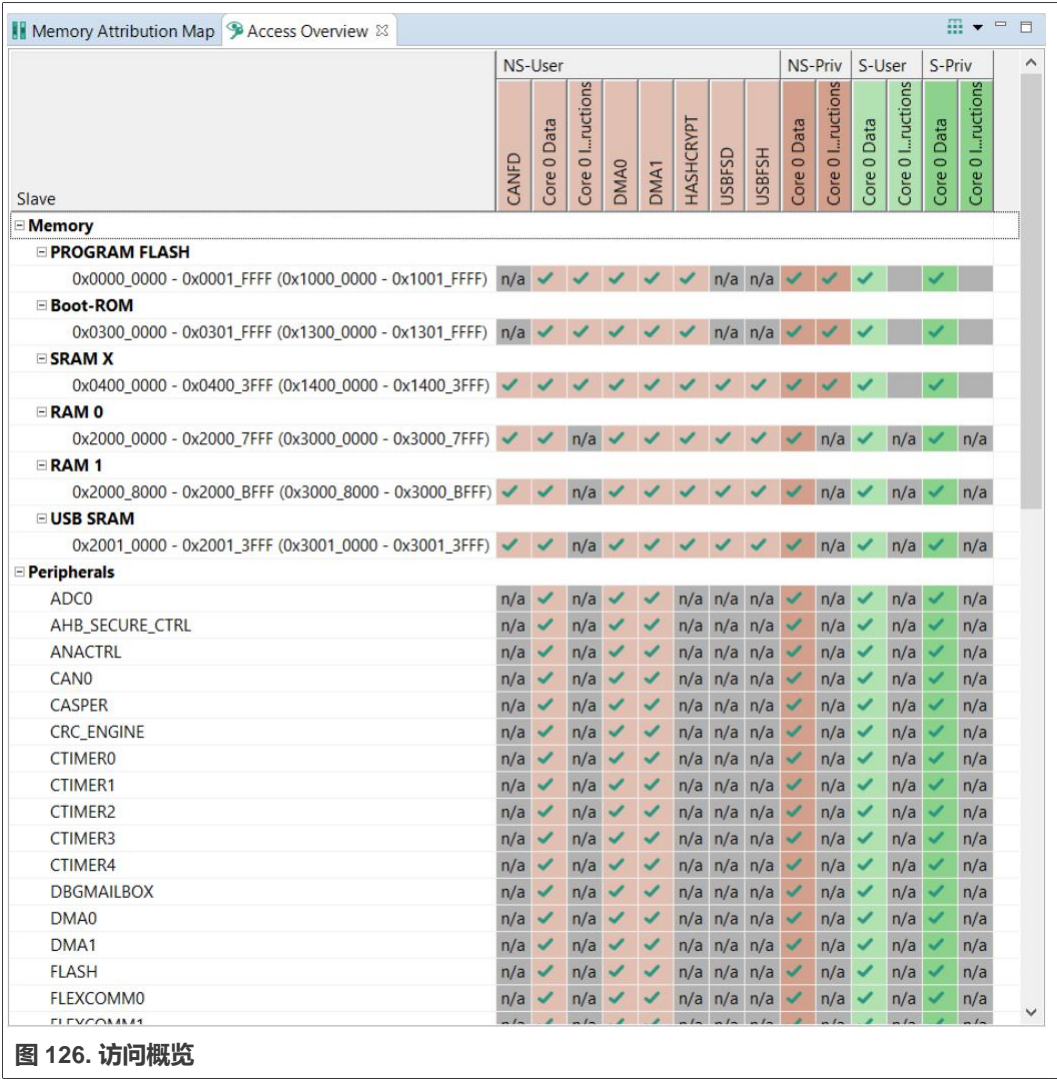


图 126. 访问概览

5.1.5 代码生成

如果设置正确并且没有报错，则代码生成引擎重新生成源代码。您可以在 **Trusted Execution Environment (可信执行环境)** 工具的 **Code Preview (代码预览)** 视图中查看生成的代码。

**Code Preview (代码预览)** 会自动高亮显示当前代码迭代与前一个迭代之间的差异。您可以单击 **Set viewing style for source differences (设置源代码差异的查看样式)**，在两种高亮显示模式之间进行选择。您还可以从同一下拉菜单中完全禁用高亮显示。

一些支持安全扩展功能的 AHBSC 或 TRDC 器件支持 ROM 预设和 C 语言代码。您可以在 **Miscellaneous (杂项)** 子视图中选中在 ROM 预设中生成代码。

5.2 支持 RDC 的器件

TEE 工具的功能和外观基于所加载器件的安全模型。

本节介绍加载具有 RDC（资源域控制器）和 XRDC2（扩展资源控制器 2）器件时工具的功能和外观。

目前支持以下类型的器件：

- RT1170
  - 双核（Cortex-M7 + Cortex-M4）：MIMXRT1176, MIMXRT1175, MIMXRT1173
  - 仅单核（Cortex-M7）：MIMXRT1172, MIMXRT1171
- KW45
- RT118x
- i.MX93

5.2.1 用户内存区域视图

在 **User Memory Regions（用户内存区域）** 视图中，您可以创建和维护内存区域及其访问模板的高级配置。您可以创建区域、为其命名、指定其地址、大小和安全级别，并为其添加说明。然后，您可以借助 **Problems（问题）** 视图来修复设置中的任何错误。

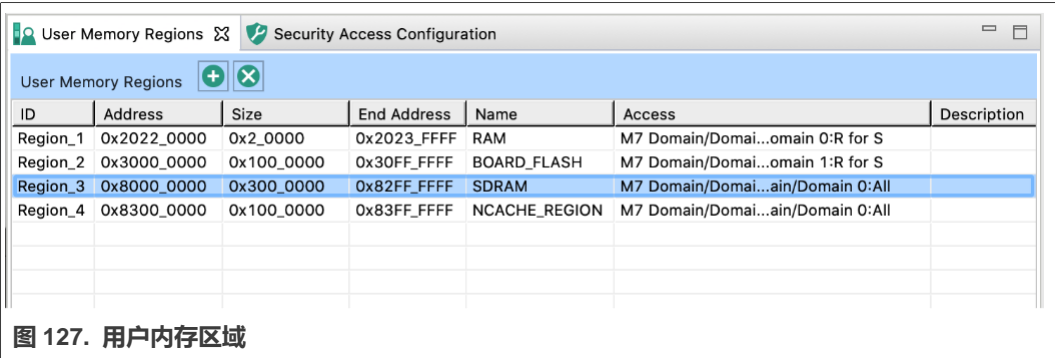


图 127. 用户内存区域

单击视图标题栏中 **Add new memory region button（添加新内存区域按钮）**，创建新内存区域。

单击该行的单元格，输入/更改内存区域的参数。

单击 **Access（访问）** 列中的单元格，修改内存区域的访问策略。此操作将打开[访问模板](#)对话框。

配置错误会在相关单元格中以红色图标高亮显示。如果问题可以轻松修复，可以右击该单元格，显示所提供解决方案的下拉列表。

选择表格行并单击视图标题栏中的 **Remove selected memory region(s)（删除所选内存区域）** 按钮，即可删除内存区域。

5.2.1.1 访问模板

在 **Access templates（访问模板）** 对话框中，您可以修改设备域的访问模板。该对话框显示设备 RDC 域以及所有用户创建的 XRDC2 域。

**注意：**请确保首先在 **M4 Domain/M7 Domain > Domains（M4 域/M7 域 > 域）** 中指定域的数量。

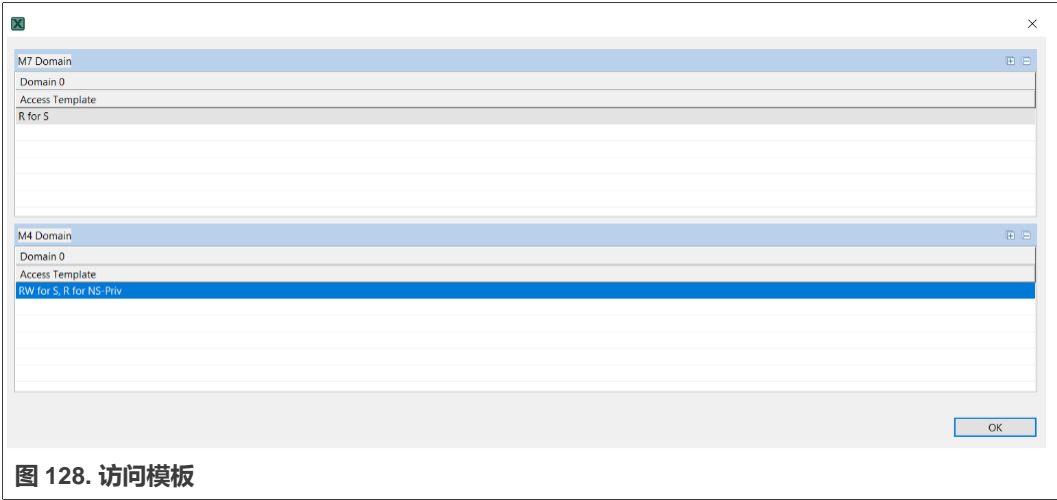


图 128. 访问模板

通过单击域列的最上方单元格，打开含所有选项的下拉列表，从中选择访问模板。  
为所有域选择访问模板后，单击 **OK（确定）** 返回 **User Memory Regions（用户内存区域）** 视图。

5.2.2 安全访问配置视图

在 **Security Access Configuration（安全访问配置）** 视图中，可以通过多种方式配置应用程序的安全策略。如需了解有关详情，请参阅以下章节。

5.2.2.1 RDC

在 **RDC** 子视图中，您可以为域分配主设备，并为每个域指定从设备的访问规则。

5.2.2.1.1 RDC 主设备

在 **RDC Masters（RDC 主设备）** 子视图中，您可以查看可用的总线主设备，将其分配给可用的域（核），并锁定/解锁分配。

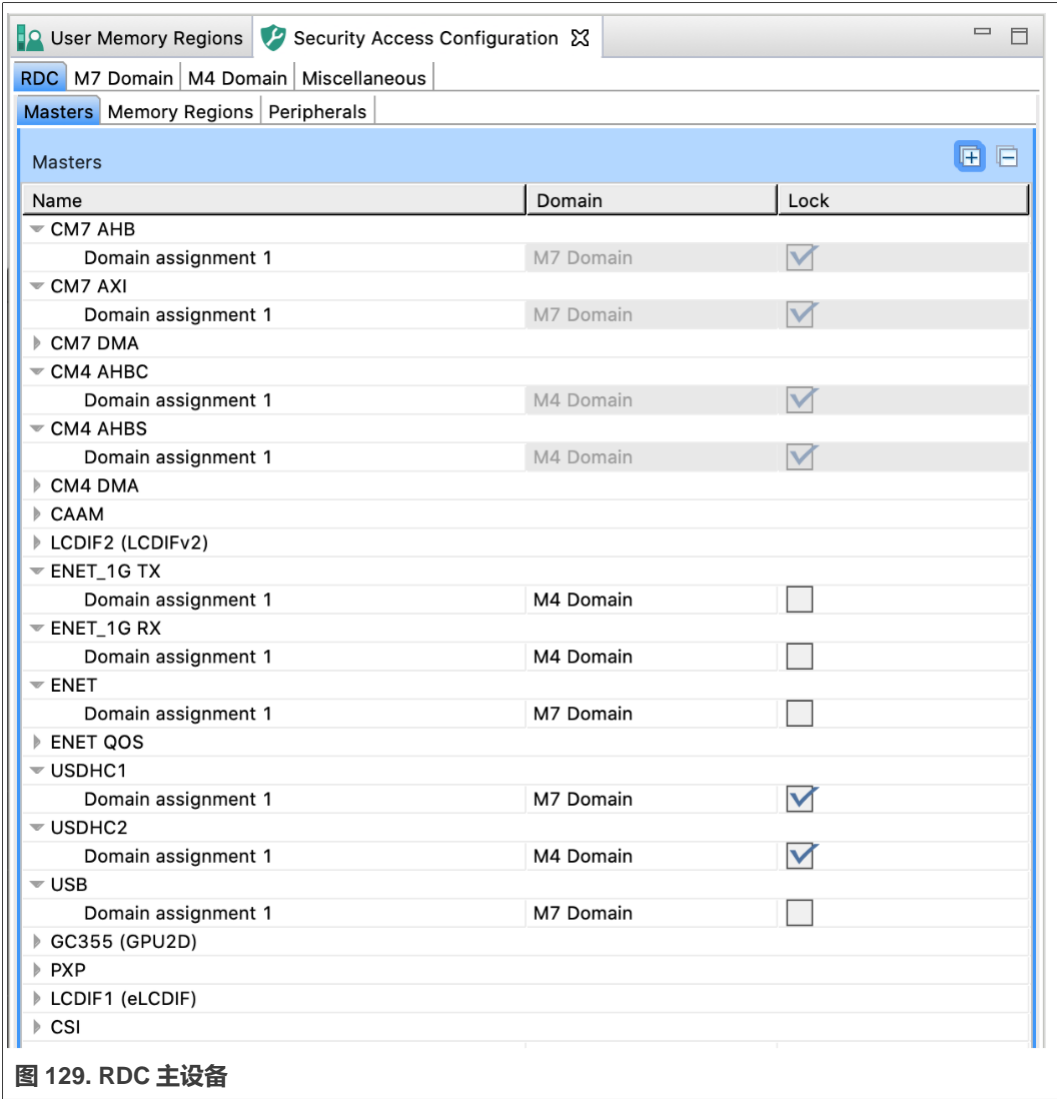


图 129. RDC 主设备

单击 **Masters (主设备)** 表中 **Domain (域)** 列的单元格，然后从下拉列表中选择域，将主设备分配给域。

选中 **Lock (锁定)** 复选框，防止进一步修改寄存器。

或者，您也可以右键单击主设备，并使用下拉列表选择选项。

**注意：**有些主设备默认分配给特定的域，不能重新分配。

5.2.2.1.2 内存区域

在 **Memory Regions (内存区域)** 子视图中，您可以查看、启用/禁用和配置 MRC（内存区域控制器）总线从设备及其域访问。

根据预编程的内存区域描述符寄存器，内存区域控制器可实现对从设备内存的访问控制。

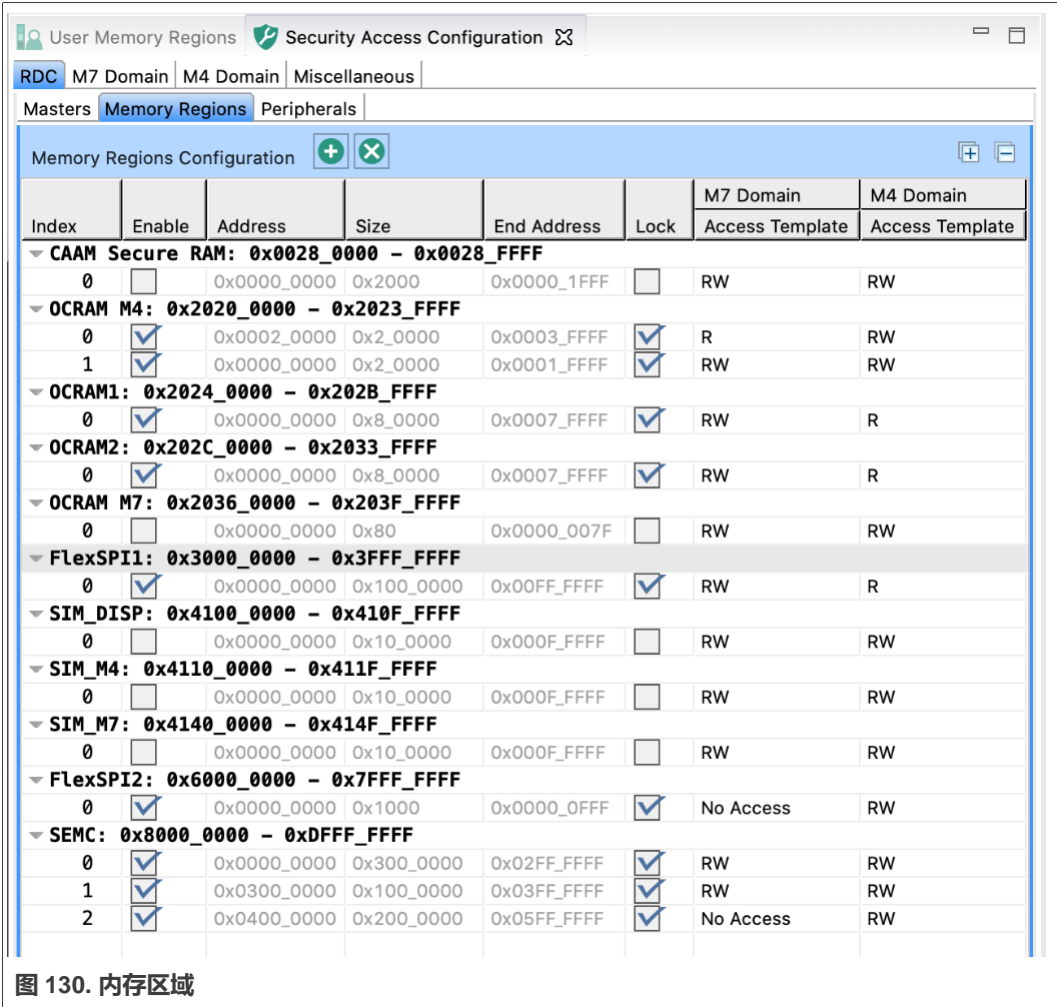


图 130. 内存区域

使用 **Memory Regions Configuration (内存区域配置)** 表格启用和配置 MRC 从设备：

- 1. **Enable (启用)** 区域。
- 2. 指定 **Address (地址)** 。
- 3. 指定 **Size (大小)** 或 **End Address (结束地址)** 。
- 4. 可选：**Lock (锁定)** 设置，防止进一步修改寄存器。
- 5. 为可用域设置 **Access Template (访问模板)** 。

您也可以右键单击主设备，使用下拉列表选择选项。

5.2.2.1.3 外设

在 **Peripherals (外设)** 子视图中，您可以查看和配置外设的 PDAP (外设域访问权限) 。



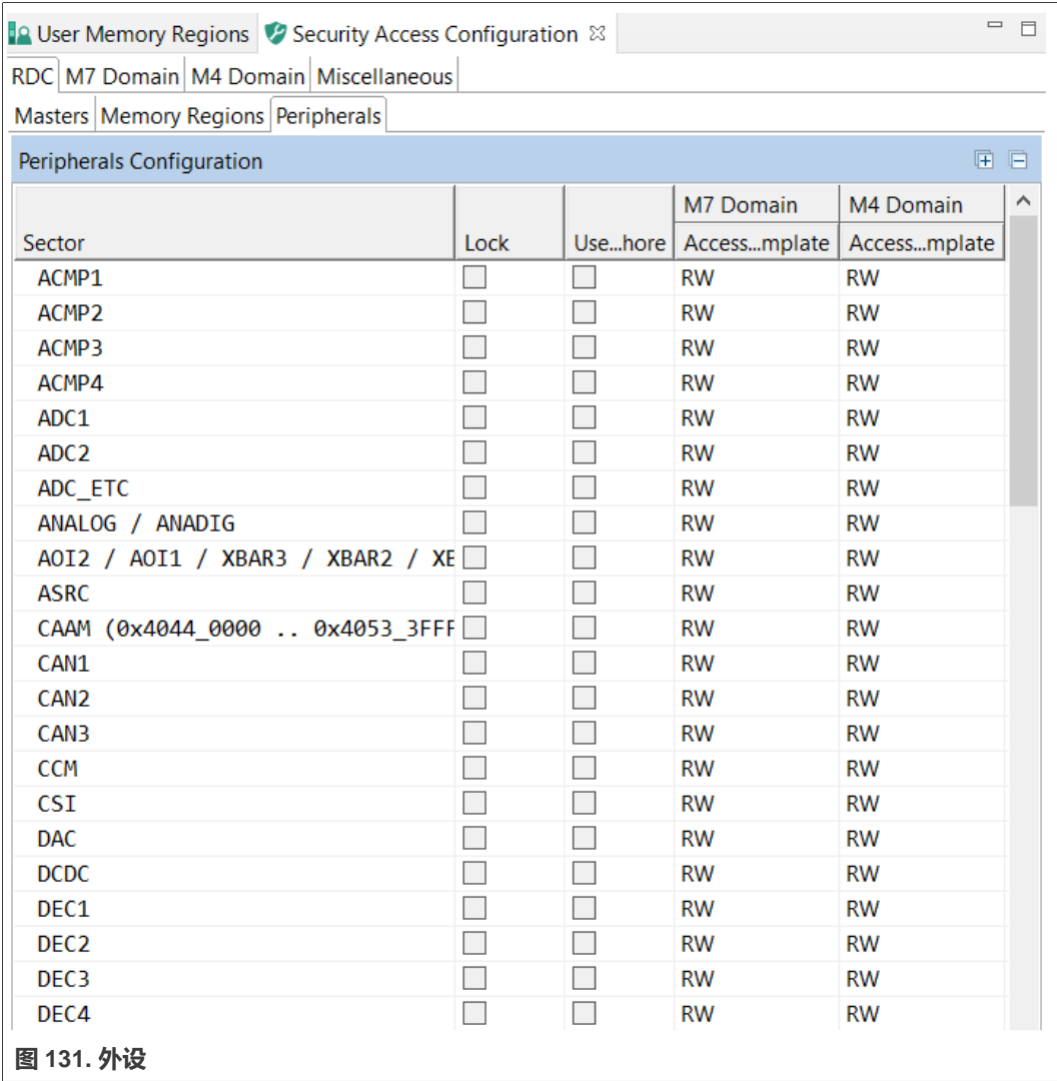


图 131. 外设

使用 **Peripherals Configuration (外设配置)** 表格启用和配置 PDAP:

- 1. 可选: **Lock (锁定)** 设置, 防止进一步的寄存器输入。
- 2. 选择 **Use semaphore (使用信号量)**, 启用外设的信号量功能。  
*注意: 启用后, 主设备在获得信号量之前无法访问该外设。在域拥有信号量期间, 其总线主设备对外设拥有独占访问权。*
- 3. 为可用域设置 **Access Template (访问模板)**。

5.2.2.2 XRDC2 域视图

在 **M7/M4 Domain (M7/M4 域)** 子视图中, 您可以查看和配置 XRDC2 (扩展资源域控制器 2) 域的安全策略。每个 CPU 最多可包含 16 个域。

5.2.2.2.1 MPU

在 **MPU** 子视图中, 您可以启用和配置 MPU (内存保护单元), 也可以创建区域、指定其地址、大小和其他参数。



MPU 可以执行权限规则、分离进程和执行内存访问规则，并支持标准的 ARMv7 受保护的内存系统架构模型。

MPU 默认是禁用的，必须选择 **Enable MPU (启用 MPU)** 选项才能启用它。

**注意：**并不是每个设备都支持 MPU。

User Memory RegionsSecurity Access Configuration

RDCM7 DomainM4 DomainMiscellaneous

MPUDomainsMastersPeripheralsMemory RegionsMemory Slots

Options

☐ Enable MPU☐ Enable MPU during HardFault and NMI handlers☐ Enable privileged software access to the default memory map☐ Generate sources for disabled regions

MPU Memory Attributes

Index	ID	Memory Type	Inner Attributes			Outer Attributes		
			C	B	W	C	B	W
0	0	Device	n/a	n/a	n/a	n/a	n/a	n/a
1	1	Device	n/a	n/a	n/a	n/a	n/a	n/a
10	10	Device	n/a	n/a	n/a	n/a	n/a	n/a
11	11	Device	n/a	n/a	n/a	n/a	n/a	n/a
12	12	Device	n/a	n/a	n/a	n/a	n/a	n/a
13	13	Device	n/a	n/a	n/a	n/a	n/a	n/a
14	14	Device	n/a	n/a	n/a	n/a	n/a	n/a
15	15	Device	n/a	n/a	n/a	n/a	n/a	n/a
2	2	Device	n/a	n/a	n/a	n/a	n/a	n/a
3	3	Device	n/a	n/a	n/a	n/a	n/a	n/a
4	4	Device	n/a	n/a	n/a	n/a	n/a	n/a
5	5	Device	n/a	n/a	n/a	n/a	n/a	n/a

MPU Memory Regions

Index	Enable	Address	Size	End Address	Exec	Permissions	SRD	Shareability	Mem
0	<input type="checkbox"/>	0x0000_0000	0x20	0x0000_001F	<input type="checkbox"/>	RW priv	0b0	No	0 (0)
1	<input type="checkbox"/>	0x0000_0000	0x20	0x0000_001F	<input type="checkbox"/>	RW priv	0b0	No	1 (1)
10	<input type="checkbox"/>	0x0000_0000	0x20	0x0000_001F	<input type="checkbox"/>	RW priv	0b0	No	10 (10)
11	<input type="checkbox"/>	0x0000_0000	0x20	0x0000_001F	<input type="checkbox"/>	RW priv	0b0	No	11 (11)
12	<input type="checkbox"/>	0x0000_0000	0x20	0x0000_001F	<input type="checkbox"/>	RW priv	0b0	No	12 (12)
13	<input type="checkbox"/>	0x0000_0000	0x20	0x0000_001F	<input type="checkbox"/>	RW priv	0b0	No	13 (13)
14	<input type="checkbox"/>	0x0000_0000	0x20	0x0000_001F	<input type="checkbox"/>	RW priv	0b0	No	14 (14)
15	<input type="checkbox"/>	0x0000_0000	0x20	0x0000_001F	<input type="checkbox"/>	RW priv	0b0	No	15 (15)
2	<input type="checkbox"/>	0x0000_0000	0x20	0x0000_001F	<input type="checkbox"/>	RW priv	0b0	No	2 (2)
3	<input type="checkbox"/>	0x0000_0000	0x20	0x0000_001F	<input type="checkbox"/>	RW priv	0b0	No	3 (3)
4	<input type="checkbox"/>	0x0000_0000	0x20	0x0000_001F	<input type="checkbox"/>	RW priv	0b0	No	4 (4)

图 132. MPU

使用 **MPU Memory Attributes (MPU 内存属性)** 表格命名和配置 MPU 内存属性集。单击 **Memory Type (内存类型)** 和 **Inner/Outer Attributes (内部/外部属性)** 列的单元格，显示可用选项。

使用 **MPU Memory Regions (MPU 内存区域)** 表格启用和配置 MPU 内存区域。

- 1. **Enable (启用)** 区域。
- 2. 指定 **Address (地址)** 。
- 3. 指定 **Size (大小)** 或 **End Address (结束地址)** 。
- 4. 如果希望区域能够运行代码，请设置 **Exec (执行)** 选项。
- 5. 设置 **Permissions (权限)** 。
- 6. 设置 **SRD (子区域禁用)** 位。
- 7. 设置 **Shareability (共享性)** 或缓存选项。

5.2.2.2.2 域

在 **Domains (域)** 子视图中，您可以查看、添加/删除和重命名 XRDC2 域。每个 CPU 最多支持 16 个 XRDC2 域。

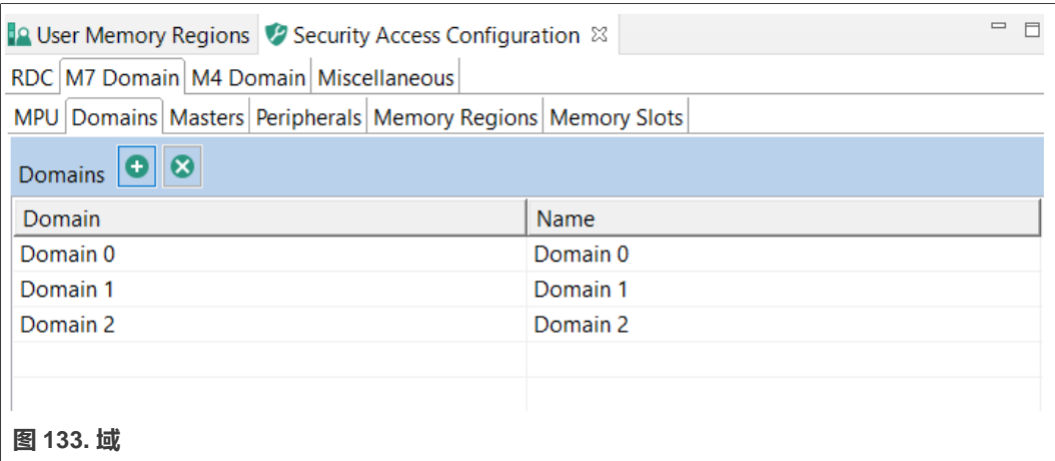


图 133. 域

单击 **Add new domain (添加新域)** 按钮，添加一个新域。

在 **Name (名称)** 列中输入新名称，重命名域。

单击 **Remove last domain (删除最后一个域)** 按钮，删除一个域。

5.2.2.2.3 主设备

在 **Masters (主设备)** 子视图中，可以添加/删除、查看、配置可用 RDC 主设备的 XRDC2 域分配。

主域分配控制器 (MDAC) 负责根据预编程的主域分配 (MDA) 寄存器为设备中的每个系统总线事务生成 DID、非安全和特权属性。

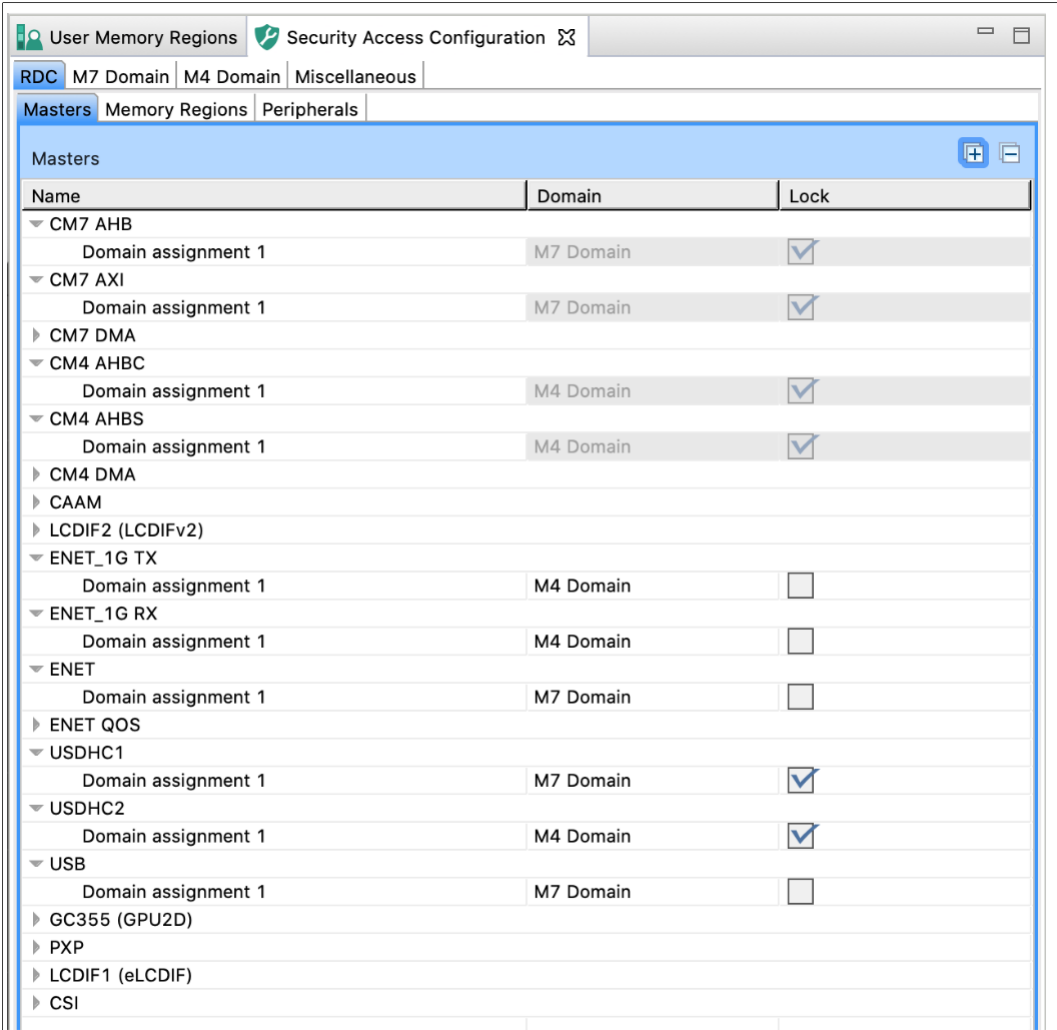


图 134. 主设备

要添加新的域分配，请按照以下步骤操作：

1. 单击 **Add new domain assignment for the selected master**（为所选主设备添加新域分配）按钮。
2. 选择 **Enable**（启用）复选框。
3. 输入 **Match Input**（匹配输入）值。  
**注意：**匹配字段指定了与 MDAC 匹配输入进行比较的参考值。匹配字段宽度因 MDAC 实例而异，从 0 到 16 位不等。未实现的位读取为 0。0 位的大小会在所有比较中生成相匹配的字段。
4. 输入 **Mask Input**（屏蔽输入）值。  
**注意：**掩码字段指定哪些位在匹配比较中有效。仅比较掩码值为 0 位的位置。掩码字段宽度与掩码字段相同，掩码字段因 MDAC 实例而异，从 0 到 16 位不等。一个全为 1 的掩码值，将生成所有比较都匹配的字段。
5. 从 **Domain**（域）列的下拉列表中选择 XRDC2 域分配。
6. 从 **Secure**（安全）列的下拉列表中选择安全访问类型。
7. 从 **Privileged**（特权）列的下拉列表中选择特权访问类型。
8. 可选：选择 **Lock**（锁定）复选框，防止进一步修改寄存器。

#### 5.2.2.2.4 外设

在 **Peripherals (外设)** 子视图中，您可以查看 PAC（外设访问控制器）的访问模板，并为所选 RDC 域上 PAC 管理的所有外设配置访问权限。

外设访问控制器子模块对连接到外设总线桥或集成到外设子系统的一组外设执行访问控制。

**Access Template (访问模板)** 表格显示所选设备上 PAC 可用的所有访问模板的 ID 和名称。该信息由数据驱动，仅用于显示。

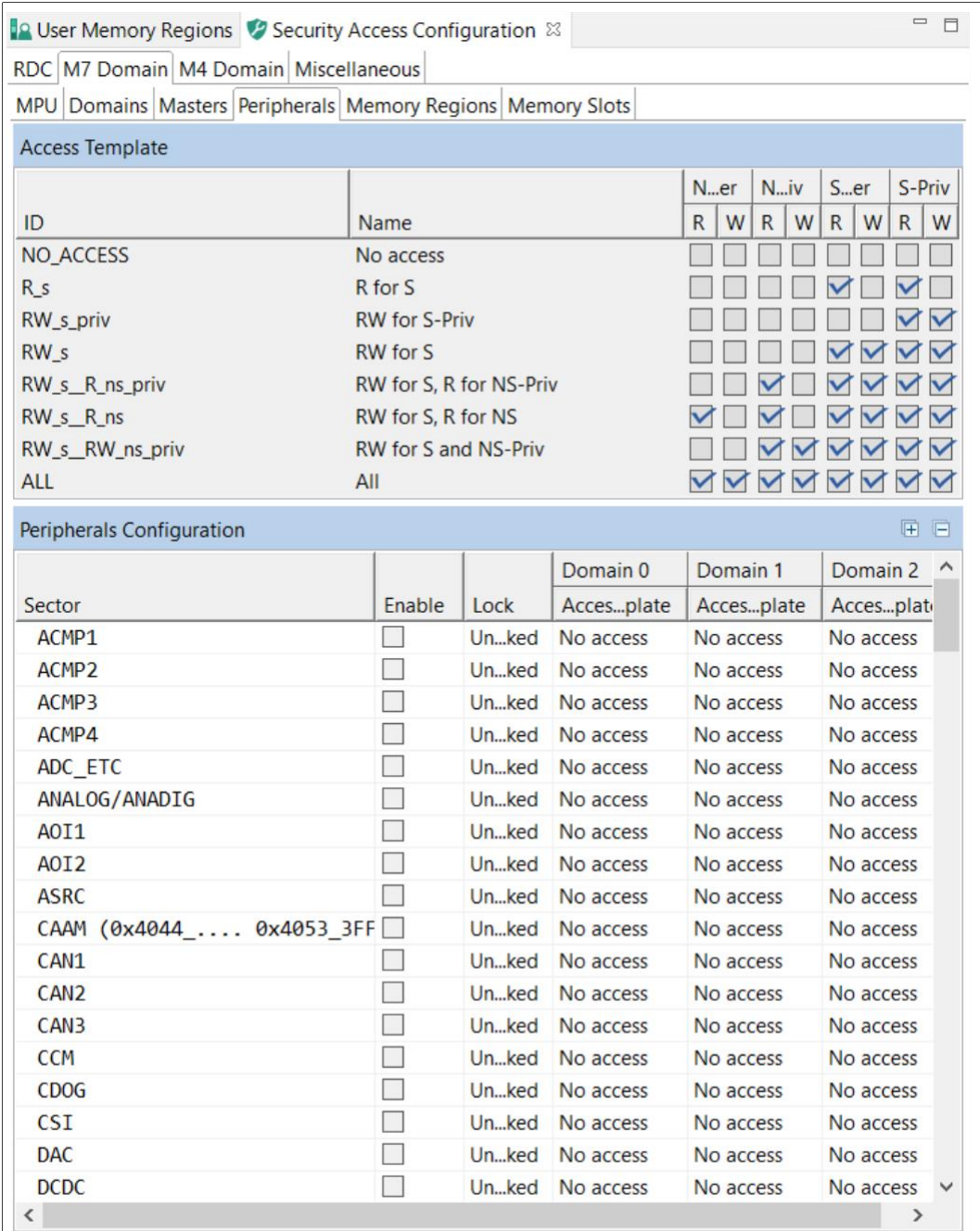


图 135. 外设

使用 **Peripherals Configuration (外设配置)** 表格配置外设的访问权限：

- 1. 选择 **Enable (启用)** 复选框。
- 2. 将 **Lock (锁定)** 设置为所需状态。
- 3. 为所有列出的域设置 **Access Template (访问模板)** 。

或者，您也可以右键单击主设备并使用下拉列表选择选项。

5.2.2.2.5 内存区域

在 **Memory Regions (内存区域)** 子视图中，您可以查看 MRC (内存区域控制器) 的访问模板，并为所选 RDC 域上 MRC 管理的所有非外设内存空间配置访问权限。

内存区域控制器 (MRC) 为针对非外设内存空间的所有系统总线引用提供基于域的硬件访问控制。

**Access Template (访问模板)** 表显示所选设备上 MRC 可用的所有访问模板的 ID 和名称。该信息是数据驱动的，仅用于显示。

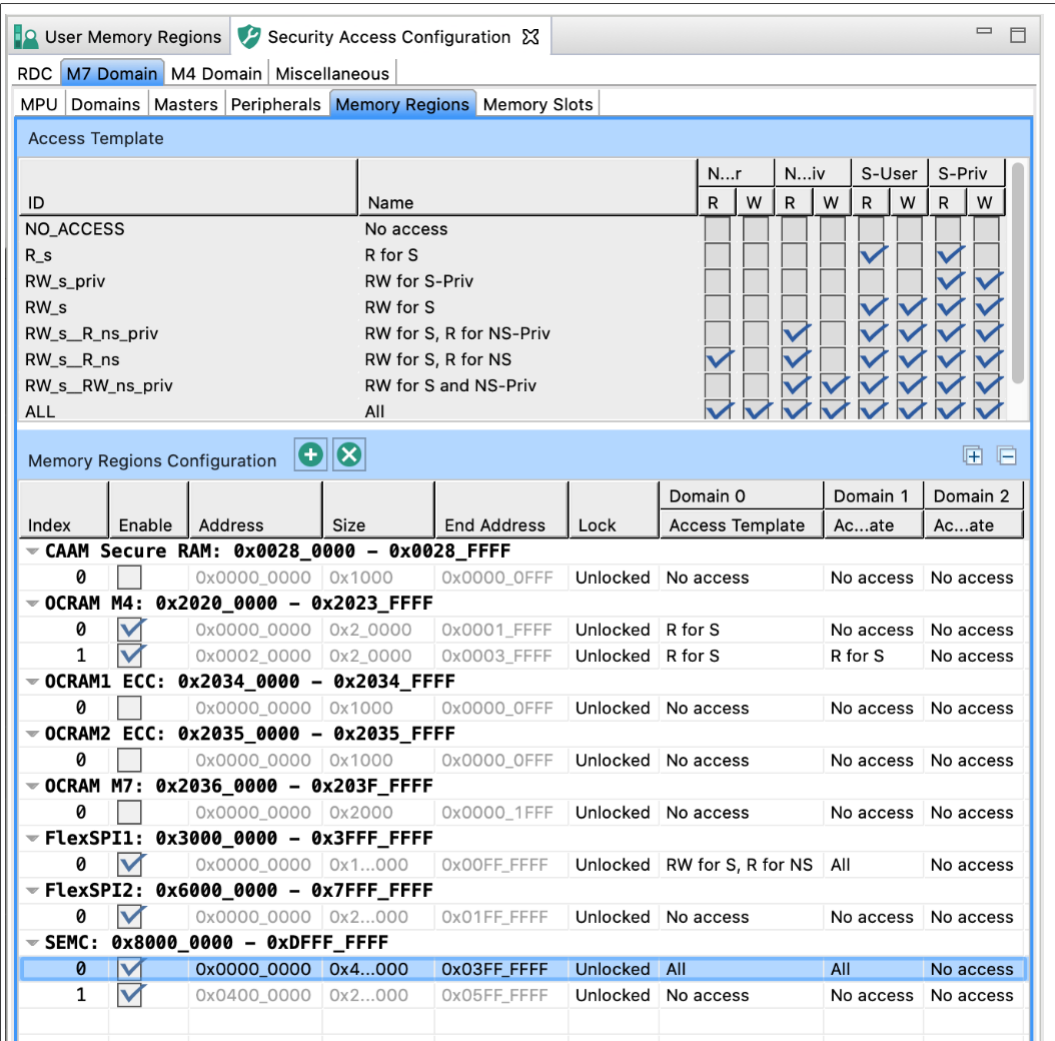


图 136. 内存区域

使用 **Memory Regions Configuration (内存区域配置)** 表格为非外设内存空间配置访问权限：

1. 选择 **Enable (启用)** 复选框。
2. 指定 **Start Address (起始地址)** 。
3. 指定 **Size (大小)** 或 **End Address (结束地址)** 。
4. 将 **Lock (锁定)** 设置为所需状态。
5. 为所有列出的域设置 **Access Template (访问模板)** 。



或者，您也可以右键单击主设备并使用下拉列表来选择选项。

5.2.2.2.6 内存插槽

在 **Memory Slots (内存插槽)** 子视图中，您可以查看 MSC（内存插槽控制器）的访问模板，并为所选 RDC 域中 MSC 管理的所有内存空间配置访问权限。

内存插槽控制器（MSC）对具有固定地址范围的外设或内存空间执行访问控制。

**Access Template (访问模板)** 表格显示所选设备上 MSC 可用的所有访问模板的 ID 和名称。该信息由数据驱动，仅用于显示。

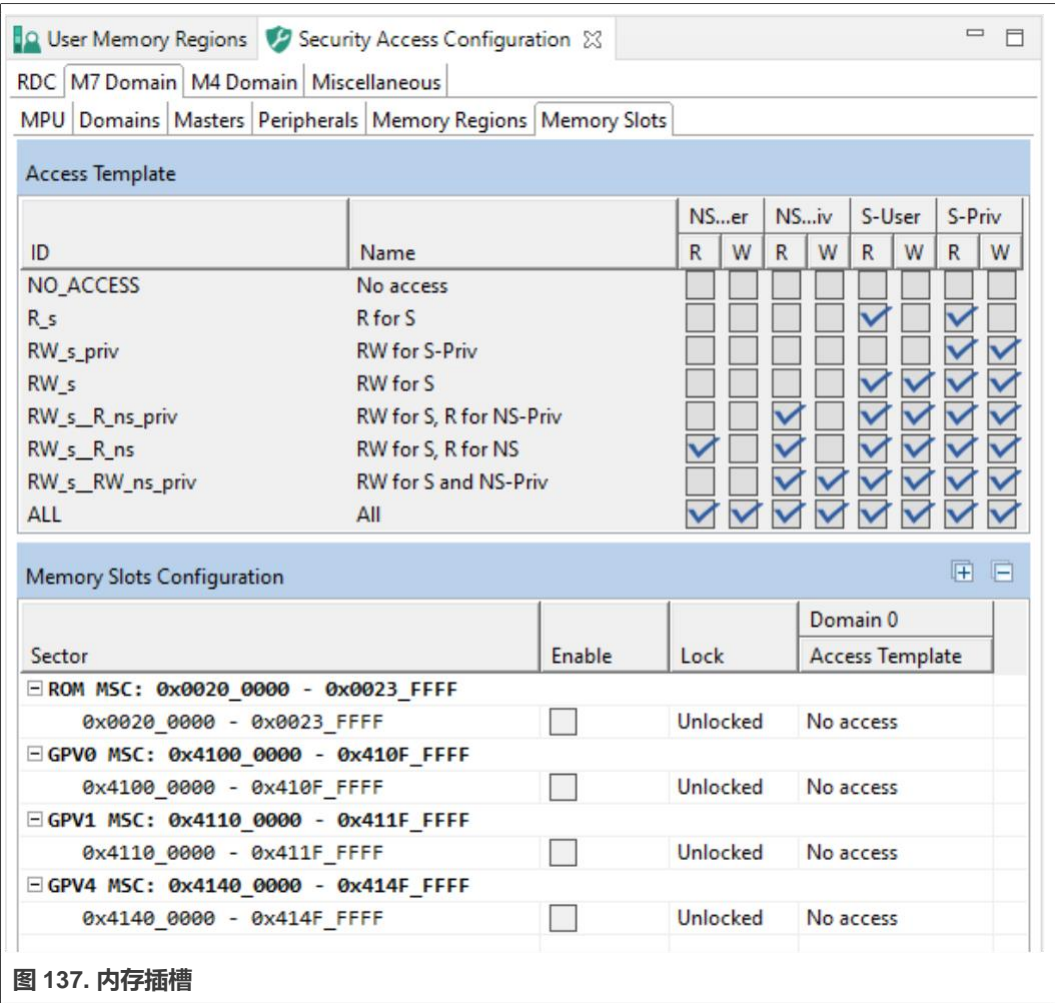


图 137. 内存插槽

使用 **Memory Slots Configuration (内存插槽配置)** 表配置内存空间的访问权限：

1. 选择 **Enable (启用)** 复选框。
2. 将 **Lock (锁定)** 设置为所需状态。
3. 为所有列出的域设置 **Access Template (访问模板)** 。

或者，您也可以右键单击主设备并使用下拉列表来选择选项。

5.2.2.3 i.MX8 ULP 中 Cortex-A35 上的 XRDC (扩展可信资源域控制器)

5.2.2.3.1 主设备

XRDC 主设备与 [TRDC 主设备](#) 类似。此外，还支持以下功能：

- **PID (Process Identifier (进程标识符))** 与 PIDM 字段相结合，以确定域命中。
- **PIDM (PID Mask (PID 掩码))** 提供掩码功能，可将多个进程标识符作为域命中确定的一部分。如果 PIDM 中的某个位被设置，则在比较中会忽略 PID 的相应位。
- **PID enable (PID 启用)** 提供了包含或排除屏蔽 PID 值集的功能。允许的值为 00b、01b、10b 和 11b。更多信息，请参阅《参考手册》(链接待提供)。

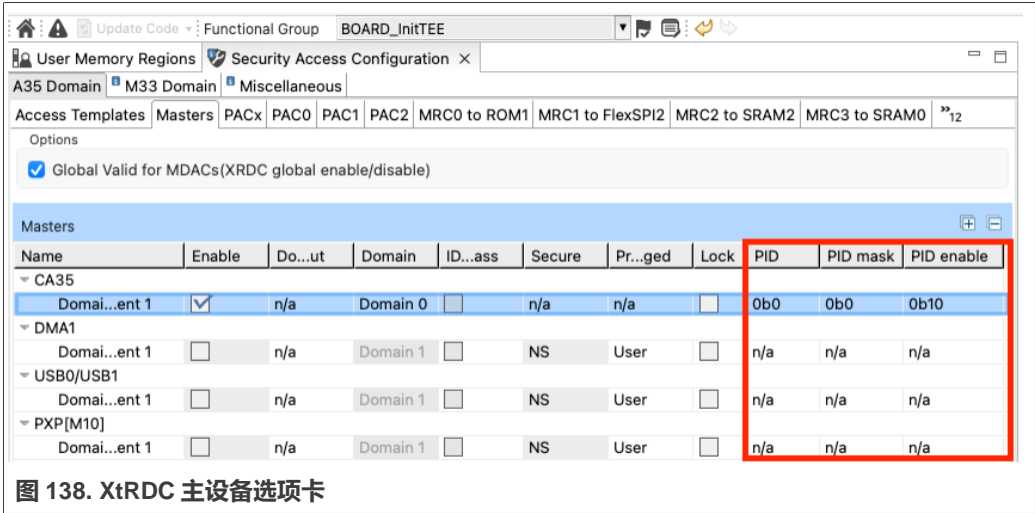


图 138. XtRDC 主设备选项卡

5.2.2.3.2 MRC

XRDC 上的 MRC 与 [TRDC 上的 MRC](#) 类似。两者之间存在一些细微差别：

1. 内存区域表只有一个实例，因为地址范围在所有域中共享。对于每个内存区域，用户可以为每个域指定一个访问模板。
2. 代码区域指定使用哪些模板（0=数据，1=代码）。现在的模板是混合模板。这意味着有两个模板具有相同的 ID 和名称，第一行用于数据区，第二行用于代码区。这些模板都有锁定字段，可以通过单击所需的访问框进行编辑。



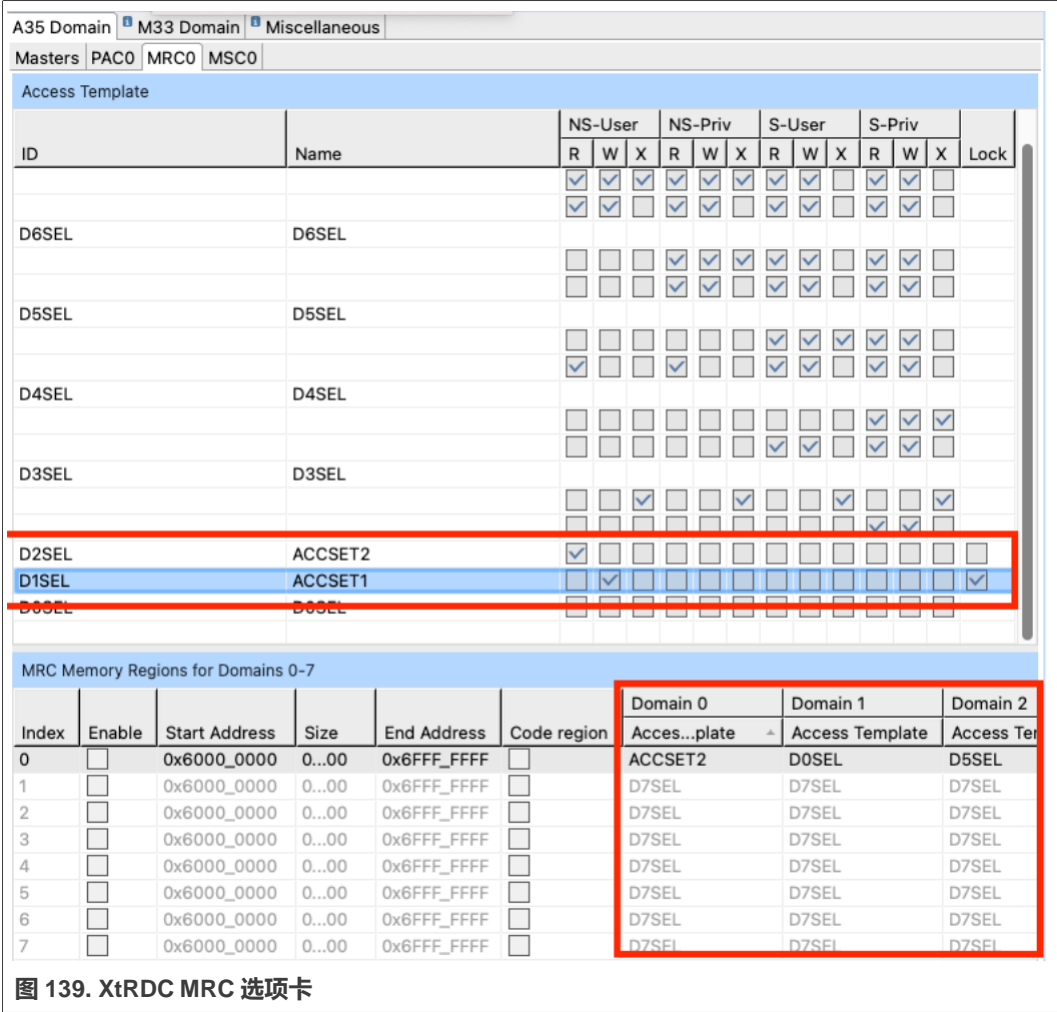


图 139. XtRDC MRC 选项卡

5.2.2.3.3 访问控制模式

PID 有两种模式可以选择。

对于仅支持 TSM（三态模式：SecurePriv、SecureUser、NonsecureUser）的处理器，在特权模式下，MDAC 子模块的非安全[n]输出信号被强制置零，以实现用户模式和特权模式之间的精确状态转换。如果启用 SP4SM（特殊四态模式），MDAC 就不会使用 MDA[DIDS,DID]字段。MDAC 会跟踪当前访问级别，为特定访问级别生成特定的域 ID。

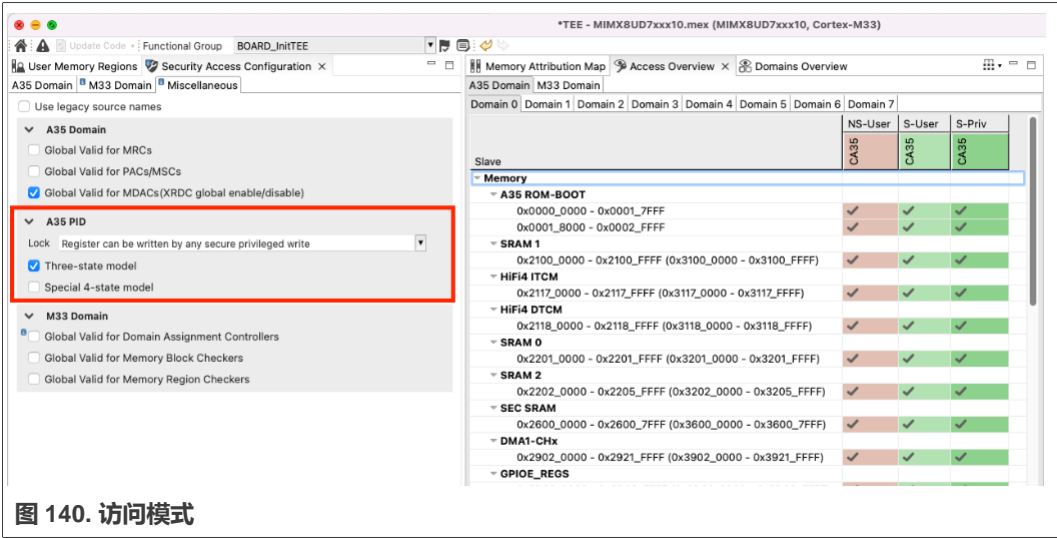


图 140. 访问模式

5.2.2.4 i.MX8 ULP 和 KW45 中 Cortex-M33 上的可信资源域控制器 (TRDC)

5.2.2.4.1 MPU

此 MPU 与其他带 Cortex-M33 的 MPU (例如 LPC55S) 相同, 或其他基于 ARMv8-M 架构或更高版本的核心 (具有安全/非安全寄存器库) 相同。

5.2.2.4.2 域

这些域与 RDC/XRDC2/XRDC 类似: 将芯片资源分配给处理 “域”, 每个处理域分配唯一的域标识符 (域 ID, DID)。支持的 DID 数量通常是 CPU 数量加 1。

5.2.2.4.3 主设备

主设备类似于 MIMXRT117x 上的 [XRDC2](#) 中的主设备。用户还可以根据主设备类型选择域 ID 输入或 ID 旁路。

5.2.2.4.4 访问模板

访问模板与 MIMXRT117x 上的 XRDC2 中的模式类似。主要区别如下: 您可以在 “全局” 模板 (适用于整个 RDC, 由所有校验器使用, 且可编辑) 和 “本地” 模板 (特定于检测工具, 且不可变) 之间切换; 同时, XRDC2 中的访问模板始终取决于校验工具, 且可编辑。

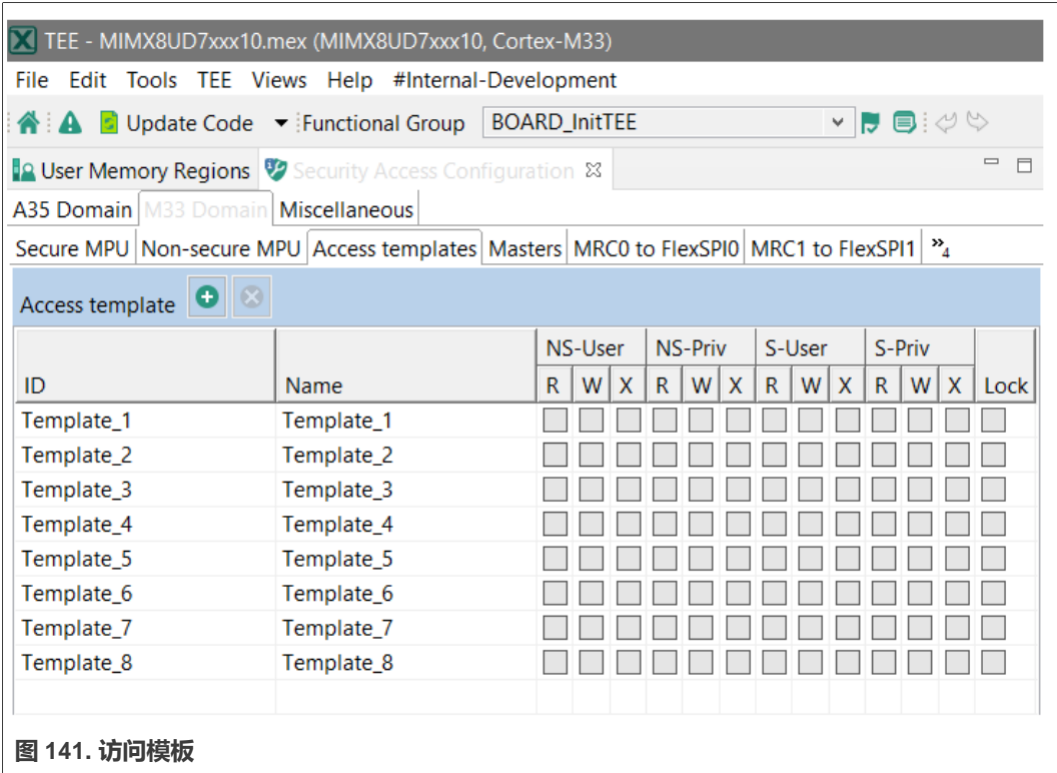


图 141. 访问模板

5.2.2.4.5 MRC

TRDC 上的 MRC 类似于 XRDC2 中的 MRC（内存区域）。

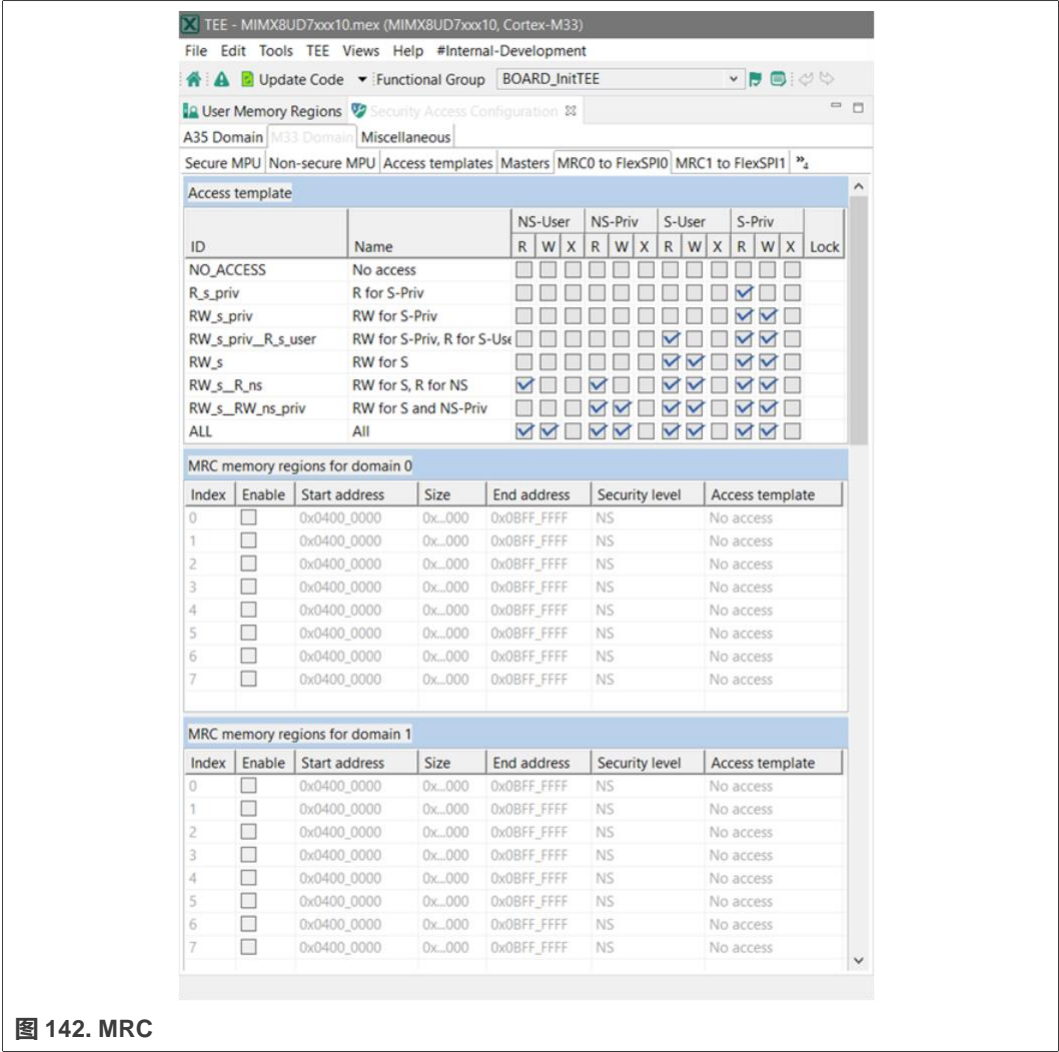


图 142. MRC

5.2.2.4.6 MBC

TRDC 中的 MBC 类似于 XRDC2 中的 MSC（内存插槽）和 XRDC 中的 MSC。

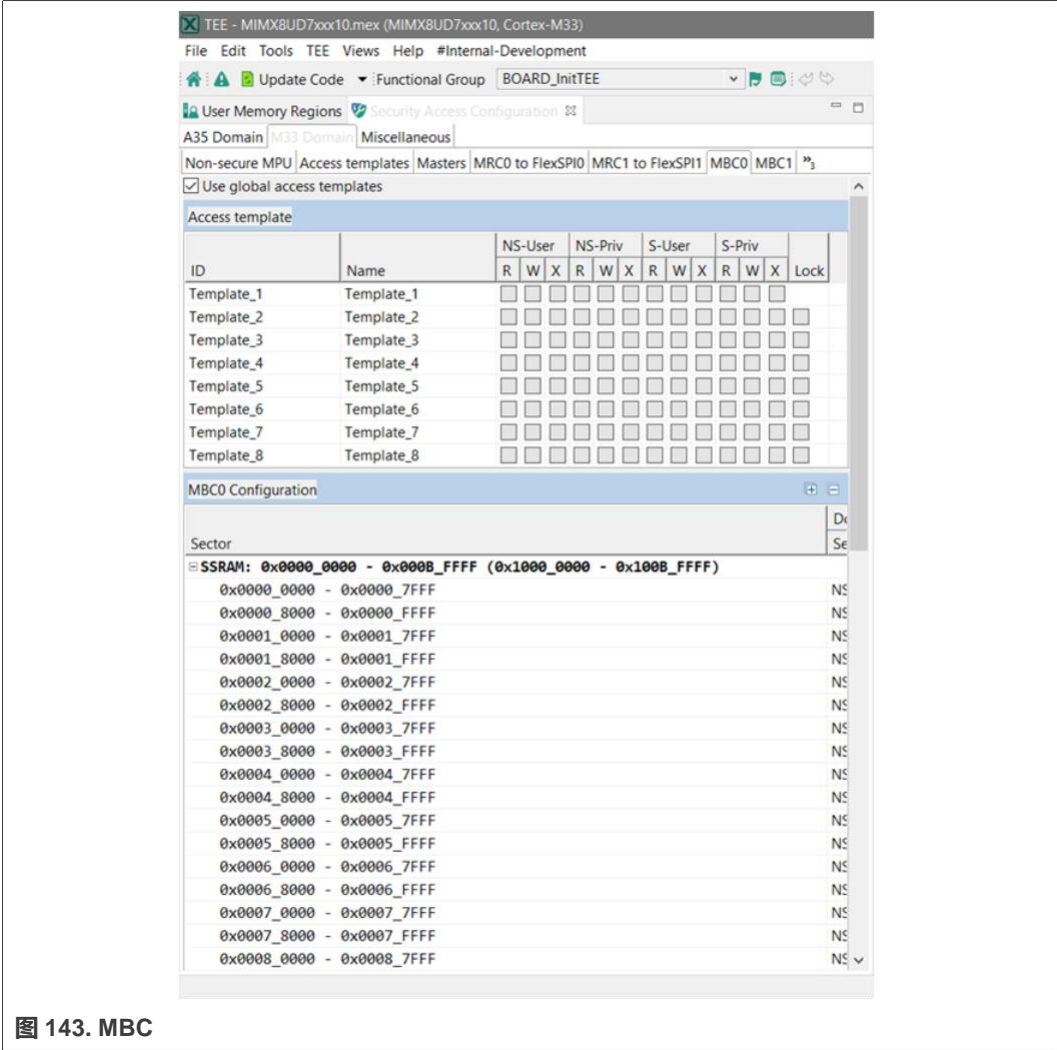
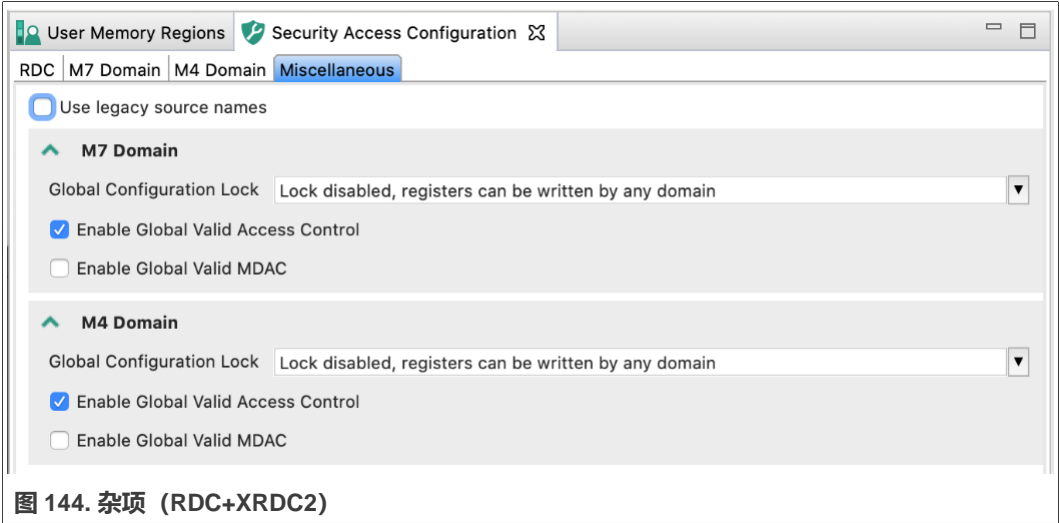


图 143. MBC

5.2.2.5 杂项

在 **Miscellaneous (杂项)** 子视图中，您可以设置各种配置选项。这些选项的列表取决于处理器数据，并且差异很大。所有选项都会影响寄存器设置，您可以在 **Register (寄存器)** 视图中查看它们。只有部分选项会直接影响您在 **Security Access Configuration (安全访问配置)** 视图中所做的配置。将光标指向各个选项可显示工具提示，解释每个选项的功能。



5.2.3 内存归属映射

在 **Memory Attribution Map (内存属性映射)** 视图中，您可以查看在域级别上为所有主设备设置的代码、数据和外设内存区域的访问级别。该表格是只读的。

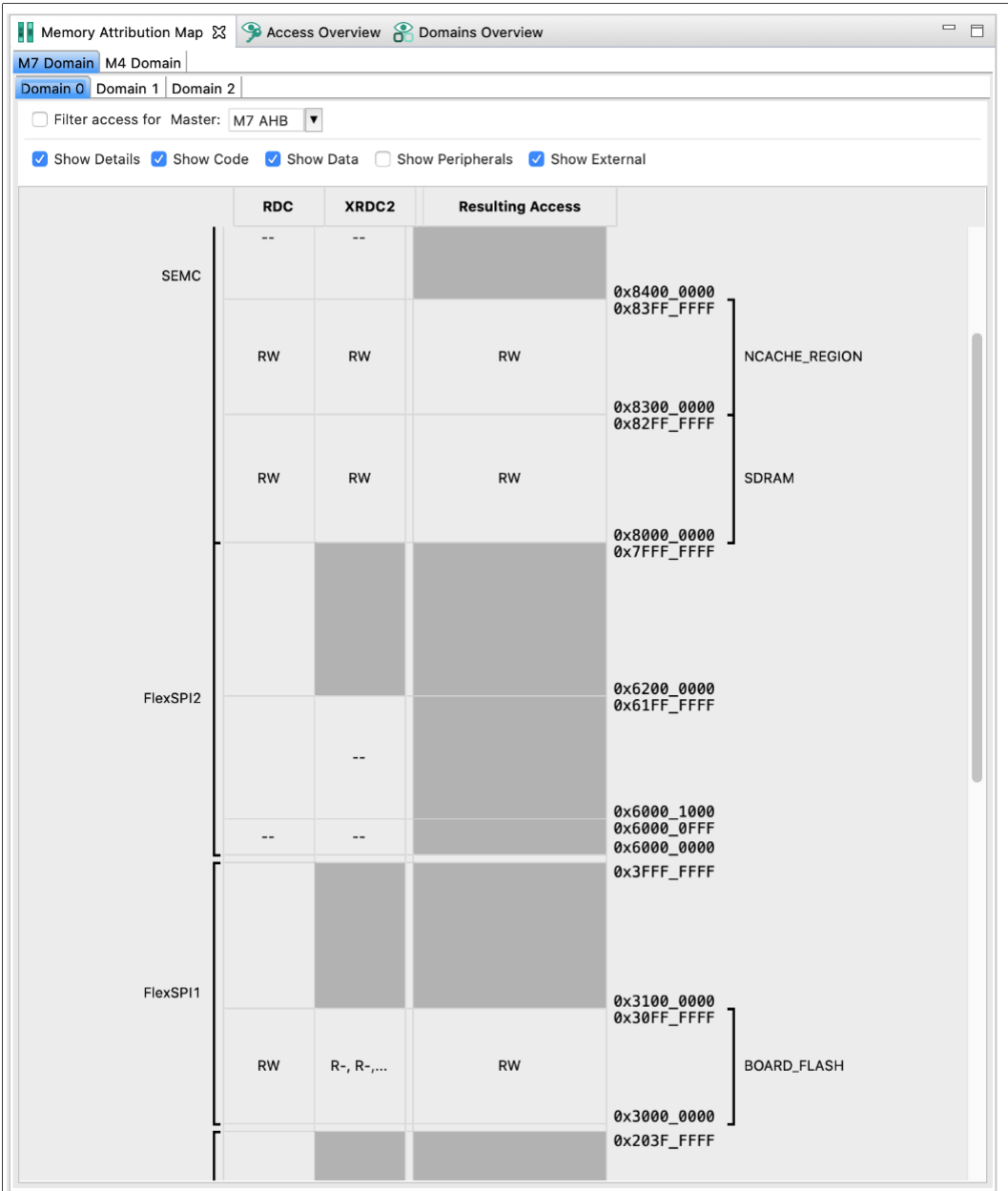


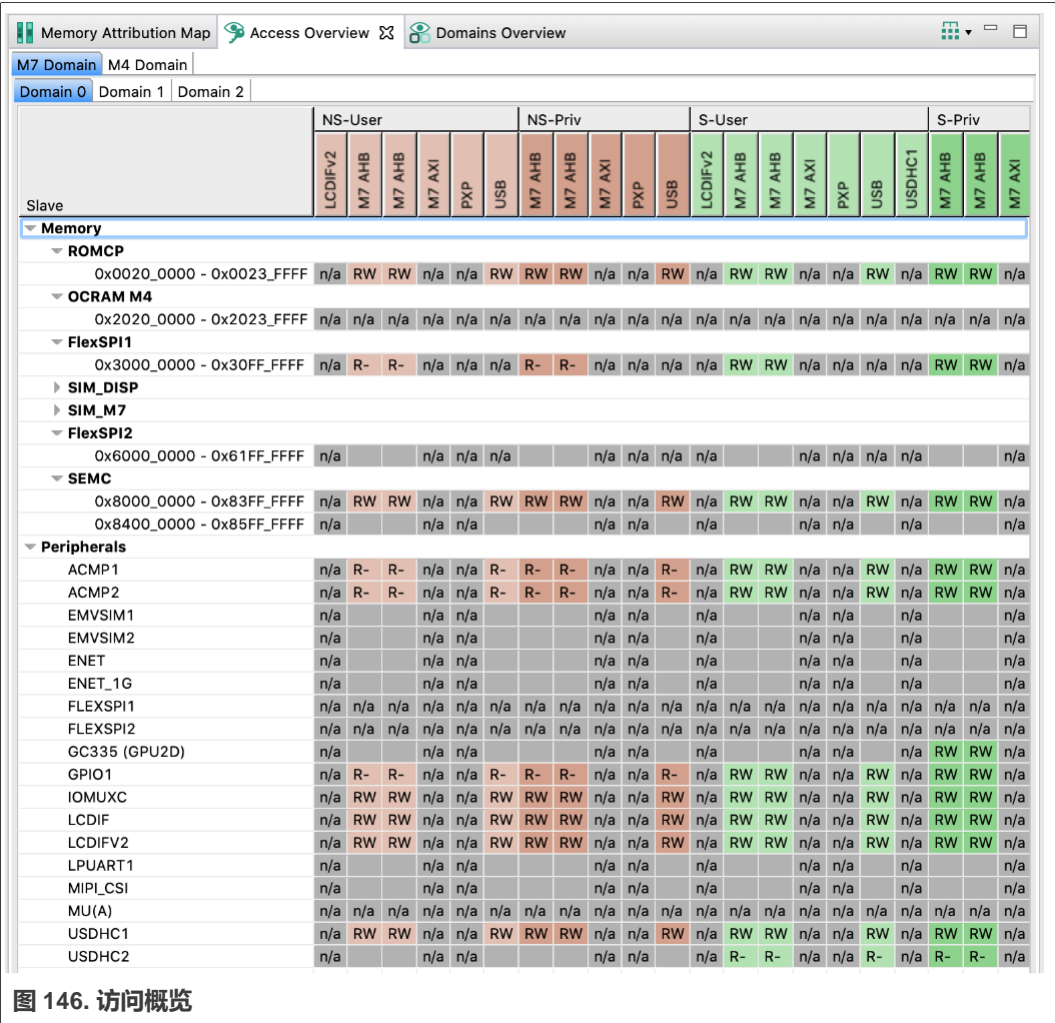
图 145. 内存属性映射

- 要设置显示选项，请按照以下步骤操作：
1. 单击 **Filter access for (过滤访问)** 复选框启用过滤选项。
  2. 从 **Master (主设备)** 下拉菜单中选择要查看的主设备。
  3. (可选) 也可从 **Security mode (安全模式)** 下拉菜单中选择，设置所选主设备的安全级别。这个设置不会影响配置。
  4. (可选) 取消选择 **Show Details (显示详细信息)**、**Show Code (显示代码)**、**Show Data (显示数据)**、**Show Peripherals (显示外设)** 和 **Show External (显示外部)** 选项来自定义输出。
  5. (可选) 在 **Filter (过滤器)** 区域过滤显示的内存区域。

将光标指向单元格，可显示一个工具提示，说明安全级别组合的信息。  
双击单元格，打开 **Security Access Configuration（安全访问配置）** 中的相关设置。

5.2.4 访问概览

在 **Access Overview（访问概览）** 中，您可以查看在 **Security Access Configuration（安全访问配置）** 视图中设置的安全策略。该视图分为多个子视图，显示特定 XRDC2 域的访问概览。  
纵轴显示所有主设备，并按其安全设置分为彩色编码组。  
横轴显示内存范围和从设备总线/外设。



将光标指向某个条目，就会显示有关该条目信息的工具提示。  
您可以使用工具栏右侧的按钮，按安全性或主设备对显示的信息进行分组。



5.2.5 域概览

在 **Domains Overview (域概览)** 中，您可以查看在 **Domain (域)** 视图的子视图中配置的 XRDC2 域的访问策略。

将光标指向单元格可显示一个工具提示，说明安全级别组合的信息。

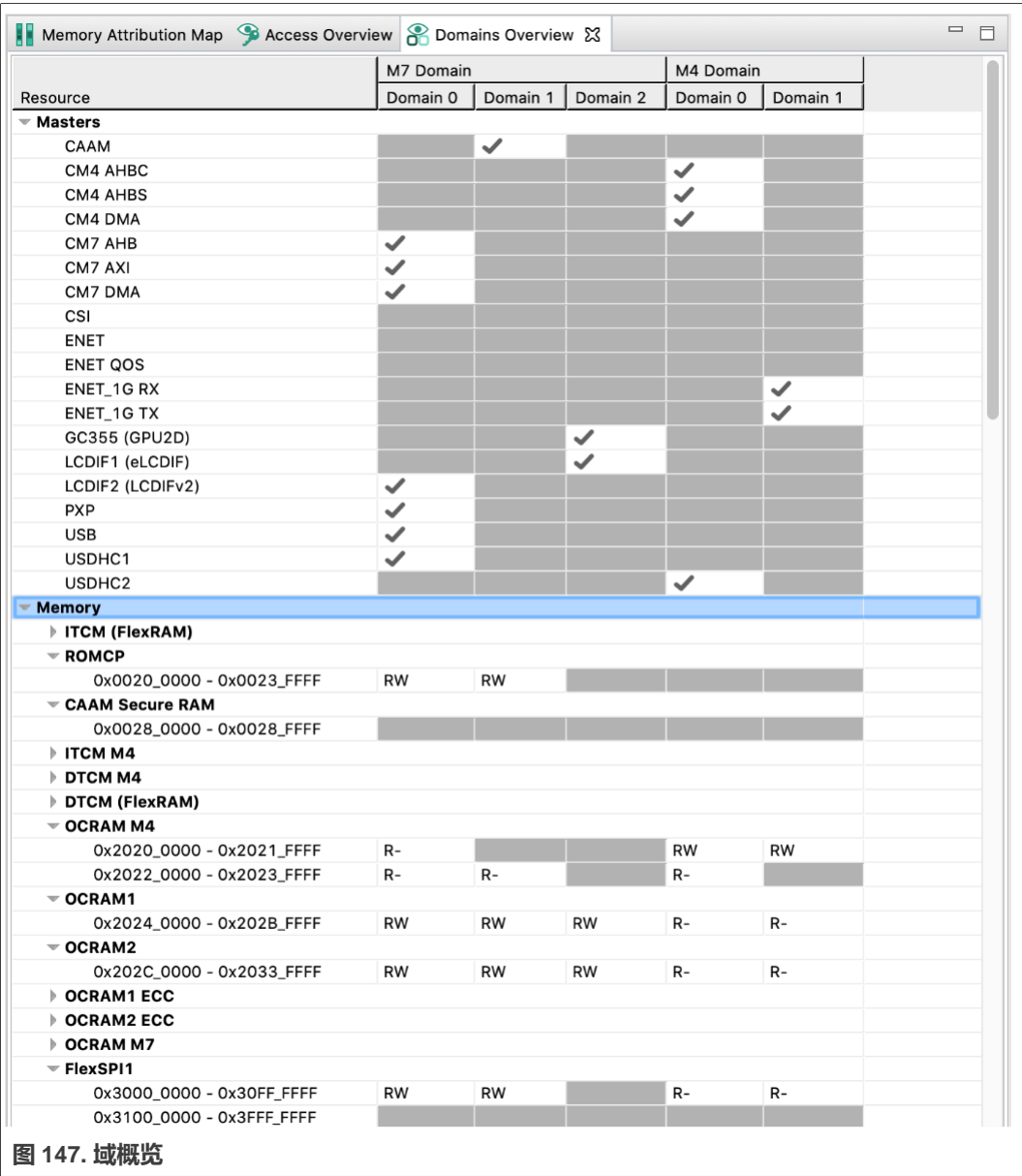


图 147. 域概览

5.2.6 代码生成

如果设置正确并且没有报错，则代码生成引擎重新生成源代码。您可以在 **Trusted Execution Environment (可信执行环境)** 工具的 **Code Preview (代码预览)** 视图中查看生成的代码。

**Code Preview (代码预览)** 会自动高亮显示当前代码迭代与前一个迭代之间的差异。

您可以单击 **Set viewing style for source differences (设置源代码差异的查看样式)**，在两种高亮显示模式之间进行选择。您还可以从同一下拉菜单中完全禁用高亮显示。

一些支持安全扩展功能的 AHBSC 或 TRDC 设备支持 ROM 预设和 C 语言代码。您可以在 **Miscellaneous (杂项)** 子视图中选择在 ROM 预设中生成代码。

6 高级功能

6.1 切换处理器

您可以将当前配置的处理器或封装切换到不同的处理器或封装。但是，切换到完全不同的处理器可能会导致各种问题，如无法访问引脚路由或时钟输出频率不稳定。在这种情况下，有必要手动解决问题。例如，转到 **Routing Details (路由详情)** 视图，重新配置所有报告错误或冲突的引脚。或者，您可能需要更改所需的时钟输出频率。

要更改所选配置中的处理器，请从 **Menu bar (菜单栏)** 中选择 **File > Switch processor (文件 > 切换处理器)**。

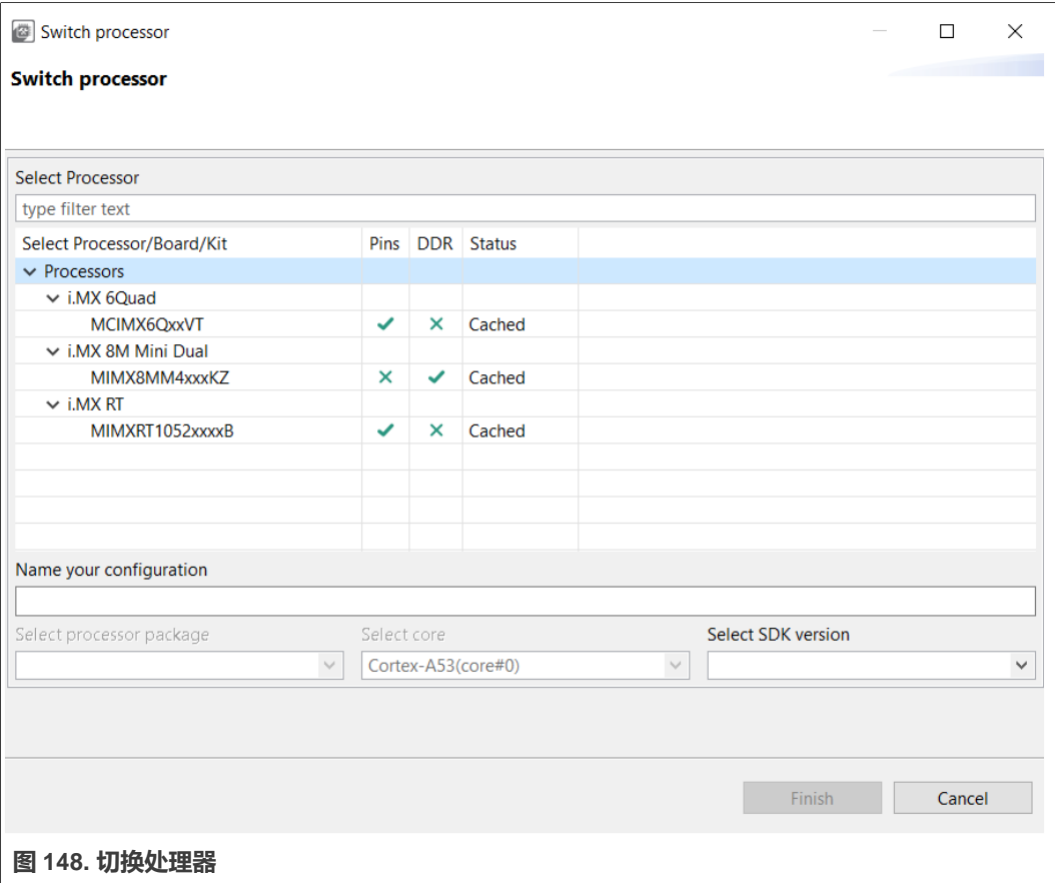
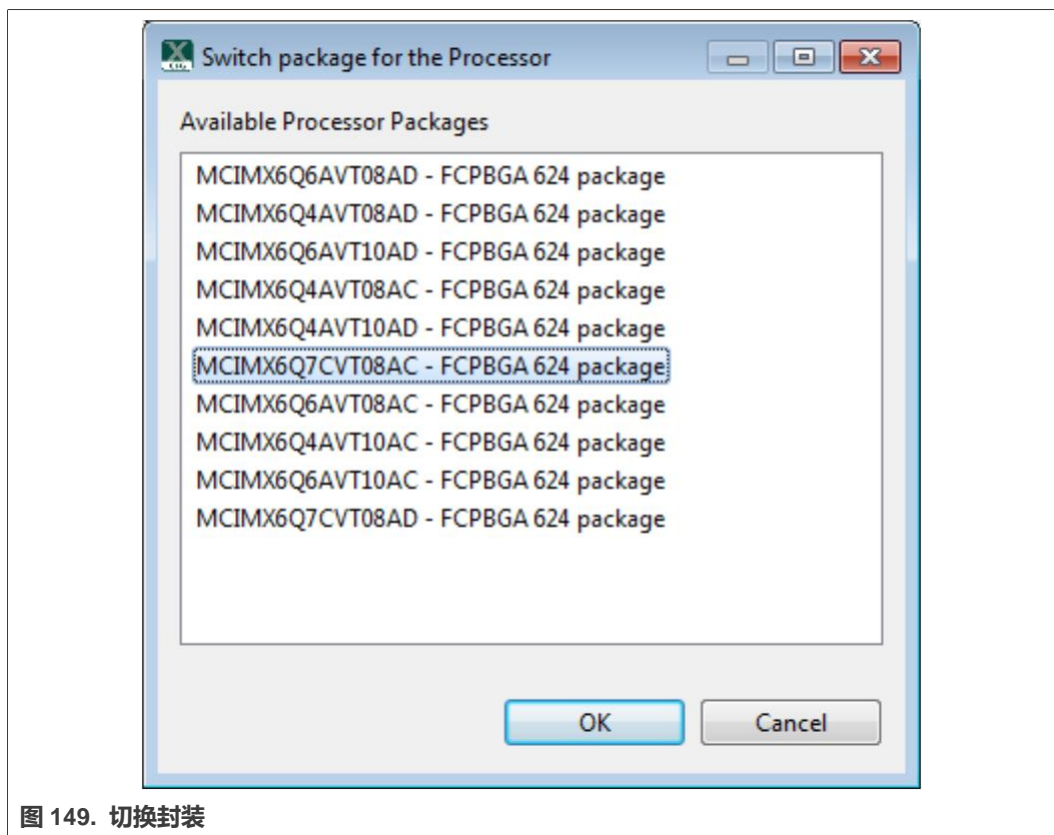


图 148. 切换处理器

要更改当前所选处理器的封装，请从 **Menu bar (菜单栏)** 中选择 **文件 > 切换处理器** **File > Switch processor (文件 > 切换处理器)**。



## 6.2 导出引脚表

要导出引脚表，请按照以下步骤操作：

1. 在 **Menu bar (菜单栏)** 中，选择 **File > Export (文件 > 导出)**。
2. 在 **Export (导出)** 向导中，选择 **Export the Pins in CSV (Comma Separated Values) Format (以 CSV (逗号分隔值) 格式导出引脚)**。
3. 单击 **Next (下一步)**。
4. 选择文件夹并指定要导出的文件名。
5. 导出的文件包含引脚视图表的内容，并列出功能和所选的路由引脚。

[illegible]

**图 150. 导出的文件内容**

导出的内容可用于其他工具进行进一步处理。例如，请在与下图中的块对齐后查看它。

[illegible]

图 151. 与块对齐

## 6.3 工具高级配置

使用 tools.ini 文件配置处理器数据目录位置。您可以定义 “com.nxp.mcudata.dir” 属性来设置数据目录的位置。

例如: -Dcom.nxp.mcudata.dir=C:/my/data/directory.

## 6.4 生成 HTML 报告

您可以生成 HTML 报告文件，显示引脚、时钟和外设工具的配置，以供将来参考。

要生成 HTML 报告，请选择 **Export > Pins > Export HTML Report**（导出 > 引脚 > 导出 HTML 报告）。

## 6.5 导出源文件

您可以使用导出向导，来导出生成的源文件。

要启动导出向导，请按照以下步骤操作：

1. 从 **Menu bar (菜单栏)** 中选择 **File > Export (文件 > 导出)**。
2. 选择 **Export Source Files (导出源文件)**。

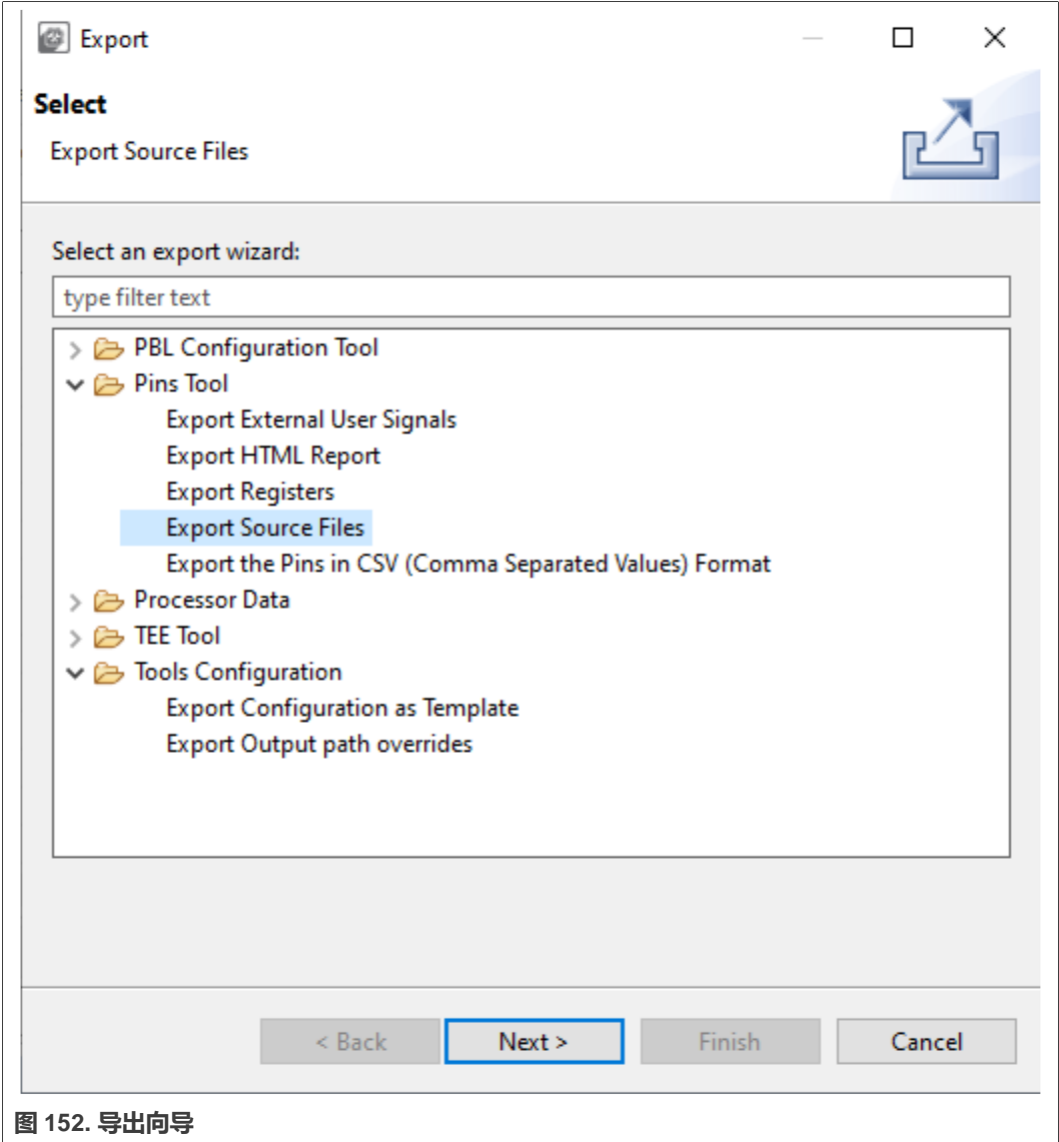


图 152. 导出向导

- 3. 单击 **Next (下一步)** 。
- 4. 选择存储生成文件的目标文件夹。

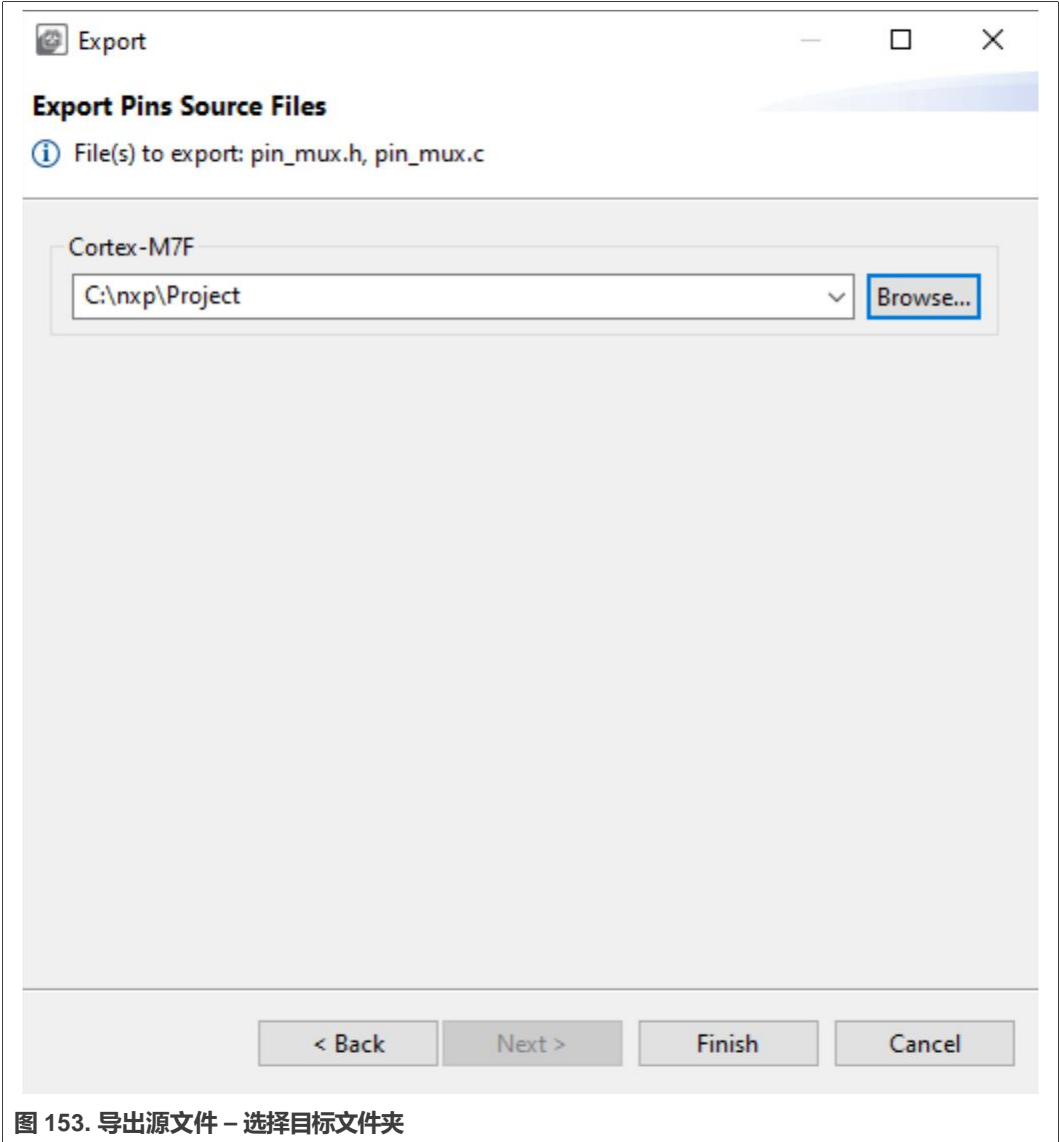


图 153. 导出源文件 – 选择目标文件夹

- 5. 如果是多核处理器，请选择要导出的处理器核心。
- 6. 单击 **Finish（完成）**。

6.6 导出寄存器

您可以使用导出向导导出工具修改的寄存器数据内容。

导出寄存器的步骤如下：

- 1. 从主菜单中选择 **File > Export（文件 > 导出）**。
- 2. 选择 **Pins Tool > Export Registers（引脚工具 > 导出寄存器）** 选项。
- 3. 单击 **Next（下一步）**。
- 4. 选择要导出已修改寄存器内容的目标文件路径。
- 5. 单击 **Finish（完成）**。

6.7 管理数据和脱机工作

使用 **Data Manager（数据管理器）**，您可以下载、导入和导出处理器数据。如果您想在离线状态下充分利用工具，这项功能尤其有用。

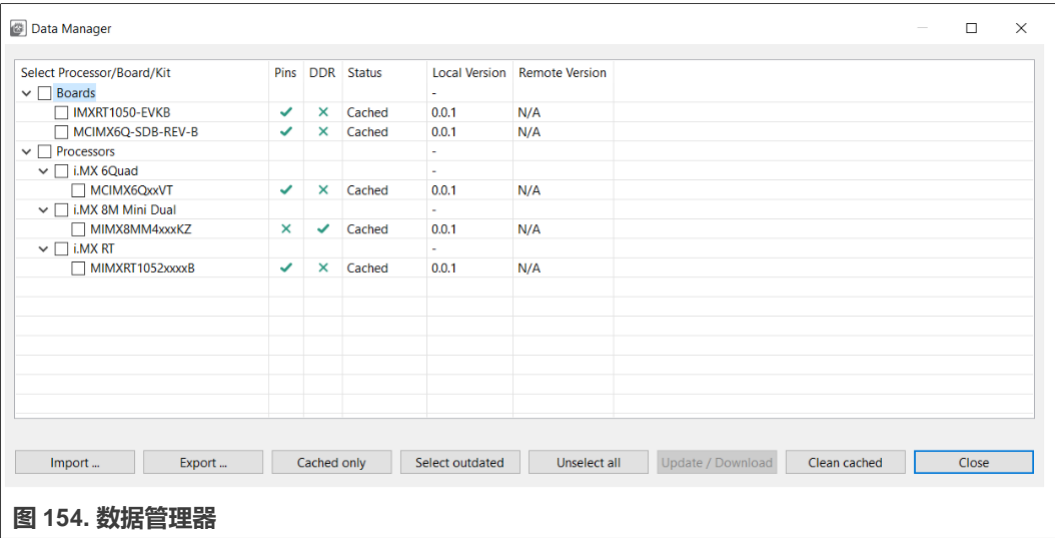


图 154. 数据管理器

6.7.1 脱机工作

即使在无法上网的情况下，您也可以使用缓存的处理器数据创建新配置。为此，您必须在脱机前下载处理器特定的数据，或导入从在线计算机下载和导出的数据。

要脱机工作，请从 **Menu bar（菜单栏）** 中选择 **Edit > Preferences > Work offline（编辑 > 首选项 > 脱机工作）**。

6.7.2 下载数据

您可以使用 **Data Manager（数据管理器）** 下载所需的处理器数据。

**注意：**默认情况下，在 **Creating a new standalone configuration for processor, board, or kit（为处理器、电路板或套件创建新的独立配置）** 过程中会自动下载并缓存数据。

要下载处理器数据，请按照以下步骤操作：

**注意：**下载数据需要连接互联网。

1. 在 **Menu bar（菜单栏）** 中，选择 **File>Data Manager（文件 > 数据管理器）**。
2. 在 **Data Manager（数据管理器）** 中，从列表中选择要处理的处理器/电路板/套件。
3. 单击 **Update / Download（更新/下载）** 并确认。

数据已下载到本地计算机上，如 **Data Manager（数据管理器）** 中的 **Cached（缓存）** 状态所示。

现在您可以关闭互联网连接，并在 **Start development（启动开发）** 向导中选择 **File > New...>Create new standalone configuration for processor, board, or kit（文件 > 新建...>为处理器、板或套件创建新的独立配置）** 来处理数据。

6.7.3 导出数据

使用 **Data Manager（数据管理器）**，您可以 ZIP 格式导出下载的处理器数据。



要导出数据，请按照以下步骤操作：

1. 在 **Menu bar (菜单栏)** 中，选择 **File>Data Manager (文件 > 数据管理器)**。
2. 在 **Data Manager (数据管理器)** 中，单击 **Export (导出)**。
3. 在 **Export Processor Data (导出处理器数据)** 窗口中，选择要导出的处理器数据。
4. 单击 **Browse (浏览)** 指定生成的 ZIP 文件的位置和名称。
5. 单击 **Finish (完成)**。

数据现在以 ZIP 格式保存在本地计算机上。您可以使用 U 盘等实体部件将数据移动到脱机计算机上。

**注意：**您也可以从 **Menu bar (菜单栏)** 中选择 **File > Export > Processor Data > Export Processor Data (文件 > 导出 > 处理器数据 > 导出处理器数据)** 来导出下载的数据。

#### 6.7.4 导入数据

您可以使用 **Data Manager (数据管理器)** 从另一台计算机导入处理器数据，前提是这些数据在本地可用。

要导入数据，请按照以下步骤操作：

1. 在 **Menu bar (菜单栏)** 中，选择 **File>Data Manager (文件 > 数据管理器)**。
2. 在 **Data Manager (数据管理器)** 中，单击 **Import (导入)**。
3. 在 **Import Processor Data (导入处理器数据)** 对话框中，单击 **Browse (浏览)**。
4. 指定要导入的 ZIP 文件的位置，然后单击 **OK (确定)**。
5. 勾选表中的复选框，可选择要导入的数据。
6. 单击 **Finish (完成)**。

数据现在导入到脱机计算机，如 **Data Manager (数据管理器)** 中的 **Cached (缓存)** 状态所示。现在，您可以在 **Start development (启动开发)** 向导中选择 **New...>Create new standalone configuration for processor, board, or kit (新建...>为处理器、板或套件创建新的独立配置)** 来处理数据。

**注意：**您也可以通过从 **Menu bar (菜单栏)** 中选择 **File>Import>Import Processor Data (文件>导入>导入处理器数据)** 来导入数据。

#### 6.7.5 更新数据

您可以使用 **Data Manager (数据管理器)** 更新缓存数据。

**注意：**如果在 **菜单栏的编辑>首选项** 中选择相关选项，数据将自动更新或在提示后更新。

**注意：**数据更新需要互联网连接。

要更新缓存数据，请按照以下步骤操作：

1. 在 **Menu bar (菜单栏)** 中，选择 **File>Data Manager (文件 > 数据管理器)**。
2. 在 **Data Manager (数据管理器)** 中，单击 **Select outdated (选择过时数据)** 过滤过时数据。
3. 单击 **Update / Download (更新/下载)** 并确认。

单击 **Cached only (仅缓存)** 并比较 **Local Version (本地版本)** 和 **Remote Version (远程版本)** 列中的版本信息，即可随时检查数据的版本。

选择 **Clean cached (清除缓存)** 可清除所有缓存数据。它会删除计算机中的所有处理器、电路板、套件和组件数据以及 SDK 信息文件。

**注意：**此操作不会影响用户模板。

## 6.8 输出路径覆盖

本节包含覆盖工具生成的输出文件路径（包括名称）的规则。这些规则将应用于更新代码、导出向导和命令行导出命令。这些规则存储在 MEX 配置中。

**注意：**无效路径会被记录为警告，并使用未被覆盖的原始路径。

您可在配置设置中的“输出路径覆盖”对话框中编辑规则。新规则会添加到列表末尾，并对所选元素进行移除。规则按定义的顺序应用于路径，该顺序可以更改。规则包含：

- 启用 - 定义应用路径是使用规则还是跳过规则。
- 描述 - 作为规则的用户友好描述
- 正则表达式 - 匹配整个输出路径中的覆盖部分。格式取自 Java 正则表达式。
- 替换表达式 - 用于替换路径中的所有匹配内容。可以使用占位符 \$1、2 等来引用子串组。

输出路径覆盖规则可通过向导导出到 yaml 文件中。yaml 文件的结构与对话框类似。

输出路径覆盖 yaml 文件内容示例：

```
outputPathOverrides:
-description: Rule group.h
enabled:true
regex: (bo)ar(d) (/.*\..h)
replacement: $2ar$1$3
-description:Rule2
...
```

设置规则的第二种方法是使用向导或命令行覆盖 yaml 文件中的输出路径来替换规则。只有当所有规则都有效时，才会使用规则。空列表会删除当前规则。输出路径中的空列表会覆盖 yaml 文件。

```
outputPathOverrides: [
]
```

## 6.9 从传统工具项目导入引脚配置

**Config Tools for i.MX (i.MX 配置工具)** 中的引脚工具可帮助从现有传统工具项目中导入引脚配置，前提条件如下。

- 从传统工具项目导入引脚配置之前，先将所需的 i.MX 处理器引脚工具数据下载到本地目录。
- 导入前，为给定封装变体的相应 i.MX 处理器创建新配置，尽量减少手动修正导入引脚配置的需要。

### 6.9.1 从 IO Mux 工具设计配置文件导入

您可以从 IO Mux 工具设计配置项目文件 (XML) 中导入现有的引脚配置，该文件用于在传统 IO Mux 工具中保留引脚路由配置。导入向导用于从与 IO Mux Tool v3.4.0.3 版兼容的现有项目 XML 文件中导入引脚路由配置。

要从 IO Mux Tool 设计配置文件导入，请按照以下步骤操作：

1. 从主菜单中选择 **File > Import (文件 > 导入)**。
2. 选择 **Import Legacy IOMux Tool Design Configuration (XML) Format (导入传统 IOMux 工具设计配置 (XML) 格式)** 选项。
3. 单击 **Next (下一步)**。
4. 在 **Import (导入)** 对话框中使用 **Browse (浏览)** 按钮，选择要导入的源文件 (XML)。

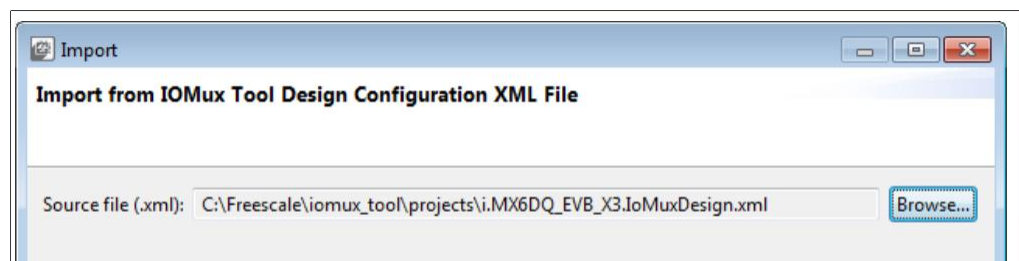


图 155. 从 IOMux Tool 设计配置 XML 文件导入

5. 单击 **Finish (完成)**。

处理选定的源文件，并将每个外设路由的现有引脚配置导入引脚工具。使用函数名 “configure\_<peripheral name>\_pins” 为每个外设实例创建一个包含所有配置引脚的新函数，并添加到 **Routed Pins (路由引脚)** 视图中。

### 6.9.2 从 Processor Expert 项目导入

您可以从 Processor Expert (PEX) 项目文件 (PE) 中导入现有引脚配置。PE 文件是传统 i.MX 引脚复用器组件的主项目文件，可在 Processor Expert for i.MX 工具中使用。

**Import (导入)** 向导可从现有的 PEX 项目文件中导入传统 i.MX 引脚配置。

要从 Processor Expert 项目导入，请按照以下步骤操作：

1. 从主菜单中选择 **File > Import (文件 > 导入)**。
2. 选择 **Import Legacy i.MX Pins Configuration (PEX for i.MX) Format (导入传统 i.MX 引脚配置 (PEX for i.MX) 格式选项)**。
3. 单击 **Next (下一步)**。
4. **Import Pins Configuration from Processor Expert Project (从 Processor Expert 项目导入引脚配置)** 对话框中使用 **Browse (浏览)** 按钮，选择要导入的源文件 (.pe)。

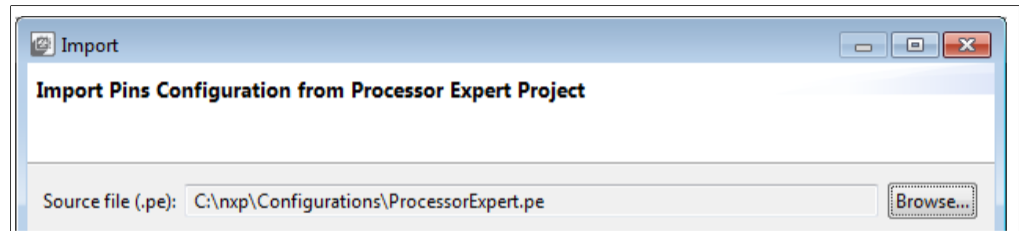


图 156. 从 Processor Expert 项目导入引脚配置

5. 单击 **Finish (完成)**。

处理选定的源文件，并将每个外设路由的现有引脚配置导入引脚工具。使用函数名“configure\_<peripheral name>\_pins”为每个外设实例创建一个包含所有配置引脚的新函数，并将其添加到路由引脚中。

## 6.10 命令行执行

本节介绍桌面应用程序支持的命令行接口 (CLI) 命令。

出错时应用程序退出：

- Tools v4.1 及更早版本：
  - 出现‘123321’错误代码。应记录原因。
- Tools v5.0 及更新版本：
  - 缺失参数时为 1
  - 工具出错时为 2

您可以在 CLI (命令行接口) 中执行命令序列。

命令行执行的注意事项：

- 命令 **-HeadlessTool** 用作每个命令链的分隔符。
- 每个命令链都独立运行。
- 每条命令链都以 **-HeadlessTool** 命令开始，并持续到下一条 **-HeadlessTool** 命令或结束。(唯一例外的是不需要 **-HeadlessTool** 命令的框架命令)。
- 不需要 **-HeadlessTool** 命令的命令，如果是链式命令，可以放在第一个 **-HeadlessTool** 命令之前；如果不是链式命令，则不带 **-HeadlessTool** 命令。
- 每个工具的命令都按指定顺序执行。
- 框架中的命令 **不按指定顺序执行**。
- 以下命令不按指定顺序执行：
  - ImportProject
  - Export MEX
  - ExportAll
- 出现意外行为时，应用程序会以下列代码退出：参数缺失：
  - 参数缺失时：1
  - 工具出错时：2

命令示例：

```
-HeadlessTool Clocks -MCU MK64FX512xxx12 -SDKVersion ksdk2_0 -
ExportSrc C:/exports/src -HeadlessTool Pins -MCU MK64FX512xxx12
-SDKVersion ksdk2_0 -ExportSrc C:/exports/src -HeadlessTool
Peripherals -MCU MK64FX512xxx12 -SDKVersion ksdk2_0 -ExportSrc
C:/exports/src
```

为了提高性能，当预计使用同一处理器多次运行 CLI 时，只有在**磁盘上没有数据时**才会加载数据。如果服务器上有更新的数据，则**不会更新**。

长时间运行的任务共享数据，所以它们不会在执行过程中更新。要更新可能在服务器上有新版本的本地数据，请使用-updateData 参数。

推荐的用法如下：

- 手动运行一次命令时，请在 CLI 中加上-updateData 参数。
  - 多次执行命令时，例如持续集成设置任务，可执行以下操作：
    - 使用带有-updateData 的命令，它会更新可能过时的数据。
    - 使用软件包中的所有其他命令时不使用此参数：copy /Y eclipse.exe toolsc.exe @rem 如果存在较新数据，则会更新所有本地数据
- ```
tools.exe -updateData -consoleLog -HeadlessTool Pins
@rem 现在多次运行工具
tools.exe -consoleLog -HeadlessTool Pins -Load some.mex -ExportAll c:/directory
tools.exe -consoleLog -HeadlessTool Pins -Load other.mex -ExportAll c:/other_directory
@rem 等等。
```

该框架支持以下命令：

表 17. 框架支持的命令

| 命令名称   | 定义和参数       | 说明                                                | 限制规定                                            | 示例               |
|--------|-------------|---------------------------------------------------|-------------------------------------------------|------------------|
| 产品版本   | -version    | 将产品的构建版本显示到 stdout 中，并继续分析其他参数。（自 6.0 起）          |                                                 | -version         |
| 强制设置语言 | -nl {lang}  | 强制设置语言 {lang}符合 <a href="#">ISO-639-1</a> 标准      | 建议从主目录中删除'.npx' 文件夹，因为可能会缓存某些文本<br>仅支持'zh'和'en' | -nl zh           |
| 显示控制台  | -consoleLog | 日志输出也会发送到 Java 的 System.out（如果有的话，通常会返回到命令 shell） | 无                                               |                  |
| 选择 MCU | -MCU        | MCU 将由框架选择<br>更改上一命令链结果配置中的处理器                    | 需要-SDKversion 命令                                | -MCU MCIMX6QxxVT |

表 17. 框架支持的命令 (续)

| 命令名称           | 定义和参数                 | 说明                                                                     | 限制规定                               | 示例                                                              |
|----------------|-----------------------|------------------------------------------------------------------------|------------------------------------|-----------------------------------------------------------------|
| 选择电路板          | -Kit                  | 由框架选择的电路板 (MCU 也是自动选择的) (自 6.0 起)                                      | 需要-SDKversion 命令                   | -Board FRDM-K22F-SDKversion ksdk2_0                             |
| 选择套件           | -SDKversion           | 由框架选择的套件 (MCU 和板也自动选择) (自 6.0 起)                                       | 需要-SDKversion 命令                   | -Kit FRDM-K22F-AGM01 -SDKversion ksdk2_0                        |
| 选择 SDK 版本      | -SDKversion           | 由框架选择的 MCU 版本                                                          | 需要-MCU 命令                          | -SDKversion i_mx_1_0                                            |
| 选择部件号          | -PartNum              | 选择 MCU 的特定封装                                                           | 需要-MCU 和-SDKversion 命令             | -PartNum MCIMX6Q6 AVT10AD                                       |
| 配置名称           | -ConfigName           | 新建配置的名称 - 用于导出                                                         | 使用-MCU 和-SDKversion 命令创建新配置时会使用该名称 | -ConfigName "My Config"                                         |
| 选择工具           | -HeadlessTool         | 选择应在无头模式下运行的工具                                                         | 无                                  | -HeadlessTool Pins                                              |
| 加载配置           | -Load                 | 从(*.mex)文件中加载现有配置                                                      | 无                                  | -Load C:/conf/conf.mex                                          |
| 导出 Mex         | -ExportMEX            | 工具运行后导出.mex 配置文件<br>参数应为文件夹名称                                          | 无                                  | -MCU xxx -SDKversion xxx -ExportMEX C:/exports/my_config_folder |
| 导出所有生成的文件      | -ExportAll            | 导出生成的文件 (含源代码等)。导出前重新生成代码<br>包括-ExportSrc 和框架中的-ExportMEX<br>参数应为文件夹名称 | 需要-HeadlessTool 命令                 | -HeadlessTool Pins -ExportAll C:/exports/generated              |
| 通过导入工具链项目创建新配置 | -ImportProject {path} | 通过导入工具链项目创建新配置<br>参数为工具链项目根目录的路径                                       | 需要-HeadlessTool 命令                 | -HeadlessTool Pins -ImportProject c:\test\myproject             |
| 生成具有自定义版权的源文件  | -CustomCopyright      | 文件内容作为版权文件标题注释插入到生成的源文件(.c, .h, .dts, .dtsi)中, 不包含版权。                  | 需要-HeadlessTool 命令                 | -CustomCopyright c:\test\copyright.txt                          |

表 17. 框架支持的命令 (续)

| 命令名称        | 定义和参数                 | 说明                                      | 限制规定 | 示例                                                        |
|-------------|-----------------------|-----------------------------------------|------|-----------------------------------------------------------|
| 覆盖生成文件的输出路径 | -OutputPath Overrides | 包含规则的文件路径，用于覆盖生成文件的输出路径。空规则列表会删除已设置的规则。 |      | -OutputPathOverrides c:\test\outputPathOverrideRules.yaml |
| 更新本地下载的数据   | -updateData           | 如果本地下载的数据有更新，则下载这些数据。                   |      | -updateData                                               |

6.10.1 命令行执行 – 引脚工具

本节介绍引脚工具中支持的命令行接口 (CLI) 命令。

表 18. 引脚支持的命令

| 命令名称                       | 定义和参数       | 说明                                                                                           | 限制规定               | 示例                                                                         |
|----------------------------|-------------|----------------------------------------------------------------------------------------------|--------------------|----------------------------------------------------------------------------|
| 启用工具                       | -Enable     | 如果在当前配置中禁用该工具，则启用该工具                                                                         | 需要-HeadlessTool 引脚 | -HeadlessTool Pins - Enable                                                |
| 导入 C 文件                    | -ImportC    | 将.c 文件导入配置在加载 mex 之后和生成输出之前完成导入                                                              | 需要-HeadlessTool 引脚 | -HeadlessTool Pins -ImportC C:/imports/file1.c C:/imports/file2.c          |
| 导入 DTSI 文件                 | -ImportDTSI | 将.dtsi 文件导入配置在加载 mex 之后和生成输出之前完成导入                                                           | 需要-HeadlessTool 引脚 | -HeadlessTool Pins -ImportDTSI C:/imports/file1.dtsi C:/imports/file2.dtsi |
| 导出所有生成的文件 (将所有导出命令简化为一个命令) | -ExportAll  | 导出生成的文件 (带有源代码等) 导出前将重新生成代码 包括-ExportSrc、 -ExportCSV、 -ExportHTML 和框架中的-Export MEX 参数应为文件夹名称 | 需要-HeadlessTool 引脚 | -HeadlessTool Pins -ExportAll C:/exports/generated                         |



表 18. 引脚支持的命令（续）

| 命令名称         | 定义和参数            | 说明                                             | 限制规定               | 示例                                                         |
|--------------|------------------|------------------------------------------------|--------------------|------------------------------------------------------------|
| 导出源文件        | -ExportSrc       | 导出生成的源文件。<br>导出前将重新生成代码<br>参数应为文件夹名称           | 需要-HeadlessTool 引脚 | -HeadlessTool Pins -<br>ExportSrc C:/exports/<br>src       |
| 导出 CSV 文件    | -ExportCSV       | 导出生成的 csv 文件。<br>导出前将重新生成代码<br>参数应为文件夹名称       | 需要-HeadlessTool 引脚 | -HeadlessTool Pins -<br>ExportSrc C:/exports/<br>src       |
| 导出 HTML 报告文件 | -ExportHTML      | 导出生成的 html 报告文件。<br>导出前将重新生成代码<br>参数应为文件夹名称    | 需要-HeadlessTool 引脚 | -HeadlessTool Pins<br>-ExportHTML C:/<br>exports/html      |
| 导出寄存器        | -ExportRegisters | 将寄存器选项卡导出到<br>文件夹中。<br>导出前将重新生成代码<br>参数应为文件夹名称 | 需要-HeadlessTool 引脚 | -HeadlessTool Pins<br>-ExportRegisters C:/<br>exports/regs |

6.10.2 命令行执行 – TEE 工具

本节介绍 TEE 工具支持的命令行接口（CLI）命令。

表 19. TEE 工具支持的命令

| 命令名称                              | 定义和参数      | 说明                                                                                                                | 限制规定                     | 示例                                                        |
|-----------------------------------|------------|-------------------------------------------------------------------------------------------------------------------|--------------------------|-----------------------------------------------------------|
| 启用工具                              | -Enable    | 如果在当前配置中禁用<br>该工具，则启用该工具                                                                                          | 需要 -Headless<br>Tool TEE | -HeadlessTool TEE -<br>Enable                             |
| 导出所有生成的文件<br>（将所有导出命令简<br>化为一个命令） | -ExportAll | 导出生成的文件（带有<br>源代码等）<br>导出前将重新生成代码<br>包括-ExportSrc,<br>-ExportHTML 和框架中<br>的 framework –<br>ExportMEX<br>参数应为文件夹名称 | 需要 -<br>HeadlessTool TEE | -HeadlessTool TEE -<br>ExportAll C:/exports/<br>generated |

表 19. TEE 工具支持的命令（续）

| 命令名称  | 定义和参数      | 说明                                      | 限制规定                     | 示例                                                  |
|-------|------------|-----------------------------------------|--------------------------|-----------------------------------------------------|
| 导出源文件 | -ExportSrc | 导出生成的源文件<br>导出前将重新生成<br>代码<br>参数应为文件夹名称 | 需要 -<br>HeadlessTool TEE | -HeadlessTool TEE -<br>ExportSrc C:/exports/<br>src |

7 支持

如果您有任何问题或需要其他帮助，请在论坛上搜索或发布新问题。请访问 [community.nxp.com/community/imx](https://community.nxp.com/community/imx) 获得帮助。

8 修订历史

表 20. 修订历史

| 版本号 | 日期               | 实质性变更                              |
|-----|------------------|------------------------------------|
| 0   | 2020 年 4 月 27 日  | 初版发布                               |
| 1   | 2021 年 7 月 1 日   | 微小改动                               |
| 2   | 2021 年 12 月 22 日 | 添加了新功能，<br>更新了截图。                  |
| 3   | 2022 年 6 月 30 日  | 更新至第 12 版                          |
| 4   | 2022 年 9 月 20 日  | 更新了第 1.8.3.1 和 1.8.3.2 节，<br>略有更正。 |
| 5   | 2023 年 1 月 2 日   | 更新了屏幕截图，<br>更新了第 4 章。              |

## 9 Legal information

### 9.1 Definitions

**Draft** — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

### 9.2 Disclaimers

**Limited warranty and liability** — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

**Right to make changes** — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

**Suitability for use** — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

**Applications** — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

**Terms and conditions of commercial sale** — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at <http://www.nxp.com.cn/profile/terms>, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

**Export control** — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

**Suitability for use in non-automotive qualified products** — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

**Evaluation products** — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out of the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US\$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

**Translations** — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

**Security** — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at [PSIRT@nxp.com](mailto:PSIRT@nxp.com)) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

### 9.3 Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

**NXP** — wordmark and logo are trademarks of NXP B.V.

**AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamiQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, uVision, Versatile** — are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

**Airfast** — is a trademark of NXP B.V.

**Bluetooth** — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

**Cadence** — the Cadence logo, and the other Cadence marks found at [www.cadence.com/go/trademarks](http://www.cadence.com/go/trademarks) are trademarks or registered trademarks of Cadence Design Systems, Inc. All rights reserved worldwide.

**CodeWarrior** — is a trademark of NXP B.V.

**ColdFire** — is a trademark of NXP B.V.

**ColdFire+** — is a trademark of NXP B.V.

**EdgeLock** — is a trademark of NXP B.V.

**EdgeScale** — is a trademark of NXP B.V.

**EdgeVerse** — is a trademark of NXP B.V.

**eIQ** — is a trademark of NXP B.V.

**FeliCa** — is a trademark of Sony Corporation.

**Freescale** — is a trademark of NXP B.V.

**HITAG** — is a trademark of NXP B.V.

**ICODE and I-CODE** — are trademarks of NXP B.V.

**Immersiv3D** — is a trademark of NXP B.V.

**I2C-bus** — logo is a trademark of NXP B.V.

**Kinetis** — is a trademark of NXP B.V.

**Layerscape** — is a trademark of NXP B.V.

**Mantis** — is a trademark of NXP B.V.

**MIFARE** — is a trademark of NXP B.V.

**NTAG** — is a trademark of NXP B.V.

**Processor Expert** — is a trademark of NXP B.V.

**QorIQ** — is a trademark of NXP B.V.

**SafeAssure** — is a trademark of NXP B.V.

**SafeAssure** — logo is a trademark of NXP B.V.

**Synopsys** — Portions Copyright © 2021 Synopsys, Inc. Used with permission. All rights reserved.

**Tower** — is a trademark of NXP B.V.

**UCODE** — is a trademark of NXP B.V.

**VortiQa** — is a trademark of NXP B.V.

## 目录

|          |                 |           |            |                                                 |            |
|----------|-----------------|-----------|------------|-------------------------------------------------|------------|
| <b>1</b> | <b>介绍</b>       | <b>2</b>  | <b>4.1</b> | <b>新建 DDR 工具项目</b>                              | <b>56</b>  |
| 1.1      | 特性              | 2         | 4.2        | DDR 配置                                          | 56         |
| 1.2      | 版本              | 2         | 4.2.1      | 导入初始化脚本                                         | 57         |
| 1.3      | 工具本地化           | 3         | 4.2.2      | 从目标导入                                           | 57         |
| <b>2</b> | <b>用户界面</b>     | <b>3</b>  | 4.2.3      | 启用手动配置                                          | 58         |
| 2.1      | 启动开发向导          | 3         | 4.2.4      | 用户界面配置                                          | 58         |
| 2.2      | 创建、保存和打开配置      | 4         | 4.2.5      | 代码生成                                            | 61         |
| 2.2.1    | 新建配置            | 4         | 4.3        | DDR 验证                                          | 62         |
| 2.2.1.1  | 新建独立配置          | 4         | 4.3.1      | 连接                                              | 63         |
| 2.2.2    | 保存配置            | 5         | 4.3.1.1    | 支持串行下载模式/制造模式的电路板                               | 63         |
| 2.2.3    | 打开现有配置          | 5         | 4.3.1.2    | 支持 JTAG 连接的电路板                                  | 64         |
| 2.2.4    | 用户模板            | 6         | 4.3.2      | 测试场景                                            | 64         |
| 2.2.5    | 导入源代码文件         | 7         | 4.3.2.1    | 检查                                              | 66         |
| 2.2.5.1  | 导入配置            | 9         | 4.3.2.2    | 优化                                              | 66         |
| 2.2.5.2  | 导入电路板/套件配置      | 10        | 4.3.2.3    | vTSA                                            | 67         |
| 2.2.6    | 导出源文件           | 11        | 4.3.2.4    | 压力测试                                            | 68         |
| 2.3      | 菜单栏             | 12        | 4.4        | 常见问题解答                                          | 70         |
| 2.4      | 工具栏             | 14        | <b>5</b>   | <b>可信执行环境工具</b>                                 | <b>72</b>  |
| 2.4.1    | 配置工具概览          | 15        | 5.1        | AHBSC 支持安全扩展功能的器件                               | 74         |
| 2.4.2    | 更新代码            | 16        | 5.1.1      | 用户内存区域视图                                        | 74         |
| 2.4.3    | 函数组             | 16        | 5.1.2      | 安全接入配置视图                                        | 75         |
| 2.4.3.1  | 函数组属性           | 17        | 5.1.2.1    | SAU                                             | 75         |
| 2.4.4    | 撤销/恢复操作         | 18        | 5.1.2.2    | 中断                                              | 76         |
| 2.5      | 首选项             | 19        | 5.1.2.3    | 安全/非安全 MPU                                      | 77         |
| 2.6      | 配置首选项           | 20        | 5.1.2.4    | MPC                                             | 79         |
| 2.7      | 问题视图            | 21        | 5.1.2.5    | 主设备/从设备                                         | 80         |
| 2.8      | 寄存器视图           | 22        | 5.1.2.6    | 引脚                                              | 81         |
| 2.9      | 日志视图            | 23        | 5.1.2.7    | 杂项                                              | 83         |
| <b>3</b> | <b>引脚工具</b>     | <b>23</b> | 5.1.3      | 内存属性映射                                          | 84         |
| 3.1      | 引脚路由原理          | 24        | 5.1.3.1    | 内核 0                                            | 84         |
| 3.1.1    | 从选择引脚/内部信号开始    | 24        | 5.1.3.2    | 简单主设备和智能主设备                                     | 85         |
| 3.1.2    | 外设信号的路由         | 25        | 5.1.4      | 访问概览                                            | 86         |
| 3.2      | 使用示例            | 30        | 5.1.5      | 代码生成                                            | 87         |
| 3.3      | 用户界面            | 32        | 5.2        | 支持 RDC 的器件                                      | 87         |
| 3.3.1    | 引脚视图            | 33        | 5.2.1      | 用户内存区域视图                                        | 88         |
| 3.3.2    | 封装              | 34        | 5.2.1.1    | 访问模板                                            | 88         |
| 3.3.3    | 外设信号视图          | 36        | 5.2.2      | 安全访问配置视图                                        | 89         |
| 3.3.3.1  | 引脚和外设信号视图中的过滤控件 | 38        | 5.2.2.1    | RDC                                             | 89         |
| 3.3.4    | 路由详情视图          | 39        | 5.2.2.2    | XRDC2 域视图                                       | 92         |
| 3.3.4.1  | 标签和标识符          | 41        | 5.2.2.3    | i.MX8 ULP 中 Cortex-A35 上的 XRDC (扩展可信资源域控制器)     | 100        |
| 3.3.5    | 扩展头             | 42        | 5.2.2.4    | i.MX8 ULP 和 KW45 中 Cortex-M33 上的可信资源域控制器 (TRDC) | 102        |
| 3.3.5.1  | 扩展板             | 46        | 5.2.2.5    | 杂项                                              | 105        |
| 3.3.6    | 电源组             | 48        | 5.2.3      | 内存归属映射                                          | 106        |
| 3.3.7    | 外部用户信号视图        | 49        | 5.2.4      | 接入概览                                            | 108        |
| 3.3.8    | 函数              | 50        | 5.2.5      | 域概览                                             | 109        |
| 3.3.9    | 高亮显示和颜色编码       | 50        | 5.2.6      | 代码生成                                            | 109        |
| 3.4      | 错误和警告           | 52        | <b>6</b>   | <b>高级功能</b>                                     | <b>110</b> |
| 3.4.1    | 不完整路由           | 53        | 6.1        | 切换处理器                                           | 110        |
| 3.4.2    | 电源组电压电平冲突       | 53        | 6.2        | 导出引脚表                                           | 111        |
| 3.5      | 代码生成            | 54        | 6.3        | 工具高级配置                                          | 112        |
| 3.6      | 在代码中使用引脚定义      | 54        |            |                                                 |            |
| 3.7      | 引脚完全初始化         | 55        |            |                                                 |            |
| 3.8      | 创建默认路由          | 55        |            |                                                 |            |
| <b>4</b> | <b>DDR 工具</b>   | <b>55</b> |            |                                                 |            |

6.4 生成 HTML 报告 .....112

6.5 导出源文件 .....112

6.6 导出寄存器 .....114

6.7 管理数据和脱机工作 .....115

6.7.1 脱机工作 .....115

6.7.2 下载数据 .....115

6.7.3 导出数据 .....115

6.7.4 导入数据 .....116

6.7.5 更新数据 .....116

6.8 输出路径覆盖 .....117

6.9 从传统工具项目导入引脚配置 .....117

6.9.1 从 IO Mux 工具设计配置文件导入 .....118

6.9.2 从 Processor Expert 项目导入 .....118

6.10 命令行执行 .....119

6.10.1 命令行执行 – 引脚工具 .....122

6.10.2 命令行执行 – TEE 工具 .....123

7 支持 .....124

8 修订历史 .....124

9 法律声明 .....125

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.