M16C Family, R8C Family

I^{2}C-bus Interface Using UARTi Special Mode 1

REJ05B1349-0102
Rev.1.02
Dec. 20, 2010

1. Abstract

This document describes the functions and usage of UARTi special mode 1 to provide I^{2}C-bus interface using the M16C Family and R8C Family serial interface (UARTi) special mode 1 (I^{2}C mode).

2. Introduction

The application example described in this document applies to the following microcomputers (MCUs):

- MCUs: M16C Family, R8C Family

In this document, "i" (e.g. UARTi, UiMR register) indicates the number of serial interface channels available in special mode 1. The number of UARTi channels that can be used in special mode 1 is dependent on the MCU. Refer to individual hardware manuals for details.

The simplified I^{2}C bus communication is enabled by controlling additional functions for I^{2}C bus communication added to the UARTi clock synchronous circuit for I^{2}C bus interface using UARTi special mode 1. The I^{2}C bus interface using UARTi special mode 1 has more limitations for software processing time and timing than the I^{2}C bus interface hardware module. Careful verification and evaluation of your system are recommended, including the interaction between the I^{2}C bus communication program and programs other than the I^{2}C bus communication program.

Quick Reference by Title

Refer to the corresponding master or slave column depending on use.

Title	Master Transmitter	Master Receiver	Slave Transmitter	Slave Receiver
3. Initialization				
3.1 I^{2}C Mode Setting	Yes	Yes	Yes	Yes
3.2 SCL Clock Generation				
3.2.1 SCL Clock in Master Mode	Yes	Yes	N/A	N/A
3.2.2 SCL Clock in Slave Mode	N/A	N/A	Yes	Yes
3.2.3 Counting SCL Clock	Yes	Yes	Yes	Yes
3.2.4 SCL Clock Frequency	Yes	Yes	N/A	N/A
3.3 Other Bits Settings				
3.3.1 IICM2 Bit Setting (Selecting an Interrupt Source)	Yes	Yes	Yes	Yes
3.3.2 STAC Bit Setting (UARTi Initialization)	N/A	N/A	Yes	Yes
3.3.3 DL2 to DL0 Bit Settings (SDA Digital Delay)	Yes	Yes	Yes	Yes
3.3.4 CSC Bit Setting (Clock Synchronization)	Yes	Yes	N/A	N/A
4. Condition Generation and Detection				
4.1 Condition Generation in Master Mode	Yes	Yes	N/A	N/A
4.2 Start Condition and Stop Condition Detection in Slave Mode	N/A	N/A	Yes	Yes
4.3 CKPH Bit Setting (Clock Delay)	Yes	Yes	Yes	Yes
4.4 SCLHI Bit Setting (SCL Output Stop)	Yes	Yes	N/A	N/A
5. Data Transmission/Reception (Including Slave Address Transmission)				
5.1 Transmitting Byte Data	Yes	Yes	Yes	N/A
5.2 Byte Data Reception	N/A	Yes	N/A	Yes
5.3 SWC Bit and SWC9 Bit Settings (SCL Wait Function)	N/A	N/A	Yes	Yes
6. ACK and NACK Generation and Detection				
6.1 Generating ACK and NACK	N/A	Yes	Yes	Yes
6.2 Detecting ACK and NACK	Yes	Yes	Yes	N/A
7. Interrupts				
7.1 Start and Stop Condition Interrupts	Yes	Yes	Yes	Yes
7.2 Receive/Transmit Interrupts				
7.2.1 Receive Interrupt	N/A	N/A	Yes	Yes
7.2.2 Transmit Interrupt	Yes	Yes	Yes	Yes
8. Notes on Using UARTi Special Mode 1 (I^{2}C Mode)				
8.1 Setting Procedure for UiTB Register Data at Slave Transmit/Receive	N/A	N/A	Yes	Yes
8.2 Electrical Characteristics				
8.2.1 Low/High-level Input Voltage and Low-level Output Voltage	Yes	Yes	Yes	Yes
8.2.2 Set-up and Hold Time in When Detecting a Condition	N/A	N/A	Yes	Yes
8.2.3 Set-up and Hold Times When Generating a Condition	Yes	Yes	N/A	N/A
8.3 Maximum Transfer Speed Using the UiBRG Count Source	N/A	N/A	Yes	Yes
8.4 Function Limitations				
8.4.1 SWC2 bit in the UiSMR2 Register (SCL Wait)	N/A	N/A	N/A	N/A
8.4.2 SDHI bit in the UiSMR2 Register (SDA Output Disable)	N/A	N/A	N/A	N/A
8.4.3 Restart Condition in Slave Mode	N/A	N/A	Yes	Yes

3. Initialization

3.1 I2C Mode Setting

Set the IICM bit in the UiSMR register to 1 (I2C mode) and then set bits SMD2 to SMD0 in the UiMR register to 010b (I2C mode) to enter I2C mode.

3.2 SCL Clock Generation

3.2.1 SCL Clock in Master Mode

When using master mode, first set the SCL clock transfer rate.
After writing data to the UiTB register, the SCL clock is output from the SCLi pin within 1.5 cycles.

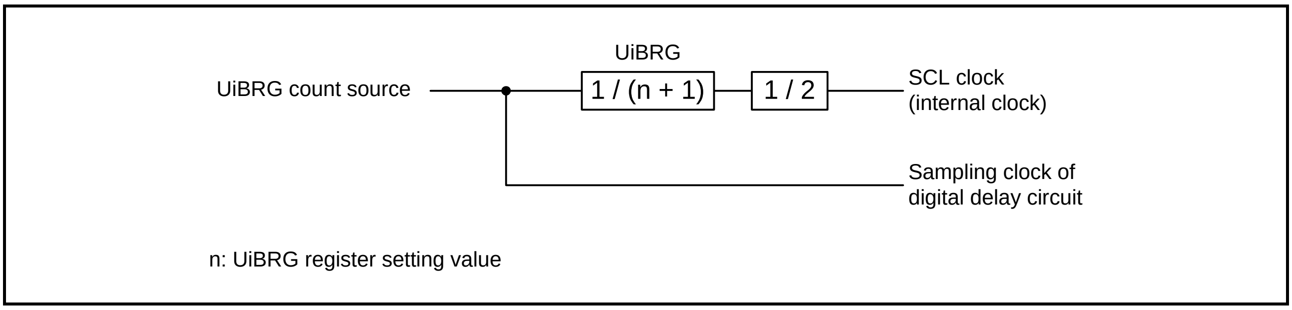


Figure 3.1 Internal Clock Configuration

3.2.2 SCL Clock in Slave Mode

When using slave mode, set the CKDIR bit in the UiMR register to 1 (external clock). This setting disables the UiBRG register.
Since the UiBRG count source is used as a sampling clock of the digital delay circuit, set bits CLK1 and CLK0 in the UiC0 register (see 3.3.3 ?DL2 to DL0 Bit Settings (SDA Digital Delay)?).

3.2.3 Counting SCL Clock

Counting the SCL clock in this application note is shown in Figure 3.2.

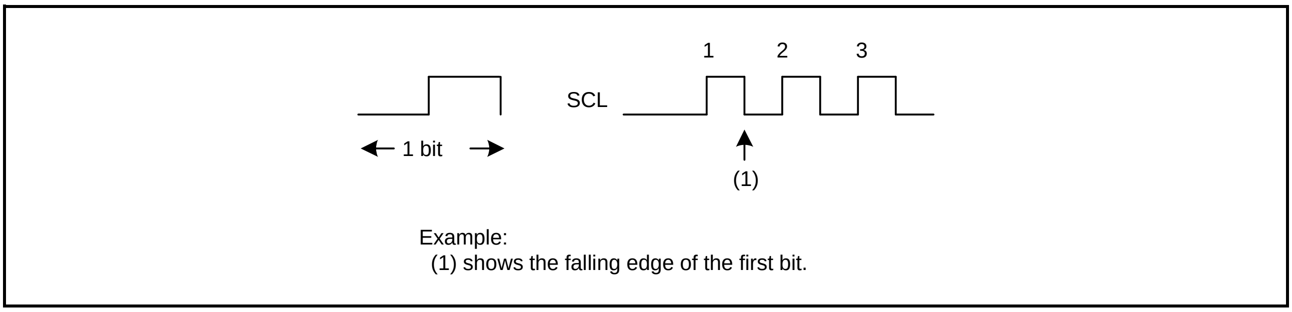


Figure 3.2 Counting th SCL Clock

3.2.4 SCL Clock Frequency

The SCL clock duty generated in I2C mode is 50%. The low-level width of the SCL clock is 1.25 µs when the I2C-bus setting is Fast-mode maximum SCL clock (400 kbps). This value does not satisfy the Fast-mode

I2C-bus specification (f_{LOW} = Min. 1.3 µs). Set the SCL clock to 384.6 kbps or less to satisfy the SCL clock low-level width of 1.3 µs or more.

When the clock synchronous function is enabled, there is a sampling delay of the noise filter plus 1 to 1.5 cycles of UiBRG count source.

There is also a delay of the SCL clock when high is determined and the SCL clock high width is extended. Therefore, the actual SCL clock becomes slower than SCL clock transfer rate setting.

To calculate the actual SCL clock, the SCL clock rise time (t_R) also needs to be taken into consideration.

The following is an example of an SCL clock calculation.

Example of an actual SCL clock calculation at 384.6 kbps
- UiBRG count source: f1 = 20 MHz
- UiBRG register setting value: n = 26 - 1
- SCL clock rise time: t_R = 100 ns
- SCL clock fall time: t_F = 0 ns
- Noise filter width: t_{NF} = 100 ns (1)
- Sampling delay: t_{SD} = 1 cycle

f_{SCL} (theoretical value) = f1 / (2(n + 1)) = 20 MHz / (2(25 + 1)) = 384.6 kbps
t_{LOW} = 1 / (2f_{SCL} (theoretical value)) = 1 / (2 × 384.6 kbps) = 1.3 ms
t_{HIGH} = 1 / (2f_{SCL} (theoretical value)) + t_{NF} + (t_{SD} × 1 / f1)
= 1 / (2 × 384.6 kbps) + 100 ns + (1 × 1 / 20 MHz)
= 1.45 ms
f_{SCL} (actual value) = 1 / (t_F + t_{LOW} + t_R + t_{HIGH}) = 1 / (0 ns + 1.3 ms + 100 ns + 1.45 ms) ≈350.8 kbps

Note:
1. Maximum 200 ns.

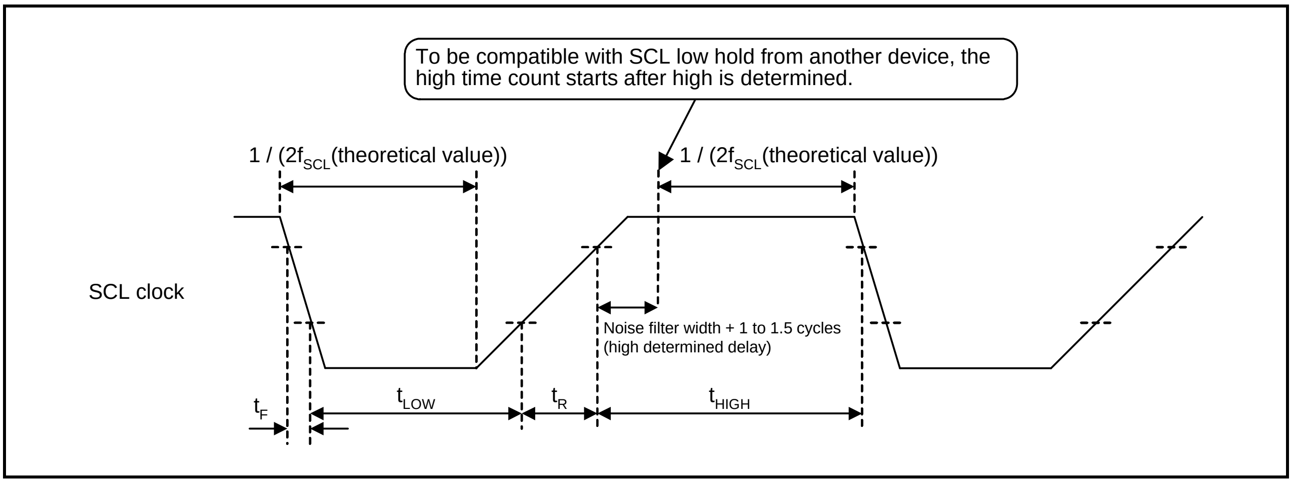

To be compatible with SCL low hold from another device, the high time count starts after high is determined.
1 / (2fSCL(theoretical value))
1 / (2fSCL(theoretical value))
SCL clock
Noise filter width + 1 to 1.5 cycles (high determined delay)
tF
tLOW
tR
tHIGH

Figure 3.3 SCL Clock

3.3 Other Bits Settings

3.3.1 IICM2 Bit Setting (Selecting an Interrupt Source)

The IICM2 bit in the UiSMR2 register is the I^{2}C mode interrupt source select bit. In general, set the IICM2 bit to 1 (UART transmit/receive interrupt).

3.3.2 STAC Bit Setting (UARTi Initialization)

UARTi initialization automatically initializes UARTi when a start condition is detected. Use this function in slave mode. Set the STAC bit in the UiSMR2 register to 1 to enable this function or 0 to disable it.
While using the UARTi initialization function in slave mode, UARTi is automatically initialized when a start condition is detected. Therefore, an interrupt is not necessary when a start condition is detected.
When the STAC bit is 1 and a start condition is detected, the following initialization procedure is executed.

(1) The transmit shift register is initialized, and the UiTB register value is transferred to a transmit shift register. Consequently, there is no need to reset data to the UiTB register when receiving data, and transfer starts with the next input clock as the first bit. As the transmit data is the same as the last data transmitted, set the ACKD bit in the UiSMR4 register to 1 (NACK) and the ACKC bit to 1 (ACK data output) to disable the transmit data from being output.
(2) The receive shift register is initialized, and slave address reception starts with the next input clock as the first bit. Even if UARTi is initialized and data reception starts before the UiRB register is read, an overrun error will not occur.
(3) The SWC bit in the UiSMR2 register automatically becomes 1 (enabled). Consequently, the SCL wait function is enabled and the SCLi pin becomes fixed low at the falling edge of the eighth bit of the SCL clock.

3.3.3 DL2 to DL0 Bit Settings (SDA Digital Delay)

When transferring data with the I2C-bus, change the data while the SCL clock is a low. If SDA is changed when the SCL clock is a high, the change is recognized as one of the corresponding conditions (see 8.2.3 ?Set-up and Hold Times When Generating a Condition?).

The SDA digital delay function delays output from the SDAi pin. By delaying the change of the SDA, the change in data can be read while the SCL clock is low.

The SDA digital delay function is enabled by setting bits DL2 to DL0 in the UiSMR3 register to 001b to 111b, and no delay by setting them to 000b.

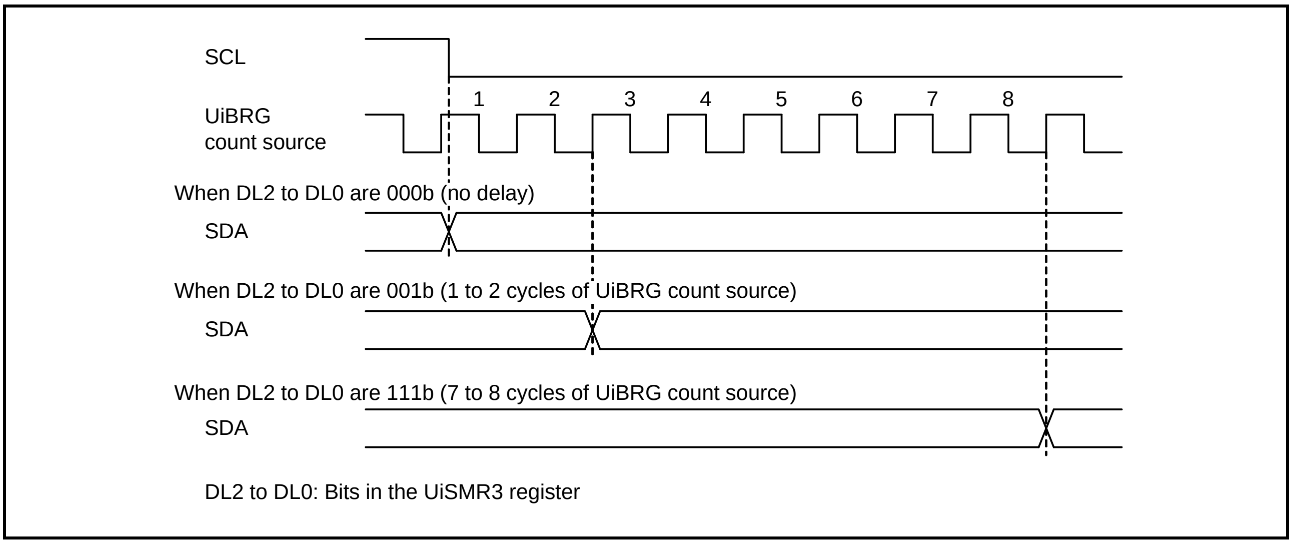

SCL
UiBRG
count source
1
2
3
4
5
6
7
8
When DL2 to DL0 are 000b (no delay)
SDA
When DL2 to DL0 are 001b (1 to 2 cycles of UiBRG count source)
SDA
When DL2 to DL0 are 111b (7 to 8 cycles of UiBRG count source)
SDA
DL2 to DL0: Bits in the UiSMR3 register

Figure 3.4 SDA Output Selection by Setting Bits DL2 to DL0

3.3.4 CSC Bit Setting (Clock Synchronization)

In master mode, set the CSC bit in the UiSMR2 register to 1 (enabled) to enable clock synchronization. In slave mode, set the CSC bit to 0 (disabled).

Clock synchronization enters a wait state automatically by the low hold of the SCLi pin from another device and leaves the wait state by releasing the low hold of the SCLi pin.

When using clock synchronization, the actual SCL clock is delayed compared despite the setting of the SCL clock. Refer to 3.2.4 ?SCL Clock Frequency? for details.

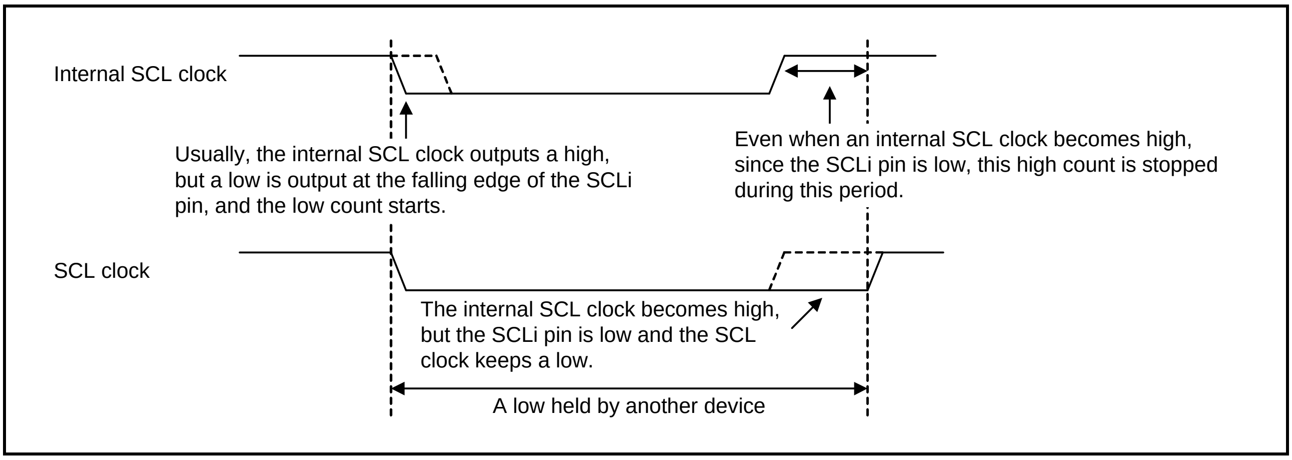

Internal SCL clock
Usually, the internal SCL clock outputs a high, but a low is output at the falling edge of the SCLi pin, and the low count starts.
Even when an internal SCL clock becomes high, since the SCLi pin is low, this high count is stopped during this period.
SCL clock
The internal SCL clock becomes high, but the SCLi pin is low and the SCL clock keeps a low.
A low held by another device

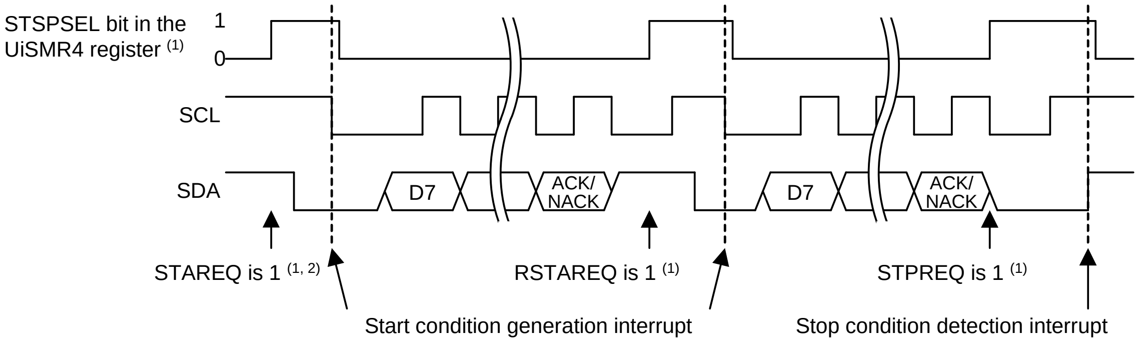
Figure 3.5 Clock Synchronization

4. Condition Generation and Detection

4.1 Condition Generation in Master Mode

In master mode, start, stop and restart conditions are generated by hardware.
A start condition is generated by setting the STAREQ bit in the UiSMR4 register to 1 (start).
A stop condition is generated by setting the STPREQ bit in the UiSMR4 register to 1 (start) after the SCLi pin has been released.
A restart condition is generated by setting the RSTAREQ bit in the UiSMR4 register to 1 (start) after the SCLi pin has been released.
Bits STAREQ, STPREQ, and RSTAREQ automatically become 0 when their respective conditions are generated.
When the STSPSEL bit in the UiSMR4 register is set to 1 (start and stop conditions output), the conditions corresponding to the above bits are generated. When generating a condition, set bits STAREQ, STPREQ, and RSTAREQ to 1 before setting the STSPSEL bit to 1.
Figure 4.2 shows the Register Setting Procedures for Condition Generation.

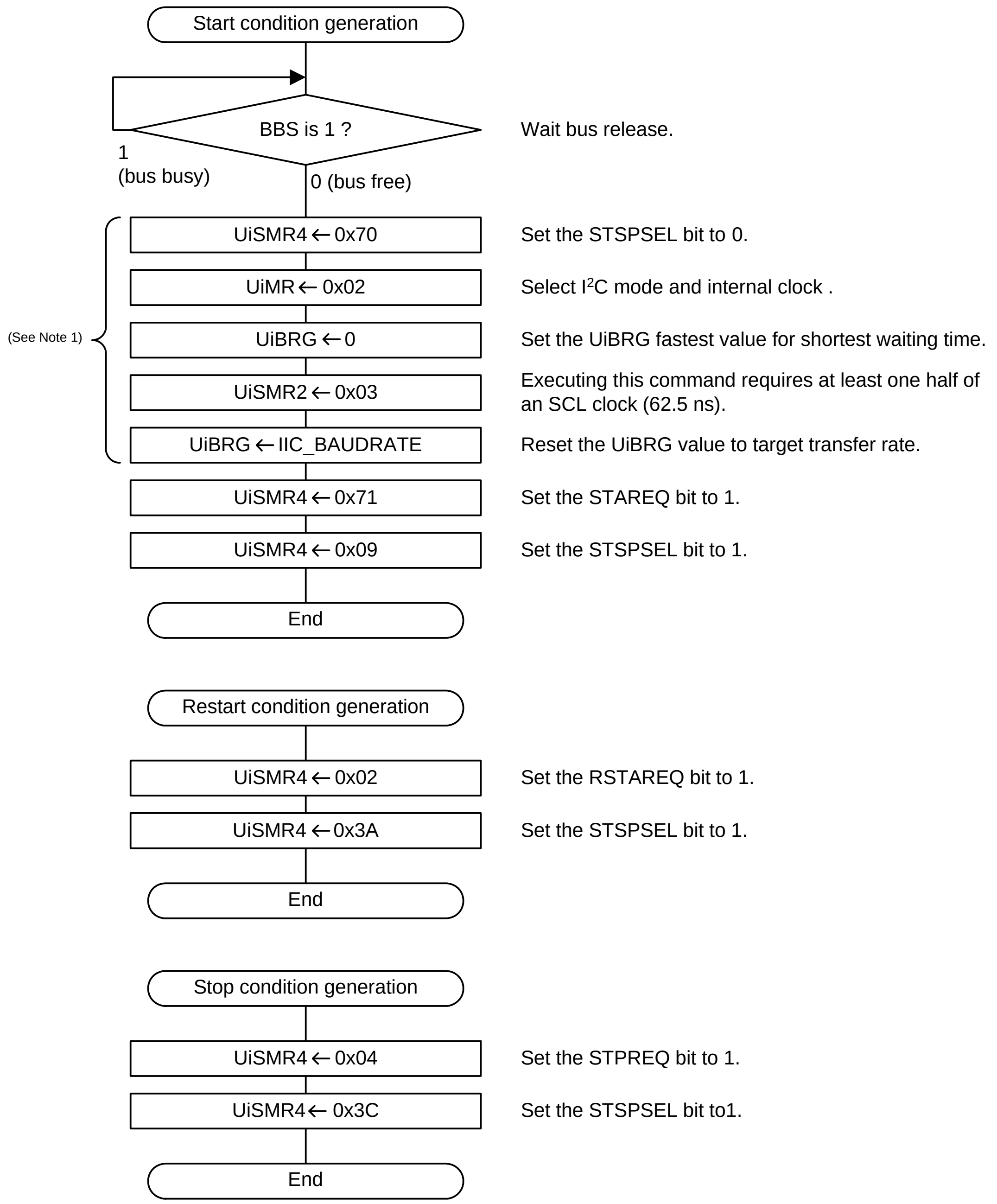
For details on set-up and hold times when generating start and stop conditions, refer to 8.2.3 ?Set-up and Hold Times When Generating a Condition?.



Notes:

1. Set to 0 or 1 by a program. However bits STAREQ, RSTAREQ, and STPREQ become 0 automatically when each of the conditions is generated.
2. When generating a start condition after a stop condition is generated, set the STSPSEL bit in the UiSMR4 register to 0, wait half an SCL clock or more, then set the STAREQ bit to 1.

Figure 4.1 Operation Example of Bits STAREQ, RSTAREQ, STPREQ, and STSPSEL


Start condition generation
BBS is 1 ?
Wait bus release.
1
(bus busy)
0 (bus free)
UiSMR4 ← 0x70
Set the STSPSEL bit to 0.
UiMR ← 0x02
Select I^{2}C mode and internal clock .
(See Note 1)
UiBRG ← 0
Set the UiBRG fastest value for shortest waiting time.
UiSMR2 ← 0x03
Executing this command requires at least one half of an SCL clock (62.5 ns).
UiBRG ← IIC_BAUDRATE
Reset the UiBRG value to target transfer rate.
UiSMR4 ← 0x71
Set the STAREQ bit to 1.
UiSMR4 ← 0x09
Set the STSPSEL bit to 1.
End
Restart condition generation
UiSMR4 ← 0x02
Set the RSTAREQ bit to 1.
UiSMR4 ← 0x3A
Set the STSPSEL bit to 1.
End
Stop condition generation
UiSMR4 ← 0x04
Set the STPREQ bit to 1.
UiSMR4← 0x3C
Set the STSPSEL bit to1.
End

The above assumes the following:
XIN = 16 MHz, main clock divided by 1 (no division), UiBRG count source = f1

Note:
1. After a stop condition is generated, when generating the next start condition, after setting the STSPSEL bit in the UiSMR4 register to 0 and waiting at least half of an SCL clock, then set the STAREQ bit to 1.

Figure 4.2 Register Setting Procedures for Condition Generation

4.2 Start Condition and Stop Condition Detection in Slave Mode

In slave mode, start and stop conditions can be detected by start and stop condition detecting interrupts. Refer to 7.1 ?Start and Stop Condition Interrupts? for details.

When detecting a start condition or stop condition, the set-up and hold times may differ from the I^{2}C-bus specification. Refer to 8.2.2 ?Set-up and Hold Time in When Detecting a Condition? for details.

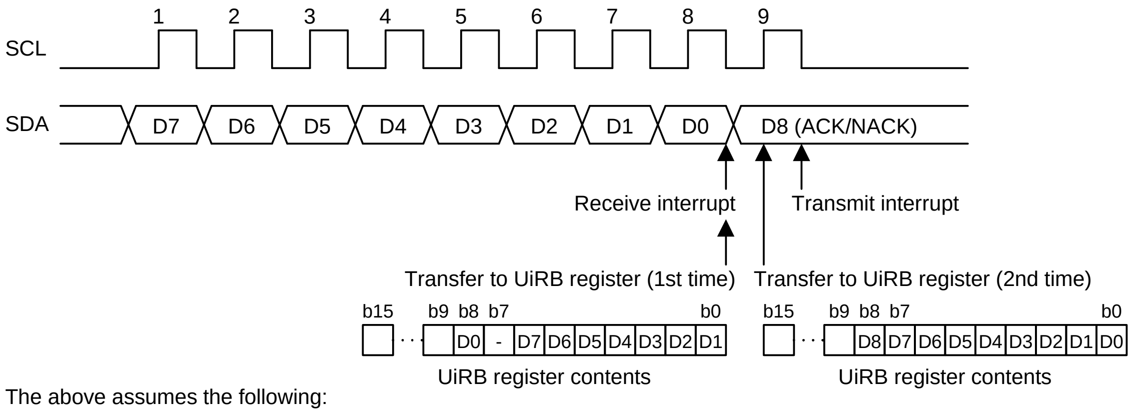
4.3 CKPH Bit Setting (Clock Delay)

Use the clock delay function in I^{2}C mode.

The clock delay function is enabled by setting the CKPH bit in the UiSMR3 register to 1 (with clock delay). When using the clock delay function, data is transmitted twice from the receive shift register to the UiRB register.

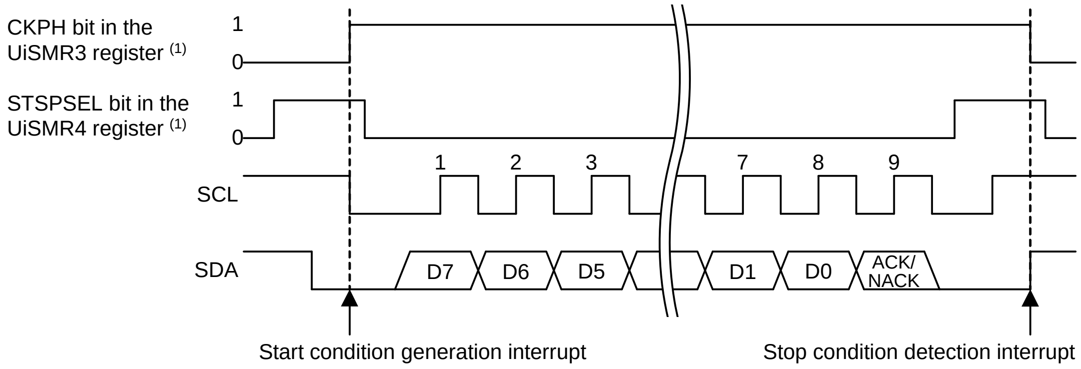
For contents of UiRB register, refer to Figure 4.3 ?Clock Delay Function?.

In master mode, set the CKPH bit when the STSPSEL bit in the UiSMR4 is 1 (start and stop conditions output). Before generating a start condition, set the CKPH bit to 0 (no clock delay). After generating a start condition, set the CKPH bit to 1 before setting the STSPSEL bit to 0 (start and stop conditions not output). After generating a stop condition is generated, set the CKPH bit to 0 before setting the STSPSEL bit to 0.



The above assumes the following:
- The IICM2 bit in the UiSMR2 register is 1 (UART transmit/UART receive interrupt).
- The CKPH bit in the UiSMR3 register is 1 (with clock delay).

Figure 4.3 Clock Delay Function



Note:
1. Set to 0 or 1 by a program.

Figure 4.4 CKPH Bit Setting in Master Mode

4.4 SCLHI Bit Setting (SCL Output Stop)

In master mode, the SCLHI bit must be set when generating a start condition. Refer to Figure 4.2 ?Register Setting Procedures for Condition Generation? for details.
In slave mode, set the SCLHI bit to 0 (disabled).

5. Data Transmission/Reception (Including Slave Address Transmission)

5.1 Transmitting Byte Data

When transmitting byte data, the SDAi pin outputs transmit data for the first to eighth bits, and it is released to receive an acknowledgement for the ninth bit.

In I^{2}C mode, by setting the UiTB register to 9-bit data, transmit data can be output and the SDAi pin can be released. In 9-bit data, set the transmit data to bits b7 to b0 and set b8 to 1 to release the SDAi pin.

By setting the UFORM bit in the UiC0 register to 1 (MSB first) and 9-bit data to the UiTB register, transmit data is output from the SDAi pin in the following order: b7, b6, b5 b4, b3, b2, b1, b0 and b8. By setting b8 to 1, the SDAi pin becomes high-impedance for the ninth bit and an acknowledgement can be received.

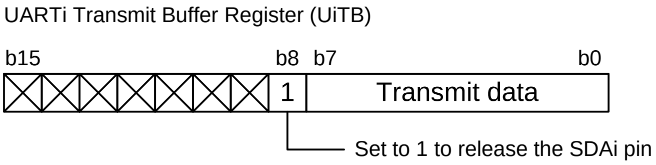

UARTi Transmit Buffer Register (UiTB)
b15
b8
b7
b0
1
Transmit data
Set to 1 to release the SDAi pin

Figure 5.1 UiTB Register Setting

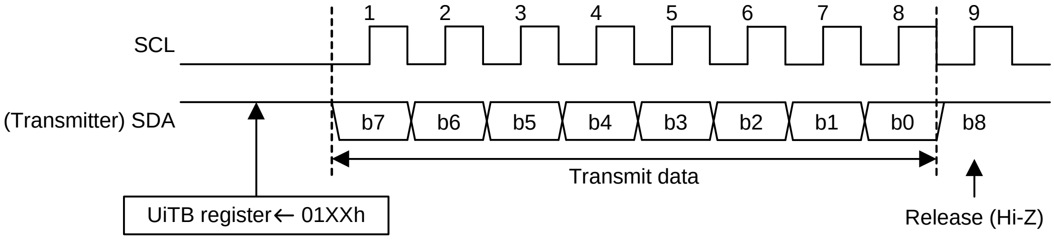


Figure 5.2 Byte Data Transmission

5.2 Byte Data Reception

When receiving byte data, the SDAi pin is released for the first to eighth bits to receive data, and an acknowledgement is generated for the ninth bit. NACK is generated as an acknowledgement when the last byte data is received in master mode, or when the slave address does not match in slave mode. In all other cases, ACK is generated.

In I2C mode, by setting 9-bit data to the UiTB register, data can be received and an acknowledgment can be generated. In 9-bit data, set FFh to bits b7 to b0 to release the SDAi pin and set bit b8 to 0 to generate ACK or 1 to generate NACK.

By setting 00FFh or 01FFh as 9-bit data to the UiTB register, the SDAi pin becomes high-impedance for the first to eighth bits, and data can be received. ACK or NACK is generated for the ninth bit depending on the setting.

The received data can be read from the UiRB register. When the clock delay function is used, data transfer to the UiRB register occurs twice and each UiRB register value is different. Refer to 4.3 ?CKPH Bit Setting (Clock Delay)? for details.

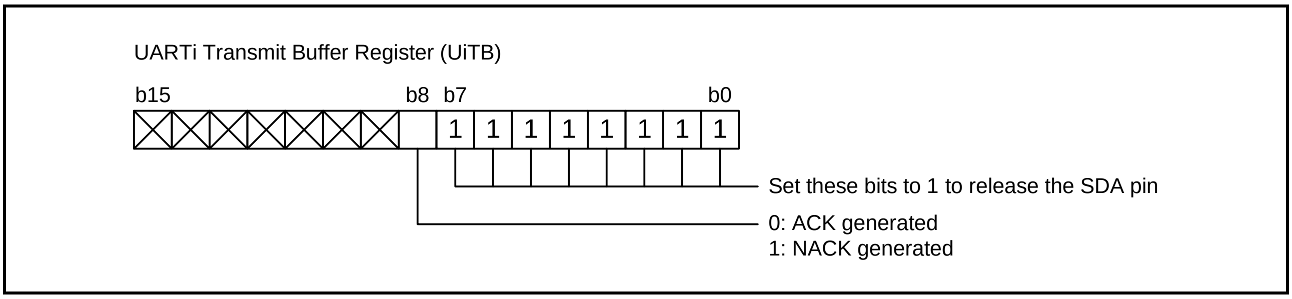

UARTi Transmit Buffer Register (UiTB)
b15
b8
b7
b0
1 1 1 1 1 1 1 1
Set these bits to 1 to release the SDA pin
0: ACK generated
1: NACK generated

Figure 5.3 UiTB Register Setting

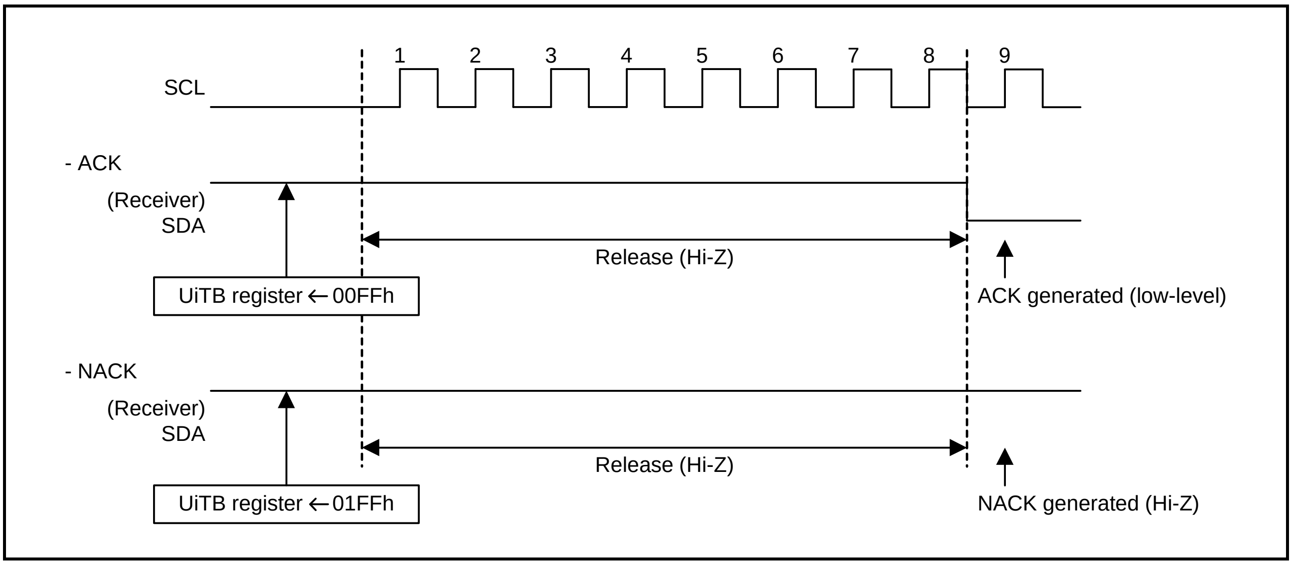


Figure 5.4 Byte Data Reception

5.3 SWC Bit and SWC9 Bit Settings (SCL Wait Function)

To secure the necessary amount of time for generating an acknowledgement or preparing data, the SCL wait function forces other devices to wait. Use the SCL wait function in slave mode.

Set the SWC bit in the UiSMR2 register to insert a wait for acknowledgement generation.
When the SWC bit is 1 (after 8 bit receive, SCLi pin low hold), the SCLi pin becomes fixed low at the falling edge of the eighth bit of the SCL clock. When the SWC bit is set to 0 (disabled), the SCLi pin is released.

Set the SWC9 bit in the UiSMR4 register to insert a wait to judge the received acknowledgement. When the CKPH bit in the UiSMR4 register is 1 (with clock delay) and the SWC9 bit is set to 1 (after 9 bit receive, SCLi pin low hold), the SCLi pin becomes fixed low at the falling edge of the ninth bit of the SCL clock. When the SWC9 bit is set to 0 (SCL low hold disabled), the SCLi pin is released.

Refer to 7.2 ?Receive/Transmit Interrupts? for details on transmit and receive interrupt timing.

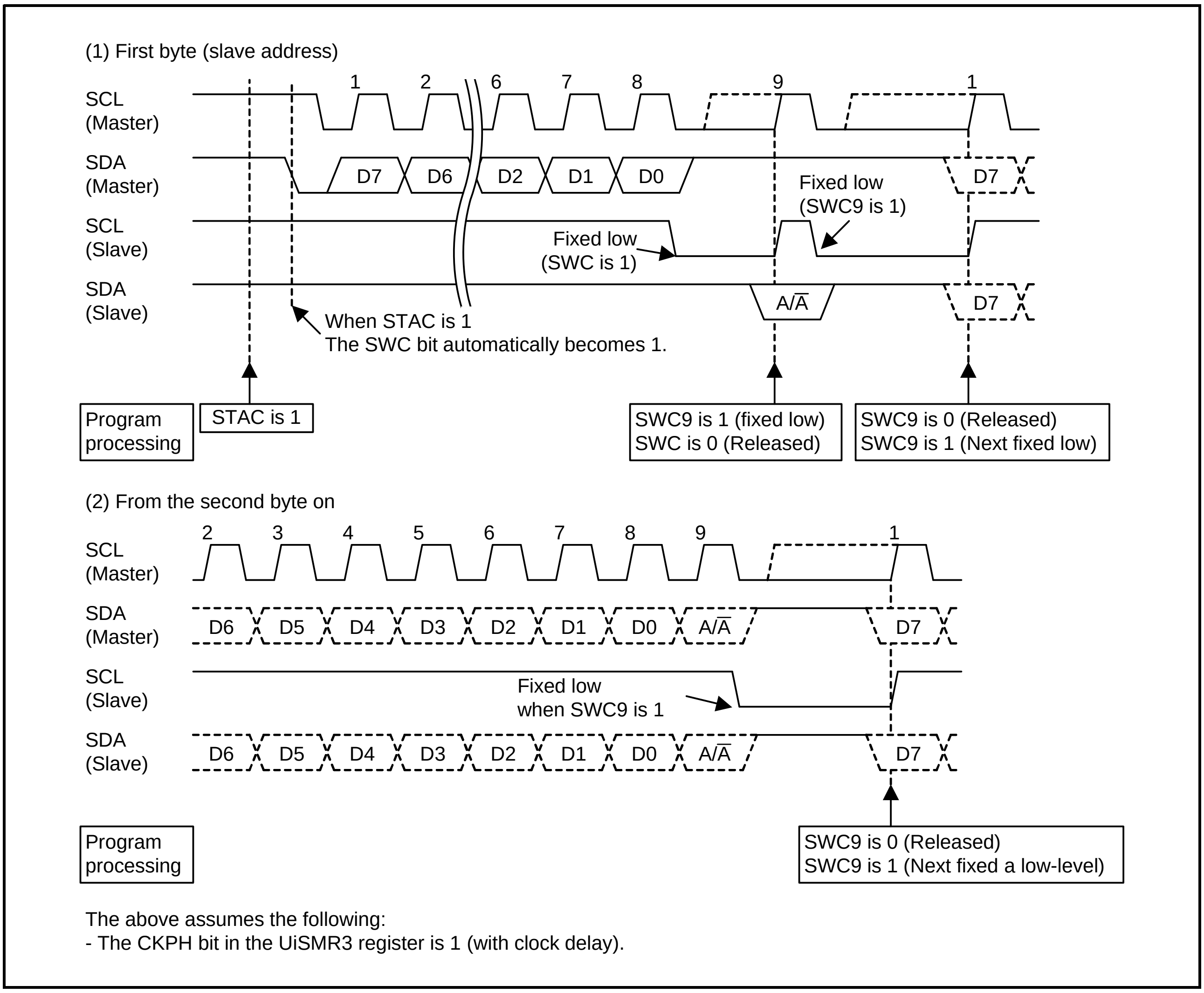


Figure 5.5 SCL Wait Function

6. ACK and NACK Generation and Detection

6.1 Generating ACK and NACK

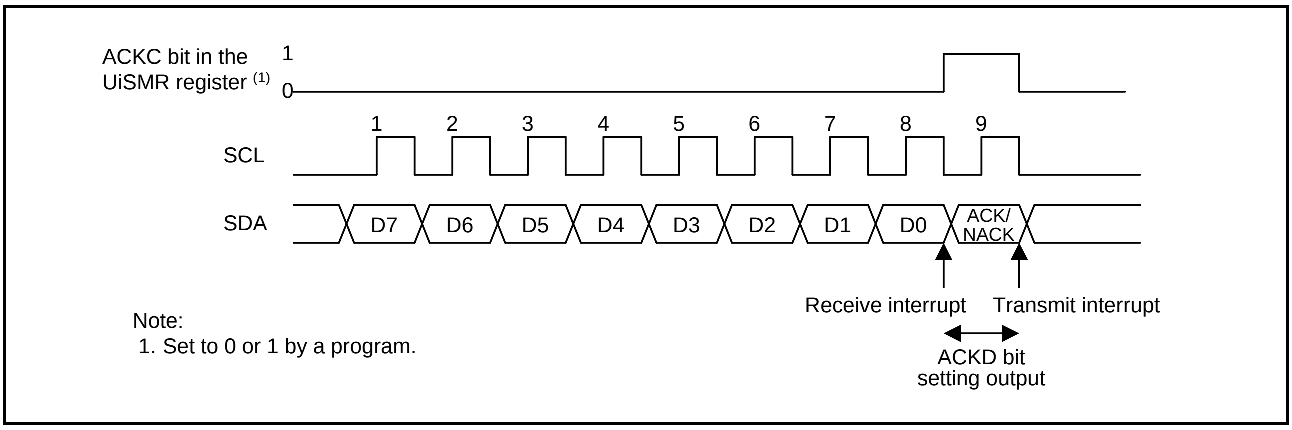
When receiving data that includes a slave address, the receiver generates an acknowledgement at the ninth bit.

An acknowledgement is generated by setting bits ACKD and ACKC in the UiSMR4 register. When the ACKC bit set to 1 (ACK data output), the UiTB register setting is not output and the ACKD bit setting is output. When the ACKD bit is 0 (ACK), the SDAi pin outputs a low. When the ACKD bit is 1 (NACK), the SDAi pin becomes high-impedance.

If ACK and NACK are generated after data reception, ACK is generated when the ACKD bit set to 0 and the ACKD bit set to 1 at the falling edge of the eighth bit of the SCL clock. NACK is generated when the ACKD bit set to 1 and the ACKC bit set to 1. Set ACKC bit set to 0 (serial interface data output) at the falling edge of the ninth bit of the SCL clock (see Figure 6.1 ?ACKC Bit and ACKD Bit Settings?).

When ACK or NACK is determined before data receive starting time, acknowledge is generated by the UiTB register set to 00FFh (ACK) or 01FFh (NACK) (see 5.2 ?Byte Data Reception?). At this point, set the ACKC bit to 0.

In slave mode, ACK or NACK is generated according slave address matching, in the receive interrupt handler, set the ACKD bit to 0 or 1 and the ACKC bit to 1 (see 7.2 ?Receive/Transmit Interrupts?).



Note:
1. Set to 0 or 1 by a program.

Figure 6.1 ACKC Bit and ACKD Bit Settings

6.2 Detecting ACK and NACK

When transmitting data that includes a slave address, the transmitter receives an acknowledgement at the ninth bit.

In the transmit interrupt handler, read b8 in the UiRB register to determine the received acknowledgement (see 5.1 ?Transmitting Byte Data?).

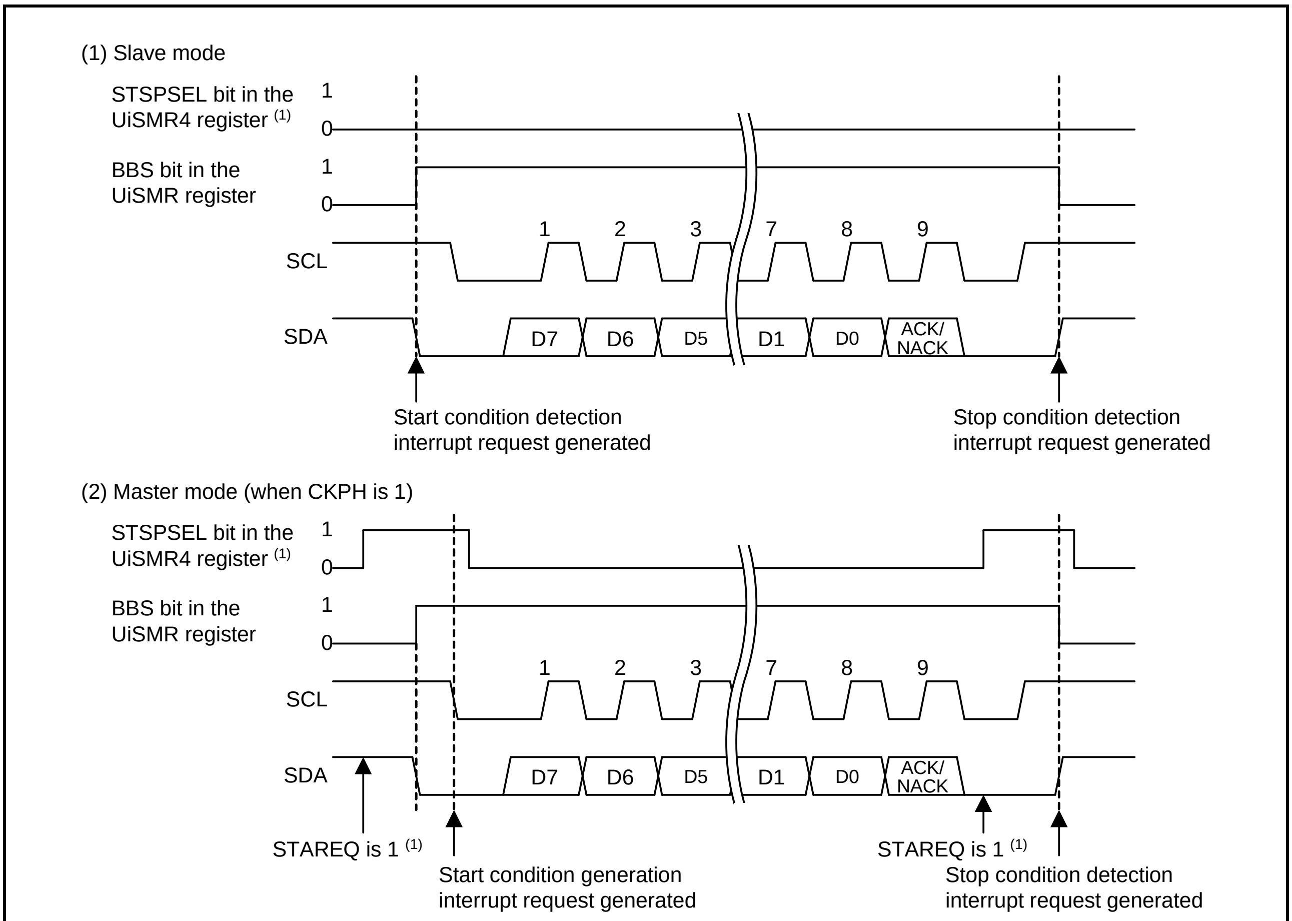
7. Interrupts

7.1 Start and Stop Condition Interrupts

The interrupt request generation timing of the start condition interrupt depends on the setting of the STSPSEL bit in the UiSMR4 register.
When the STSPSEL bit is 0 (start and stop conditions not output), an interrupt request is generated when a start condition or stop condition is detected.
When the STSPSEL bit is 1 (start and stop conditions output), an interrupt request is generated when a start condition is generated or a stop condition is detected.
In master mode, set the STSPSEL bit to 0 in the start and stop condition generation interrupt handler.

The BBS bit in the UiSMR register is changed to 1 (start condition detected) at the falling edge of SDA in a start condition. The BBS bit is changed to 0 (stop condition detected) at the rising edge of SDA in a stop condition. When an interrupt request is generated, read the BBS bit and execute start condition or stop condition processing.

While in slave mode and while using the UARTi initialization function, UARTi is automatically initialized when a start condition is detected, so an interrupt is not necessary when detecting a start condition (see 3.3.2 ?STAC Bit Setting (UARTi Initialization)?.

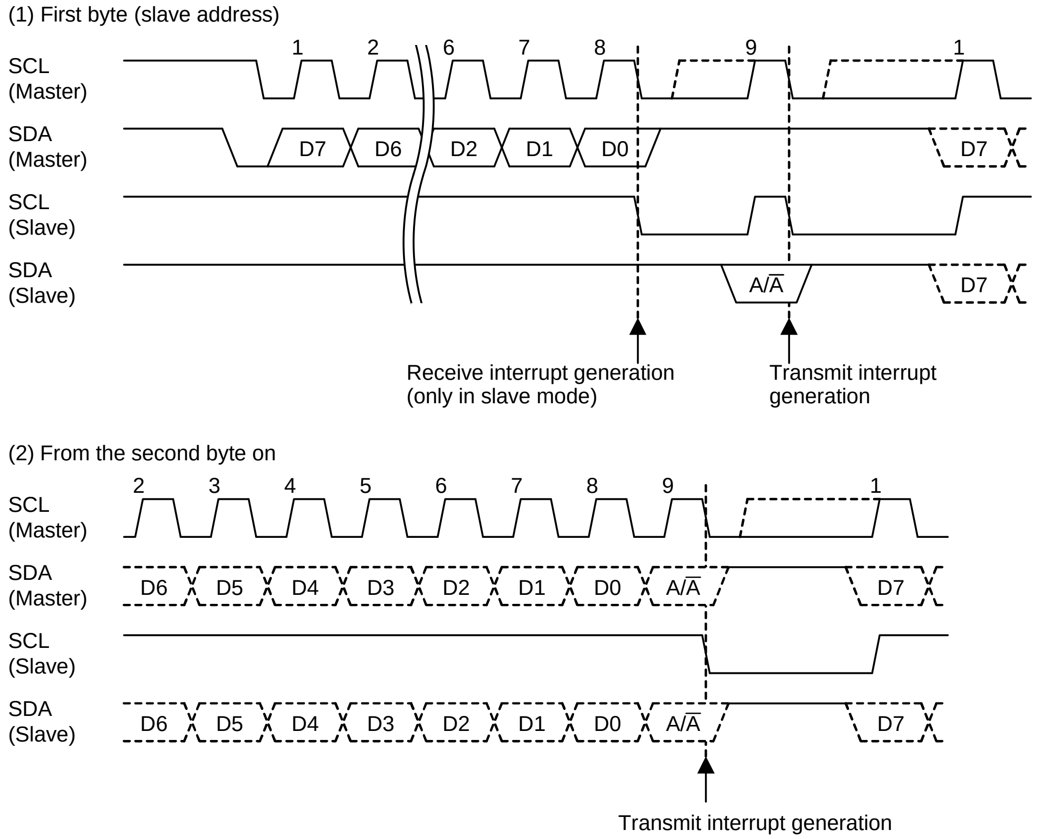

(1) Slave mode
STSPSEL bit in the UiSMR4 register (1)
1
0
BBS bit in the UiSMR register
1
0
SCL
1
2
3
7
8
9
SDA
D7
D6
D5
D1
D0
ACK/ NACK
Start condition detection interrupt request generated
Stop condition detection interrupt request generated
(2) Master mode (when CKPH is 1)
STSPSEL bit in the UiSMR4 register (1)
BBS bit in the UiSMR register
SCL
SDA
STAREQ is 1 (1)
Start condition generation interrupt request generated
STAREQ is 1 (1)
Stop condition detection interrupt request generated

Note:
1. Set to 0 or 1 by a program.

Figure 7.1 Start and Stop Condition Interrupts

7.2 Receive/Transmit Interrupts

Figure 7.2 shows the Receive/Transmit Interrupt Timing.



The above assumes the following:
- The CKPH bit in the UiSMR3 register is 1 (with clock delay).

Figure 7.2 Receive/Transmit Interrupt Timing

7.2.1 Receive Interrupt

A receive interrupt is generated at the falling edge of the eighth bit of the SCL clock.
When the CKPH bit in the UiSMR3 register is 1 (with clock delay), by reading the UiRB register in the receive interrupt handler, the bit position of the received data is changed (see Figure 4.3 ?Clock Delay Function?).

Examples of receive interrupt handling are below.
Additional processing can be added as needed.

Slave transmission and reception

(1) First byte (slave address)

- Read the slave address.

When slave address is matched

- Generate ACK (ACKD is 0, ACKC is 1).
- Fix SCL low after receiving 9 bits (SWC9 is 1). Release SCL after receiving 8 bits (SWC is 0).

When slave address does not match

- Generate NACK (ACKD is 1, ACKC is 1)
- Receive and transmit interrupts disabled.
- Start condition and stop condition interrupts enabled.
- SCL low hold disabled after receiving 9 bits (SWC9 is 0). Release SCL after receiving 8 bits (SWC is 0).

7.2.2 Transmit Interrupt

In I^{2}C mode, set the UiIRS bit in the UiC1 register to 1 (UARTi transmit interrupt source is transmission completed (TXEPT is 1)).
When the CKPH bit in the UiSMR3 register is 1 (with clock delay), a transmit interrupt is generated at the falling edge of the ninth bit of the SCL clock.

Examples of transfer interrupt handling are below.
Additional processing can be added as needed.

Master transmission

(1) First byte (slave address)

- Read ACK and NACK (check slave address matching).
- Set second byte transmit data (when ACK).
- Generate a stop condition or restart condition (when NACK).

(2) From second byte on

- Read ACK and NACK.
- Set next byte transmit data (when ACK).
- Generate a stop condition or restart condition (when NACK, restart, or last byte).

Master reception

(1) First byte (slave address)

- Read ACK and NACK (check slave address matching).
- Set second byte of ACK data (when ACK) (prepare for next reception).
- Generate a stop condition or restart condition (when NACK).

(2) From second byte on

- Set next byte of ACK data (prepare for next reception).

(3) (Last byte - 1) byte

- Set next byte of NACK data (prepare for next reception).

(4) Last byte

- Generate a stop condition or restart condition.

Slave transmission

(1) First byte (slave address)

- Disable ACK data output set by the receive interrupt handling (ACKC is 0).
- Set the second byte of transmit data.
- Immediately after disabling SCL low hold (SWC9 bit is 0), enable the SCL low hold (SWC9 bit is 1) after receiving 9 bits from the second byte.

(2) From the second byte on

- Read ACK and NACK.
- Set the next byte of the transfer data (when ACK).
- Immediately after disabling SCL low hold (SWC9 bit is 0), enable the SCL low hold (SWC9 bit is 1) after receiving 9 bits from the next byte.

Slave reception

(1) First byte (slave address)

- Disable ACK data output set by the receive interrupt handling (ACKC is 0).
- Set the second byte of ACK data (prepare for next reception).
- Immediately after disabling SCL low hold (SWC9 bit is 0), enable the SCL low hold (SWC9 bit is 1) after receiving 9 bits from the second byte.

(2) From second byte on

- Set the next byte of ACK data (prepare for next reception).
- Immediately after disabling SCL low hold (SWC9 bit is 0), enable the SCL low hold (SWC9 bit is 1) after receiving 9 bits from the next byte.

8. Notes on Using UARTi Special Mode 1 (I^{2}C Mode)

8.1 Setting Procedure for UiTB Register Data at Slave Transmit/Receive

Write data to the UiTB register at slave transmit/receive according to the following procedure.

- When receiving the first byte data (slave address):

  (1) Write the second byte data to the UiTB register in the receive interrupt handling.

  (2) Write the third byte data to the UiTB register in the transmit interrupt handling.

- When receiving the second byte data onwards

  Each time a transmit interrupt handling occurs, write 1-byte data sequentially to the UiTB register starting with the fourth byte.

8.2 Electrical Characteristics

The electrical characteristic of the M16C Family and R8C Family and the I^{2}C-bus electrical specifications differ.

8.2.1 Low/High-level Input Voltage and Low-level Output Voltage

The low-level input voltage, high-level input voltage, and low-level output voltage may differ from the I^{2}C-bus specification.

Each MCU Series' electrical characteristics differ. Refer to individual hardware manuals for details.

M16C/62P Group
When V_{CC} = 2.7 to 5.5 V
High level input voltage (V_{IH}) = min. 0.8 V_{CC} (guaranteed value)
Low level input voltage (V_{IL}) = max. 0.2 V_{CC} (guaranteed value)
When V_{CC} = 5 V and I_{OL} = 5 mA
Low level output voltage (V_{OL}) = max. 2.0 V_{CC} (guaranteed value)

I^{2}C-bus specification
High level input voltage (V_{IH}) = min. 0.7 V_{CC}
Low level input voltage (V_{IL}) = max. 0.3 V_{CC}
When I_{OL} = 6 mA
Low level output voltage (V_{OL}) = max. 0.6 V

8.2.2 Set-up and Hold Time in When Detecting a Condition

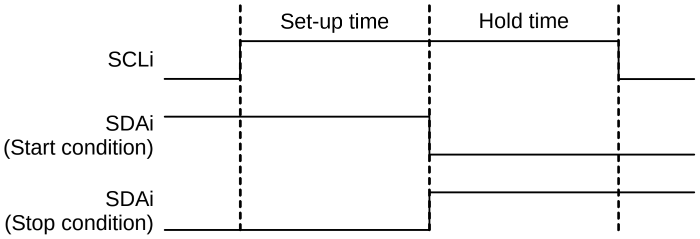
In slave mode, set-up and hold times in the start and stop condition detection may differ from the I^2C-bus specification.

In the I^2C-bus specification, Fast-mode set-up and hold times in the start and stop condition are a minimum of 600 ns (minimum of 4.0 µs in Standard-mode). In contrast, the set-up and hold times of the M16C Family are a minimum of six cycles of the UiBRG count source (the cycle count depends on the MCU type).

In Fast-mode, the minimum time is 600 ns and is within the I^2C-bus specification when using the UiBRG count source operated at 10 MHz, but it becomes out of spec when operated at less than 10 MHz.

In Standard-mode, the minimum time is 4.0 µs and is within the I^2C-bus specification when using the UiBRG count source operated at 1.5 MHz, but it becomes out of spec when operated at less than 1.5 MHz.

6 cycles of UiBRG count source [(1)] < Set-up time
6 cycles of UiBRG count source [(1)] < Hold time



Note:
1. The number of cycles depends on MCU type. Refer to the hardware manual for details.

Figure 8.1 Set-up and Hold Times When Detecting Start and Stop Conditions

8.2.3 Set-up and Hold Times When Generating a Condition

When generating a start condition, the hold time (t_{HD}:STA) is a half cycle of the SCL clock. When generating a stop condition, the set-up time (t_{SU}:STO) is a half cycle of the SCL clock.
When the SDA digital delay function is enabled, delay time must be taken into consideration (see 3.3.3 ?DL2 to DL0 Bit Settings (SDA Digital Delay)?).
The following shows a calculation example of hold and set-up times when generating a condition.

Calculation example when setting 100 kbps
- UiBRG count source: f1 = 20 MHz
- UiBRG register setting value: n = 100 - 1
- SDA digital delay setting value: DL2 to DL0 are 101b (5 or 6 cycles of UiBRG count source)

f_{SCL} (theoretical value) = f1 / (2(n+1)) = 20 MHz / (2 × (99 + 1)) = 100 kbps
t_{DL} = delay cycle count / f1 = 6 / 20 MHz = 0.3 μs
$t_{HD:STA}$ (theoretical value) = 1 / (2f_{SCL} (theoretical value)) = 1 / (2 × 100 kbps) = 5 μs
$t_{SU:STO}$ (theoretical value) = 1 / (2f_{SCL} (theoretical value)) = 1 / (2 × 100 kbps) = 5 μs
$f_{HD:STA}$ (actual value) = $t_{HD:STA}$ (theoretical value) - t_{DL} = 5 μs - 0.3 μs = 4.7 μs
$f_{SU:STO}$ (actual value) = $t_{SU:STO}$ (theoretical value) - t_{DL} = 5 μs - 0.3 μs = 5.3 μs

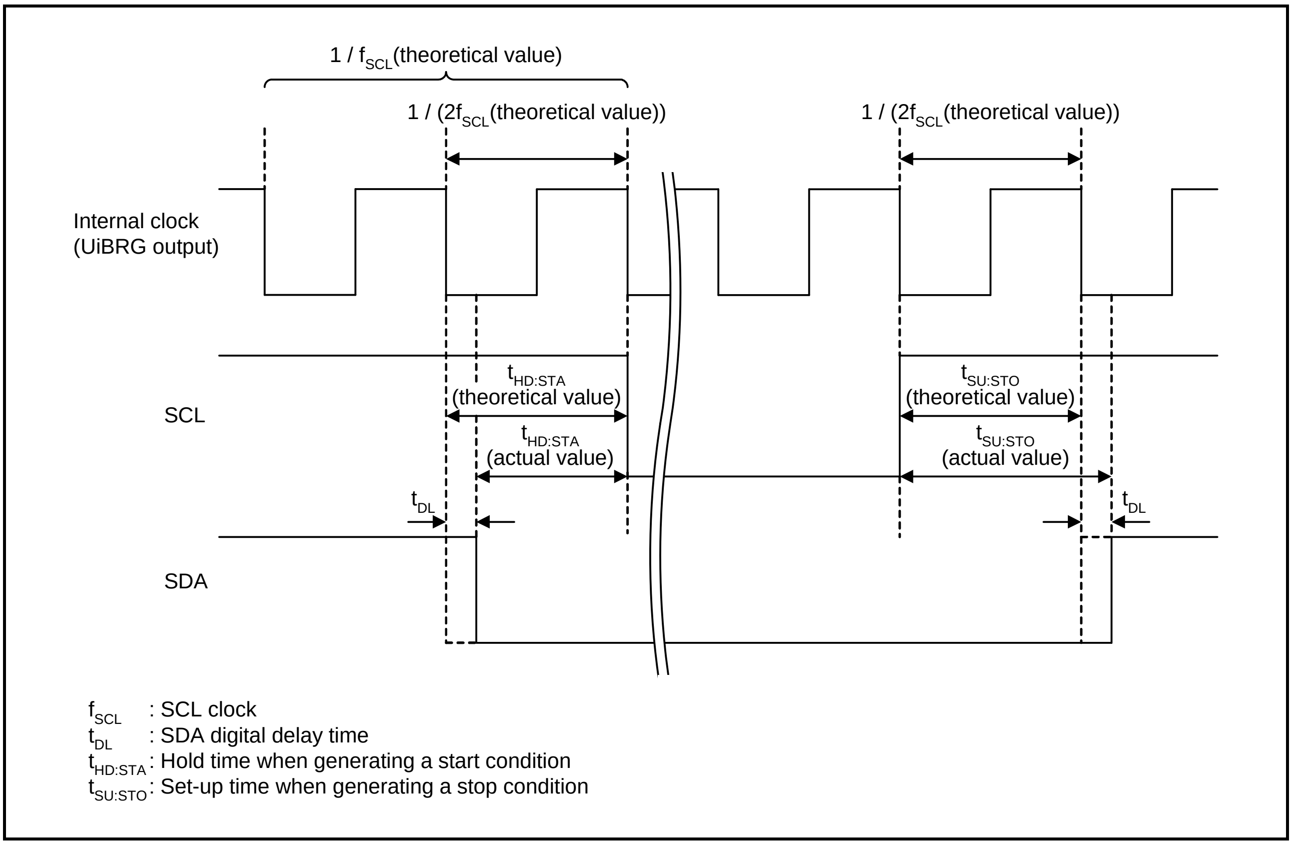

1 / f_{SCL}(theoretical value)
1 / (2f_{SCL}(theoretical value))
1 / (2f_{SCL}(theoretical value))
Internal clock
(UiBRG output)
SCL
SDA
$t_{HD:STA}$
(theoretical value)
$t_{HD:STA}$
(actual value)
$t_{SU:STO}$
(theoretical value)
$t_{SU:STO}$
(actual value)
t_{DL}
t_{DL}
f_{SCL} : SCL clock
t_{DL} : SDA digital delay time
$t_{HD:STA}$: Hold time when generating a start condition
$t_{SU:STO}$: Set-up time when generating a stop condition

Figure 8.2 Set-up and Hold Times When Generating Start and Stop Conditions

8.3 Maximum Transfer Speed Using the UiBRG Count Source

The time necessary to recognize the SCL clock level is dependent on the sampling frequency, with a maximum of three clock cycles of the UiBRG count source. Therefore, the maximum transfer speed of an I^2C-bus connectable to the M16C Family and R8C Family is limited by the main clock frequency and speed of the UiBRG count source, which is selected by setting bits CLK1 and CLK0 in the UiC0 register. There is a possibility of bit slippage if not used with a transfer speed that meets the following conditions:

I^2C-bus interface maximum transfer speed (Hz) < UiBRG count source (Hz) / 3

Example 1: When the source frequency is 10 MHz and f32 is selected as the UiBRG count source:
Maximum transfer speed without bit slippage (Hz) < (10 MHz / 32) / 3 = 104 kbps
In this case, the maximum transfer speed of the I^2C-bus is 104 kbps.

Example 2: When the source frequency is 10 MHz and f8 is selected as the UiBRG count source:
Maximum transfer speed without bit slippage (Hz) < (10 MHz / 8) / 3 = 416 kbps
In this case, the maximum transfer speed of the I^2C-bus is 400 kbps (maximum value of Fast-mode).

8.4 Function Limitations

8.4.1 SWC2 bit in the UiSMR2 Register (SCL Wait)

In general, the SWC2 bit is not needed.
When the SWC2 bit is set to 1 (low-level output), the SCLi pin can be fixed low while transmitting/receiving. When the SWC2 bit is set to 0 (transmit/receive clock), the SCLi pin is released, and an SCL clock is output.

8.4.2 SDHI bit in the UiSMR2 Register (SDA Output Disable)

In general, the SDHI bit is not needed.
In slave mode, disable SDAi pin output when the slave address does not match. In this case, the SDAi pin becomes high-impedance when the UiTB register is set to 01FFh or when using the SDA output disable function.

The SDA output disable function changes the SDAi pin to a high-impedance state. Set the SDHI bit to 1 (disabled) to enable this function. When the SDHI bit is set to 0 (enabled), the value set to the UiTB register is output from the SDAi pin.

8.4.3 Restart Condition in Slave Mode

In I^2C mode, the restart condition detection in slave mode is not supported.

9. Reference Documents

M16C Family, R8C Family User's Manual: Hardware
The latest versions can be downloaded from the Renesas Electronics website.

Technical News/Technical Update
The latest information can be downloaded from the Renesas Electronics website.

Website and Support

Renesas Electronics website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry

Revision History	M16C Family, R8C Family I^{2}C-bus Interface Using UARTi Special Mode 1

Rev.	Date	Description	
		Page	Summary
1.00	Jan. 25, 2010	—	First edition issued
1.01	Sep. 1, 2010	—	R8C Family added
		1	Sentences revised
		3	3.2.3, Counting SCL Clock added and Figure 3.2 added
		14	6.2, the ninth bit (b8) revised as b8
		19	Setting Procedure for UiTB Register Data at Slave Transmit/Receive added as 8.1
1.02	Dec. 20, 2010	3	3.1 I^{2}C Mode Setting revised

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to one with a different part number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.0