

## 1 引言

本应用笔记介绍了如何使用 FlexIO 模块来模拟 6800 并行总线。

6800 总线通常又称 Motorola 总线，是异步智能 LCD 控制器通常使用的并行总线接口。

FlexIO 是 i.MX RT 系列板载的一种外设。它是一个非常灵活并且可以随意配置的模块，不仅可以模拟出 UART, I2C, SPI 等常用通信接口，用户还可以使用它来模拟 6800 总线。

i.MX RT1010 处理器基于 Arm<sup>®</sup> Cortex<sup>®</sup>-M7 平台。它具有较高的 CPU 性能和最佳的实时响应，并具有丰富的外围设备。本应用笔记基于 i.MXRT1010-EVB 开发相关应用以验证通过 FlexIO 模拟的 6800 总线。

## 2 FlexIO 概述

### 2.1 特性

i.MX RT1010 的 FlexIO 模块提供以下主要特性：

- 具有发送，接收和数据匹配功能的 32 位移位寄存器组。
- 双缓冲移位器操作，可进行连续数据传输。
- 移位器串联以支持较大的数据传输。
- 自动开始/停止位生成。
- 支持并行接口的 1, 2, 4, 8, 16 或 32 位移位宽度。
- 中断，DMA 或轮询的发送/接收操作。
- 与总线时钟频率无关的可编程波特率，支持停止模式下的异步操作。
- 高度灵活的 16 位定时器，支持各种内部或外部触发，复位，启用和禁用条件。
- 可编程逻辑模式，用于在片上集成外部数字逻辑功能或组合引脚/移位器/定时器功能以生成复杂的输出。
- 可编程状态机，用于从 CPU 分流基本的系统控制功能，每个状态最多支持 8 种状态，8 个输出和 3 个可选输入。

图 1 给出了 FlexIO 模块定时器和移位器的配置。

### 目录

1 引言.....	1
2 FlexIO 概述.....	1
2.1 特性.....	1
2.2 并行传输.....	2
3 6800 总线时序.....	3
4 6800 总线模拟.....	4
4.1 开发平台.....	4
4.2 6800 总线写配置.....	5
4.3 6800 总线读配置.....	7
4.4 示例运行.....	9
5 参考资料.....	11



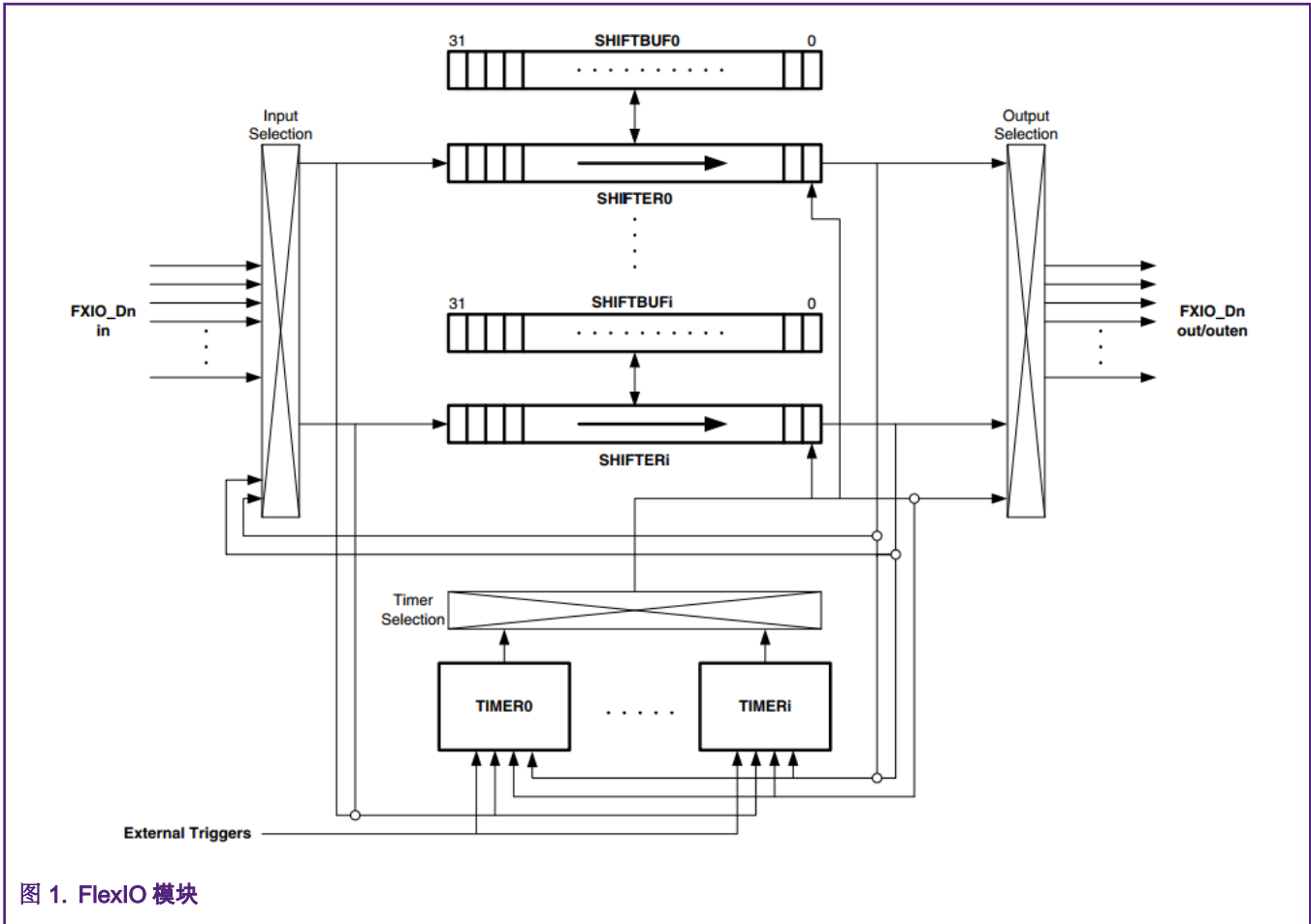


图 1. FlexIO 模块

i.MXRT 1010 上的 FlexIO 模块有 8 个移位器和 8 个定时器以及 27 个引脚。移位器负责缓冲数据并将其移入或移出 FlexIO。16 位定时器用于控制移位寄存器的加载，移位和存储。每个定时器和移位器的引脚配置都可以配置为使用任何极性的 FlexIO 引脚。

## 2.2 并行传输

i.MX RT1010 的 FlexIO 模块支持串行和并行传输模式。可以使用 SHIFTCFG [PWIDTH] 字段将移位器配置为使用多个 FlexIO 引脚进行并行传输。PWIDTH 是总线宽度，用于配置移位器的以下设置：

- 每个移位时钟移位的位数。
- 每个移位时钟由移位器驱动的引脚数。
- 每个移位时钟由移位器采样的引脚数。

当配置成并行传输时，每个移位时钟可以移位 4, 8, 16 或者 32 位。为了支持较大的数据传输，可以将多个移位器串联在一起，并使用 DMA 方法访问移位器缓冲寄存器以进行高速传输和低功耗操作。

对于并行发送，8 个移位器中只有 SHIFTER0 和 SHIFTER4 支持输出到 FlexIO 引脚。但是除 SHIFTER0 以外的所有移位器均支持输出到与其串联的低位移位器。

类似的，对于并行接收，8 个移位器中只有 SHIFTER3 和 SHIFTER7 支持 FlexIO 引脚的输入。但是除 SHIFTER7 以外的所有移位器均支持从与其串联的高位移位寄存器进行输入。

任何 FlexIO 引脚都可以用作并行输出/输入引脚。但是，对于 8 位的并行总线，必须使用连续的 FlexIO 引脚，例如使用引脚 0~7 或者引脚 1~8 等。

### 3 6800 总线时序

6800 总线又称 Motorola 总线。一般来说，6800 总线包含 4 根控制信号线：片选信号(CS，读/写选择信号(R/W，读/写使能信号(EN，数据/命令选择信号(RS，以及 8 位或 16 位的双向数据总线。

CS 和 RS 均为低有效信号。CS 信号为低时选中从设备。EN 信号线的上升沿或者是下降沿是数据读/写锁存信号（时钟）。RS 是数据/命令选择信号，低电平表示命令（或地址）传输，高电平表示数据传输。R/W 是读/写选择信号，低电平表示写操作，高电平表示读操作。

在数据读/写传输开始前，需用命令/地址写时序来指定目标地址。数据传输可以是一个或多个 beats。

图 2 给出了 6800 总线的写时序图。

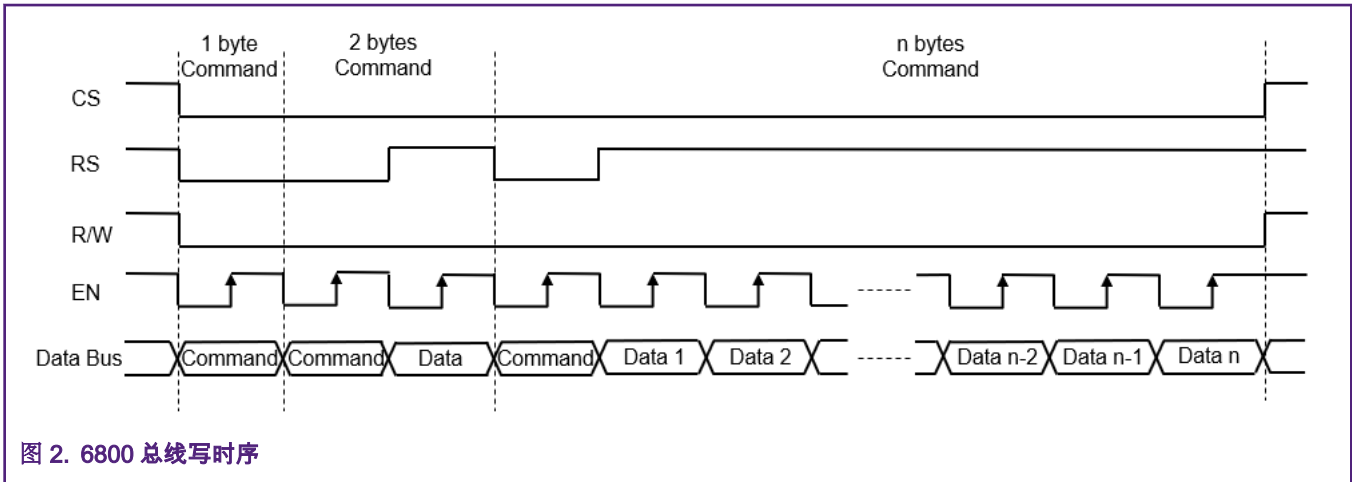
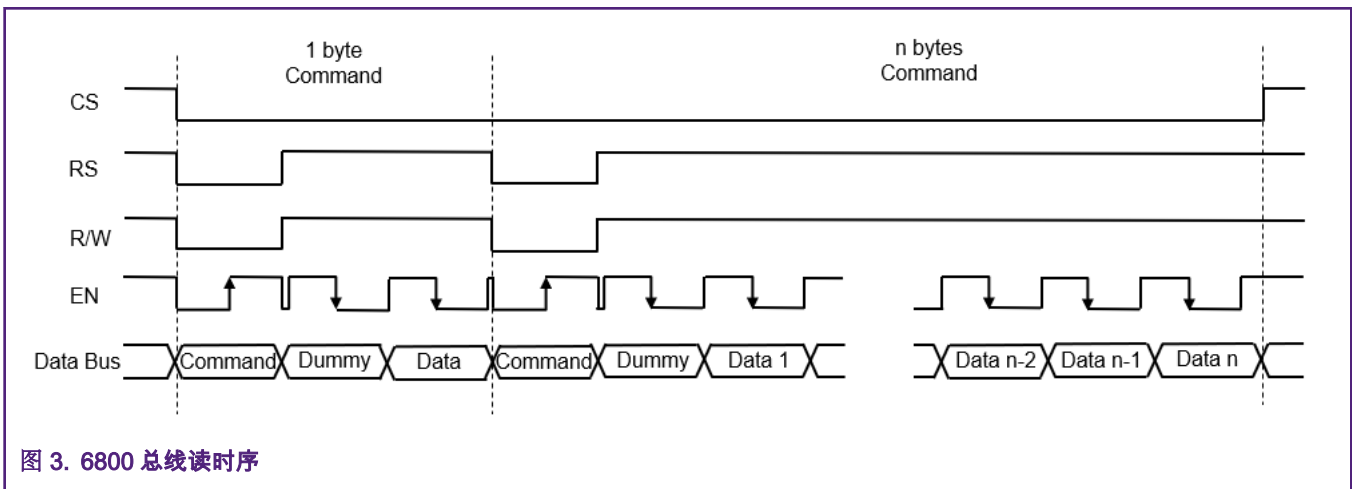


图 3 给出了 6800 总线的读时序图。根据总线从设备的不同，在命令写入 beat 和第一个数据读取 beat 之间可能会出现虚拟读取 beat。



通常来说，对 6800 总线从设备的任何操作都将从一个命令写周期开始，然后是一个或多个数据读/写周期。具体操作流程如下：

1. 将 FlexIO 配置成 single-beat 写模式。
2. 配置 GPIO 以拉低 CS, RS 以及 R/W 信号。
3. 将命令数据写入移位器缓冲区。
4. 配置 GPIO 以拉高 RS 信号(读取数据时同时需要拉高 R/W 信号)。
5. 将 FlexIO 配置成所需的读/写模式（如 single 或 32-beats 模式）。
6. 使用移位器状态标志来触发由中断或 DMA 驱动的数据传输（写入或读出移位器缓冲区）。

7. 配置 GPIO 以拉高 CS 信号。

## 4 6800 总线模拟

本章节主要介绍了如何使用 FlexIO 模块来模拟如 图 2 和 图 3 所示的 6800 并行总线读写时序。

### 4.1 开发平台

为了模拟 6800 并行总线，本应用以 i.MXRT1010-EVB 板为例。该板上有一个连接到 FlexIO 部分引脚的接头 J46，这使该应用程序的硬件连接更为方便。图 4 即为 i.MXRT1010-EVB 开发平台。

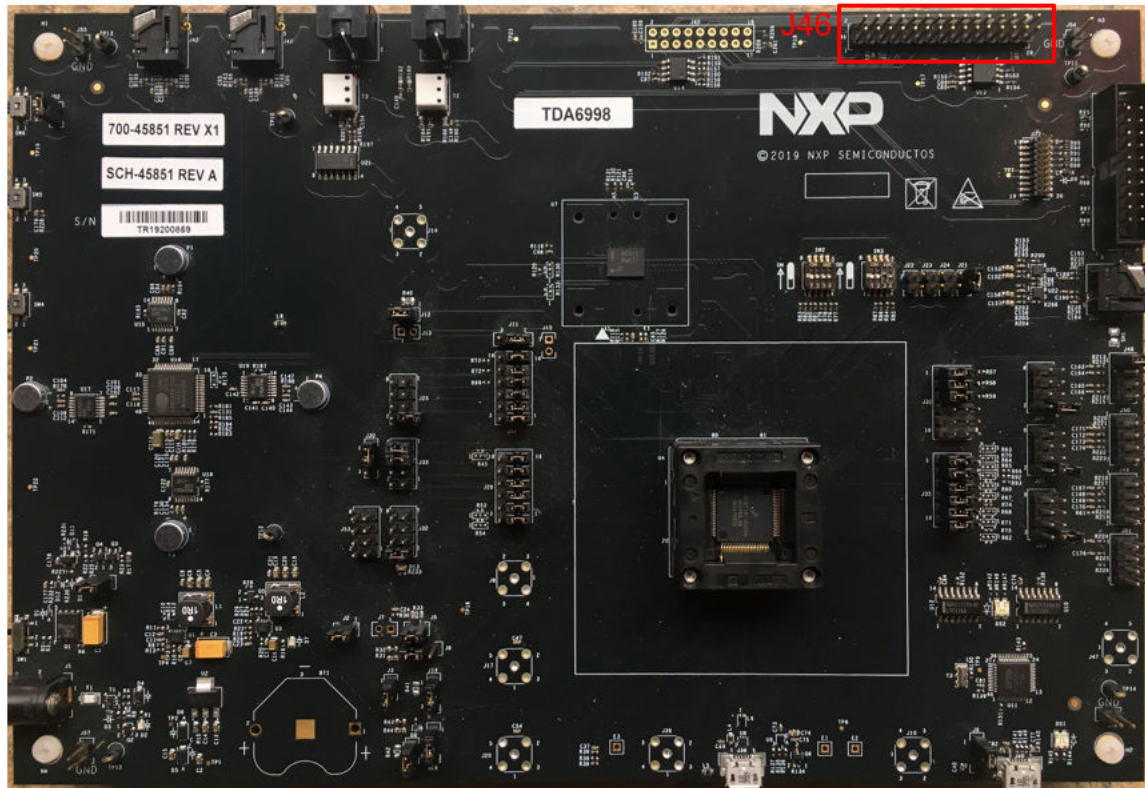


图 4. i.MXRT1010-EVB 开发板

在 i.MX RT1010 上，FlexIO 模块总共有 27 个引脚。本应用中使用了 9 个 FlexIO 引脚来模拟 EN 以及 D0~D7 信号。此外，使用三个额外的 GPIO 引脚来生成 RS，CS 和 R/W 信。

表 1 给出了在 i.MXRT1010-EVB 板上模拟 6800 总线的详细引脚分配。

表 1. 引脚分配

引脚	引脚位置	6800 总线信号
FlexIO1_IO3	J46-6	D0
FlexIO1_IO4	J46-7	D1
FlexIO1_IO5	J46-8	D2
FlexIO1_IO6	J46-9	D3
FlexIO1_IO7	J46-10	D4

Table continues on the next page...

表 1. 引脚分配 (continued)

引脚	引脚位置	6800 总线信号
FlexIO1_IO8	J46-11	D5
FlexIO1_IO9	J46-12	D6
FlexIO1_IO10	J46-13	D7
FlexIO1_IO1	J46-3	EN
GPIO1_10	J46-4	R/W
GPIO1_8	J46-2	RS
GPIO1_7	J46-1	CS

## 4.2 6800 总线写配置

6800 写操作可以通过将移位器配置成发送模式来实现。6800 并行总线写功能包含 single-beat 写和 multi-beats 写两种模式。6800 总线位宽是 8 位或者 16 位，由于 8 位和 16 位并行总线的模块配置很相似，因此以下各节仅描述 8 位并行总线的具体实现。

Single-beat 写模式用于小数据发送，例如配置 LCD 驱动寄存器，少量帧数据以及命令数据等。该模式下只需要用到一个移位器来发送数据，发送数据时，8 位数据将同时移出到分配的 8 个 FlexIO 引脚上。一个 Single-beat 发送时序仅需要定时器产生一个移位时钟。在这种情况下，使用轮询方法来驱动从移位器缓冲区到移位器的数据传输。

Multi-beats 写模式用于大数据传输，例如传送帧数据到 LCD 模块。将移位器串联起来传输数据，其中，一个发送时序的 beat 总数与串联的移位器数量和总线位宽有关。对于 8 位并行总线，一个 32 位移位器至少支持 4-beats 的传输。本应用中使用了 8 个移位器进行串联，因此对 8 位总线可以支持 32-beats 的传输。一个 Multi-beats 发送时序需要定时器产生多个移位时钟。在这种情况下，使用 DMA 方式来驱动从移位器缓冲区到移位器的数据传输。

本应用中使用了一个移位器和一个定时器来模拟 single-beat 写模式。定时器 0 用于产生移位时钟 和 EN 信号。移位器 0 在移位时钟的每个上升沿将数据传输到 D0~D7 引脚。

例程中使用 `FLEXIO_MCULCD_SetSingleBeatWriteConfig(FLEXIO_MCULCD_Type *base` 函数将 FlexIO 模块配置成 single-beat 写模式。

表 2 提供了 single-beat 写模式的详细寄存器配置。

表 2. Single-beat 写模式配置

寄存器	取值	备注
SHIFTCFG0	0x0007_0100	配置总线位宽为 8bit，禁用移位器停止和开始位。
SHIFTCTL0	0x0003_0302	配置为发送模式，使用 Timer 0 来产生移位时钟，数据在移位时钟的上升沿输出到 FlexIO1_[10:3]引脚。
TIMCMP0	0x0000_0105	$TIMCMP[15:8] = (\text{beats 总数} \times 2) - 1 = (1 \times 2) - 1$ $TIMCMP[7:0] = (\text{波特率分频数} / 2) - 1$  <p style="text-align: center;"><b>NOTE</b> 波特率分频数 = FlexIO 频率 / 波特率</p>
TIMCFG0	0x0000_2200	配置定时器输出逻辑 1 并不受复位影响，在 FlexIO 时钟递减，定时器永不复位，定时器比较时禁用，触发信号为高时启用，禁用开始和停止位。
TIMCTL0	0x01C3_0181	配置 Shifter 0 状态标志作为定时器的内部触发信号，触发电平低有效，定时器引脚为输出，引脚索引为 1 (EN)，引脚电平低有效，定时器为双 8bit 波特率计数器模式。

本应用中使用了 8 个移位器和 1 个定时器来模拟 multi-beats 写模式。定时器 0 用于产生移位时钟和 EN 信号。移位器 0~7 在移位时钟的每个上升沿将数据传输到 D0~D7 引脚。此外，3 个 GPIO 引脚用于驱动 CS，RS 和 R/W 信号。图 5 给出了 multi-beats 写模式的配置框图。

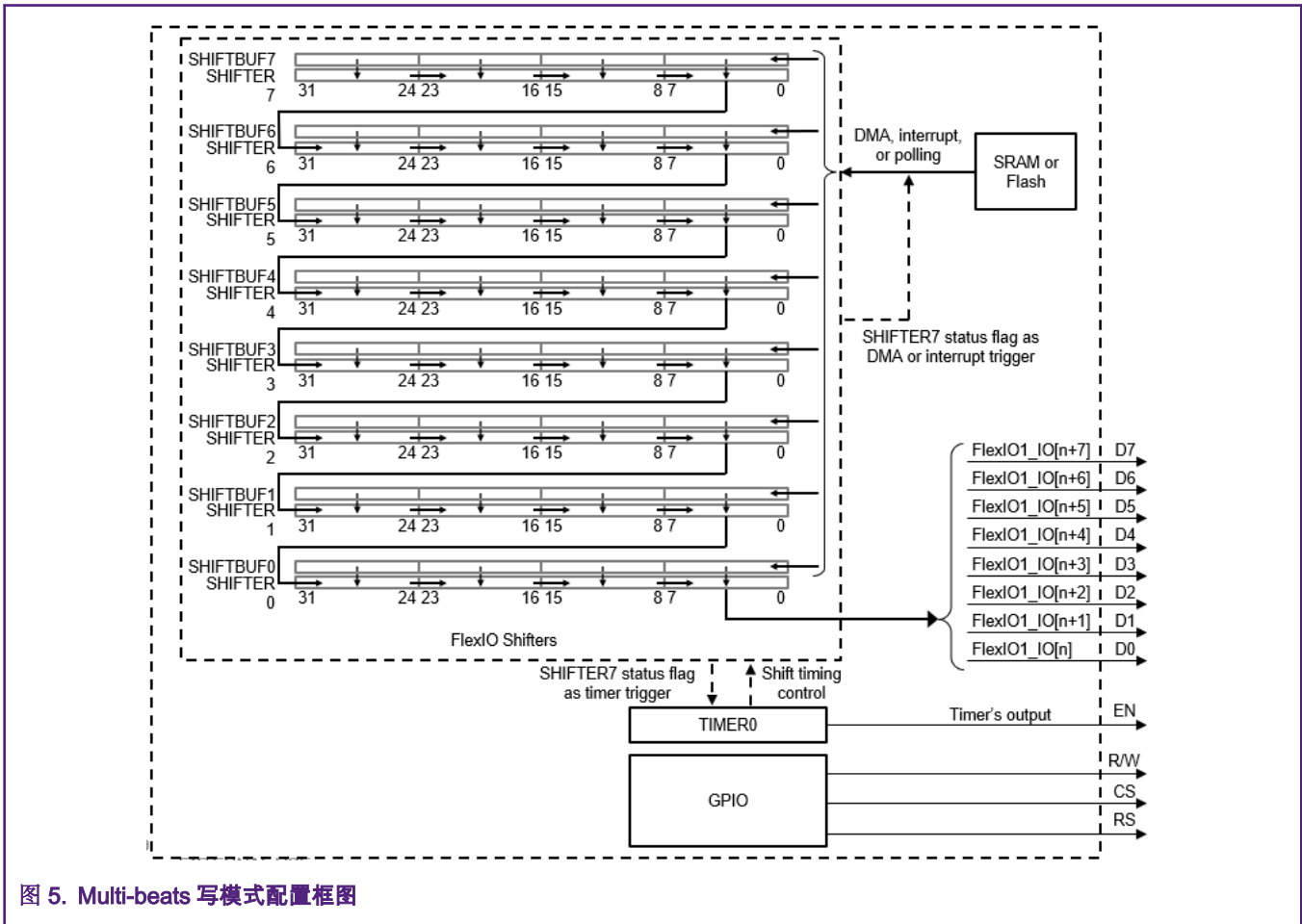


图 5. Multi-beats 写模式配置框图

例程中使用 `FLEXIO_MCULCD_SetMultiBeatsWriteConfig(FLEXIO_MCULCD_Type *base)` 函数将 FlexIO 配置成 multi-beats 写模式。

表 3 提供了 multi-beats 写模式的详细寄存器配置。

表 3. Multi-beats 写模式配置

寄存器	取值	备注
SHIFTCFG0~7	0x0007_0100	配置总线位宽为 8bit，禁用移位器停止和开始位，移位器从移位器 N+1 输入数据。
SHIFTCTL0	0x0003_0302	配置为发送模式，使用 Timer 0 来产生移位时钟，数据在移位时钟的上升沿输出到 FlexIO1_[10:3] 引脚。
SHIFTCTL1~7	0x0000_0002	配置为发送模式，使用 Timer 0 来产生移位时钟，禁用移位器引脚输出功能。
TIMCMP0	0x0000_3F05	$TIMCMP[15:8] = (\text{beats 总数} \times 2) - 1 = (32 \times 2) - 1$ $TIMCMP[7:0] = (\text{波特率分频数} / 2) - 1$ 注: 波特率分频数 = FlexIO 频率 / 波特率

Table continues on the next page...



表 3. Multi-beats 写模式配置 (continued)

TIMCFG0	0x0000_2200	配置定时器输出逻辑 1 并不受复位影响，在 FlexIO 时钟递减，定时器永不复位，定时器比较时禁用，触发信号为高时启用，禁用开始和停止位。
TIMCTL0	0x1DC3_0181	配置 Shifter 7 状态标志作为定时器的内部触发信号，触发电平低有效，定时器引脚为输出，引脚索引为 1 (EN)，引脚电平低有效，定时器为双 8bit 波特率计数器模式。

### 4.3 6800 总线读配置

6800 读操作可以通过将移位器配置成接收模式来实现。6800 并行总线读功能包含 single-beat 读和 multi-beats 读两种模式。

Single-beat 读模式用于小数据接收。该模式下只需要用到一个移位器来接收数据，接收数据时，8 位数据将同时从分配的 8 个 FlexIO 引脚上输入。一个 Single-beat 接收时序仅需要定时器产生一个移位时钟。在这种情况下，使用轮询方法来驱动从移位器到移位器缓冲区的数据传输。

Multi-beats 读模式用于大数据接收。与写模式类似，将移位器串联起来接收数据。一个 Multi-beats 接收时序需要定时器产生多个移位时钟。在这种情况下，使用 DMA 方式来驱动从移位器到移位器缓冲区的数据传输。

本应用中使用了一个移位器和一个定时器来模拟 single-beat 读模式。定时器 0 用于产生移位时钟 和 EN 信号。移位器 7 在移位时钟的每个下降沿从 D0~D7 引脚移入数据。

例程中使用 `FLEXIO_MCULCD_SetSingleBeatReadConfig(FLEXIO_MCULCD_Type *base)` 函数将 FlexIO 模块配置成 single-beat 读模式。

表 4 提供了 single-beat 读模式的详细寄存器配置。

表 4. Single-beat 读模式配置

寄存器	取值	备注
SHIFTCFG7	0x0007_0000	配置总线位宽为 8bit，禁用移位器停止和开始位，移位器从引脚输入数据。
SHIFCTL7	0x0080_0301	配置为接收模式，使用 Timer 0 来产生移位时钟，数据在移位时钟的下降沿从 FlexIO1_[10:3] 引脚输入。
TIMCMP0	0x0000_0105	TIMCMP[15:8] = (beats 总数 x 2) - 1 = (1 x 2) - 1 TIMCMP[7:0] = (波特率分频数 / 2) - 1 注: 波特率分频数 = FlexIO 频率 / 波特率
TIMCFG0	0x0000_2220	配置定时器输出逻辑 1 并不受复位影响，在 FlexIO 时钟递减，定时器永不复位，定时器比较时禁用，触发信号为高时启用，禁用开始位，当定时器禁用时启用停止位(用于内部信号同步以及时禁用计时器)。
TIMCTL0	0x1DC3_0101	配置 Shifter 7 状态标志作为定时器的内部触发信号，触发电平低有效，定时器引脚为输出，引脚索引为 1 (EN)，引脚电平高有效，定时器为双 8bit 波特率计数器模式。

本应用中使用了 8 个移位器和 1 个定时器来模拟 multi-beats 读模式。定时器 0 用于产生移位时钟和 EN 信号。移位器 0~7 在移位时钟的每个下降沿从 D0~D7 引脚移入数据。此外，3 个 GPIO 引脚用于驱动 CS，RS 和 R/W 信号。图 6 给出了 multi-beats 读模式的配置框图。

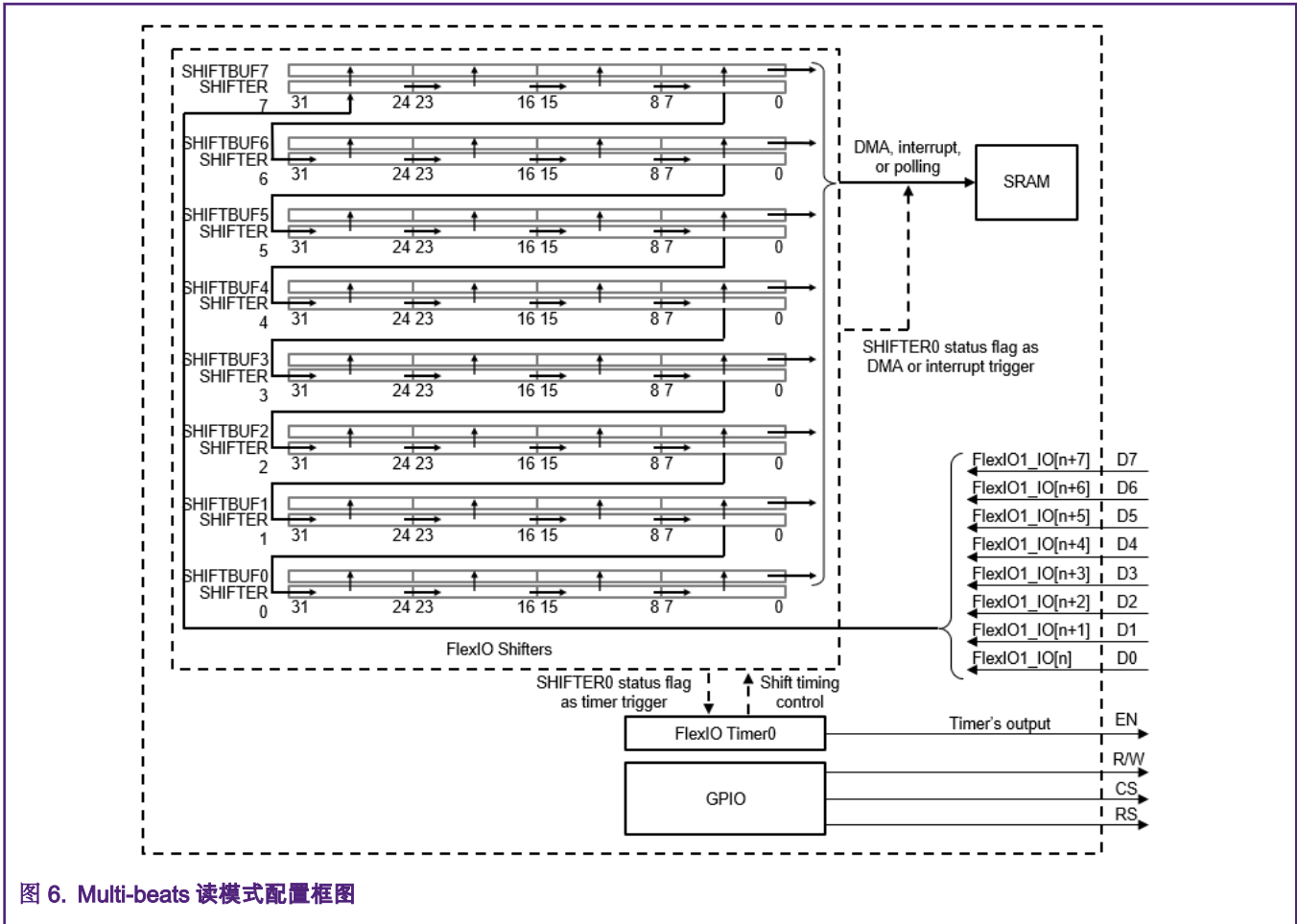


图 6. Multi-beats 读模式配置框图

例程中用 `FLEXIO_MCULCD_SetMultiBeatsReadConfig(FLEXIO_MCULCD_Type *base)` 函数将 FlexIO 模块配置成 multi-beats 读模式。

表 5 提供了 multi-beats 读模式的详细寄存器配置。

表 5. Multi-beats 读模式配置

寄存器	取值	备注
SHIFTCFG0~6	0x0007_0100	配置总线位宽为 8bit，禁用移位器停止和开始位，移位器从移位器 N+1 输入数据。
SHIFTCFG7	0x0007_0000	配置总线位宽为 8bit，禁用移位器停止和开始位，移位器从引脚输入数据。
SHIFTCTL0~7	0x0080_0301	配置为接收模式，使用 Timer 0 来产生移位时钟，数据在移位时钟的下降沿从 FlexIO1_[10:3] 引脚输入。
TIMCMP0	0x0000_3F05	$TIMCMP[15:8] = (\text{beats 总数} \times 2) - 1 = (32 \times 2) - 1$ $TIMCMP[7:0] = (\text{波特率分频数} / 2) - 1$

**NOTE**  
波特率分频数 = FlexIO 频率 / 波特率

Table continues on the next page...



表 5. Multi-beats 读模式配置 (continued)

寄存器	取值	备注
TIMCFG0	0x0000_2220	配置定时器输出逻辑 1 并不受复位影响, 在 FlexIO 时钟递减, 定时器永不复位, 定时器比较时禁用, 触发信号为高时启用, 禁用开始位, 当定时器禁用时启用停止位(用于内部信号同步以及时禁用计时器)。
TIMCTL0	0x01C3_0101	配置 Shifter 0 状态标志作为定时器的内部触发信号, 触发电平低有效, 定时器引脚为输出, 引脚索引为 1 (EN), 引脚电平高有效, 定时器为双 8bit 波特率计数器模式。

#### 4.4 示例运行

由于目前很难找到通过 6800 总线驱动的 LCD 模块, 因此使用逻辑分析仪捕获实际的总线信号以验证 6800 总线时序。逻辑分析仪按照 表 1 中的连接方法连接到 RT1010-EVB 板上。

用户可以从 [nxp.com](http://nxp.com) 网站上下载相关软件代码。找到 IAR 工程 *flexio\_6800*, 编译下载到 i.MXRT1010-EVB 板上并运行。然后, 可以看到逻辑分析仪捕获的所有信号的波形。开发板与逻辑分析仪之间的连接如 图 7 所示。

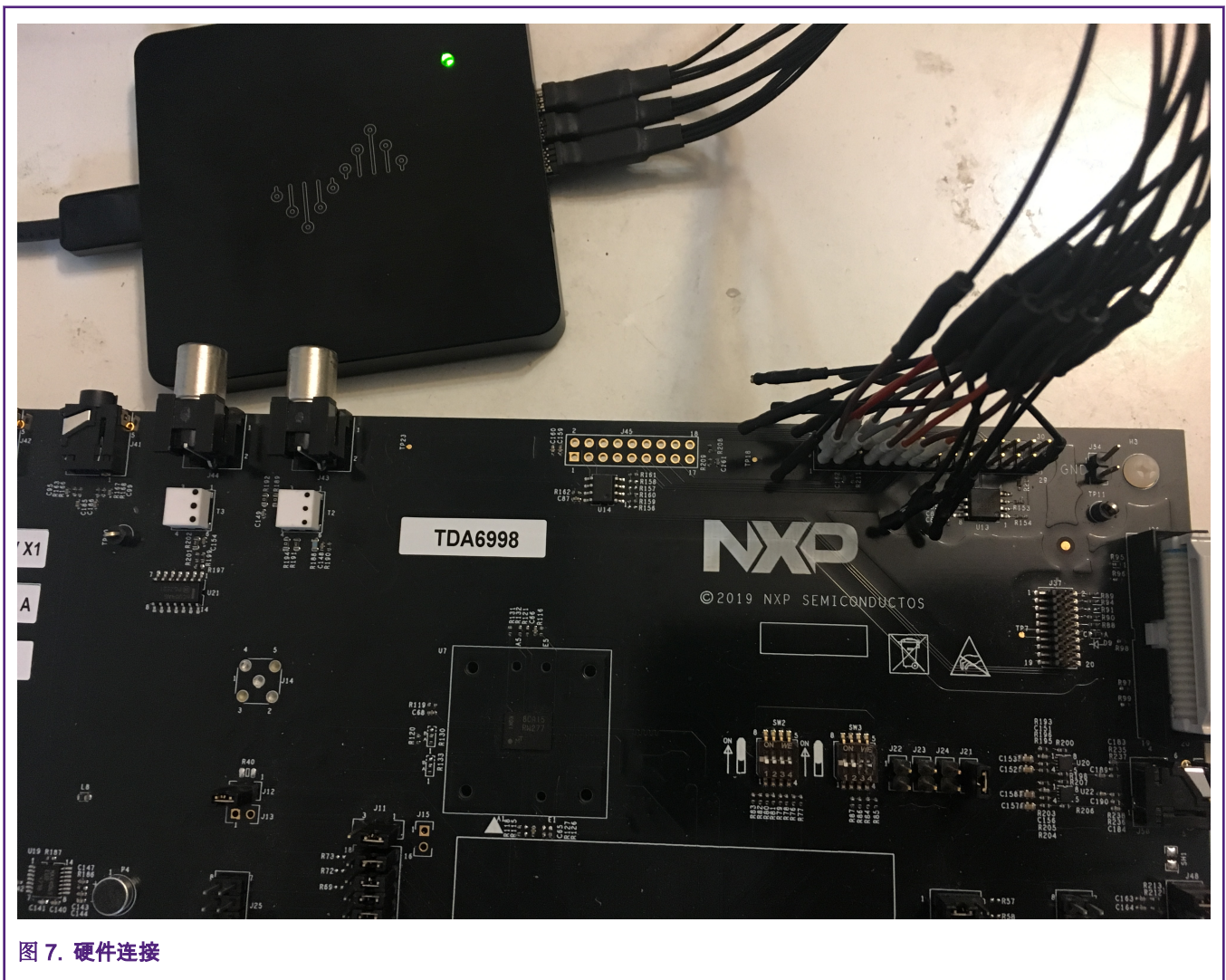


图 7. 硬件连接

图 8 给出了 6800 总线写时序波形。从图中可以看出, 6800 总线写时序从一个写命令周期开始, 然后是一个或多个写数据周期。

在写操作过程中，R/W 信号始终为低。写数据开始之前，应拉低 CS 信号，在完成数据写入之后，拉高 CS 信号。写命令时，RS 信号应为低电平，而写数据时，RS 信号应为高电平。写命令操作使用 single-beat 写模式，在单个移位时钟（由 EN 信号生成）的上升沿，将命令移出至 FlexIO1\_[10:3]引脚。写数据使用 single-beat 或者 multi-beats 写模式，在单个或多个移位时钟的上升沿，将数据移出至 FlexIO1\_[10:3]引脚。

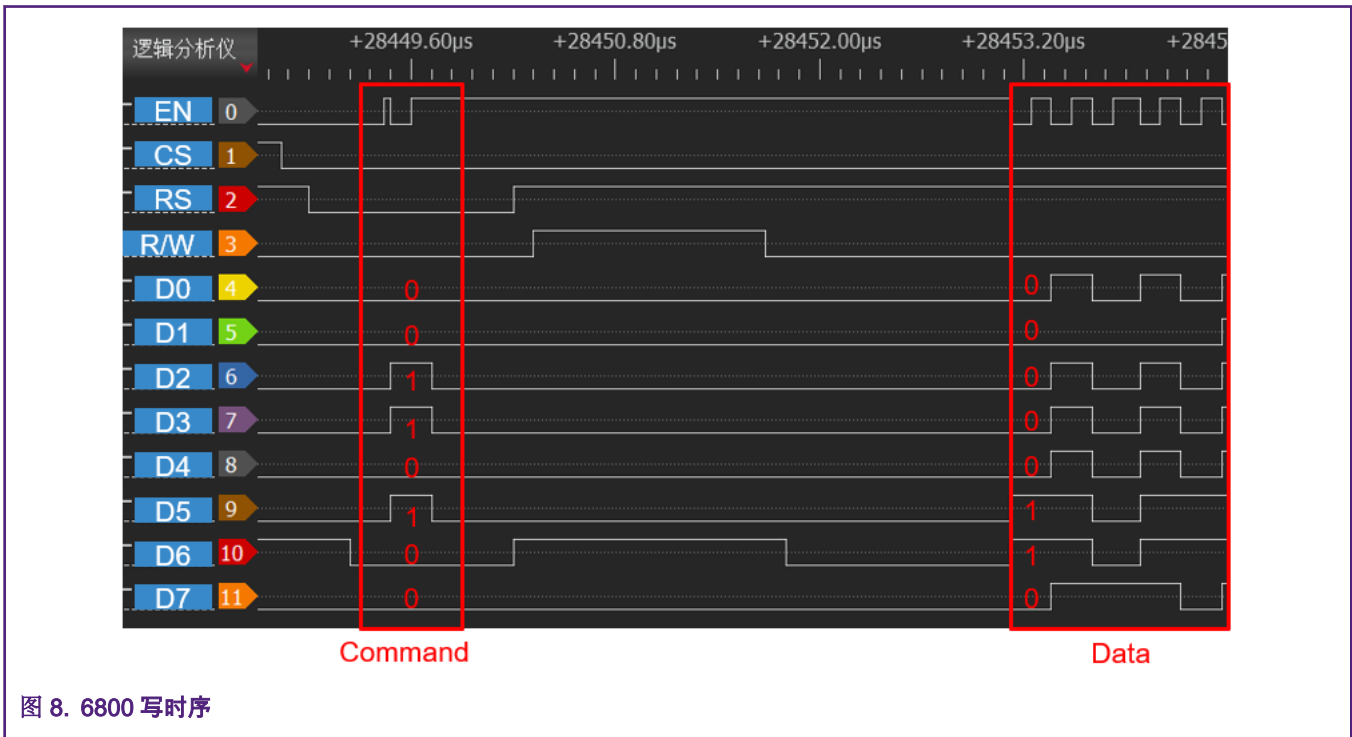
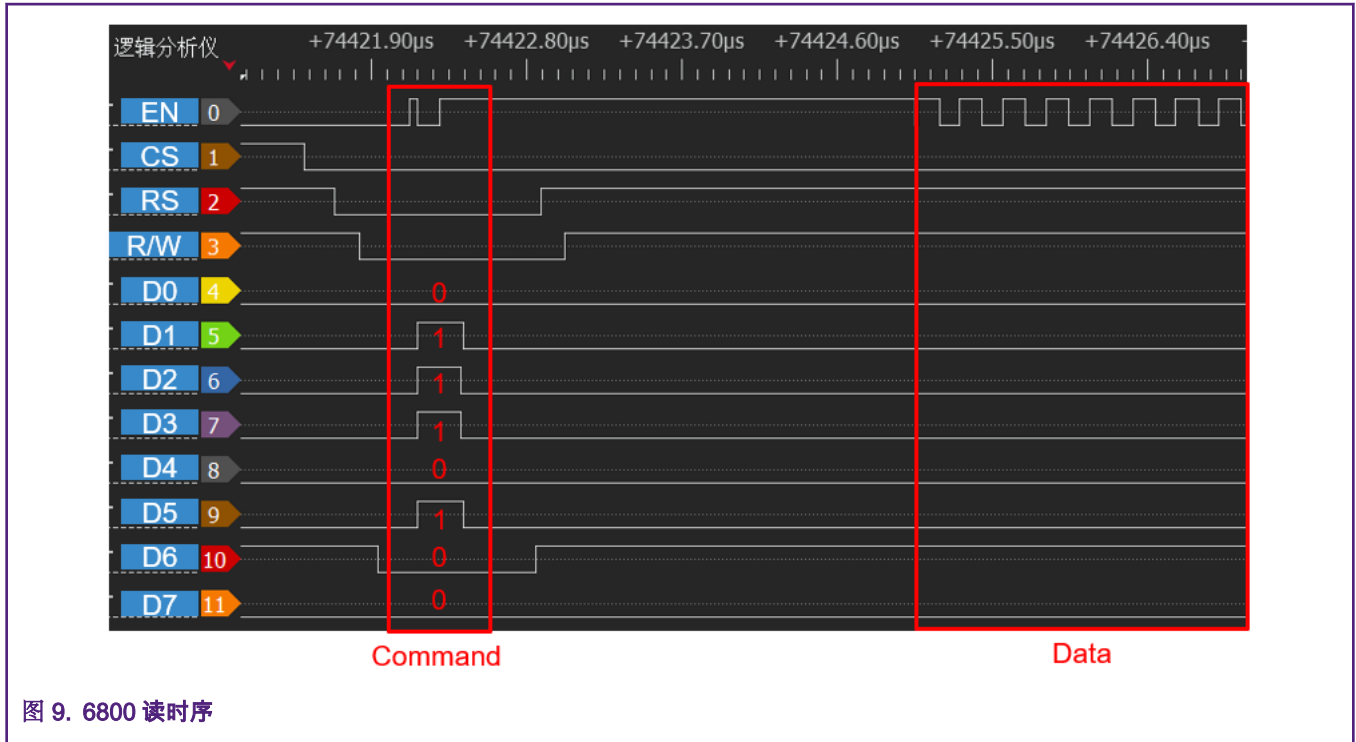


图 8. 6800 写时序

图 9 给出了 6800 总线读时序波形。类似的，6800 总线读时序从一个写命令周期开始，然后是一个或多个读数据周期。

在读操作过程中，R/W 信号始终为高。读数据开始之前，应拉低 CS 信号，在完成读数据之后，拉高 CS 信号。写命令时，RS 信号应为低电平，而读数据时，RS 信号应为高电平。写命令操作使用 single-beat 写模式，在单个移位时钟（由 EN 信号生成）的上升沿，将命令移出至 FlexIO1\_[10:3]引脚。读数据使用 single-beat 或者 multi-beats 读模式，在单个或多个移位时钟的下降沿，将数据从 FlexIO1\_[10:3]引脚移入。



## 5 参考资料

1. *i.MX RT1010 Processor Reference Manual (Rev. 0, 09/2019)* (document [IMXRT1010RM](#))
2. *Using FlexIO to Drive 8080 Bus Interface LCD Module* (document [AN5313](#))

## How To Reach Us

### Home Page:

[nxp.com](http://nxp.com)

### Web Support:

[nxp.com/support](http://nxp.com/support)

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](http://nxp.com/SalesTermsandConditions).

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, 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, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, UMEMS, 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.

© NXP B.V. 2020.

All rights reserved.

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

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

Date of release: April 2020  
Document identifier: AN12813

