Application Note

MSPM0 硬件编程和调试程序指南

Henry Nguyen

摘要

本应用手册介绍了 MSPM0 的调试子系统、复位，以及不同器件系列之间的 flashctl 差异。调试子系统是与 MSPM0 中的 M0+ 内核分离的实体。通过正确利用 MSPM0 中的调试子系统，用户可在低功耗状态下访问 M0+ 内核，并在错误配置时将其恢复，因为该子系统是与 M0+ 内核分离的。通过了解 MSPM0 的调试子系统、flashctl 和复位，可利用 MSPM0 实现理想的调试和编程环境。

内容

插图清单

表格清单

商标

EnergyTrace™ is a trademark of Texas Instruments.

1 调试子系统和 MSPM0 简介

MSPM0 是一款低功耗 MCU，以低成本提供密集功能集。为了平衡外设的性能和功耗，将外设分为两个单独的电源域 [1] 称为 PD0 和 PD1。PD1 域中的外设由 CPU、存储器和高性能外设组成，而 PD0 由低速、低功耗外设组成。进入比 SLEEP 更强的低功耗模式时，会禁用 PD1 外设以降低功耗。这会导致不能发现 AHB 总线，但是 MSPM0 中包含一个称为调试子系统的外设，其支持再次发现 AHB。通过将调试子系统与 M0+ 内核分离，为调试程序或编程程序提供了一种在错误配置或低功耗状态的场景下重新访问器件的方法。在低功耗状态下访问器件或将其重新配置为“已知状态”是通过一组称为访问端口的寄存器来完成的。本应用手册深入介绍了 SEC-AP 和 PWR-AP，因为它们是实现先前讨论的功能的重要元件。除了调试子系统之外，还讨论了 flashctl 及其在器件系列之间的不同保护方案，以及通过 AIRCR 进行的独特复位。

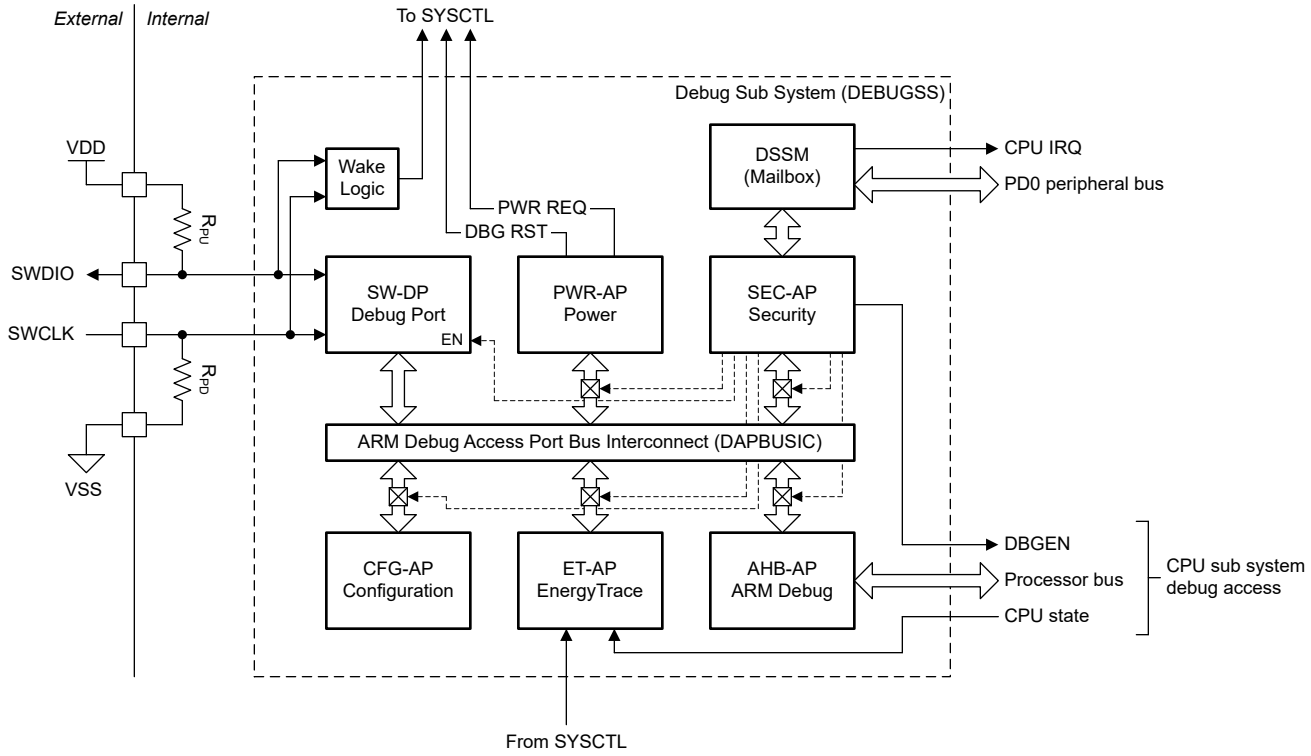

图 1-1. 调试子系统方框图

1.1 访问 MSPM0 的端口

MSPM0 器件包含 图 1-1 中所示的总共五个访问端口，用户可使用调试访问端口 (DAP) 直接与这些端口交互。每个访问端口都有自己独特的功能，使用户能够在 M0+ 内核之外对器件进行配置和读取。

1.1.1 高级高性能总线访问端口

此访问端口也称为 AHB-AP，为用户提供了从 DAP 到 M0+ 内核的桥接器，允许用户通过直接存储器存取来直接与器件交互。

1.1.2 配置访问端口

此访问端口也称为 CFG-AP，可以为用户提供器件信息及其当前运行状态。

1.1.3 安全访问端口

此访问端口也称为 SEC-AP，使用户能够与器件引导代码通信，以擦除闪存存储器并重新启用对器件的访问。

[1] 有关电源域的更多信息，请参阅 *MSPM0L110x 混合信号微控制器数据表*。

1.1.4 EnergyTrace™ 访问端口

此访问端口也称为 ET-AP，使用户能够读取电源状态数据。

1.1.5 电源访问端口

此访问端口也称为 PWR-AP，为用户提供了配置器件电源状态和复位行为的方法。

1.2 处于空白/低功耗状态的 MSPM0 的行为

将 MSPM0 器件置于大于 SLEEP 的低功耗模式的结果是使 AHB-AP 不可发现。这是因为 AHB 是 PD1 的一部分。进入 DEEPSLEEP（STOP 和 STANDBY）后，将禁用 PD1，从而防止 AHB-AP 处于可发现状态。除了应用程序代码将器件置于低功耗状态之外，清空闪存也会使器件置于 STANDBY0 状态。

当任何 MSPM0 器件中的主存储器为空并通电约 5 至 10 秒时，引导代码将进入并填充 SRAM 中的内容，然后开始从中执行。填充 SRAM 的程序将器件置于 STANDBY0 中，进而撤消 AHB-AP 访问。若要使 AHB-AP 再次可见，用户必须利用 PWR-AP 为器件供电。修改 PWR-AP（见 图 1-1）可在 CPU 保持低功耗状态时提供用户调试访问。

2 正确的 SWD 初始化序列

MSPM0 使用 Arm® M0+ 内核，允许用户按照 Arm 介绍的程序来将器件从 JTAG 切换到 SWD。

在执行 Arm 所介绍的序列从 JTAG 切换到 SWD 时，所看到的唤醒逻辑单元会向器件发送唤醒请求，允许读取 IDCODE，而不管器件处于何种低功耗状态，也允许访问端口可用。

最好是在开始任何操作之前使 SWJ-DP 状态机处于已知状态。在进入状态之间之前，执行线路复位，然后开始 SWD 到 JTAG 序列，这是为了确保线路处于复位状态并且已经初始化为已知状态。然后执行线路复位，再执行 JTAG 到 SWD 序列，这样做之后，唤醒逻辑单元向 CPU 发送唤醒信号，即使器件处于关断模式，也可以读取 IDCODE。若要了解实现 SWD 初始化序列时应该做什么，请参阅 图 2-1 中的流程图。

图 2-1. MSPM0 SWD 初始化序列

3 PWR-AP

PWR-AP 是包含两个寄存器（称为 DPREC0 和 SPREC）的访问端口。这些寄存器可用于启用低功耗模式调试、修改复位行为以及从外部执行复位。

3.1 使用 MSPM0 启用低功耗模式调试

MSPM0 包含一个称为 PWR-AP 的访问端口。器件处于低功耗状态时，该访问端口用于重新启用 AHB-AP 以重新访问 M0+ 内核。

若要在器件处于低功耗状态时访问器件并保持连接，用户必须将 1 写入这些位。可以在 表 3-3 中看到寄存器视图。

表 3-1. DPREC0 低功耗模式配置位

DPREC0 位	说明
FRC ACT (位 20)	强制器件退出低功耗模式
IHIB SLP (位 3)	不允许系统进入低功耗模式

写入 FRC ACT 位会强制器件退出低功耗状态，从而允许再次发现 AHB-AP，写入 IHIB SLP 甚至会在 CPU 收到进入 DEEPSLEEP 模式的请求时保持与器件的连接。添加对 MSPM0 的支持时，必须启用这些位。如果未写入 FRC ACT 和 IHIB SLP，则不能在 M0+ 处于低功耗模式时访问它。

3.2 修改 MSPM0 的复位行为

利用 DPREC0 寄存器的 RST CTL 位，可以修改复位后看到的行为。表 3-2 包含 RST CTL 位 (16:14) 的所有可能配置。

表 3-2. RST CTL 位配置

RST CTL 模式	位配置
等待调试	001b
复位时停止	100b
默认复位	000b

3.2.1 等待调试

当向任何工具链添加对 MSPM0 的支持时，等待调试都是要实现的最重要配置。在器件后置配置时执行的任何复位仍将仅复位由复位级别定义的外设，但它将保留在复位处理程序后复位中。这是为了确保器件不会返回应用程序，除非用户希望继续执行应用程序。此配置在刷写器件或调试应用程序时尤其有用，因为它会使器件返回到已知状态并使其停止。

3.2.2 复位时停止

复位时停止将在进行配置后对器件执行任何形式的复位时立即停止器件。与等待调试的方式类似，这也很有用，但不会强制器件在复位处理程序中保持停止状态。

3.2.3 INRST 行为

执行等待调试序列时，DPREC0 中的位 17 将设置为 1，以指示处于等待调试模式。若要从等待调试状态中释放，用户必须将 1 写入 INRST 位，才能从等待调试状态中释放。

3.3 寄存器视图

表 3-3. PWR-AP 寄存器视图

APSEL	AP 名称	ADDR	BANK	INDEX	寄存器名称	位 31	位 30	位 29	位 28	位 27	位 26	位 25	位 24	位 23	位 22	位 21	位 20	位 19	位 18	位 17	位 16	位 15	位 14	位 13	位 12	位 11	位 10	位 9	位 8	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
4	PWR	0x00	0	0	DPREC0	保留											IHIBSLP	保留		INRST	RST CTL			保留											FRCACT	保留		
		0xF0	15	0	SPREC	保留																															SYSRST
		0xFC	15	3	IDR	访问点 ID																															

4 SEC-AP

当用户错误配置其时钟、禁用通过非主存储器的调试访问 [2] 或者将错误值编程到非主存储器的 CRC 寄存器中时，这将禁用对 MSPM0 的访问。在添加对 MSPM0 的支持时，了解 SEC-AP 至关重要，因为它用于在外设或非主存储器错误配置的情况下恢复器件。之前所说的恢复是通过如下来完成的：向邮箱发送调试子系统邮箱 (DSSM) 命令，然后通过执行 BOOTRST 来执行该命令。在调试端口访问仍然启用时，也可以在使用或不使用密码来禁用调试访问的情况下解锁调试访问。

4.1 DSSM 命令

当向邮箱发送 DSSM 命令时，它由在执行 BOOTRST 且仅执行 BOOTRST 时开始执行的引导代码提供服务。如果在邮箱中存在 DSSM 命令时执行 BOR 或 POR，该命令将被擦除并且得不到服务，因为复位级别将对电源域进行下电上电。请务必注意，如果为任何 DSSM 命令启用了密码，除非密码序列已经完全执行，否则不会完全执行命令。将命令发送到邮箱后，用户有两秒的窗口来发送与在非主存储器中设置的密码匹配的密码。只有成功，才会完全处理命令。对于所有可能的 DSSM 命令 [3] 请参阅 表 4-1。有关实现对 DSSM 命令和寄存器视图的支持时的流程，请参阅 表 4-2。

表 4-1. DSSM 命令表

DSSM 命令	DSSM 值
恢复出厂设置	0x020Ah
批量擦除	0x020Ch
密码身份验证	0x030Eh
数据交换	0x00EEh
等待调试	0x0206h

4.1.1 恢复出厂设置

在将“恢复出厂设置”命令的值发送到邮箱时进行处理。主存储器和非主存储器中的所有内容都将被擦除，然后非主存储器重新填充其默认内容。

此命令在以下情况下很有用：

- 非主存储器错误配置，但不太严重。
- 禁用调试访问。
- 外设/器件错误配置（例如，超频 PLL、非服务式 wwdt 或双硬故障）。

4.1.2 批量擦除

在将“批量擦除”命令的值发送至邮箱时进行处理。主存储器中的所有内容都将被擦除，而非主存储器则保持不变

与恢复出厂设置类似，此命令在以下情况下很有用：

- 外设/器件错误配置（例如，超频 PLL、非服务式 wwdt 或双硬故障）。

4.1.3 密码身份验证

在将“密码身份验证”命令的值发送到邮箱时进行处理。仅在处理密码之后才会解锁调试访问。

4.1.4 数据交换

数据交换是不需要复位即可处理命令的唯一 DSSM 命令。如果场景（例如，恢复出厂设置、批量擦除或密码身份验证）需要密码。将命令发送到邮箱并执行复位之后，用户必须开始向 TXDATA 寄存器发送密码。在每个字之后，必须将 0x00EEh 写入 TXCTL 寄存器。

[2] 有关非主存储器的更多信息，请参阅 *MSPM0 G 系列 80MHz 微控制器技术参考手册*。
[3] MSPM0C 和 MSPS 器件系列无法执行批量擦除、密码身份验证和数据交换。

4.1.5 等待调试

在将“等待调试”命令的命令发送到邮箱时进行处理。与之前在 节 3.2 中介绍的等待调试类似，它将复位由复位级别定义的外设，然后强制器件进入复位处理程序。然而，当通过邮箱执行等待调试序列时，在执行命令之后，为了完全执行命令，需要清除 INRST 位。

4.1.6 自定义 DSSM 命令

用户也可以创建自己的 DSSM 命令并让它执行用户所定义的操作。这可通过让调试程序通过 TXDATA 和 TXCTL 与 M0+ 内核通信来完成。然后，内核可从调试程序接收消息，并且使用 RXDATA 和 RXCTL 寄存器发回响应以供调试程序读取。还可以为 TX_DATA 缓冲器、RX_DATA 缓冲器和 DAP 连接中的活动配置 CPU 中断事件。

4.2 DSSM 流程

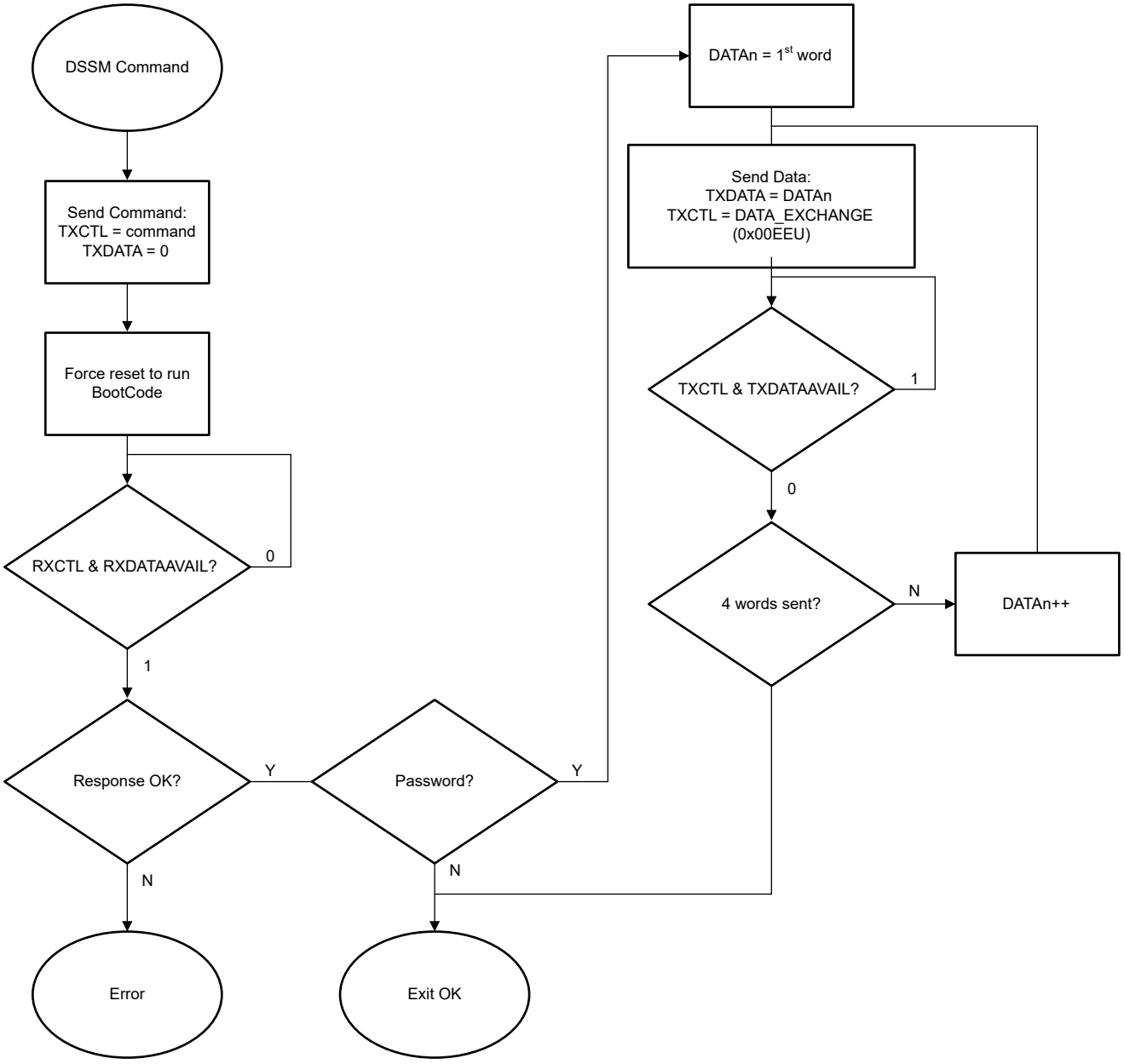

图 4-1. DSSM 命令流程

4.3 寄存器视图

表 4-2. SEC-AP 寄存器视图

APSEL	AP名称	ADDR	BANK	INDEX	寄存器名称	位31	位30	位29	位28	位27	位26	位25	位24	位23	位22	位21	位20	位19	位18	位17	位16	位15	位14	位13	位12	位11	位10	位9	位8	位7	位6	位5	位4	位3	位2	位1	位0
2	SEC	0x00	0	0	TXDATA	TX 数据																															
		0x04	0	1	TXCTL	TX 控制																															TXVLD
		0x08	0	2	RXDATA	RX 数据																															
		0x0C	0	3	RXCTL	RESERVED																									RX 控制						RXVLD
		0xFC	15	3	IDR	访问点 ID																															

5 了解 MSPM0 中的闪存

MSPM0 器件中的存储器被组织为多个组，每个存储器组包含各自大小为 1kB 的存储器扇区。对于某些 MSPM0 型号，它们也可以包含多个存储器组。实现 MSPM0 闪存的编程算法时，了解这一点非常重要[4]，因为充分了解器件之间的细微差别至关重要。同样重要的是，将不同存储器空间（如非主存储器和主存储器）之间的任何闪存操作隔离到各自的编程操作中。本应用手册的此部分进一步详细介绍了如何在常规调试到生产环境中处理 MSPM0 系列的闪存。

5.1 保护 MSPM0 上的闪存存储器

在 MSPM0 中闪存存储器的所有区域中，具有一个称为 CMDWEPROTx 的保护寄存器。它们是动态寄存器，将根据用户设置为零的位取消对存储器扇区的保护。在用户执行闪存操作之前，存储器保持非保护状态。执行任何闪存操作时，所使用的 CMDWEPROTx 寄存器会将所有位重新设置为 1，从而自动重新保护存储器。表 5-1 展示了用于所有器件的保护寄存器。

表 5-1. 用于 MSPM0 Flashctl 的保护寄存器

CMDWEPROTx 寄存器	MSPM0L11XX/ MSPM0L13XX	MSPM0G1X0X/ MSPM0G3X0X	MSPM0C110X/ MSPS003FX	MSPM0L122X/ MSPM0L222X	所有未来推出的 MSPM0 SOC
CMDWEPROTA	x	x	x	x	
CMDWEPROTB	x	x		x	x
CMDWEPROTNM	x	x	x	x	x

5.2 清除 STATCMD 寄存器

随着 MSPM0 器件系列通过更新的 SOC 进行扩展，添加闪存支持时的最佳实践也在不断发展。对于 MSPM0 的所有型号，最好在执行任何闪存操作之前清除 STATCMD 寄存器。通过将清除状态命令 (0x00000005h) 写入 CMDTYPE 寄存器并执行，可清除寄存器。执行命令时，将清除 STATCMD 寄存器中的所有既有内容。需要注意的是，清除 STATCMD 寄存器也会复位之前讨论的 CMDWEPROTx 寄存器。

在尝试执行扇区擦除时清除 STATCMD 寄存器的示例可参见以下步骤：

1. 向 CMDTYPE 寄存器写入清除状态 (0x00000005h) 命令。
2. 通过将执行密钥 (0x00000001h) 写入 CMDEXEC 寄存器来执行命令。
3. 通过检查 STATCMD 寄存器来轮询完成情况。
   a. 完成时，STATCMD 将清空。
4. 通过将 CMDWEPROTx 中所需的位设置为零来取消对存储器扇区的保护。
5. 设置 CMDTYPE 寄存器执行大小为一个扇区的擦除 (0x00000042h)。
6. 将 CMDADDR 寄存器设置为等于所需的操作地址。
7. 通过将执行密钥 (0x00000001h) 写入 CMDEXEC 寄存器来执行命令。
8. 通过检查 STATCMD 寄存器来轮询完成情况。

[4] 要更广泛地了解 flashctl 和 SDK 示例代码，请参阅 MSPM0。

5.3 MSPM0 的理想编程流程

为任何 MSPM0 器件创建闪存加载程序时，闪存算法对每个存储器区域单独执行修改始终至关重要。例如，如果用户的应用程序包含主存储器和非主存储器的内容。然后，该工具必须负责分别对每个存储器区域进行擦除和编程。这是为了尽可能限制非主存储器区域暴露于噪声等其他因素。由于所有安全配置都位于非主存储器区域内，因此在尝试修改它时必须注意这些事项。有关编程时的理想流程，请参阅 图 5-1。

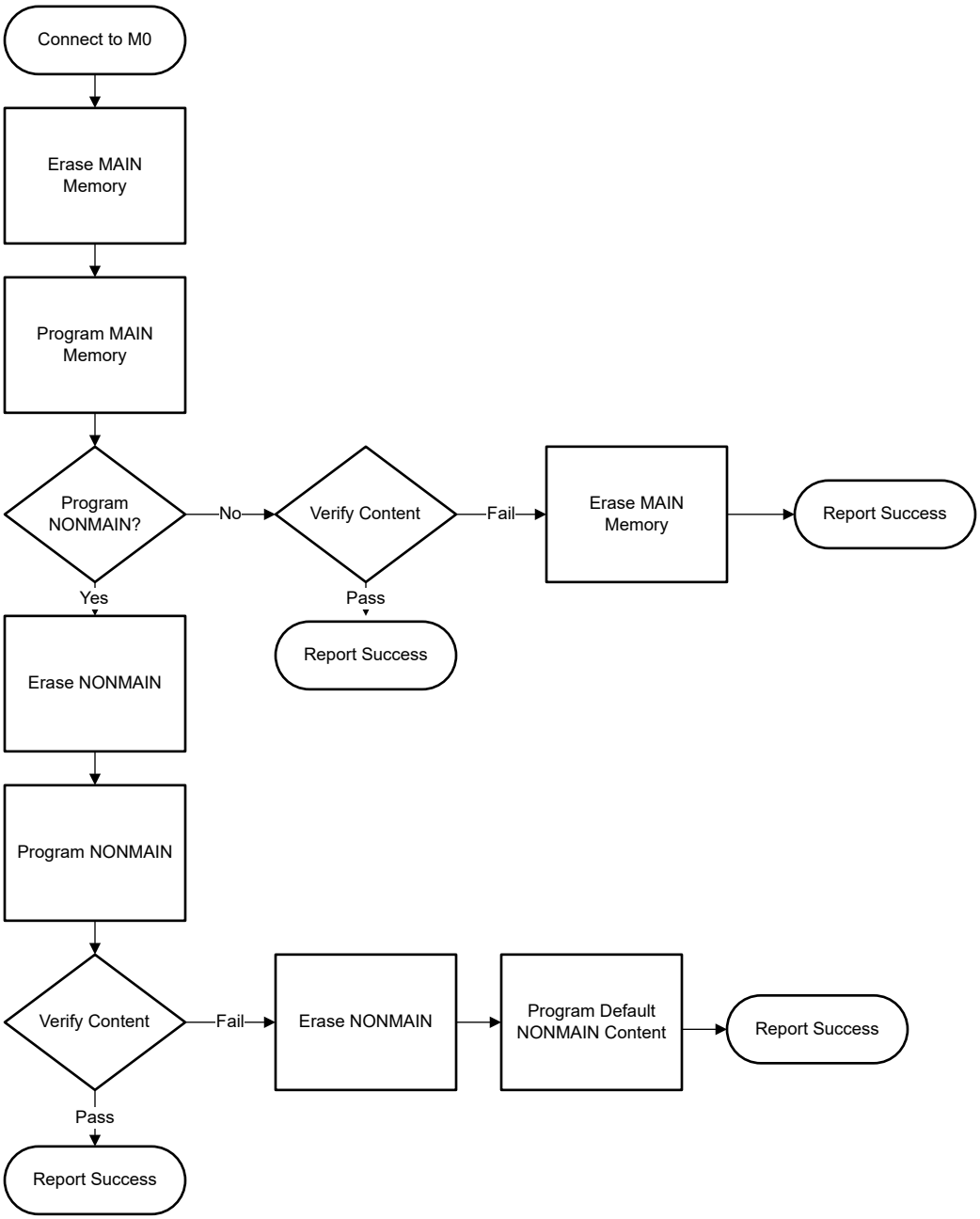

图 5-1. MSPM0 编程序列

6 MSPM0 的复位

添加支持时，了解 MSPM0 内复位的独特实现非常重要。通过 AIRCR 执行复位时，仅对 CPU 执行复位，而不对其他外设执行复位。若要执行系统复位，最好使用 SYSCTL 模块来执行复位。可使用 表 6-1 并按以下步骤完成此操作：

1. 将系统级复位 (0x00000000h) 写入复位级别寄存器。这是系统复位的值。
2. 将复位密钥 (0xE4000001h) 写入复位命令寄存器以执行复位。

添加支持是为了确保器件能够在空置任意时间之后退出引导代码。也可通过写入 PWR-AP 内 SPREC 的 SYS RST 位，从外部执行系统复位。由 SPREC 执行的系统复位低于通过 SYSCTL 进行的系统复位，但高于 CPU 复位。

表 6-1. Sysctl 复位寄存器

寄存器	地址
复位级别	0x400B0300h
复位命令	0x400B0304h

7 总结

本应用手册介绍了 MSPM0 的调试子系统、flashctl 和独特复位。通过本文档中提供的理解，任何第三方都可以轻松选择 MSPM0 并为其工具链实施强大的解决方案。这确保了客户使用该工具的体验尽可能理想。

8 参考资料

- 德州仪器 (TI)：*MSPM0L110x 混合信号微控制器数据表*
- 德州仪器 (TI)：*MSPM0 G 系列 80MHz 微控制器技术参考手册*

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做出任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 TI 的销售条款或 ti.com 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265