标题：i.MX RT1060 中的增强功能

网址：https://www.nxp.com/docs/en/application-note/AN12240.pdf

目录

1、介绍 2

2.概述 2

3. 增强功能 4

3.1. 片内 RAM 4

3.2 平台 5

3.2.1. 闪存地址重新映射设置 5

3.2.2. 紧密耦合的 GPIO 访问 7

3.3. 外部内存 8

3.3.1 两个相同的 FlexSPI 接口 8

3.3.2 SDRAM 增强功能 8

3.3.3. SEMC 支持具有同步模式的 PNOR / PSRAM / NAND 9

3.4. 连接性 10

3.4.1. 两个 10M / 100M 以太网控制器 10

3.4.2. Flexible Data-rate Controller 局域网 12

3.4.3. Flexible I/O 15

3.5. ROM 17

3.5.1. 自动探测 17

3.5.2. 闪存重映射设置 18

4. Conclusion····· 23

5.修订历史·····23

1、介绍

i.MX RT1060 是业界首个跨界处理器系列的最新成员。i.MXRT1060 将片上 SRAM 提高了一倍，达到 1 MB，同时保持了与 i.MX RT1050 的引脚间兼容性。这个新系列引入了非常适合实时应用的其他特性，例如高速 GPIO，CAN-FD 和同步并行 NAND / NOR / PSRAM 控制器。 i.MX RT1060 在 Arm®Cortex®-M7 内核上以 600 MHz 运行。

本文档旨在介绍与 i.MX RT1050 相比的这些增强功能。

2.概述

i.MX RT1060 保持了与 i.MX RT1050 的引脚间兼容性，并添加了一些功能来提高它的性能。下面是 i.MX RT1060 的模块框图：

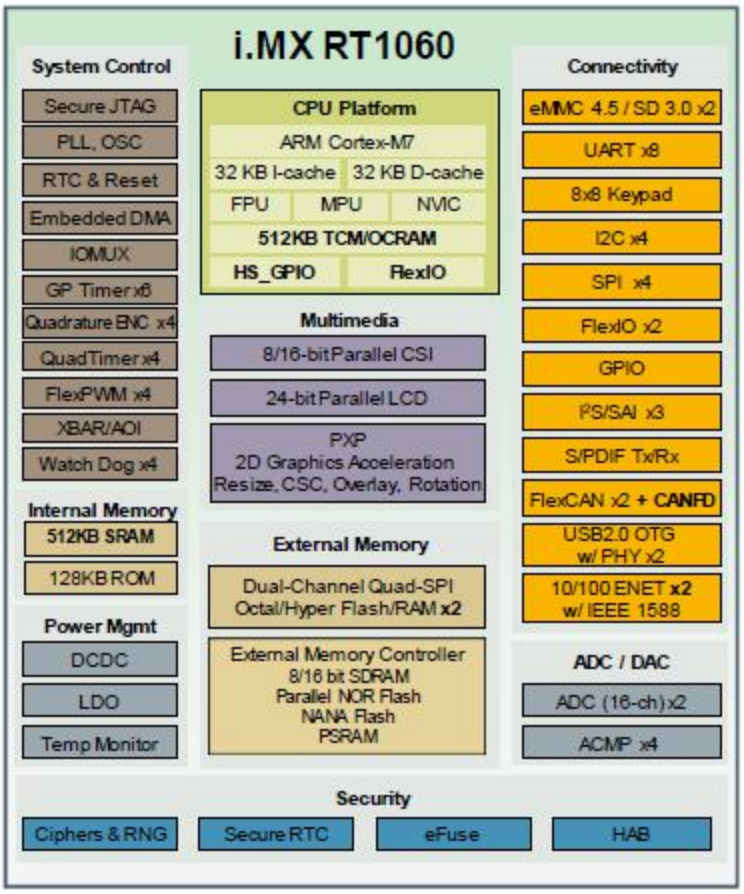

图 1. i.MX RT1060 模块框图

表 1 列出了 i.MX RT1060 与 i.MXRT1050 相比的增强功能：

表 1. I.MXRT1060 的增强功能

对象	I.MXRT1060 的增强功能
片上 RAM（1MB）	ITCM / DTCM 和 OCRAM 之间共享 512 KB 的 OCRAM
	专用 512 KB OCRAM
平台	支持闪存重映射地址设置
	紧密耦合的 GPIO，工作频率与 ARM 相同
外部存储	两个支持 XIP 的单/双通道 Quad SPI

	FLASH
	支持 8 列或 2 个存储区的小型 SDRAM
	为通过 SEMC 接口连接的并行 NAND 闪存/ 并行 NOR / PSRAM，增加了同步传输功能
互联性	两个 10M / 100M 以太网控制器，支持 IEEE1588
	增加一个 CANFD 模块
	三个 FlexIO 模块
ROM	支持自动探测
	支持闪存重映射设置

这些功能使 i.MXRT1060 与 i.MXRT1050 为通过 SEMC 接口连接的并行 NAND 闪存/ 并行 NOR / PSRAM，增加了同步传输功能。以下各节提供了新功能的详细信息以及它们是如何提高性能的。

3. 增强功能

3.1. 片内 RAM

i.MX RT1050 提供 512 KB 的 FlexRAM，可以灵活配置为 ITCM，DTCM 和

OCRAM。用户可以将应用程序代码放置到 ITCM，将数据放置到 DTCM 以获得高性能。通过 DMA 访问 OCRAM 可以获得更高的性能，因此 FlexRAM 提供了基于不同应用，灵活配置 RAM 类型的能力，从而帮助提高性能。

i.MX RT1060 不仅具有相同的 FlexRAM 功能，而且还添加了专用的 512 KB OCRAM，因此它具有 1 MB RAM 空间，供用户将关键代码/数据放置到 TCM 或 OCRAM。

片内 RAM 的存储映射如下：

表 2. 片内 RAM 的存储映射

存储器类型	起始地址	结束地址	分配
OCRAM	2020_0000	2027_FFFF	OCRAM
	2028_0000	202F_FFFF	FlexRAM (OCRAM)
DTCM	2000_0000	2007_FFFF	DTCM
ITCM	0000_0000	0007_FFFF	ITCM

3.2 平台

i.MX RT1060 还通过改进平台来获得高性能。

3.2.1. 闪存地址重新映射设置

i.MXRT1060 提供了一种重新映射 FlexSPI1 和 FlexSPI2 地址的功能，这意味着它可以使用相同的地址映射到不同的闪存物理地址，该地址接口为 FlexSPI1 和

FlexSP2。

为了实现这一点，它提供了三个寄存器。

• IOMUXC_GPR_GPR30

指定 flexspi1 和 flexspi2 的起始地址

• IOMUXC_GPR_GPR31

指定 flexspi1 和 flexspi2 的结束地址

• IOMUXC_GPR_GPR32

指定 flexspi1 和 flexspi2 的偏移地址 ,当 ADDR_START [31:12]≤Addr_i [31:12] <ADDR_END [31:12]时，重新映射地址 Addr_o = Addr_i [31:12] + {OFFSET [31:12]，12'h0}；否则 Addr_o = Addr_i，

Addr_i：原始访问地址

Addr_o：重新映射的地址

例如：

不设置任何 FlexSPI 重新映射寄存器，它会获取相应访问地址的 Flash 内容，不进行任何重新映射。

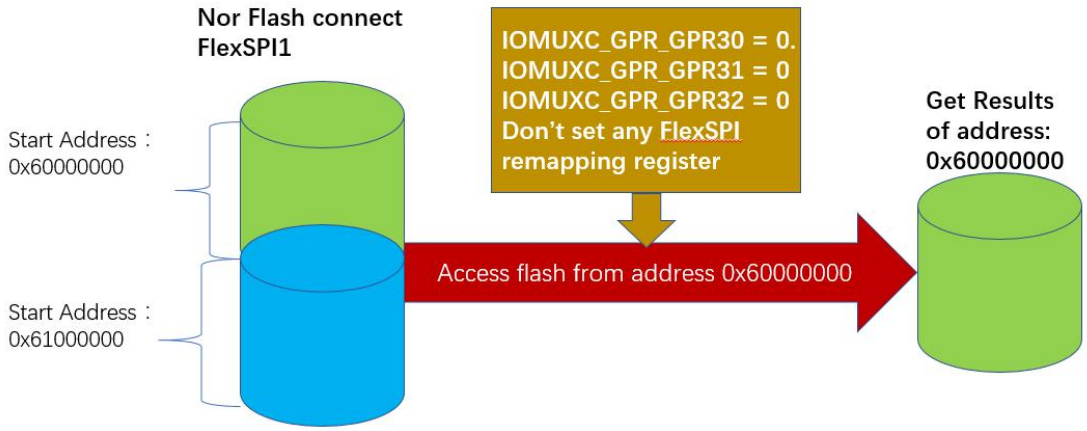

图 2. 无需任何重新映射设置即可访问 FlexSPI

设置重映射寄存器后，尝试访问相同的闪存地址的时候，将会获得重映射地址的

内容。

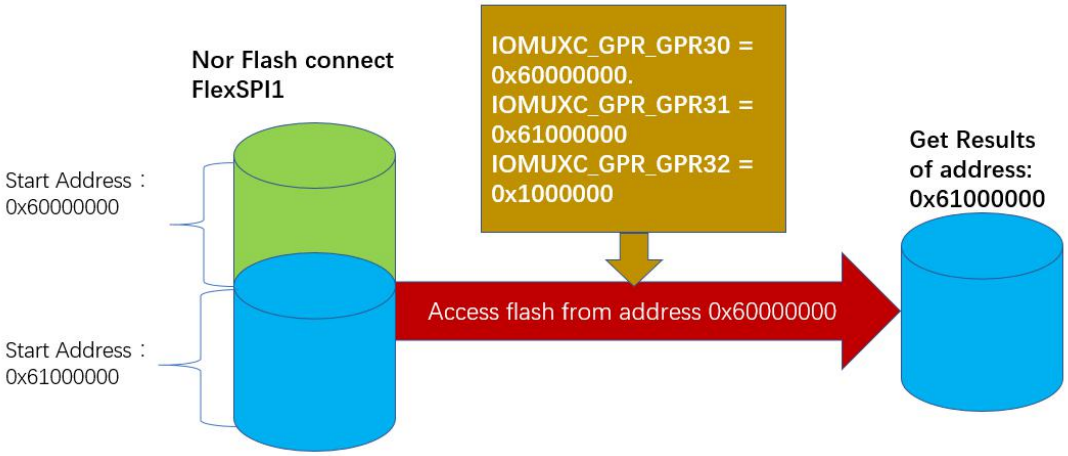

图 3. 通过重新映射的地址设置访问 FlexSPI

通过此功能，用户可以轻松切换位于不同闪存空间中的固件，并且多个固件可以共享同一链接文件，这有利于 OTA 的应用。它可用于接收固件并保存到不同的闪存地址，在接收到的固件被验证后，可以通过闪存重映射功能轻松地切换到新固件来运行，而且易于升级并在固件升级过程中保持高可靠性，用户无需使用不同的链接文件来构建固件，这样可以降低操作的复杂性并避免问题的发生。

i.MXRT1060 的嵌入式 ROM 还支持闪存重映射，该闪存重映射允许将多个固件设置为闪存，并通过提供的 API 来切换固件，有关详细信息，请参阅以下章节“ 3.5.2”。

3.2.2. 紧密耦合的 GPIO 访问

i.MX RT 提供紧密耦合的 GPIO，可高频地进行访问。

它提供了两组 GPIO 寄存器来控制引脚输入输出。GPIO1 至 GPIO4 是普通 GPIO，

而 GPIO6 至 GPIO9 是紧密 GPIO，但是它们共享相同的引脚，这意味着 gpio 引脚可以从 GPIO1/2/3/4 和 GPIO6/7/8/9 中进行选择。

寄存器 IOMUXC_GPR_GPR26 ， IOMUXC_GPR_GPR27 ， IOMUXC_GPR_GPR28 和 IOMUXC_GPR_GPR29 用于 GPIO 选择。

以下是 IOMUXC_GPR_GPR26 寄存器的说明：

表 3. IOMUXC_GPR_GPR26 说明

Field	Description
GPIO_MUX1_ GPIO_SEL	GPIO1 and GPIO6 share same IO MUX function, GPIO_MUX1 selects one GPIO function. This register controls GPIO_MUX1 to select GPIO1 or GPIO6. For bit *n*, • 0: GPIO1[*n*] is selected; • 1: GPIO6[*n*] is selected.

使用该寄存器，用户可以选择一般的 GPIO（慢速）或快速 GPIO 连接到相应的引脚上，当选择 GPIO6/7/8/9 时，引脚翻转的输出频率可以高达 150Mhz。

3.3. 外部存储器

3.3.1 两个相同的 FlexSPI 接口

i.MX RT1060 添加了一个 FlexSPI 接口，提供了与多个闪存或 SRAM 器件连接的能力。一个典型的用例是使用一个 flexspi 接口连接串行 NOR 闪存来放置代码，而另一个接口则连接 hyper RAM。与 SDRAM 相比它节省了引脚，仅需要大约 11 个引脚。

3.3.2 SDRAM 增强功能

由于 i.MXRT1050 将列地址宽度限制为 9 位或更多，因此无法支持较小的地址

宽度，但是对于某些小尺寸的 SDRAM，要求列地址宽度为 8 位，因此它无法支持这些小尺寸的 SDRAM。

为此，i.MXRT1060 在寄存器 SEMC_SDRAMCR0 中添加两个位字段，详细说明如下：

针对选择 2 个存储区和 4 个存储区，添加位字段 BANK2。

0-SDRAM 设备具有 4 个存储区。

1-SDRAM 设备有 2 个存储区。

列地址定义有两个寄存器，一个是位域 COL，与 i.MX RT1050 相同。

00b - 12 bit

01b - 11 bit

10b - 10 bit

11b - 9 bit

对于 i.MX RT1060，另一个寄存器字段（CLO8）是新的。

0b-列地址位数由 COL 字段决定。

1b-列地址的位数为 8。COL 字段将被忽略。

3.3.3. SEMC 支持具有同步模式的 PNOR / PSRAM / NAND

i.MX RT1060 对 SEMC IP 进行了改进以支持同步模式，可以通过同步模式访问 NAND 闪存，Nor 闪存和 SRAM，从而进一步提高性能。

3.4. 互联

i.MX RT1060 在连接性方面的增强功能包括以太网接口，可变数据率的控制器局域网(CAN FD)和 Flexible I/O。

3.4.1. 两个 10M / 100M 以太网控制器

有两个 10M / 100M 以太网控制器以支持两个以太网接口，这有利于互联性的应用。

以及新加入的以太网接口的管脚复用如下：

表 4. 以太网 2 的 Muxing 选项

实例	端口	引脚	模式
ENET2	ENET2_MDC	GPIO_EMC_38	ALT8
		GPIO_B0_00	ALT8
	ENET2_MDIO	GPIO_EMC_39	ALT8
		GPIO_B0_01	ALT8
	ENET2_TDATA0	GPIO_EMC_30	ALT8
		GPIO_B0_12	ALT8
		GPIO_B1_14	ALT8
	ENET2_TDATA1	GPIO_EMC_31	ALT8
		GPIO_B0_13	ALT8
		GPIO_B1_15	ALT8

	ENET2_TDATA2	GPIO_B0_05	ALT8
	ENET2_TDATA3	GPIO_B0_04	ALT8
	ENET2_TX_CLK	GPIO_EMC_33	ALT8
		GPIO_B0_15	ALT8
		GPIO_SD_B0_01	ALT8
	ENET2_TX_EN	GPIO_EMC_32	ALT8
		GPIO_B0_14	ALT8
		GPIO_SD_B0_00	ALT8
	ENET2_TX_ER	GPIO_B0_07	ALT8
	ENET2_RDATA0	GPIO_EMC_35	ALT8
		GPIO_B1_01	ALT8
		GPIO_SD_B0_03	ALT8
	ENET2_RDATA1	GPIO_EMC_36	ALT8
		GPIO_B1_02	ALT8
		GPIO_SD_B0_04	ALT8
	ENET2_RDATA2	GPIO_B0_09	ALT8
	ENET2_RDATA3	GPIO_B0_08	ALT8
	ENET2_RX_CLK	GPIO_B0_06	ALT8
	ENET2_RX_EN	GPIO_EMC_37	ALT8
		GPIO_B1_03	ALT8
		GPIO_SD_B0_05	ALT8
	ENET2_RX_ER	GPIO_EMC_34	ALT8

		GPIO_B1_00	ALT8
		GPIO_SD_B0_02	ALT8
	ENET2_REF_CLK2	GPIO_EMC_33	ALT8
		GPIO_B0_15	ALT8
		GPIO_SD_B0_01	ALT8
	ENET2_1588_EVENT0_IN	GPIO_AD_B1_01	ALT8
		GPIO_B0_03	ALT8
	ENET2_COL	GPIO_B0_11	ALT8
	ENET2_CRS	GPIO_B0_10	ALT8
	ENET2_1588_EVENT1_IN	GPIO_AD_B1_11	ALT8
	ENET2_1588_EVENT2_IN	GPIO_AD_B1_13	ALT8
	ENET2_1588_EVENT3_IN	GPIO_AD_B1_15	ALT8
	ENET2_1588_EVENT0_OUT	GPIO_AD_B1_00	ALT8
		GPIO_B0_02	ALT8
	ENET2_1588_EVENT1_OUT	GPIO_AD_B1_10	ALT8
	ENET2_1588_EVENT2_OUT	GPIO_AD_B1_12	ALT8
	ENET2_1588_EVENT3_OUT	GPIO_AD_B1_14	ALT8

3.4.2. 可变数据率的控制器局域网(CAN FD)

i.MXRT1060 添加了一个新模块 “可变数据率的控制器局域网（CANFD / FlexCAN）”，该模块用于需要高速和可靠性的工业控制总线的应用，CANFD

是 CAN 协议规范的完整实现，同时支持标准和扩展消息帧以及传输速率高达 8 Mbps 且长度达到 64 字节的有效载荷。消息缓冲区存储在专用于 CANFD / FlexCAN 模块的嵌入式 RAM 中。框图如下：

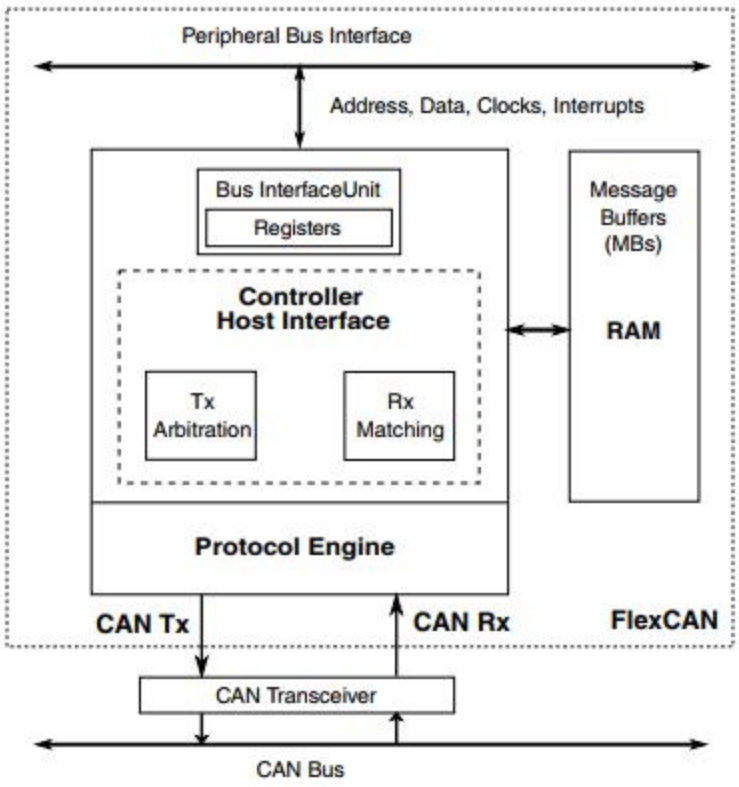

图 4. CANFD / FlexCAN 框图

CANFD / FlexCAN 模块包括以下的独特功能：

•完全实现具有可变数据率的控制器局域网（CAN FD）协议规范和 CAN 协议规范，版本 2.0 B

•标准数据帧

•扩展数据帧

•0~64 字节的数据长度

•可编程位速率（有关特定的最大速率配置，请参见芯片特定的 FlexCAN 信息）

•与内容相关的寻址

•符合 ISO 11898-1 标准

•灵活的邮箱可配置为存储 0 到 8、16、32 或 64 字节的数据长度

•每个 mailbox 可配置为接收或发送，都支持标准消息和扩展消息

•每个 mailbox 有单独的 Rx 屏蔽寄存器

•全功能的旧版 Rx FIFO 存储最多可支持六个帧，并具有 DMA 支持的自动内部指针处理功能；增强型 Rx FIFO，可存储多达 32 个 CAN FD 帧，并具有 DMA 支持的自动内部指针处理功能。

•传输中止功能

•灵活的消息缓冲区（MB），总共 64 个消息缓冲区，每个消息缓冲区的数据长度为 8 个字节，可配置为 Rx 或 Tx

• CAN 协议接口的时钟源可编程，来自外设时钟或振荡器时钟

•接收或发送未使用的 RAM 可用作通用 RAM 空间

•支持仅侦听模式

•可编程的回环模式支持自测操作

•可编程传输优先级方案：最低 ID，最低缓冲区号或最高优先级

•基于 32 位自由运行计时器的时间戳，带有可选的外部时钟节拍

•通过特定消息同步的全局网络时间

•可屏蔽的中断

•与传输介质的独立性（假定使用外部收发器）

•高优先级消息的仲裁方案使等待时间更短

•根据总线活动可编程唤醒的低功耗模式

•以更快的数据速率传输 CAN FD 消息时具有收发器延迟补偿功能

•远程请求帧可以自动或通过软件来处理

•CAN 位时间设置和配置位只能在 Freeze 模式时写入

•Tx 邮箱状态（最低优先级缓冲区或空缓冲区）

•针对接收到的帧的标识符接受过滤器命中指示器（IDHIT）寄存器

•错误和状态 1 寄存器中的 SYNCH 位，用于通模块与 CAN 总线同步

•传输消息的 CRC 状态

•旧版 Rx FIFO 全局屏蔽寄存器

•匹配过程中，可在邮箱和 Rx FIFO 之间选择优先级

•强大的旧版 Rx FIFO ID 过滤功能，能够匹配传入的 ID 和扩展的 128 位，或标准 256 位，或 512 位的部分（8 位）ID，最多具有 32 个单独的屏蔽功能

•强大的增强型 Rx FIFO ID 过滤功能，能够通过三种过滤方案：屏蔽+过滤器、范围、两个不带屏蔽的过滤器，将传入的 ID 与扩展的 64 位或标准 128 位 ID 过滤元素进行匹配。

•100%向后兼容以前的 FlexCAN 版本

3.4.3. Flexible I/O

i.MX RT1060 最多支持三个 Flexible I/O（FlexIO），而 i.MX RT1050 仅支持两个，FlexIO3 可以支持与核心时钟相同的快速时钟源 ahb_clock_root，而另外两个 FlexIO 由 ipg_clk_root 提供时钟，频率限制为 120MHz。

有关 FlexIO 模块，请参见下面的时钟分配。

表 5. FlexIO 时钟选项

模块	模块时钟	时钟源	模块时钟门控使能
FLEXIOn	flexio1_ipg_clk	ipg_clk_root	CCGR5[CG1] (flexio1_clk_enable)
	flexio1_ipg_clk_s	ipg_clk_root	CCGR5[CG1] (flexio1_clk_enable)
	flexio1_flexio_clk	flexio1_clk_root	CCGR5[CG1] (flexio1_clk_enable)
	flexio2_ipg_clk	ipg_clk_root	CCGR3[CG0] (flexio2_clk_enable)
	flexio2_ipg_clk_s	ipg_clk_root	CCGR3[CG0] (flexio2_clk_enable)
	flexio2_flexio_clk	flexio2_clk_root	CCGR3[CG0] (flexio2_clk_enable)
	flexio3_ipg_clk	ahb_clk_root	CCGR7[CG6] (flexio3_clk_enable)
	flexio3_ipg_clk_s	ipg_clk_root	CCGR7[CG6] (flexio3_clk_enable)
	flexio3_flexio_clk	flexio2_clk_root	CCGR7[CG6] (flexio3_clk_enable)

3.5. ROM

i.MX RT1060 将 ROM 的大小增加到 128 KB，而 RT1050 仅支持 96 KB。此外，它在 RT1060 上实现了自动探测和闪存重新映射功能，请参见下面的详细介绍。

3.5.1. 自动探测

通常，为了使外部闪存和无闪存处理器正常工作，需要通过读取用户指定的闪存配置块来获取闪存配置信息。例如，RT1050 要求客户输入闪存配置位字段（32 位）以生成闪存配置块，而 ROM 可以通过这些配置块信息初始化闪存，因此需要将闪存配置信息填充到 bd 文件中，例如：

```
# The section block specifies the sequence of boot commands to be written to the SB file
section (0) {

    #1. Prepare Flash option
    # 0xc0233007 is the tag for Serial NOR parameter selection
    # bit [31:28] Tag fixed to 0x0C
    # bit [27:24] Option size fixed to 0
    # bit [23:20] Flash type option
    #             0 - QuadSPI SDR NOR
    #             1 - QUadSPI DDR NOR
    #             2 - HyperFLASH 1V8
    #             3 - HyperFLASH 3V
    #             4 - Macronix Octal DDR
    #             6 - Micron Octal DDR
    #             8 - Adesto EcoXIP DDR
    # bit [19:16] Query pads (Pads used for query Flash Parameters)
    #             0 - 1
    #             2 - 4
    #             3 - 8
    # bit [15:12] CMD pads (Pads used for command)
    #             0 - 1
    #             2 - 4
    #             3 - 8
    # bit [11: 08] fixed to 0
    # bit [07: 04] fixed to 0
    # bit [03: 00] Flash Frequency, device specific
    #
    # In this example, the 0xc0233007 repres
    #     HyperFLASH 1V8, Query pads: 8 pads, Cmd pads: 8 pads, Frequency: 133MHz
    load 0xc0233007 > 0x2000;
    # Configure HyperFLASH using option a address 0x2000
    enable flexspinor 0x2000;
```

图 5. bd 文件中的 Hyperflash 配置

RT1060 ROM 具有自动探测功能，可以自动探测闪存类型并获取 bd 文件中没有上述配置信息的参数，并且一旦启用了自动探测功能，无需用户输入即可配置闪存。

可以通过以下两种方式启用自动探测功能。

•烧写熔断位

烧写熔断位“ FLASH_AUTO_PROBE_EN”以启用 eFUSE，还需要编程 eFUSE 位“ BT_FUSE_SEL”以启用熔断位来配置启动模式。

它还通过 BOOT_CFG1 [3：2]确定闪存类型。

➢00-四线 NOR 闪存

➢01-Macronix 四线 NOR 闪存

➢10-Micron 八进制闪存

➢11-Adesto 八进制闪存

•通过设置配置引脚的电平启用它

当 eFUSE 位“ BT_FUSE_SEL”为 0 时，将引脚“ GPIO_B0_04”保持为高电平以启用自动探测功能。

同样，它根据 GPIO“ GPIO_B0_6”和“ GPIO_B0_7”的状态确定闪存类型。

3.5.2. 闪存重映射设置

i.MXRT1060 的 ROM 支持闪存重新映射功能，并允许用户下载两个固件到闪存，还提供 API 供用户轻松切换固件。要启用闪存重新映射功能，需要烧写以下熔断位。

表 6. FlexSPI1 的熔断位定义

模块	地址	7	6	5	4	3	2	1	0
FlexSPI 1 - Serial NOR	0x6E0[7:0]	FLEXSPI_RESET_PIN_EN 0 - Disabled 1 - Enabled	JEDEC_HW_RESET_EN 0 - Disabled 1 - Enabled	xSPI FLASH HOLD TIME 0 - 500us / 1 - 1ms 2 - 3ms / 3 - 10ms		xSPI FLASH BOOT FREQUENCY 0 - 100MHz / 1 - 120MHz / 2 - 133MHz / 3 -166MHz / 4 - Reserved 5 -			SIP_TEST_EN

					80MHz / 6 - 60MHz	
	xSPI FLASH IMAGE SIZE 0-FLEXSPI_NOR_SEC_IMAGE_OFFSET*256KB 1-12: 1MB-12MB 13 - 256KB, 14-512KB, 15-768KB				xSPI FLASH DUMMY CYCLE 0 - Auto probe Others - Dummy cycles (for example, 8 - 8 cycles)	
	FLEXSPI_NOR_SEC_IMAGE_OFFSET[7:0] Actual offset = 256KB * fuse value					

0x6E0 [23:16]指定闪存重映射的偏移值，如果不为 0，则启用闪存重映射。例如，如果代码大小约为 512 KB，则将保险丝设置如下：

•0x6E0 [23:16]设置为 2，0x6E0 [15:12]设置为 0。

用户可以通过调用 API 函数在其应用程序代码中切换固件，ROM 开放了 API 函数以降低使用难度。

以下是引导加载程序 API 入口结构体，以供参考：

typedef struct

{

```
  const uint32_t version; //!< Bootloader version
number
  const char *copyright; //!< Bootloader Copyright
  void (*runBootloader)(void *arg); //!< Function to start the
bootloader executing
  const uint32_t *reserved0; //!< Reserved
  const flexspi_nor_driver_interface_t *flexSpiNorDriver; //!< FlexSPI NOR
Flash API
  const uint32_t *reserved1; //!< Reserved
  const clock_driver_interface_t *clockDriver;
  const rtwdog_driver_interface_t *rtwdogDriver;
  const wdog_driver_interface_t *wdogDriver;
  const uint32_t *reserved2;
} bootloader_api_entry_t;
```

用户可以通过 API 入口地址 0x0020001c 调用这些 API 函数。

下面是一个示例：

```
g_bootloaderTree = (bootloader_api_entry_t *)*(uint32_t *)0x0020001c;
```

以及 bootloader 参数如下：

```
typedef union
{
  struct
  {
```

```
  uint32_t imageIndex : 4;
  uint32_t reserved : 12;
  uint32_t serialBootInterface : 4;
  uint32_t bootMode : 4;
  uint32_t tag : 8;
  } B;
  uint32_t U;
} run_bootloader_ctx_t;
```

“ imageIndex”定义要重新映射运行的代码。

下面是一个示例：

```
run_bootloader_ctx_t boot_para;
  boot_para.B.imageIndex = 1; // specified firmware index
to 1
  boot_para.B.serialBootInterface                                    =
kEnterBootloader_SerialInterface_USB;
  boot_para.B.bootMode = kEnterBootloader_Mode_Default;
  boot_para.B.tag = kEnterBootloader_Tag;
  g_bootloaderTree->runBootloader( (void *)&boot_para ); // run the
index 1 firmware
```

用户可以轻松更改固件以按指定的固件索引运行。

4. Conclusion

本文档介绍了 RT1060 的增强功能，并讨论了与 RT1050 的区别。它还提供了有关如何使用这些新功能的指南。本文档的目的是帮助客户学习 i.MX RT1060 并更好地使用它。

5.修订历史

表 7. 修订历史

修订号	日期	内容变化
0	09/2018	初版