AN13472

RT1010 使用 LPI2C 时钟延展

版本 0 — 2021 年 12 月 3 日

应用笔记

目录

1 概述

I²C 是在产品项目开发中常用到的通讯接口，i.MXRT 系列的 MCU 提供了功能强大的 LPI2C 模块。时钟延展功能便是其中之一。RT1010 支持了四种时钟延展方式。本文将介绍从机设备如何应用该功能并且描述了相关的要点。

硬件环境使用的是 RT1010 EVK (版本 C) ，软件环境为 SDK 2.10.0 版本。

具体的例程路径为：

SDK_2_10_0_EVK-MIMXRT1010\boards\evkmimxrt1010\driver_examples\lpi2c\interrupt_b2b_transfer

2 LPI2C 的时钟延展

时钟延展是通将 SCL 保持为低电平来暂停传输，直到 SCL 再次被释放到高电平，传输才能继续进行。
时钟延展通常可以分为字节级和比特级。在字节级的时钟延展，设备通常能够以较快的速度接收数据字节，但需要更多的时间用来存储数据或者准备即将发送的数据。在接收完一个字节数据后从机设备可以将 SCL 保持为低电平，强制让主机设备进入等待状态，直到从机设备完成准备进行下一个字节的传输。

在比特级的时钟延展，每一个 bit 都是可以进行时钟延展而不需要传输整个字节后再进行时钟延展。

注意

在使用该功能前，需要确认主机设备和从机设备是否支持时钟延展功能。因为：

- 不是所有的 I²C 从机设备都支持时钟扩展功能，例如某些传感器和存储设备。
- 不是所有的 I²C 主设备支持时钟扩展，例如使用 GPIO 模拟的 I²C 或者 FPGA 上模拟的 I²C。

3 RT1010 平台 LPI2C 的时钟延展

i.MX RT1010 Processor Reference Manual（文档 IMXRT1010RM）列出了四种时钟延展功能，如 图 1 所示。

- During the 9th clock pulse of the address byte and the Address Valid flag is set.
- During the 9th clock pulse of a slave-transmit transfer and the Transmit Data flag is set.
- During the 9th clock pulse of a slave-receive transfer and the Receive Data flag is set.
- During the 8th clock pulse of an address byte or a slave-receive transfer and the Transmit ACK flag is set. In high speed mode, this is disabled.

图 1. RT1010 支持的四种时钟延展

从我个人的观点出发，我认为用下面的方式进行描述会更准确：

Using **Following the Nth clock pulse** instead of **During the Nth clock pulse**.

3.1 接收地址信息后使能时钟延展

在从机接收完主机发送的地址信息且 AVF 置位后，从机获得 SCL 的控制权，开始持续拉低 SCL 开启时钟延展。那么什么时候结束呢，从硬件状态机的角度看，当 AVF bit 被清除，时钟延展结束，主机获得 SCL 的控制权。举个例子，在接收到地址信息后，使用 delay 函数强行延时 500 μs 后再去清除 AVF bit, 图 2 为逻辑分析仪采集的波形。

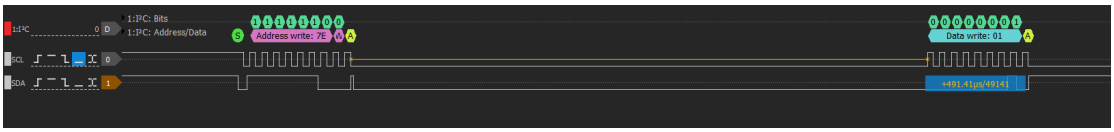

图 2. 接收地址信息后使能时钟延展

从 图 2 可以看到，两次时钟间隔约 491 μs,基本上与 500 μs 的预期相匹配。并不是精确的 500 μs 的原因是延时函数没有使用定时器精确延时。

3.2 从机发送数据前使能时钟延展

在从机接收到主机的地址信息和读指令后，TDF 将会置位，则此时从机获得 SCL 的控制权。同样是在清除 TDF bit 之后，主机再次获得 SCL 的控制权。在这个过程中，从机可以根据主机发送的信息，把要发送的数据准备好，再释放 SCL 的控制权。同样举个例子，再清除 TDF bit 之前等一等，这次等的长一点 800 μs。图 3 为波形结果，两个数据传输过程中 SCL 线被持续拉低约 800 μs。

图 3. 从机发送数据前使能时钟延展

3.3 从机接收数据前使能时钟延展

在从机接收到数据之后，RDF 会置位，此时从机会获得 SCL 的控制权。同样是在清除 TDF bit 之后，主机再次获得 SCL 的控制权。在清 TDF 标志位前增加 1000 μs 的延时，波形如 图 4 所示，两次数据传输过程中 SCL 被持续拉低约 1000 μs。

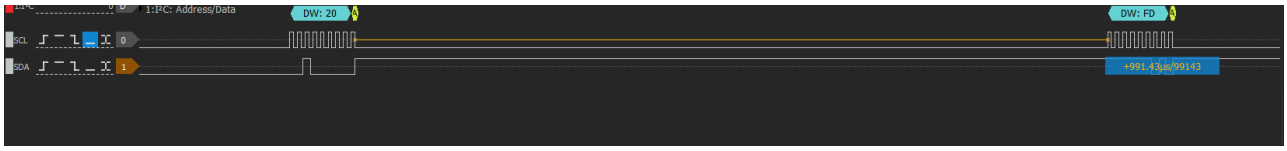

图 4. 从机接收数据前使能时钟延展

3.4 从机发送 ACK/NACK 前使能时钟延展

当主机把数据（地址信息或者数据信息）发送给从机传送完第八个 bit（或者说是第八个时钟）之后，从机获得了 SCL 的控制权限。在此时需要手动向 STAR 寄存器中最后一位写 0 或者 1，向主机反馈 ACK 或者 NACK。在手动发送 ACK/NACK 之前，可以根据主机发送的信息准备数据或者存储数据。举个例子，从机在接收到主机发来的地址信息后（前 8 个时钟），增加一个 500 μs 的延时然后向寄存器内写 0 向主机发送 ACK 的波形见 图 5，可以看到第八个和第九个 CLK 时钟的间隔被拉长。

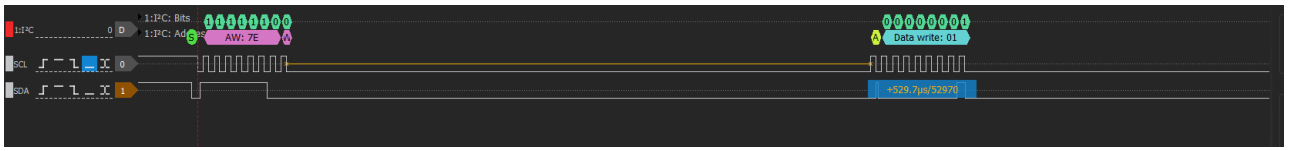

图 5. 从机发送 ACK/NACK 前使能时钟延展

从机接收数据时同样加一个 500 µs 的延时函数。波形见 图 6 同样可以看到，第八个和第九个的时钟被拉长了。

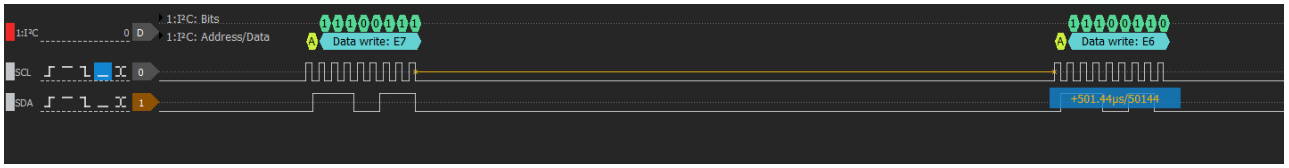

图 6. 从机接收 ACK/NACK 后使能时钟延展

3.5 如何使能时钟延展功能

在 图 7 高亮部分写 true 即可使能对应的时钟延展功能。

```
void LPI2C_SlaveGetDefaultConfig(lpi2c_slave_config_t *slaveConfig)
{
    /* Initializes the configure structure to zero. */
    (void)memset(slaveConfig, 0, sizeof(*slaveConfig));

    slaveConfig->enableSlave              = true;
    slaveConfig->address0                 = 0U;
    slaveConfig->address1                 = 0U;
    slaveConfig->addressMatchMode         = kLPI2C_MatchAddress0;
    slaveConfig->filterDozeEnable         = true;
    slaveConfig->filterEnable             = true;
    slaveConfig->enableGeneralCall        = false;
    slaveConfig->sclStall.enableAck       = false;
    slaveConfig->sclStall.enableTx        = true;
    slaveConfig->sclStall.enableRx        = true;
    slaveConfig->sclStall.enableAddress   = false;
    slaveConfig->ignoreAck                = false;
    slaveConfig->enableReceivedAddressRead = false;
    slaveConfig->sdaGlitchFilterWidth_ns  = 0U; /* TODO determine default width values */
    slaveConfig->sclGlitchFilterWidth_ns  = 0U;
    slaveConfig->dataValidDelay_ns        = 0U;
    slaveConfig->clockHoldTime_ns         = 0U;
}
```

图 7. 如何使能时钟延展功能

4 要点

在使用时钟延展的功能后，同样不能忽略的一个要点是要满足 I²C 的 AC timing，即数据的建立时间和保持时间。对于不同的工作速度，I²C 对于这两个参数的时间要求也是不同的，图 8 为 I²C 的规范上相关信息的截图。

Table 10. Characteristics of the SDA and SCL bus lines for Standard, Fast, and Fast-mode Plus I²C-bus devices[1]

Symbol	Parameter	Conditions	Standard-mode		Fast-mode		Fast-mode Plus		Unit
			Min	Max	Min	Max	Min	Max	
f_{SCL}	SCL clock frequency		0	100	0	400	0	1000	kHz
$t_{HD;STA}$	hold time (repeated) START condition	After this period, the first clock pulse is generated.	4.0	-	0.6	-	0.26	-	µs
t_{LOW}	LOW period of the SCL clock		4.7	-	1.3	-	0.5	-	µs
t_{HIGH}	HIGH period of the SCL clock		4.0	-	0.6	-	0.26	-	µs
$t_{SU;STA}$	set-up time for a repeated START condition		4.7	-	0.6	-	0.26	-	µs
$t_{HD;DAT}$	data hold time[2]	CBUS compatible masters (see Remark in Section 4.1)	5.0	-	-	-	-	-	µs
		I²C-bus devices	0[3]	-[4]	0[3]	-[4]	0	-	µs
$t_{SU;DAT}$	data set-up time		250	-	100[5]	-	50	-	ns
t_r	rise time of both SDA and SCL signals		-	1000	20	300	-	120	ns
t_f	fall time of both SDA and SCL signals[3][6][7][8]		-	300	20 × (V_{DD} / 5.5 V)	300	20 × (V_{DD} / 5.5 V)[9]	120[8]	ns
$t_{SU;STO}$	set-up time for STOP condition		4.0	-	0.6	-	0.26	-	µs
t_{BUF}	bus free time between a STOP and START condition		4.7	-	1.3	-	0.5	-	µs
C_b	capacitive load for each bus line[10]		-	400	-	400	-	550	pF
$t_{VD;DAT}$	data valid time[11]		-	3.45[4]	-	0.9[4]	-	0.45[4]	µs
$t_{VD;ACK}$	data valid acknowledge time[12]		-	3.45[4]	-	0.9[4]	-	0.45[4]	µs
V_{nL}	noise margin at the LOW level	for each connected device (including hysteresis)	0.1V_{DD}	-	0.1V_{DD}	-	0.1V_{DD}	-	V
V_{nH}	noise margin at the HIGH level	for each connected device (including hysteresis)	0.2V_{DD}	-	0.2V_{DD}	-	0.2V_{DD}	-	V

图 8. AC timing parameters from I²C spec

RT1010 中：

- 位于 SCFGR2 寄存器中的 CLKHOLD 可以用来设置建立时间（data set up time）。
- 位于 SCFGR2 寄存器中的 DATAVD 可以用来设置保持时间（data hold time）。

—	
13-8 DATAVD	Data Valid Delay Configures the SDA data valid delay time for the I2C slave, and is equal to FILTSCL+DATAVD+3 cycles. • The data valid delay must be configured to be less than the minimum SCL low period • The I2C slave data valid delay time is not affected by the PRESCALE configuration, and the I2C slave data valid delay time is disabled in high speed mode
7-4 —	Reserved
3-0 CLKHOLD	Clock Hold Time Configures the minimum clock hold time for the I2C slave, when clock stretching is enabled. • The minimum hold time is equal to CLKHOLD+3 cycles • The I2C slave clock hold time is not affected by the PRESCALE configuration, and the I2C slave clock hold time is disabled in high speed mode

图 9. 建立时间和保持时间设置寄存器

数据的建立时间和保持时间长度 t 可以用下面的公式进行计算：

$$t = (CLKHOLD + 3) * T_{clk}$$

t 的具体时间取决于两个变量，即 CLKHOLD 的数值和 T_{clk}。T_{clk} 是 LPI2C 的时钟周期。如 图 8 所示，数据的保持时间对于常规的 I²C 设备来说通常是 0，因此使用默认数值即可，不需要改动。

5 修订记录

版本号	日期	说明
0	2021 年 12 月 3 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.

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

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

Date of release: 2021 年 12 月 3 日

Document identifier: AN13472

arm