

1 介绍

使用 RT 系列的芯片时，不管在调试阶段还是在量产阶段，如果下载到外部 NOR FLASH 的 image 特别大，那整个下载过程会非常耗时，这大大影响了调试效率或者量产效率。为了解决这个痛点，本应用笔记介绍了使用 JLink 下载 image 时，如何优化 Flashloader 算法以提升下载速度。

2 Flashloader 算法简介

当在 RT 平台上使用 JLink 下载 image 时，目前几乎所有的 Flashloader 都是基于 SEGGER 提供的 Open Flashloader 模型开发的。Segger Flashloader 的框架以及开发使用流程在 [Segger](#) 官网上有详细的介绍。

2.1 SEGGER 支持用户自行添加目标设备

SEGGER 支持用户能够自行添加对新设备的支持，不依赖于 SEGGER 和 J-Link 软件的版本。默认情况下，J-Link DLL 带有一个内置设备数据库，该数据库定义了哪些设备名称是已知的。对于已知的设备，SEGGER 已经为其提供了默认的烧录下载算法。对于未知的设备，可以通过名为 JLinkDevices.xml 的 XML 文件添加其他设备。打开 SEGGER 的安装目录...\\Device\\NXP\\，添加 Flashloader 算法文件到指定路径，在 XML 文件中以 SEGGER 规定的脚本语法进行添加即可。如果往 XML 文件添加已知设备，JLink 会优先调用 XML 文件中的 Flashloader 算法文件。

```
<Database>
  <Device>
    <ChipInfo Vendor="..." Name="..." WorkRAMAddr="..." WorkRAMSize="..." Core="..." />
    <FlashBankInfo Name="..." BaseAddr="..." MaxSize="..." Loader="..." LoaderType="..."
  AlwaysPresent="..." />
  </Device>
</Database>
```

本文以 i.MX RT500 和 i.MX RT1060 平台，SEGGER 7.54d 版本为例，来演示如何提升 Flashloader 的烧录下载速度，因此我在 NXP 文件夹下添加了全新的 iMXRT5xx 和 iMXRT106x 文件夹，里面包含了 MIMXRT5XX_FLEXSPI.elf 和 MIMXRT106X_FLEXSPI.elf 文件。这两个文件就是优化过后的 Flashloader 可执行文件，它们可以是 .elf 文件，也可以是 .FLM 文件。

目录

1	介绍.....	1
2	Flashloader 算法简介.....	1
2.1	SEGGER 支持用户自行添加目标设备.....	1
2.2	Flashloader 算法文件制作平台....	2
2.3	Flashloader 算法文件制作方法....	2
3	影响 Flashloader 下载速度的几个因素.....	3
4	Flashloader 下载速度优化.....	7
4.1	基于 RT500 的 Flashloader 优化..	8
4.2	基于 RT1060 的 Flashloader 优化.....	14
4.3	Flashloader 下载速度优化结果..	15
5	参考资料.....	16
6	修订记录.....	16



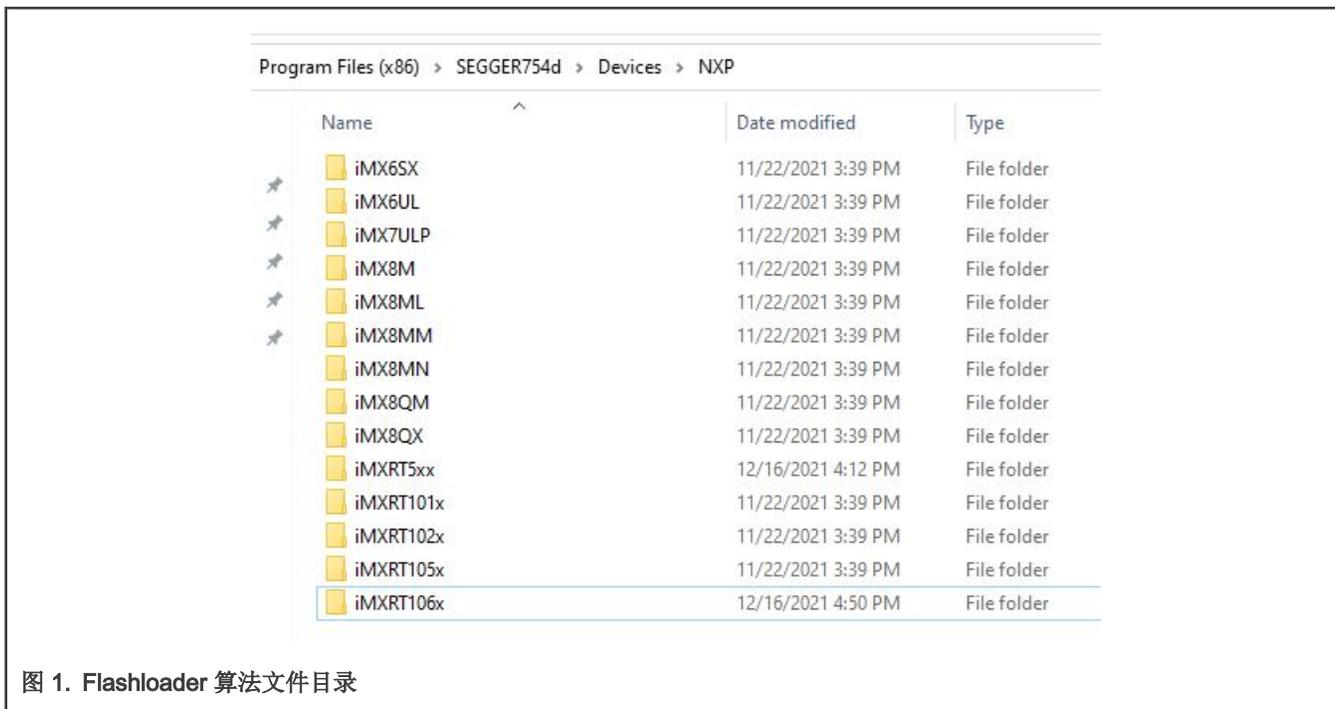


图 1. Flashloader 算法文件目录

2.2 Flashloader 算法文件制作平台

SEGGER 的下载烧录算法文件制作有两种方法：

- 使用 Keil uVision IDE。缺点有三：
 - 需要 license。
 - 没有试用版。
 - 仅支持 Cortex-M 内核的设备。
- 使用 SEGGER Embedded Studio IDE。有两种优势：
 - 基本上任何在商业范围内使用 SEGGER Embedded Studio 都需要有效的许可证，但是 Open Flashloader 是个例外。评估版 license 就可用于调试和创建 Flashloader，不需要有效的 license。
 - 支持 Cortex-M，Cortex-A/R，和 RISC-V 等多种内核。

2.3 Flashloader 算法文件制作方法

SEGGER Embedded Studio 提供了 Open Flashloader 的模板，利用 SEGGER Embedded Studio 就能生成最终的 Flashloader 可执行文件。Flashloader 模型有几个重要的 API 函数构成，如表 1 所示。这部分 API 需要用户实现对应 MCU 设备的 flash 接口代码，具体到 i.MX RT 平台上就是基于 FLEXSPI 外设和 NOR FLASH 之间的交互代码。

表 1. Opening Flashloader API

Function name	Function
Init	To handle the initialization of the Flash module.
UnInit	To handle the deinitialization of the Flash module.
Erase	To erase one Flash sector. The EraseSector function can erase a single sector.

下页继续

表 1. Opening Flashloader API (续上页)

Function name	Function
Program	To program one Flash page. The ProgramPage function can program a single page.
EraseChip	To erase the entire chip (Flash bank).
Verify	To compare a specified byte number of a provided data buffer with the content of the device.
BlankCheck	To check whether a memory region is blank.

3 影响 Flashloader 下载速度的几个因素

本章节介绍了影响 Flashloader 下载速度的因素，以及利用 SEGGER Embedded Studio 来优化 Open Flashloader 模型的一些思路。本章节中所有的实验数据都是基于 RT500 EVK 板子以及外部 NOR Flash (GD25LE64C) 展开的。

1. 下载时间组成部分

- a. 使用 Jlink 下载器通过 Commander 窗口往 RT500 EVK 板子下载一个 2 M bytes 的 image 到外部 flash。
- b. 下载结束后能在 Commander 窗口看到如 图 2 所示的 log。

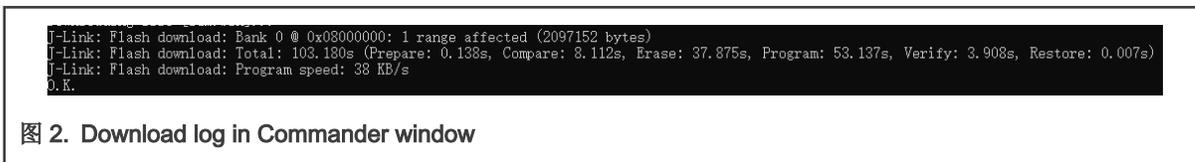


图 2. Download log in Commander window

显而易见，影响下载速度的因素有如 表 2 中所列。

- Erase 和 Program 耗时占比最大。
- Compare 和 Verify 耗时占比较小。
- Prepare 和 Restore 耗时几乎可以忽略。

因此优化下载速度的重点是 Erase 和 Program。

表 2. 下载时长组成

Image size	Prepare	Compare	Erase	Program	Verify	Restore	Total
2 M bytes	0.138 s	8.112 s	37.875 s	53.137 s	3.908 s	0.007 s	103.180 s

2. 下载速度相关因素。

以 4 线的 QSPI NOR Flash 为例，对于 Flash 的操作主要有读，写，擦这几个部分。表 3 是 NOR Flash 的 CMD 详情。

表 3. Flash CMD

Command name	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	n-Bytes
Write Enable	06H						
Write Disable	04H						
Volatile SR Write Enable	50H						
Read Status Register	05H	(S7-S0)					(continuous)

下一页继续

表 3. Flash CMD (续上页)

Command name	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	n-Bytes
Read Status Register-1	35H	(S15-S8)					(continuous)
Write Status Register	01H	S7-S0	S15-S8				
Read Data	03H	A23-A16	A15-A8	A7-A0	(D7-D0)	(Next byte)	(continuous)
Fast Read	0BH	A23-A16	A15-A8	A7-A0	dummy	(D7-D0)	(continuous)
Dual Output Fast Read	3BH	A23-A16	A15-A8	A7-A0	dummy	(D7-D0)	(continuous)
Dual I/O Fast Read	BBH	A23-A8	A7-A0 M7-M0	(D7-D0)			(continuous)
Quad Output Fast Read	6BH	A23-A16	A15-A8	A7-A0	dummy	(D7-D0)	(continuous)
Quad I/O Fast Read	EBH	A23-A0 M7-M0	dummy	(D7-D0)			(continuous)
Quad I/O Word Fast Read	E7H	A23-A0 M7-M0	dummy	(D7-D0)			(continuous)
Page Program	02H	A23-A16	A15-A8	A7-A0	D7-D0	Next byte	
Quad Page Program	32H	A23-A16	A15-A8	A7-A0	D7-D0		
Sector Erase	20H	A23-A16	A15-A8	A7-A0			
Block Erase (32 K)	52H	A23-A16	A15-A8	A7-A0			
Block Erase (64 K)	D8H	A23-A16	A15-A8	A7-A0			
Chip Erase	C7/60H						
Enable QPI	38H						
Enable Reset	66H						
Reset	99H						
Set Burst with Wrap	77H	W6-W4					
Program/Erase Suspend	75H						
Program/Erase Resume	7AH						

Read mode 对于 Flashloader 来说影响的是 Compare 和 Verify 的效率。Read 的 CMD 有 Read Data , Fast Read , Quad Output Fast Read , Quad I/O Fast Read, Quad I/O Word Fast Read 等，如表 4 所示。

区别是在 MCU 和 Flash 通信过程中：

- 发 CMD 命令是几线；
- 发地址信息是几线；
- 数据交互是几线。

一般来说如果是 4 线的 QSPI NOR Flash，推荐用 1_4_4 read mode。

表 4. Read CMD modes

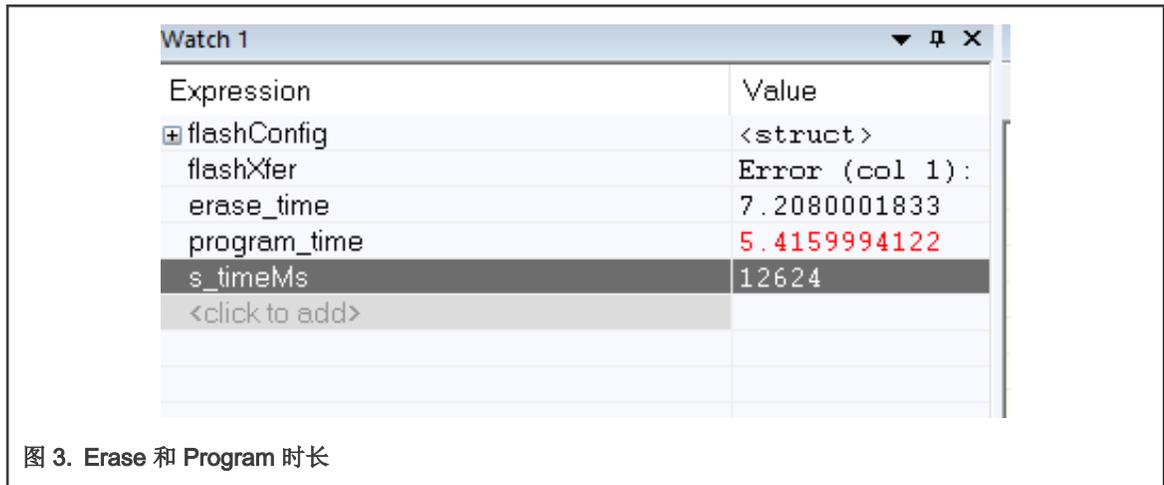
CMD	Read mode
Read Data(0x03)/Fast Read	1_1_1
Quad Output Fast Read	1_1_4
Quad I/O Fast Read	1_4_4

Erase CMD 除去 Chip Erase 有三种：Sector Erase，Block Erase (32 K)，Block Erase (64 K)。Erase 模式可以工作在 SPI mode，也可以工作在 QPI mode，一般工作在 SPI mode 就行。从擦的效率角度而言，擦除一块很大的区域，采用 Block Erase 比 Sector Erase 更有效率。

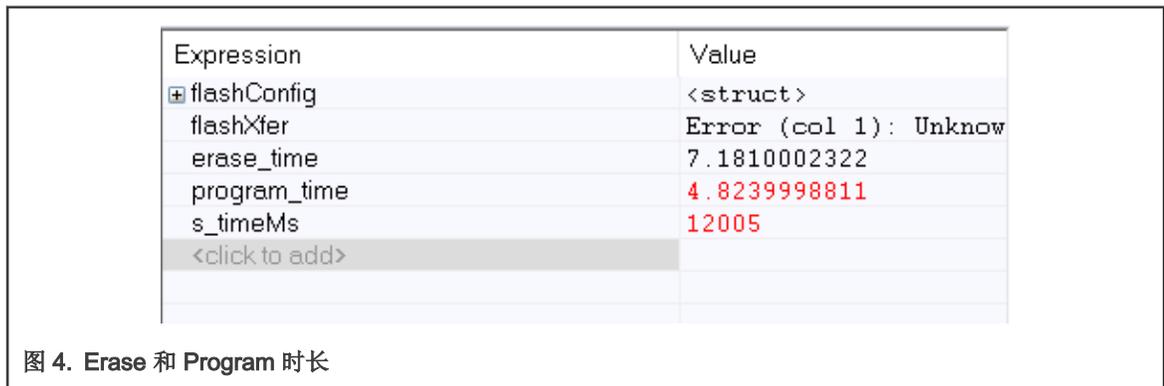
Program CMD 有两种，分别是 Page Program 和 Quad Page Program，一般使用 Page Program 就行。

影响 Flashloader 下载效率的因素除了 Read mode，Erase mode, Program mode 外，另一个重要的因素是 Flash 的时钟频率，但是这个因素对下载速度的影响是有限的，为了证明这一点做了如下实验。

- a. 在 RT500 EVK 板子上，外部 NOR Flash 更换为 GD25LE64C，在 IAR 下面去调试一个事先准备好的 Flashloader 算法，利用 MCU 的系统时钟来统计擦写时长，擦写的区域为 2MB 的外部 NOR Flash 空间。为了保证实验结果的可对比性，Flash 的起始地址一致，2MB 的 image 使用同一份 .bin 文件。Erase CMD 使用的是 Block Erase(64K)，Program CMD 使用的是 Page Program，只改变 Flexspi 的 CLK，可以得到如图 3 和 图 4 所示结果。
 - 设 Flexspi 的 CLK 频率为 30 M，可以得到如下结果：擦写时长为 12.624 s。



- 设 Flexspi 的 CLK 频率为 100 M，可以得到如下结果：擦写时长为 12.005 s。



因此得到的第一个结论是：Flash 的工作频率对于擦写速度影响不大。

- b. 设 Flexspi 的 CLK 频率为 100 M，只将 Program CMD 方式从 Page Program 变为 Quad Page Program，其他条件不变，可以得到如图 5 所示结果：擦写时长为 11.952 s。

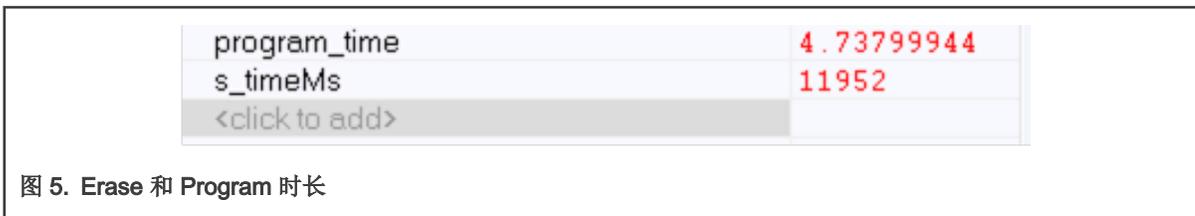


图 5. Erase 和 Program 时长

因此得到第二个结论是：Flash 的 Program 模式对于写一个 page 的整个过程，速度影响不大。

- c. 通过 Flash 的 datasheet 可以找到这款 flash 的 erase，program 的 typical 值，如表 5 所示。

表 5. Erase 和 Program 操作时长典型值

t _{PP}	Page Programming Time	0.7	5	ms
t _{SE}	Sector Erase Time	90	600	ms
t _{BE1}	Block Erase Time (32 K Bytes)	0.3	1.5	s
t _{BE2}	Block Erase Time (64 K Bytes)	0.45	3.0	s
t _{CE}	Chip Erase Time (GD25LE64C)	30	90	s

将理论值，IAR debug 测试数据，和 JLink Commander 测试数据进行比较，可以看到 Flashloader 算法在 IAR 中 debug 是非常接近甚至小于 flash 手册的 typical 值，但是当 Flashloader 算法被 Jlink Commander 调用时，擦写效率是非常低的，如表 6 所示。

表 6. Erase 和 Program 操作时长对比

Tool chain	Image size (MB)	Erase (s)	Program (s)
SEGGER commander	2	37.875	53.137
Test data on IAR	2	7.208	5.416
Typical value/Maximum value	2	14.4 (typical)/96.0 (max)	5.734 (typical)/40.960 (max)

因此得到的第三个结论是：影响 Flashloader 下载速度的最大瓶颈在于 Jlink Commander 对于 Flashloader 算法的调度，或者说是 Jlink Commander 与运行在 MCU 端的 Flashloader 程序之间的交互过程。

3. Open Flashloader 模型框架

从 Flash 本身已经找出了影响 Flash 下载速率的因素，那接下来就要了解 Jlink 通过 SWD/JTAG 接口对运行在 MCU 端的 Flashloader 的调度过程，具体主要是 Compare，Erase，Program，Verify 这几个部分。

上文提到，Flashloader 可执行文件可以由 Keil 或者 SEGGER Embedded Studio 编译生成。但是，Keil 的 Flashloader 模型中 Erase 和 Program 函数接口对于下载效率来说是有局限的，因此推荐使用 SEGGER Embedded Studio。

SEGGER Embedded Studio 提供的 Open Flashloader 模型提供了 SEGGER_OPEN_Program 和 Program 接口，SEGGER_OPEN_Program 接口与传统的 Program 接口相比可以实现更高的吞吐量，如图 6 所示。

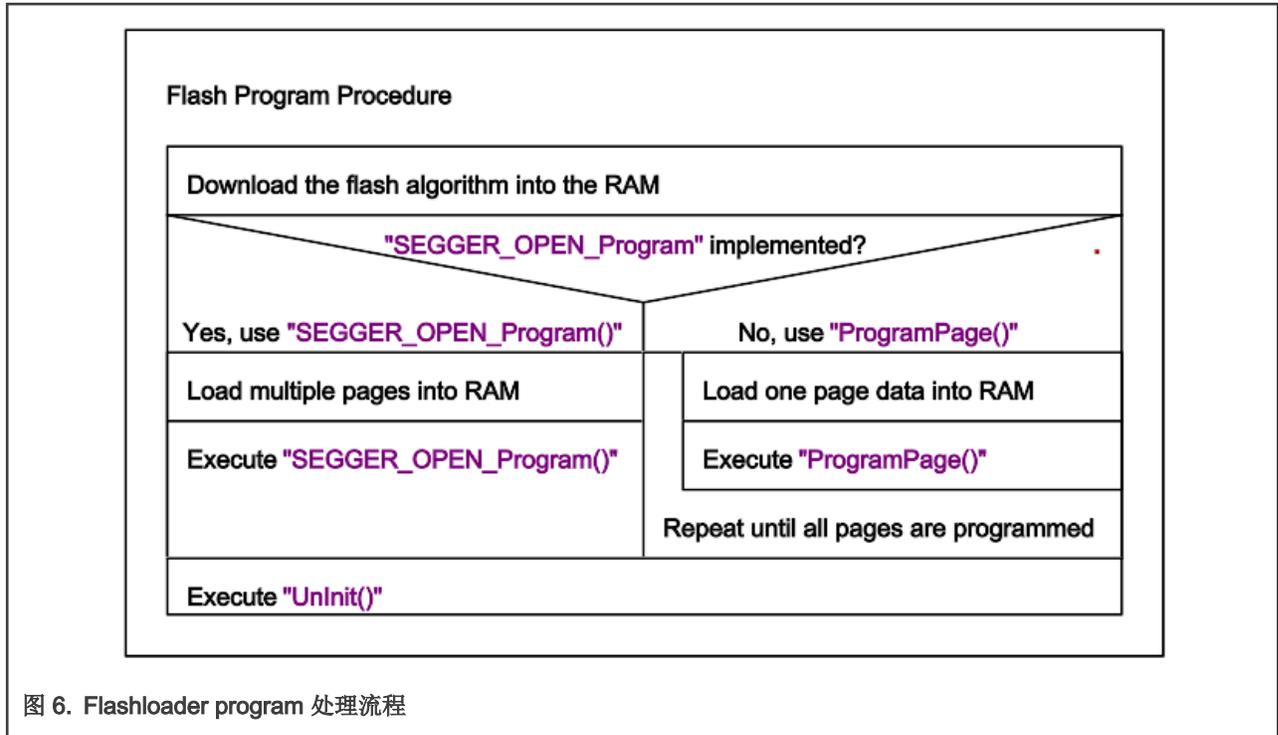


图 6. Flashloader program 处理流程

同样，SEGGER Embedded Studio 提供的 Open Flashloader 也提供了 SEGGER_OPEN_Erase 和 Erase 接口，这里的 Erase 接口是用于擦除一个或多个 sector 的。考虑到不同类型的 Flash 有不同的 sector size，因此 sector size 是用户可配置的。

通过上文分析，擦除一块很大的区域，采用 Block 擦会比 Sector 擦更有效率，而 Erase 接口函数是用户自定义的，所以用户可以将 Open Flashloader 中提供的 sector size 当作 block size 来使用，在底层代码中采用 block 去擦而不是 sector 擦，这会是一个提升 erase 效率的方案。

Open Flashloader 模型还提供了 Verify，Read，Erase_Chip 等 API 接口，另外针可以启用 TURBO Mode 去优化下载算法，这些都是可以优化下载速度的切入点。

4 Flashloader 下载速度优化

本章节基于 RT500 EVK 板子和 RT1060 EVKB 板子来验证提升下载速度的一些优化策略。为了比较 Flashloader 算法的优化结果，将 RT500 EVK 板子和 RT1060 EVKB 板子上的 Flash 都更换成 IS25WP064。表 7 是 IS25WP064 的 datasheet 中关于 Erase 和 Program 的理论值。根据 Typical 值，对 2 MB 的 Flash 空间进行 Sector Erase，Block Erase (64 K)。Program 的典型时长分别为 35.84 s，4.8 s，1.638 s。下面会将实验测试数据与理论数据进行比较。当然，实际用 JLink 对 Flash 的下载过程中肯定还有其它操作的时长需要考虑进去。

表 7. IS25WP064 Erase 和 Program 操作时长典型值

Symbol	Parameter	Min.	Type	Max.	Unit
t _{ec}	Sector Erase Time (4 kB)		70	300	ms
	Block Erase Time (32 kB)		0.1	0.5	s
	Block Erase Time (64 kB)		0.15	1.0	s
	Chip Erase Time	32 MB		8	23
64 MB			16	45	

下页继续

表 7. IS25WP064 Erase 和 Program 操作时长典型值 (续上页)

Symbol	Parameter	Min.	Type	Max.	Unit
	128 MB		30	90	
t _{pp}	Page Program Time		0.2	0.8	ms

4.1 基于 RT500 的 Flashloader 优化

Flashloader 可执行文件由 SEGGER Embedded Studio 或 Keil 生成，Erase 接口采用 Erase 接口，Program 接口采用 SEGGER_OPEN_Program 接口。IS25WP064 的 sector size 为 4 KB，page size 为 256 字节。SEGGER_OPEN_Program 可以一次加载多个 page 到 RAM 中进行缓存，这个数值是可以配置。本实验中：

- 将 SEGGER_OPEN_Program 接口中的一个 page 大小定义为变量 multi_page_size。
- 将 Program 接口中的一个 page 大小定义为变量 single_page_size。
- 将 Erase 接口中每次擦的大小定义为变量 single_erase_size。
- 将 Flexspi 的 CLK 时钟设置为 30 M。
- 擦的区域为 [0x0800 0000，0x0820 0000]，大小为 2 MB。

具体测试如下：

- Test 1
 - 平台：Keil
 - Erase 接口：Erase
 - Program 接口：Program
 - single_page_size：256 Byte
 - single_erase_size：4 kB
 - TURBO MODE：Disable

```
J-Link: Flash download: Bank 0 @ 0x08000000: 1 range affected (2097152 bytes)
J-Link: Flash download: Total: 95.803s (Prepare: 0.172s, Compare: 8.244s, Erase: 28.690s, Program: 54.772s, Verify: 3.917s, Restore: 0.007s)
J-Link: Flash download: Program speed: 37 KB/s
O.K.
```

图 7. Test 1

- Test 2
 - 平台：SEGGER Embedded Studio (SES)
 - Erase 接口：Erase
 - Program 接口：Program
 - single_page_size：256 Byte
 - single_erase_size：4 kB
 - TURBO MODE：Disable

```
J-Link: Flash download: Bank 0 @ 0x08000000: 1 range affected (2097152 bytes)
J-Link: Flash download: Total: 80.340s (Prepare: 0.243s, Compare: 2.551s, Erase: 28.270s, Program: 48.257s, Verify: 0.989s, Restore: 0.027s)
J-Link: Flash download: Program speed: 42 KB/s
O.K.
```

图 8. Test 2

• Test 3

- 平台 : SEGGER Embedded Studio (SES)
- Erase 接口 : Erase
- Program 接口 : Program
- single_page_size : 1 kB
- single_erase_size : 4 kB
- TURBO MODE : Disable

```
J-Link: Flash download: Bank 0 @ 0x08000000: 1 range affected (2097152 bytes)
J-Link: Flash download: Total: 43.063s (Prepare: 0.284s, Compare: 2.257s, Erase: 28.413s, Program: 16.089s, Verify: 0.989s, Restore: 0.028s)
J-Link: Flash download: Program speed: 127 KB/s
O.K.
```

图 9. Test 3**• Test 4**

- 平台 : SEGGER Embedded Studio (SES)
- Erase 接口 : Erase
- Program 接口 : Program
- single_page_size : 4 kB
- single_erase_size : 4 kB
- TURBO MODE : Disable

```
J-Link: Flash download: Bank 0 @ 0x08000000: 1 range affected (2097152 bytes)
J-Link: Flash download: Total: 40.490s (Prepare: 0.183s, Compare: 2.230s, Erase: 28.524s, Program: 8.528s, Verify: 0.985s, Restore: 0.033s)
J-Link: Flash download: Program speed: 239 KB/s
O.K.
```

图 10. Test 4**• Test 5**

- 平台 : SEGGER Embedded Studio (SES)
- Erase 接口 : Erase
- Program 接口 : Program
- single_page_size : 8 kB
- single_erase_size : 4 kB
- TURBO MODE : Disable

```
J-Link: Flash download: Bank 0 @ 0x08000000: 1 range affected (2097152 bytes)
J-Link: Flash download: Total: 40.811s (Prepare: 0.182s, Compare: 2.224s, Erase: 28.699s, Program: 8.677s, Verify: 0.988s, Restore: 0.039s)
J-Link: Flash download: Program speed: 235 KB/s
O.K.
```

图 11. Test 5**• Test 6**

- 平台 : SEGGER Embedded Studio (SES)
- Erase 接口 : Erase

- Program 接口 : Program
- single_page_size : 16 kB
- single_erase_size : 4 kB
- TURBO MODE : Disable

```
J-Link: Flash download: Bank 0 @ 0x08000000: 1 range affected (2097152 bytes)
J-Link: Flash download: Total: 41.656s (Prepare: 0.190s, Compare: 2.320s, Erase: 28.969s, Program: 9.100s, Verify: 1.025s, Restore: 0.051s)
J-Link: Flash download: Program speed: 225 KB/s
O.K.
```

图 12. Test 6

• Test 7

- 平台 : SEGGER Embedded Studio (SES)
- Erase 接口 : Erase
- Program 接口 : SEGGER_OPEN_Program
- multi_page_size : 256 Byte
- single_erase_size : 4 kB
- TURBO MODE : Disable

```
J-Link: Flash download: Bank 0 @ 0x08000000: 1 range affected (2097152 bytes)
J-Link: Flash download: Total: 41.244s (Prepare: 0.234s, Compare: 2.339s, Erase: 28.981s, Program: 8.652s, Verify: 0.985s, Restore: 0.051s)
J-Link: Flash download: Program speed: 236 KB/s
O.K.
```

图 13. Test 7

• Test 8

- 平台 : SEGGER Embedded Studio (SES)
- Erase 接口 : Erase
- Program 接口 : SEGGER_OPEN_Program
- multi_page_size : 1 kB
- single_erase_size : 4 kB
- TURBO MODE : Disable

```
J-Link: Flash download: Bank 0 @ 0x08000000: 1 range affected (2097152 bytes)
J-Link: Flash download: Total: 41.350s (Prepare: 0.209s, Compare: 2.336s, Erase: 29.091s, Program: 8.666s, Verify: 0.993s, Restore: 0.052s)
J-Link: Flash download: Program speed: 235 KB/s
O.K.
```

图 14. Test 8

• Test 9

- 平台 : SEGGER Embedded Studio (SES)
- Erase 接口 : Erase
- Program 接口 : SEGGER_OPEN_Program
- multi_page_size : 256 Byte
- single_erase_size : 4 kB

— TURBO MODE : Enable

```
J-Link: Flash download: Bank 0 @ 0x08000000: 1 range affected (2097152 bytes)
J-Link: Flash download: Total: 35.097s (Prepare: 0.209s, Compare: 2.140s, Erase: 26.911s, Program: 4.833s, Verify: 0.945s, Restore: 0.051s)
J-Link: Flash download: Program speed: 423 KB/s
```

图 15. Test 9

• Test 10

- 平台 : SEGGER Embedded Studio (SES)
- Erase 接口 : Erase
- Program interface : SEGGER_OPEN_Program
- multi_page_size : 4 kB
- single_erase_size : 32 kB
- TURBO MODE : Disable

```
J-Link: Flash download: Bank 0 @ 0x08000000: 1 range affected (2097152 bytes)
J-Link: Flash download: Total: 18.340s (Prepare: 0.216s, Compare: 2.353s, Erase: 5.798s, Program: 8.911s, Verify: 1.004s, Restore: 0.051s)
J-Link: Flash download: Program speed: 229 KB/s
O.K.
```

图 16. Test 10

• Test 11

- 平台 : SEGGER Embedded Studio (SES)
- Erase 接口 : Erase
- Program 接口 : SEGGER_OPEN_Program
- multi_page_size : 4 kB
- single_erase_size : 64 kB
- TURBO MODE : Disable

```
J-Link: Flash download: Bank 0 @ 0x08000000: 1 range affected (2097152 bytes)
J-Link: Flash download: Total: 16.175s (Prepare: 0.214s, Compare: 2.193s, Erase: 3.947s, Program: 8.772s, Verify: 0.990s, Restore: 0.051s)
J-Link: Flash download: Program speed: 233 KB/s
O.K.
```

图 17. Test 11

• Test 12

- 平台 : SEGGER Embedded Studio (SES)
- Erase 接口 : Erase
- Program 接口 : SEGGER_OPEN_Program
- multi_page_size : 4 kB
- single_erase_size : 64 kB
- TURBO MODE : Enable

```
J-Link: Flash download: Bank 0 @ 0x08000000: 1 range affected (2097152 bytes)
J-Link: Flash download: Total: 11.218s (Prepare: 0.236s, Compare: 2.073s, Erase: 3.880s, Program: 4.021s, Verify: 0.953s, Restore: 0.052s)
J-Link: Flash download: Program speed: 509 KB/s
O.K.
```

图 18. Test 12

• Test 13

- 平台 : SEGGER Embedded Studio (SES)
- Erase 接口 : Erase
- Program 接口 : SEGGER_OPEN_Program
- multi_page_size : 16 kB
- single_erase_size : 64 kB
- TURBO MODE : Enable

```
J-Link: Flash download: Bank 0 @ 0x08000000: 1 range affected (2097152 bytes)
J-Link: Flash download: Total: 10.844s (Prepare: 0.229s, Compare: 2.067s, Erase: 3.865s, Program: 3.651s, Verify: 0.953s, Restore: 0.076s)
J-Link: Flash download: Program speed: 561 KB/s
O.K.
```

图 19. Test 13

表 8. 基于 RT500 EVK board 的 Flashloader 测试优化数据

Test	Generate tool	Image size (MB)	Single erase size (kB)	Open program interface	Program size/ Page mode	Turbo mode	Prepare (s)	Compare (s)	Erase (s)	Program (s)	Verify (s)	Restore (s)	Total (s)
1	Keil	2	4	Disable	256 B single	Disable	0.172	8.244	28.690	54.772	3.917	0.007	95.803
2	SES	2	4	Disable	256 B single	Disable	0.243	2.551	28.270	48.257	0.989	0.027	80.340
3	SES	2	4	Disable	1 kB single	Disable	0.284	2.257	28.413	16.089	0.989	0.028	48.063
4	SES	2	4	Disable	4 kB single	Disable	0.188	2.230	28.524	8.528	0.985	0.033	40.490
5	SES	2	4	Disable	8 kB single	Disable	0.182	2.224	28.699	8.677	0.988	0.039	40.811
6	SES	2	4	Disable	16 kB single	Disable	0.190	2.320	28.969	9.100	1.025	0.051	41.656
7	SES	2	4	Enable	256 B multiple	Disable	0.234	2.339	28.981	8.652	0.985	0.051	41.244

下页继续

表 8. 基于 RT500 EVK board 的 Flashloader 测试优化数据 (续上页)

Test	Generate tool	Image size (MB)	Single erase size (kB)	Open program interface	Program size/ Page mode	Turbo mode	Prepare (s)	Compare (s)	Erase (s)	Program (s)	Verify (s)	Restore (s)	Total (s)
8	SES	2	4	Enable	1 kB multiple	Disable	0.209	2.336	29.091	8.666	0.993	0.052	41.350
9	SES	2	4	Enable	256 B multiple	Enable	0.209	2.140	26.911	4.838	0.945	0.051	35.097
10	SES	2	32	Enable	4 kB multiple	Disable	0.216	2.358	5.798	8.911	1.004	0.051	18.340
11	SES	2	64	Enable	4 kB multiple	Disable	0.214	2.198	3.947	8.772	0.990	0.051	16.175
12	SES	2	64	Enable	4 kB multiple	Enable	0.236	2.073	3.880	4.021	0.953	0.052	11.218
13	SES	2	64	Enable	16 kB multiple	Enable	0.229	2.067	3.865	3.651	0.953	0.076	10.844

通过 表 8 的实验数据可以得到下面这几个结论：

1. SEGGER Embedded Studio 生成的 Flashloader 比 Keil 生成的 Flashloader 下载效率更高。
2. 使用 Program 接口，提高每次加载到 RAM 的 Page Size 对下载速度的提升是有限的。
3. 特别需要说明的是 Flash 的 Program 或 SEGGER_OPEN_Program 接口中写 Flash 是以一个 Page 为单位的，实验中改变了 single_page_size 或 multi_page_size 影响的是 JLink 往 MCU 内存中每一次加载缓存的大小，最终底层的 Program 操作一定是以一个 Page 为单位的。SEGGER_OPEN_Program 接口比 Program 接口提升的地方在于 JLink 往 MCU 内存中每次加载的缓存大小是原来每次加载缓存大小的数倍。
4. 开启 TURBO Mode 对 Flashloader 下载速度的提升是显著的，因此推荐用户开启这个功能。
5. 当使用 SEGGER_OPEN_Program 接口并开启 TURBO Mode 的情况下，multi_page_size 越大，program 的速度越快。一般推荐用户将 multi_page_size 设为 4 kB。
6. Erase 接口本文没有推荐使用 SEGGER_OPEN_Erase 接口，因为提升 Erase 速度的关键在于用 Block Erase 方案替代 Sector Erase 方案。不过这有局限：
 - 用户擦的 Flash 区域足够大（最好是数倍的 Block Size）
 - 下载的 Flash 起始地址是 block size 大小对齐的。

如果满足不了这两个条件也没关系，解决方案是 single_erase_size 大小还是采用 block size，但在底层 Erase 驱动中通过算法将被擦的区域拆分成多个 block 加多个 sector 的组合，然后分别调用 Block_Erase、Sector_Erase 底层驱动去操作。

4.2 基于 RT1060 的 Flashloader 优化

本小节以 RT1060 EVKB 板子和外部 NOR Flash 来测试不同策略下的 Flashloader 下载速度，RT1060 EVKB 板子上的默认 Flash 型号也是 IS25WP064。有了基于 RT500 的 Flashloader 优化中的结论，本节跳过基于 RT500 的 Flashloader 优化中的实验直接拿优化结论进行优化。将 Flexspi 的时钟设置为 30 M。擦的区域为 [0x6000 0000, 0x6080 0000]，大小为 8 MB。

具体测试如下：

- Test 1
 - 平台：Keil
 - Erase 接口：Erase
 - Program 接口：Program
 - single_page_size：256 Byte
 - single_erase_size：4 KB
 - TURBO MODE：Disable

```

J-Link Commander V7.54d
SEGGER J-Link Commander V7.54d (Compiled Sep 28 2021 16:20:01)
DLL version V7.54d, compiled Sep 28 2021 16:18:43

Connecting to J-Link via USB...FAILED: Cannot connect to J-Link.
J-Link>connect
J-Link connection not established yet but required for command.
Connecting to J-Link via USB...O.K.
Firmware: J-Link V10 compiled Nov  2 2021 12:14:50
Hardware version: V10.10
S/N: 600113866
License(s): RDI, FlashBP, FlashDL, JFlash, GDB
Vtref=3.315V
Please specify device / core. <Default>: MIMXRT106X
Type '?' for selection dialog
Device?
Please specify target interface:
  J) JTAG (Default)
  S) SWD
  D) cJTAG
Type?
Specify target interface speed [kHz]. <Default>: 4000 kHz
Speed>10000
Device "MIMXRT106X_S" selected.

Connecting to target via SWD
Found SW-DP with ID 0x0B011477
DPIDR: 0x0B011477
Scanning AP map to find all available APs
AP[1]: Stopped AP scan as end of AP map has been reached
AP[0]: AHB-AP (IDR: 0x04770041)
Iterating through AP map to find AHB-AP to use
AP[0]: Core found
AP[0]: AHB-AP ROM base: 0xE00FD000
CPUID register: 0x411FC271. Implementer code: 0x41 (ARM)
Found Cortex-M7 r1p1, Little endian.
FPUnit: 5 code (BP) slots and 0 literal slots
CoreSight components:
ROMTbl[0] @ E00FD000
[0][0]: E00FE000 CID B105100D PID 000B4C8 ROM Table
ROMTbl[1] @ E00FE000
[1][0]: E00FF000 CID B105100D PID 000B4C7 ROM Table
ROMTbl[2] @ E00FE000
[2][0]: E000E000 CID B105E00D PID 000B00C SCS-M7
[2][1]: E0001000 CID B105E00D PID 000B002 DWT
[2][2]: E0002000 CID B105E00D PID 000B00E FPE-M7
[2][3]: E0000000 CID B105E00D PID 000B001 ITM
[1][1]: E0041000 CID B105900D PID 001B975 ETM-M7
[1][2]: E0042000 CID B105900D PID 004B906 CTH
[0][1]: E0040000 CID B105900D PID 000B9A9 TPIU-M7
[0][2]: E0043000 CID B105F00D PID 001B101 TSG
Cache: Separate I- and D-cache.
I-Cache L1: 32 KB, 512 Sets, 32 Bytes/Line, 2-Way
D-Cache L1: 32 KB, 256 Sets, 32 Bytes/Line, 4-Way
Cortex-M7 identified.
J-Link>loadbin S8M.bin 0x60000000
Halting CPU for downloading file.
Downloading file [S8M.bin]...
J-Link: Flash download: Bank 0 @ 0x60000000: 1 range affected (8388608 bytes)
J-Link: Flash download: Total: 894.143s (Prepare: 0.022s, Compare: 32.361s, Erase: 126.374s, Program: 219.638s, Verify: 15.716s, Restore: 0.010s)
O.K.
J-Link>

```

图 20. Test 1

- Test 2
 - 平台：SEGGER Embedded Studio (SES)
 - Erase 接口：Erase
 - Program 接口：SEGGER_OPEN_Program
 - multi_page_size：4 KB

- single_erase_size : 64 KB
- TURBO MODE : Enable

```
J-Link: Flash download: Bank 0 @ 0x60000000: 1 range affected (8388608 bytes)
J-Link: Flash download: Total: 40.896s (Prepare: 0.239s, Compare: 2.326s, Erase: 16.373s, Program: 20.897s, Verify: 1.006s, Restore: 0.053s)
J-Link: Flash download: Program speed: 392 KB/s
O.K.
```

图 21. Test 2

表 9. 基于 RT1060 EVKB board 的 Flashloader 测试优化数据

Test	Generate tool	Image size (MB)	Single erase size (kB)	Open program interface	Program size/ Page mode	Turbo mode	Prepare (s)	Compare (s)	Erase (s)	Program (s)	Verify (s)	Restore (s)	Total (s)
1	Keil	2	4	Disable	256 B single	Disable	0.022	32.381	126.374	219.638	15.716	0.010	394.143
2	SES	2	64	Enable	4 KB multiple	Enable	0.239	2.326	16.373	20.897	1.006	0.053	40.896

当使用 [SEGGGER_OPEN_Program](#) 接口，开启 TURBO Mode，设置 multi_page_size 为 4 KB，Erase 采用 Block Erase 方案时，下载速度提升是非常显著的，具体数据如 [表 9](#) 所示。

4.3 Flashloader 下载速度优化结果

[表 10](#) 是 Flashloader 在 RT500 EVK 和 RT1060 EVKB 的下载速度优化结果，可以看到优化后的下载速度分别是原来没优化过的 8.5 倍和 9.6 倍，效果非常明显，实验结果证明优化策略是可行的。

表 10. 基于 RT500 和 RT1060 上的 Flashloader 下载速度优化结果

RT platform	Promotion	Test	Image size	Prepare	Compare	Erase	Program	Verify	Restore	Total
RT500	No optimization	1	2 MB	0.172 s	8.244 s	28.690 s	54.772 s	3.917 s	0.007 s	95.803 s
	Optimized	12	2 MB	0.236 s	2.073 s	3.880 s	4.021 s	0.953 s	0.052 s	11.218 s
	Promotion				398 %	739 %	1362 %	411 %		854 %
RT1060	No optimization	1	8 MB	0.022 s	32.381 s	126.374 s	219.638 s	15.716 s	0.010 s	394.143 s
	Optimized	2	8 MB	0.239 s	2.326 s	16.373 s	20.897 s	1.006 s	0.053 s	40.896 s
	Promotion				1392 %	772 %	1051 %	1562 %		964 %

Erase 和 Program 的提升已经介绍的很详细了，这里要说明的是 Compare 和 Verify。显然从 [表 10](#) 可以看到，Compare 和 Verify 在 RT1060 上提升的效率比 RT500 还要高很多，影响它们的主要因素是 MCU 读取 Flash 数据到内部 RAM 的性能差异，因为 RT1060 是 Cortex-M7 内核，而 RT500 是 Cortex-M33 内核，RT1060 有 32 kB、L1 级别的 I-cache 和 D-cache 以及主频更高（600M），这对 Flash 的读取效率会有很大的提升。

另外要说明的是，测试中使用的 Flashloader 算法能够自动识别并支持大部分 NOR Flash，同时也支持所有的 i.MX RT 平台，该算法用于 RT-UFL (RT-UFL 是一个适用全平台 i.MX RT 的通用 Flash 下载算法项目，涵盖绝大部分 Flash 型号) 中，已经得到了广泛验证。更多详情，请参考 <https://github.com/JayHeng/RT-UFL>。

5 参考资料

1. SEGGER Open Flashloader
2. MIMXRT500-EVK Schematic(Rev E1)
3. MIMXRT1060-EVKB Schematic(Rev B)
4. GD25LE64C Data Sheet
5. IS25WP064A Data Sheet
6. RT-UFL

6 修订记录

版本号	发布日期	说明
0	2022 年 2 月 18 日	初次发布

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Limited warranty and liability — Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

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.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. 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) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, ICODE, JCOP, LIFE, VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. 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, µVision, 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. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. M, M Mobileye and other Mobileye trademarks or logos appearing herein are trademarks of Mobileye Vision Technologies Ltd. in the United States, the EU and/or other jurisdictions.



© NXP B.V. 2022.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 2022 年 2 月 18 日

Document identifier: AN13567