32-bit RISC Microcontroller
Reference Manual

I^2C Interface Version A (EI2C-A)

Revision 1.3

2024-04

Toshiba Electronic Devices & Storage Corporation

© 2020-2024
Toshiba Electronic Devices & Storage Corporation

Contents

Preface 5
Related document 5
Conventions 6
Terms and Abbreviation 8
1. Outline 9
2. Configuration 10
3. Details of Function and Operation 11
3.1. Configuration of I²C-bus 11
3.2. Data Format 12
3.3. Functional Description 13
3.3.1. Serial Clock 13
3.3.2. Auto Selection of Master/Slave 15
3.3.3. Enable of I²C-bus 16
3.3.4. Generating START and STOP Conditions 17
3.3.5. Generation Repeated START Condition 19
3.3.6. Selection of Slave Address Match Detection and General Call Detection 19
3.3.7. Arbitration Lost Detection Monitor 21
3.3.8. Reception ACK Bit Monitor 23
3.3.9. Reception ACK Wait 23
3.3.10. Detection of Repeated START 23
3.3.11. Software Reset 24
3.3.12. Noise Filter 24
3.3.13. Timeout 25
3.3.14. One Shot SCL Output Function 25
3.3.15. Interrupt Service Request and Release 26
3.3.16. DMA Request Output Control 26
3.3.17. I²C-bus Monitor 26
3.4. Address Match Wakeup Function 27
3.4.1. Clock Stretch Function 27
3.4.2. Operation Flow of Address Match Wakeup 27
4. Register 29
4.1. Register List 29
4.2. Register Descriptions 30
4.2.1. *[I2CxARST]* (I2C Reset Register) 30
4.2.2. *[I2CxAEN]* (I2C Enable Register) 30
4.2.3. *[I2CxACR0]* (I2C Control Register 0) 31
4.2.4. *[I2CxACR1]* (I2C Control Register 1) 32
4.2.5. *[I2CxADBRT]* (I2C Transmission Data Buffer Register) 33
4.2.6. *[I2CxADBRR]* (I2C Reception Data Buffer Register) 33
4.2.7. *[I2CxASR0]* (I2C Status Register 0) 33
4.2.8. *[I2CxASR1]* (I2C Status Register 1) 34
4.2.9. *[I2CxAPRS]* (I2C Prescaler Clock Setting Register) 36

4.2.10. *[I2CxASCL]* (I2C SCL Width Setting Register) 36
4.2.11. *[I2CxAAR1]* (I2C 1st Slave Address Register) 37
4.2.12. *[I2CxAAR2]* (I2C 2nd Slave Address Register) 37
4.2.13. *[I2CxAIE]* (I2C Interrupt/DMA Setting Register) 38
4.2.14. *[I2CxAPM]* (I2C Bus Pin Monitor Register) 39
4.2.15. *[I2CSWUPCR1]* (I2C Wakeup Control Register 1) 40
4.2.16. *[I2CSWUPCR2]* (I2C Wakeup Control Register 2) 40
4.2.17. *[I2CSWUPCR3]* (I2C Wakeup Control Register 3) 40
4.2.18. *[I2CSWUPSL]* (I2C Status Register) 41
4.2.19. *[I2CSWUPCR4]* (I2C Wakeup Control Register 4) 41
4.2.20. *[I2CSWUPCR5]* (I2C Wakeup Control Register 5) 41
5. Usage Example 42
5.1. Data Transfer Procedure 42
5.1.1. Device Initialization 42
5.1.2. Master Transmission 43
5.1.3. Master Reception 45
5.1.4. Slave Transmission 50
5.1.5. Slave Reception 52
5.1.6. Repeated Start 53
5.2. Wakeup Operation and Setting Flow (Example) 54
6. Precaution for Usage 56
7. Revision History 57
RESTRICTIONS ON PRODUCT USE 58

List of figures

Figure 2.1 I²C Interface Version A Block Diagram 10
Figure 3.1 Configuration of I²C-bus 11
Figure 3.2 Data Format of I²C Interface Version A 12
Figure 3.3 Example of Clock Synchronization 15
Figure 3.4 Generating START Condition and Slave Address 17
Figure 3.5 Generating of STOP Condition 18
Figure 3.6 Change of General Call Detection 20
Figure 3.7 Arbitration Lost 21
Figure 3.8 Arbitration Lost Operation (Above-Mentioned Internal Flag Shows Master B) 22
Figure 3.9 Change of Reception ACK Bit Monitor 23
Figure 3.10 Repeated START Detection Flag 23
Figure 3.11 DNF Noise Removal Example with 1 PRSCK Width 24
Figure 3.12 Clock Stretch Function 27
Figure 3.13 Address Match Wakeup Function 28
Figure 5.1 Wakeup Initialize Setting 54
Figure 5.2 Flow After Wakeup 55

List of Table

Table 2.1 List of Signals 10
Table 3.1 Setting of <SCLH[7:0]>, <SCLL[7:0]>, <PRSCK[5:0]>, <DNF[2:0]> for Transfer Speed (Example) 14
Table3.2 Operation of *[I2CxASR0]*<TRX> in Each Mode 16
Table 3.3 List of Initializing Registers / Bits 24
Table 3.4 Interrupt Signal and Request 26
Table 7.1 Revision history 57

Preface

Related document

Document name
Product Information
Exception
Clock Control and Operation Mode

Conventions

- Numeric formats follow the rules as shown below:
  Hexadecimal: 0xABC
  Decimal: 123 or 0d123 - Only when it needs to be explicitly shown that they are decimal numbers.
  Binary: 0b111 - It is possible to omit the "0b" when the number of bits can be distinctly understood from a sentence.
- "_N" is added to the end of signal names to indicate low active signals.
- It is called "assert" that a signal moves to its active level, "deassert" to its inactive level.
- When two or more signal names are referred, they are described like as [m:n].
  Example: S[3:0] shows four signal names S3, S2, S1 and S0 together.
- The characters surrounded by ***[ ]*** defines the register.
  Example: ***[ABCD]***
- "N" substitutes suffix number of two or more same kind of registers, fields, and bit names.
  Example: ***[XYZ1]***, ***[XYZ2]***, ***[XYZ3]*** → ***[XYZn]***
- "x" substitutes suffix number or character of units and channels in the register list.
- In case of unit, "x" means A, B, and C, ...
  Example: ***[ADACR0]***, ***[ADBCR0]***, ***[ADCCR0]*** → ***[ADxCR0]***
- In case of channel, "x" means 0, 1, and 2, ...
  Example: ***[T32A0RUNA]***, ***[T32A1RUNA]***, ***[T32A2RUNA]*** → ***[T32AxRUNA]***
- The bit range of a register is written like as [m: n].
  Example: Bit[3: 0] expresses the range of bit 3 to 0.
- The configuration value of a register is expressed by either the hexadecimal number or the binary number.
  Example: ***[ABCD]***<EFG> = 0x01 (hexadecimal), ***[XYZn]***<VW> = 1 (binary)
- Word and byte represent the following bit length.
  Byte: 8 bits
  Half word: 16 bits
  Word: 32 bits
  Double word: 64 bits
- Properties of each bit in a register are expressed as follows:
  R: Read only
  W: Write only
  R/W: Read and write are possible.
- Unless otherwise specified, register access supports only word access.
- The register defined as "Reserved" must not be rewritten. Moreover, do not use the read value.
- The value read from the bit having default value of "-" is unknown.
- When a register containing both of writable bits and read-only bits is written, read-only bits should be written with their default value, In the cases that default is "-", follow the definition of each register.
- Reserved bits of the write-only register should be written with their default value. In the cases that default is "-", follow the definition of each register.
- Do not use read-modified-write processing to the register of a definition which is different by writing and read out.

All other company names, product names, and service names mentioned herein may be trademarks of their respective companies.

Terms and Abbreviation

Some of abbreviations used in this document are as follows:

ANF	Analog Noise Filter
DNF	Digital Noise Filter
EI2C	I^2C Interface Version A
INT	Interrupt
I2C	Inter-Integrated Circuit
I2CS	I^2C wake-up circuit from Stand-by mode
Fm	Fast-mode
Fm+	Fast-mode Plus
STD	Standard-mode

1. Outline

The I^2C interface version A can operate as a transceiver circuit of 1ch (SCL, SDA) in 1 unit circuit. The list of functions is shown below.

Function classification	Function	Functional description or range
Transmission speed control	Prescaler dividing selection	It is dividing about a prescaler clock to 1/1, 1/2, 1/3 to 1/30, 1/31, 1/64.
	Clock source	In master mode, SCL HIGH / LOW width can be selected and set individually
	Transfer rate	10kbits/s to 1Mbit/s (it corresponds to Fast-mode Plus (Fm+)) (fsys= 8 to 120MHz)
Communication format	I^2C-bus format	Selection of a master/slave is automatically. .
	Date length	8 bits
	Acknowledge	The existence of acknowledging can be chosen.
	START/STOP condition	Generating of START/STOP condition
	Slave address	7/10-bit addressing format is supported. 2 sets of slave addresses can be set up. (1st/2nd slave address) Detection of a general call is supported in slave mode.
Transmission and reception control	Arbitration	Multi-master Clock synchronization Existence selection of Arbitration lost detection is supported.
	WAIT function	WAIT function selectable after reception
	One shot SCL output	One shot clock of SCL can output
	Repeated START detection, generating	Detection of a repeated START of a bus line (at the time of slave mode) and generating (at the time of master mode) are supported.
	Noise cancellation	Digital, analog is selection
Interlocking control	Interrupt	Transmission buffer empty interrupt Reception buffer full interrupt Status interrupt (Factors: 11 type) (Transfer end detection, start condition detection, repeated start condition detection, stop condition detection, NACK detection, arbitration lost detection, general call detection, slave address match detection, error start condition detection, error stop condition detection, time out detection)
	DMA request	A setup according to transmission and reception
	Address match wakeup function	Slave address match detection can use the release factor for the Low-power consumption mode release.
Other	Time out detection function	16-bit timer counter to detect the SCL stop time.

Note1: It does not support HS (High-speed) mode, and a START byte.

Note2: There is a function in which it cannot support depending on products, such as slope control, I/O correspondence at the time of the power supply off, an Input voltage (VIH/VIL), and an output voltage (VOL=0.4 V, VDD>2 V,3 mA sink). Please refer to the "Electrical Characteristics" chapter of the Datasheet (DS) for details.

2. Configuration

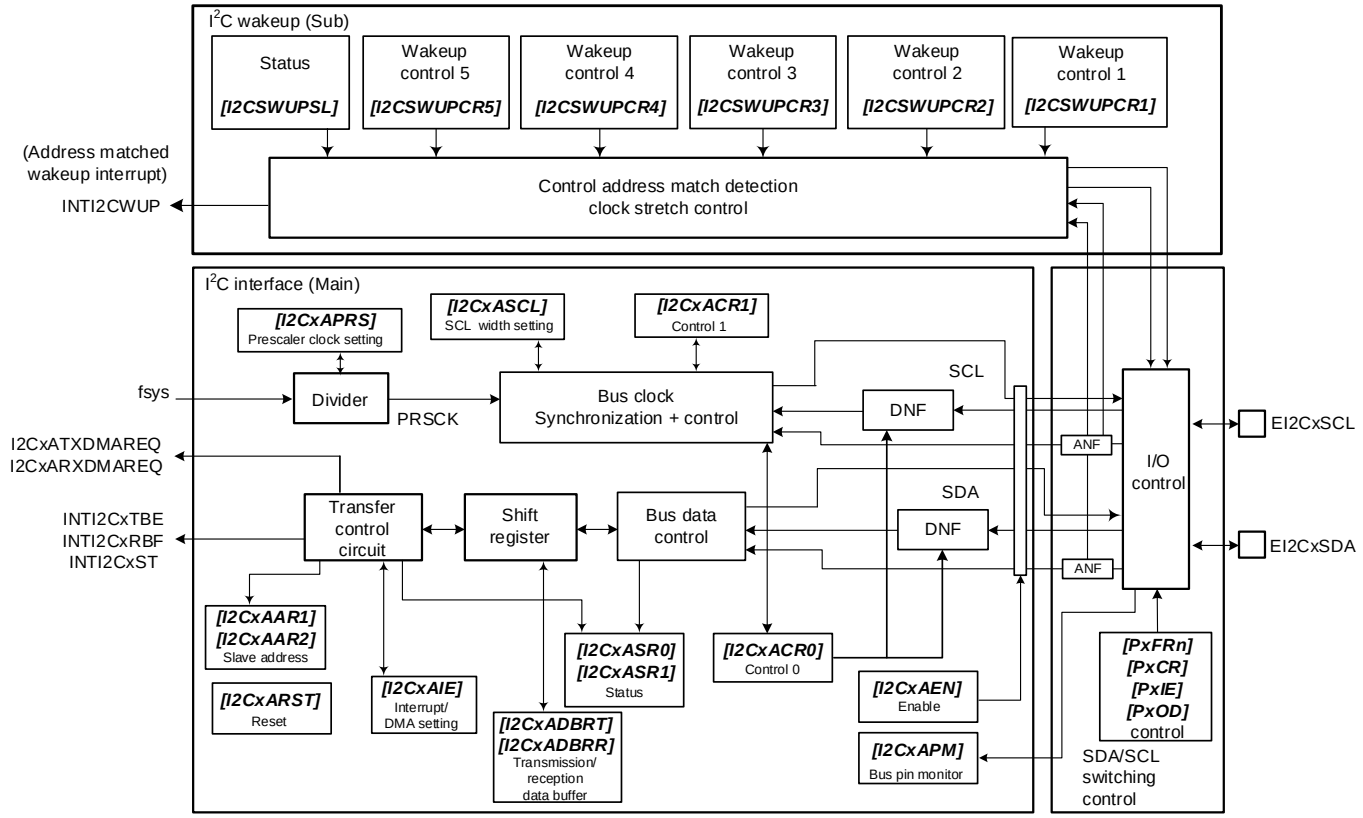


Figure 2.1 I²C Interface Version A Block Diagram

The circuit for the Wakeup is the expanded function. Analog NF (noise filter) and an expanded function, Since it does not implement depend on the specification of a product, refer to the datasheet of each product.

Table 2.1 List of Signals

No.	Symbol	Signal name	I/O	Related reference manual
1	fsys	System clock	Input	Clock Control and Operation Mode
2	EI2CxSCL	SCL signal	Input/Output	Product Information
3	EI2CxSDA	SDA signal	Input/Output	Product Information
4	INTI2CxTBE	I2C transmission buffer empty interrupt	Output	Exception
5	INTI2CxRBF	I2C reception buffer full interrupt	Output	Exception
6	INTI2CxST	I2C status interrupt	Output	Exception
7	INTI2CWUP	I2C wakeup interrupt	Output	Exception
8	I2CxATXDMAREQ	Transmission DMA request	Output	Product Information
9	I2CxARXDMAREQ	Reception DMA request	Output	Product Information

3. Details of Function and Operation

When I^{2}C interface is used, the corresponding clock enable bits should be set to "1" (Clock supply) in fsys supply stop register A (***[CGFSYSENA]*** and ***[CGFSYSMENA]***), fsys supply stop register B (***[CGFSYSENB]*** and ***[CGFSYSMENB]***), fsys supply stop register C (***[CGFSYSMENC]***), and fc supply stop register (***[CGFCEN]***).

The corresponding registers and the bit locations depend on a product. Some products do not have all registers. For the details, refer to "Clock Control and Operation Mode" in Reference manual.

3.1. Configuration of I^{2}C-bus

The I^{2}C-bus is connected to devices via the SDA and SCL pins and can communicate with multiple devices.

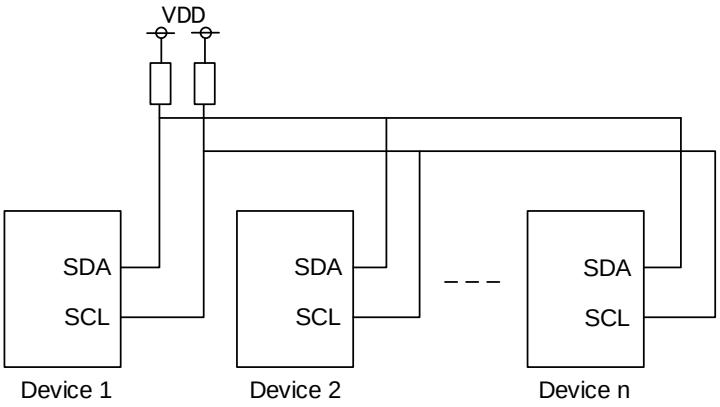


Figure 3.1 Configuration of I^{2}C-bus

This module operates as a master or slave device on the I^{2}C-bus. The master device drives the serial clock line (SCL) of the bus, send 8-bit address, and sends or receives data of 8 bits.

The slave device receives 8-bit addresses and sends or receives serial data of 8 bits in synchronization with the serial clock on the bus.

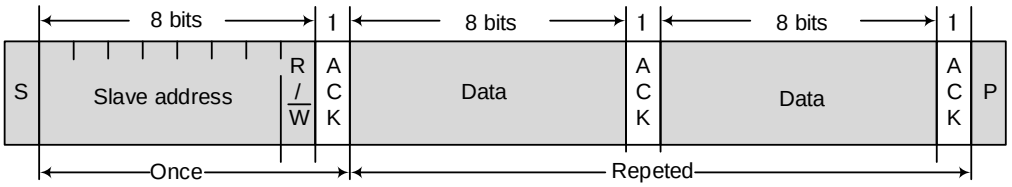
The device that operates as a receiver can output an acknowledge signal after reception of serial data and the device that operates as a transmitter can receive that acknowledge signal, regardless of whether the device is a master or slave. The master device can output a clock for the acknowledge signal.

In the multi-master mode in which multiple masters exist on the same bus, serial clock synchronization and arbitration lost to maintain consistency of serial data are supported.

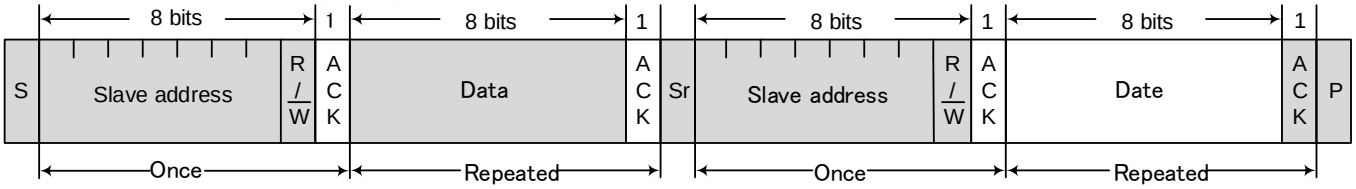
3.2. Data Format

Below shows the data formats used in the I²C interface version A.

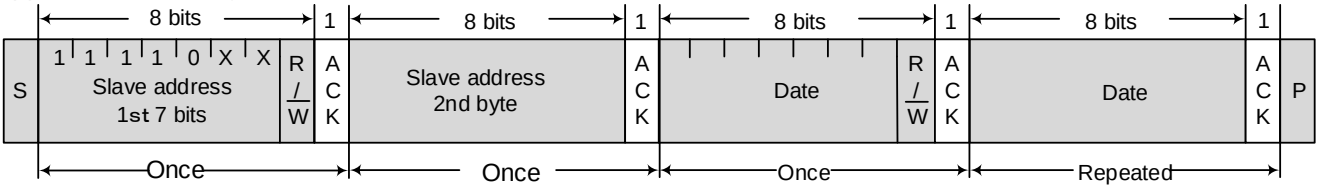
(a) 7-bit Addressing format



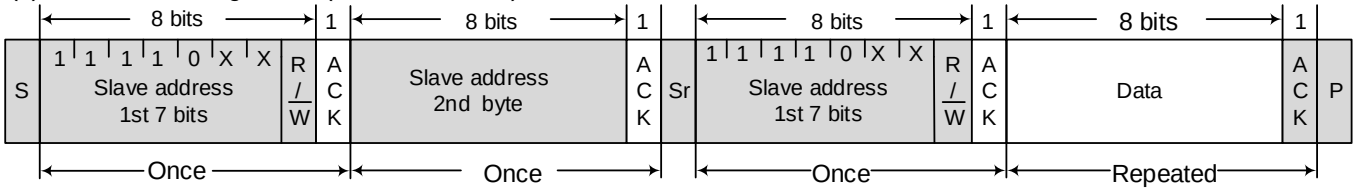
(b) 7-bit Addressing format (with repeated start condition)



(c) 10-bit addressing format



(d) 10-bit addressing format (Read from slave)



Note: S: Start condition
Sr: Repeated start condition
R/W: Direction bit
ACK: Acknowledge bit
P: Stop condition

Figure 3.2 Data Format of I²C Interface Version A

3.3. Functional Description

3.3.1. Serial Clock

(1) Reference clock

The reference clock of data transfer is a prescaler clock (PRSCK). The PRSCK is generated by dividing the operation clock. The division rate is set to 1/1, 1/2, 1/3, . . . , 1/63, or 1/64 by ***[I2CxAPRS]***<PRS[5:0]>.

Set the PRSCK clock width within the following range.

[I2CxACR0]<NFSEL> = 0:

(50ns / (<DNF[2:0]>+1)) ≤ clock width ≤ (t_{HIGH} (Min value by mode) / (<DNF[2:0]> + 2))

[I2CxACR0]<NFSEL> = 1:

clock width ≤ (t_{HIGH} (Min value by mode)) / 2

(2) Serial clock

Using the prescaler clock (PRSCK), the serial clock is generated by setting ***[I2CxASCL]***<SCLH[7:0]> and ***[I2CxASCL]***<SCLL[7:0]> to select HIGH time (hereafter, t_{HIGH}) and LOW time (hereafter, t_{LOW}), respectively.

The values of t_{HIGH} and t_{LOW} should be compliant with the corresponding standard.

The frequency of the serial clock is shown as follows. It depends on the type of the noise filter.

- Using the digital noise filer (***[I2CxACR0]***<NFSEL> = 0):

  t_{HIGH} (ns) = ((<SCLH[7:0]> + 2 + <DNF[2:0]>) × ((<PRS[5:0]> + 1) / fsys (MHz))) × 1000

  t_{LOW} (ns) = ((<SCLL[7:0]> + 2 + <DNF[2:0]>) × ((<PRS[5:0]> + 1) / fsys (MHz))) × 1000

- Using the analog noise filter (***[I2CxACR0]***<NFSEL> = 1):

  t_{HIGH} (ns) = ((<SCLH[7:0]> + 2) × ((<PRS[5:0]> + 1) / fsys (MHz))) × 1000

  t_{LOW} (ns) = ((<SCLL[7:0]> + 2) × ((<PRS[5:0]> + 1) / fsys (MHz))) × 1000

The transfer rate of the serial clock can be calculated as follows:

Serial clock rate: f_{SCL} (kHz) = (1 / (t_{HIGH} (ns) + t_{LOW} (ns) + tr (ns)+ tf (ns))) × 1000000

tr: Rise time of SCL

tf: Fall time of SCL

The tr and tf are shown bellow when each transfer speed of following Table 3.1 is calculated;

- When transfer speed is 100 kHz: tr = 1000 (ns), tf = 300 (ns)
- When transfer speed is 400 kHz: tr = 300 (ns), tf = 300 (ns)
- When transfer speed is 1000 kHz: tr = 120 (ns), tf = 120 (ns)

Table 3.1 Setting of <SCLH[7:0]>, <SCLL[7:0]>, <PRSCK[5:0]>, <DNF[2:0]> for Transfer Speed (Example)

[I2CxACR0] <NFSEL>	Transfer speed (kbps)	Operating frequency fsys (MHz)							
		20				40			
		<SCLL[7:0]>	<SCLH[7:0]>	<PRS[5:0]>	<DNF[2:0]>	<SCLL[7:0]>	<SCLH[7:0]>	<PRS[5:0]>	<DNF[2:0]>
0 (Digital)	100	91	77	0	0	91	77	1	0
	400	23	9	0	0	23	9	1	0
	1000	8	3	0	0	7	3	1	0

[I2CxACR0] <NFSEL>	Transfer speed (kbps)	Operating frequency fsys (MHz)							
		80				100			
		<SCLL[7:0]>	<SCLH[7:0]>	<PRS[5:0]>	<DNF[2:0]>	<SCLL[7:0]>	<SCLH[7:0]>	<PRS[5:0]>	<DNF[2:0]>
0 (Digital)	100	182	154	1	3	229	194	1	3
	400	98	42	0	3	59	24	1	3
	1000	7	16	1	1	19	7	1	3

[I2CxACR0] <NFSEL>	Transfer speed (kbps)	Operating frequency fsys (MHz)			
		120			
		<SCLL[7:0]>	<SCLH[7:0]>	<PRS[5:0]>	<DNF[2:0]>
0 (Digital)	100	184	156	2	1
	400	48	20	2	1
	1000	16	7	2	1

The serial clock rate may not be constant due to the clock synchronization function with other device output serial clock.

The t_{LOW} time is used to keep the setup time after release of the clock stretch in the slave mode, too. So the setting of <SCLL[7:0]> and <SCLH[7:0]> is necessary.

In master mode, the hold time when START conditions generated and the setup time when a STOP condition is generated are defined as the following.

Hold time ($t_{HD;STA}$): t_{HIGH}

Setup time ($t_{SU;STO}$): t_{HIGH}

Setup time of the Repeated START condition is as follows:

Setup time ($t_{SU;STA}$): t_{LOW}

(3) Clock synchronization

The I²C-bus is driven by using the wired-AND connection due to its pin structure. The first master that pulls its clock line to the "LOW" level overrides other masters producing the "HIGH" level on their clock lines. This must be detected and responded by the masters producing the "HIGH" level.

I2C has a clock synchronization function to ensure proper transfer operation even when multiple masters exist on a bus.

For example, the clock synchronization procedure for a bus with two masters is shown below.

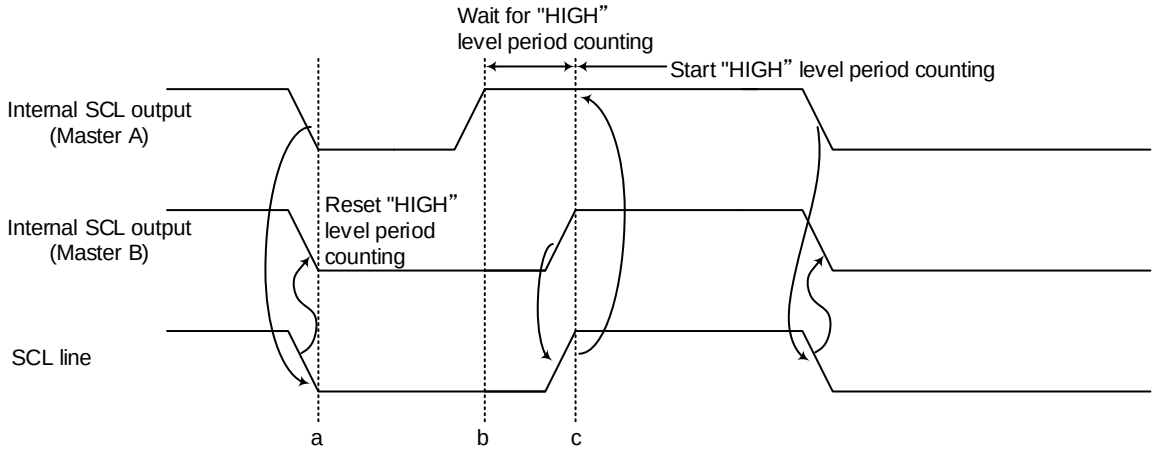


Figure 3.3 Example of Clock Synchronization

At the point a, Master A pulls its internal SCL output to the "LOW" level, bringing the SCL bus line to the "LOW" level.

Master B detects this transition, resets its "HIGH" level period counter, and pulls its internal SCL output level to the "LOW" level.

Master A completes counting of its "LOW" level period at the point b, and brings its internal SCL output to the "HIGH" level. However, Master B still keeps the SCL bus line at the "LOW" level, and Master A stops counting of its "HIGH" level period counting. After Master A detects that Master B brings its internal SCL output to the "HIGH" level and brings the SCL bus line to the "HIGH" level at the point c, it starts counting of its "HIGH" level period.

After that Master finishes counting the "HIGH" level period, the Master pulls the SCL pin to "LOW" and the SCL bus line becomes "LOW".

This way, the clock on the bus is determined by the master with the shortest "HIGH" level period and the master with the longest "LOW" level period among those connected to the bus.

3.3.2. Auto Selection of Master/Slave

This I²C module operates as a slave device at the initial setting in order to receive an address, because the system is supposed to be a multi-master one.

When ***[I2CxACR1]***<ST> is set to "1" at ***[I2CxASR0]***<BB> = 0, a START condition is generated, and the I2C module operates as a master device.

When the module operates as a master device and detects a STOP condition, an Arbitration Lost, an Error START condition, or an Error STOP condition, it changes to a slave device.

The current status can be checked by reading ***[I2CxASR0]***<MST>.

3.3.2.1. Selection of Transmitter/Receiver

When this module transmits an address in a master mode, it operates as a transmitter. When the direction bit is "1" in the transmission data, it changes to a receiver after an acknowledge returns from a slave. When the direction bit is "0", it continues to operate as a transmitter.

When an Arbitration Lost or an Error START condition is detected during transmitting an address, it changes to a slave receiver and continues to receive an address.

The mode of a slave device is determined by the direction bit from a master. When the direction bit is "0", it operates as a receiver. When it is "1", it operates as a transmitter.

The current status can be checked by reading ***[I2CxASR0]***<TRX>.

Table3.2 shows the transition condition of ***[I2CxASR0]***<TRX> in each mode and the value of ***[I2CxASR0]***<TRX> after the transition.

Table3.2 Operation of *[I2CxASR0]*<TRX> in Each Mode

Mode	Direction bit	Transition condition	TRX after transition
Slave	0	When the received slave address is the same as the value set in <SA> (<SA2>).	0
	1		1
Master	0	When ACK signal is returned.	1
	1		0

3.3.3. Enable of I²C-bus

When ***[I2CxAEN]***<I2CM> is set to "1", the I²C-bus (input and output signals of the SDA and SCL lines) is enabled.

In order to enable the I²C-bus, ***[I2CxAEN]***<I2CM> should be set to "1" after confirming the pin state is “HIGH”. And in order to change to the initial state, ***[I2CxAEN]***<I2CM> should be set to "0" after confirming the BUS free.

3.3.4. Generating START and STOP Conditions

3.3.4.1. START Condition

When ***[I2CxACR1]***<ST> is set to "1" during ***[I2CxASR0]***<BB> = 0, a START condition is generated on the bus.

After that, when ***[I2CxASR0]***<MST> and <TRX> are set to "1" and a START condition is detected, the detection flag ***[I2CxACR1]***<STCF> becomes "1", and then the module is in the WAIT state (SCL is pulled to "L"). It waits for a slave address transmission. (When an interrupt is enabled, the INTI2CxST interrupt is generated.)

Under the interrupt process and so on, write "1" to ***[I2CxASR1]***<STCF>, and then write the address and direction bit to ***[I2CxADBRT]***<DBT>.

And set ***[I2CxASR1]***<TBE> to "0" to output the address and direction bit on the bus.

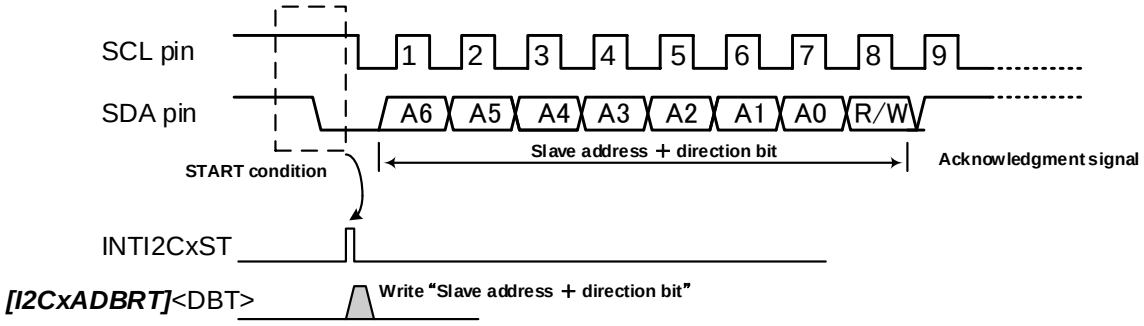


Figure 3.4 Generating START Condition and Slave Address

3.3.4.2. 10-bit Address (2nd byte)

After the first byte address is transmitted, ***[I2CxASR1]***<TBE> is set to "1". The module is in the WAIT state. When an interrupt is enabled, the INTI2CxST is generated.

When a remaining address is written to ***[I2CxADBRT]***<DBT> by an interrupt process and others, ***[I2CxASR1]***<TBE> is set to "0". Then an address transmission starts. When the data is transferred to the shift register, ***[I2CxASR1]***<TBE> is set to "1", which is different from the result of the first byte transmission.

3.3.4.3. Generating STOP Condition

When ***[I2CxACR1]***<SP> is set to "1" during ***[I2CxASR0]***<BB> and <MST> = 1, the sequence to generate a STOP condition starts on the bus. Then the STOP condition is generated on the bus.

When the SCL line in the bus has been pulled to "LOW" level by another device and the STOP condition is supposed to be generated, the STOP condition is actually generated after the SCL line is released.

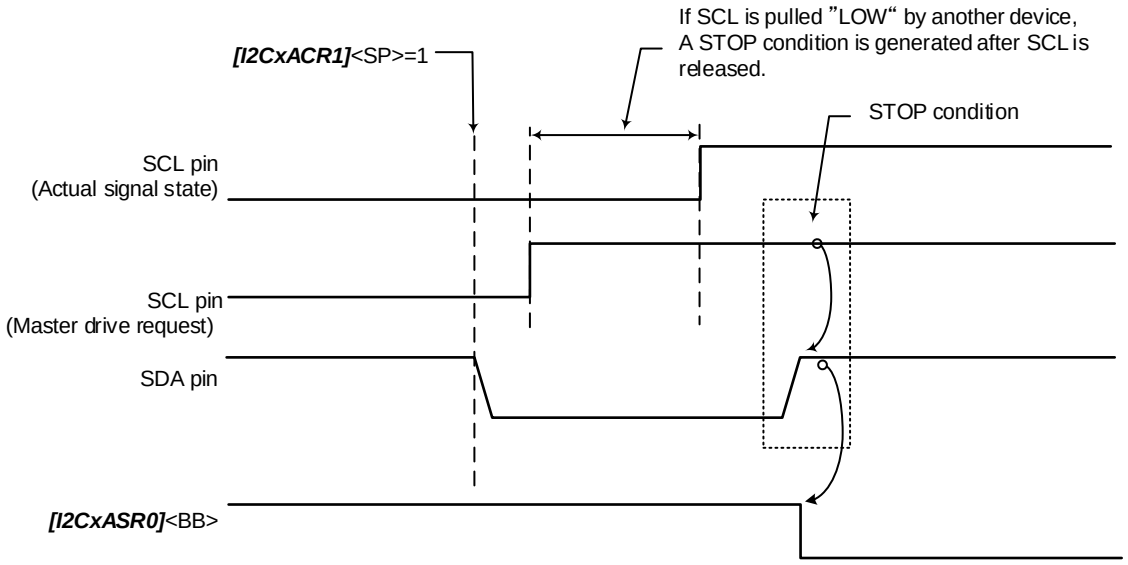


Figure 3.5 Generating of STOP Condition

The status of the bus can be checked by reading ***[I2CxASR0]***<BB>. ***[I2CxASR0]***<BB> is set to "1" when a START condition is detected on the bus (Bus busy state). It is cleared to "0" when a STOP condition is detected (Bus free state).

3.3.4.4. Error Detection

This product has a built-in detection function that detects illegal formats during communication.

(1) START condition

When ***[I2CACR0]***<ESTE> = 1 is set, the illegal formats detection of START condition is enable.

When a START condition is detected during data transfer, the Error START condition detection flag ***[I2CxASR1]***<EST> is set to "1", which shows an illegal format data is output on the bus. If an interrupt is enabled, the INTI2CxST is generated.

When an error is detected, the current process is suspended and the operation of the module changes to address reception. When an Error START condition is detected, the Repeated START condition detection flag ***[I2CxASR1]***<RSCF> is not set.

(2) STOP condition

When ***[I2CxACR0]***<ESPE> = 1 is set, the illegal formats detection of STOP condition is enable.

When a STOP condition is detected during data transfer, the Error STOP condition detection flag ***[I2CxASR1]***<ESP> is set to "1", which shows an illegal format data is output on the bus. When an interrupt is enabled, the INTI2CxST is generated.

When an error is detected, the data transfer is stopped as in the case of detecting a normal STOP condition.

When an Error STOP condition is detected, the STOP condition detection flag ***[I2CxASR1]***<SPCF> is not set.

After changed the condition detection flag to "0", detection function for Start and Stop condition is disable.

3.3.5. Generation Repeated START Condition

When ***[I2CxASR0]***<MST> and <BB> have been set to "1" and ***[I2CxACR1]***<RS> is set to "1", the SDA line is released to generate a Repeated START condition. Then a Repeated START condition is generated on the bus. When the SCL line of the bus has been pulled to "LOW" level by another device, the Repeated START condition is actually generated when the t_{LOW} interval elapses after the SCL line is released.

3.3.6. Selection of Slave Address Match Detection and General Call Detection

It supports to check that a received slave address matches a general call address or one of the two slave addresses.

In this module can set two slave addresses, the first slave address and the second slave address, which correspond a 7-bit slave address and a 10-bit slave address, respectively.

When ***[I2CxAAR1]***<SA1E> is set to "1", the setting and the match can be done for the first slave address. When a received address matches the slave address, ***[I2CxASR1]***<AAS1> is set to "1". When an interrupt is enabled, the INTI2CxST is generated.

The setting of the second slave address is also done by ***[I2CxAAR2]***<SA2E>. When the match is detected, ***[I2CxASR2]***<AAS2> is set to "1".

When the slave address does not match, a NACK is generated regardless of the value of ***[I2CxACR1]***<ACKSEL>. When the slave address matches, an ACK or a NACK is generated according to the value of ***[I2CxACR1]***<ACKSEL>. Even when the NACK is generated, the operation continues as the slave address has matched.

The operations in the cases of the 7-bit slave address and the 10-bit slave address are different for a Repeated START condition and an Error START condition.

For the 7-bit slave address, the same operation as the operation for a START condition is done. For the 10-bit slave address, only the first byte of the address is used to check the match, supposing at start that the slave address matches. So, when the slave address does not match, the address never matches. And the mismatch occurs from detecting the START condition till setting ***[I2CxASR1]***<AAS1>.

3.3.6.1. 10-bit Slave Address Match

When ***[I2CxAAR1]***<SAFS1> is set to "1" and a slave address is set to ***[I2CxAAR1]***<SA1[9:0]>, the 10-bit slave address can be used.

The reception operation of the first byte address after detecting a START condition is the same as in the case of the 7-bit address reception. When the received address is "0b11110xx0" (xx=***[I2CxAAR1]***<SA[9:8]>), an ACK is generated and the second byte address is received.

"0b11110xx" is prohibited from using as a slave address because it is a reserved word.

3.3.6.2. General Call Detection

When ***[I2CxACR0]***<GCE> is set to "1", a General call match can be detected. The operation of the match detection is the same as in the case of the slave address match. When a General call is detected, ***[I2CxASR1]***<GC> is set to "1". When an interrupt is enabled, the INTI2CxST is generated.

After detecting a General call, the operation of the normal slave reception is done.

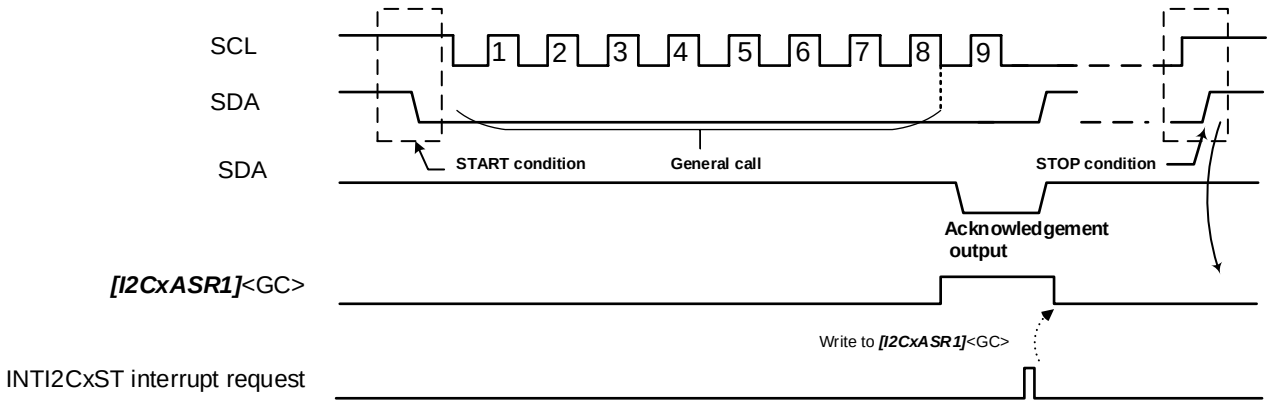


Figure 3.6 Change of General Call Detection

3.3.7. Arbitration Lost Detection Monitor

The I^2C-bus has the multi-master capability (there are two or more masters on a bus), and requires the bus arbitration procedure to ensure correct data transfer.

The I^2C-bus arbitration takes place on the SDA line.

The arbitration procedure for two masters on a bus is shown below.

Up until the “point a”, Master A and Master B output the same data. At the “point a”, Master A outputs the "LOW" level and Master B outputs the "HIGH" level.

Then Master A pulls the SDA bus line to the "LOW" level because the line has the wired-AND connection.

When the SCL line goes high at the “point b”, the slave device reads the SDA line data, i.e., data transmitted by Master A. At this time, data transmitted by Master B becomes invalid.

This condition of Master B is called "Arbitration Lost". Master B releases its SDA pin, so that it does not affect the data transfer initiated by another master. If two or more masters have transmitted exactly the same first data word, the arbitration procedure continues with the second data word.

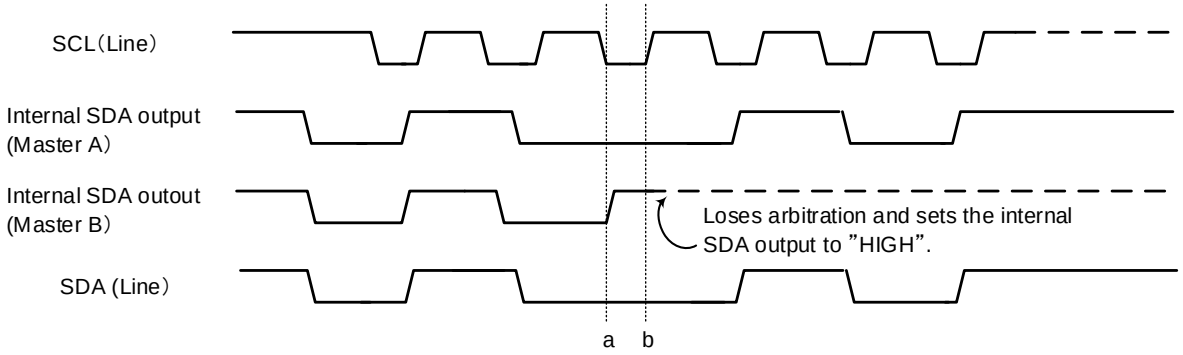


Figure 3.7 Arbitration Lost

A slave device does not detect an Arbitration Lost.

When ***[I2CxACR0]***<ALE> is set to "1" in a master device, an Arbitration Lost is generated under the following conditions. The SCL and SDA lines are released.

1. SDA = 0 or ***[I2CxASR0]***<BB> = 1 is detected when a START condition is generated.
2. SDA = 0 is detected when a transmitter transmits "1" in an address.
3. SDA = 0 is detected when a transmitter transmits "1" in data.
4. SDA = 0 is detected when a receiver transmits a NACK.

When the Arbitration Lost is generated, the successive operation depends on corresponding conditions, as shown below.

1. ***[I2CxACR1]***<ST> is cleared to "0". This module enters the slave mode, and checks the match of a received address.
2. When a 7-bit slave address or the first byte of a 10-bit slave address is transferred, this module enters the slave mode, and checks the match of the received address.

   When the first byte of a 10-bit slave address has matched and the second byte address is transferred, this module enters the slave mode, and checks the match of the received address. If the first byte does not match, this module enters the WAIT state.
3. ***[I2CxACR1]***<RS> and <SP> are cleared to "0", and this module enters the WAIT state.

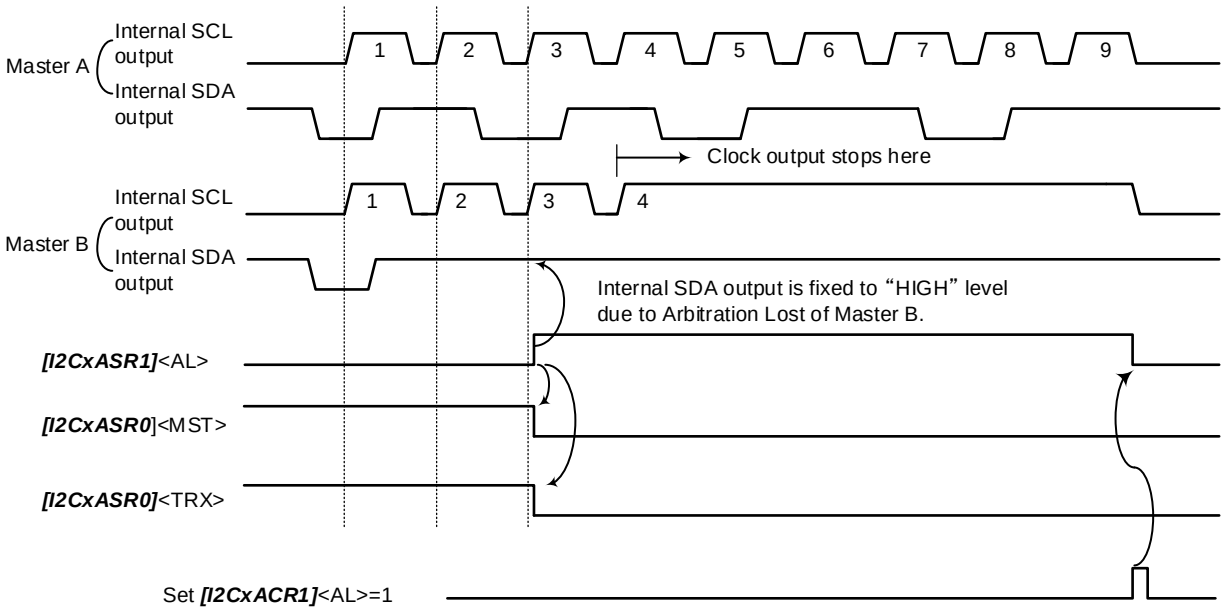


Figure 3.8 Arbitration Lost Operation (Above-Mentioned Internal Flag Shows Master B)

3.3.8. Reception ACK Bit Monitor

The 9th bit of transferred data is stored to ***[I2CxASR0]***<ACKF> at the rise edge of the SCL line on the bus. This value shows the status of the acknowledge bit.

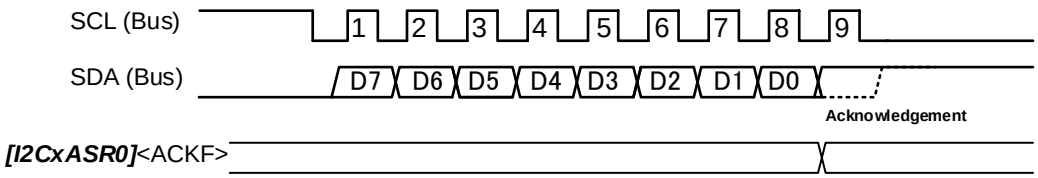


Figure 3.9 Change of Reception ACK Bit Monitor

3.3.9. Reception ACK Wait

When ***[I2CxACR1]***<ACKWAIT> = 1 has been set in the data reception mode, this module enters the WAIT state after receiving 8-bit data.

Writing ***[I2CxACR1]***<ACKSEL> releases the WAIT state.

Using this function, after reception data is read from ***[I2CxADBRR]***, the ACK/NACK can be controlled by setting ***[I2CxACR1]***<ACKSEL>.

3.3.10. Detection of Repeated START

When a Repeated START condition is detected on the bus during I2C operation, ***[I2CxASR1]***<RSCF> is set to "1". The Repeated START is used to change the direction of data transfer when a master device does not complete the data transfer from a slave device, and others.

When <RSCF> = 1 is set, this module enters the WAIT state of transmission address setting. To transmit an address, it is necessary that this bit is cleared and valid data exists in the transmission buffer.

<RSCF> is not set when an Error START condition is detected.

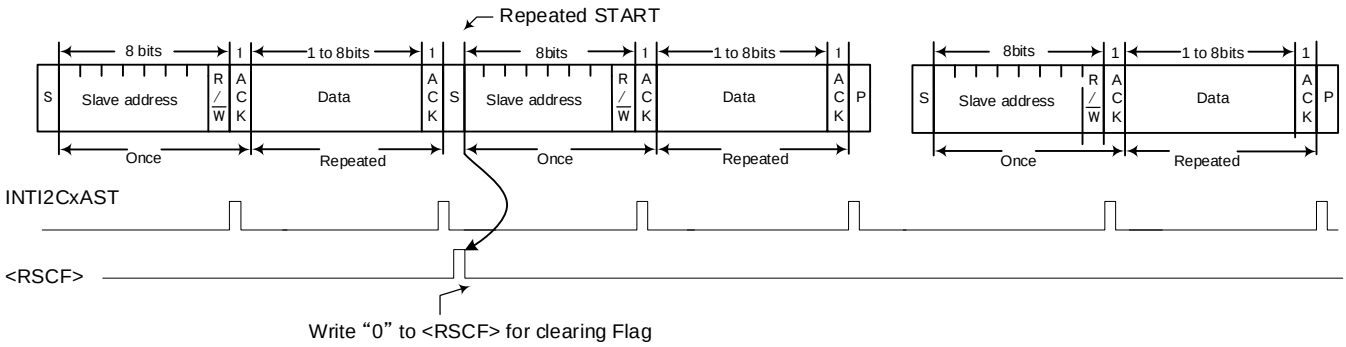


Figure 3.10 Repeated START Detection Flag

3.3.11. Software Reset

The I²C interface has a software reset function to initialize its functions. When the I2C is locked due to noise and others, the I2C can be initialized by this function.

The software reset is asserted when "10" is written to ***[I2CxARST]***<SWRES[1:0]> and "01" is written successively.

When the software reset is asserted, the I2C registers and bits except Data registers are initialized, as shown in the following table.

Table 3.3 List of Initializing Registers / Bits

Registers / <bit>	
[I2CxAEN]<I2CM>	***[I2CxASCL]***
[I2CxACR0]	***[I2CxAAR1]***
[I2CxACR1]	***[I2CxAAR2]***
[I2CxASR0]	***[I2CxAIE]***
[I2CxASR1]	***[I2CxAPM]***<SDAOUT><SCLOUT>
[I2CxAPRS]	

3.3.12. Noise Filter

The I²C interface integrates noise cancellers for the SCL and the SDA pins. The internal SCL and SDA signals are selected by ***[I2CxACR0]***<NFSEL> between the external signals via an external analog filter and the directly input signals on the corresponding ports.

When ***[I2CACR0]***<NFSEL> is set to "0", the SCL and SDA signals on the port are input to the built-in digital noise filter. The stage count of the filter is set to ***[I2CxACR0]***<DNF>. The output signals of the filter are used as the internal signals.

The noise width can be selected from among 1 to 6 times of the PRSCK width by the <DNF> setting.

The noise width which is eliminated by the noise filter should be less than the HIGH/LOW widths of the SCL signals.

In the standard of Fm/Fm+ mode, the pulse of the width of less than 50 ns maximum should be eliminated.

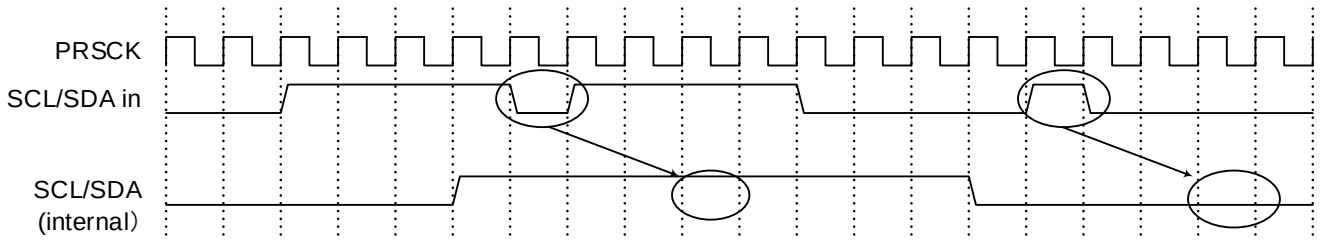


Figure 3.11 DNF Noise Removal Example with 1 PRSCK Width

3.3.13. Timeout

The timeout function detects the lock of the bus (the SCL signal stays at "HIGH" or "LOW" level and does not change). It operates in both a master and a slave device.

When ***[I2CxACR0]***<TOE> is set to "1", the timeout function is enabled. When a START condition is generated and the Bus busy ***[I2CxASR0]***<BB> is set to "1", the built-in 16-bit timer starts to operate.

The timer counts up while the SCL signal does not change. When a rising edge or a falling edge of the SCL signal is detected, the counter is cleared to "0". When the value in ***[I2CxACR0]***<TOPER> and the counter value match, the Timeout detection flag bit ***[I2CxASR1]***<TOERR> is set to "1".

When ***[I2CxAIE]***<INTTOE> is set to "1", an interrupt is enabled, the Status interrupt INTI2CxST is generated at the next fsys timing after the Timeout detection flag bit is set to "1".

The detection time can be calculated with the following formula.

Timeout [ms] = (Setting value of <TOPER[1:0]>) × (<PRSCK> / (fsys (MHz) × 1000))

Example: <TOPER[1:0]> = 11, <PRSCK> = 2 division, and fsys = 80 MHz:

Timeout [ms] = 2^16 × (2 / (80 × 1000)) = 1.64 ms

3.3.14. One Shot SCL Output Function

There is the case that a master device requests to generate a STOP condition but the STOP condition cannot be generated because a slave device pulls the SDA line to "LOW" level. This function generates a SCL pulse to release the "LOW" level generated by the slave device.

When ***[I2CxACR1]*** <OMC> = 1 is set, SCL is output once.

(***[I2CxACR1]*** = 0x0400)

To generate a stop condition after using One shot SCL output, set ***[I2CxACR1]*** <OMC> <SP> = 1 at the same time.

(***[I2CxACR1]*** = 0x0404)

In case SDA is not released when stop condition is generated, the stop condition does not occur and waits for detection of stop condition. In this case, SCL single output can be performed again to release SDA.

It is supposed that this function is used to generate a STOP condition after the release of the SDA line. So it cannot be used except generating a SCL pulse and a successive STOP condition. It cannot be used during data transfer, either.

While the One shot SCL pulse is generated, the detection of an Arbitration Lost becomes disabled. And ***[I2CxASR0]***<BC> does not change, either.

3.3.15. Interrupt Service Request and Release

The I²C interface has 3 interrupts; Transmission Buffer empty interrupt (INTI2CxTBE), Reception Buffer full interrupt (INTI2CxRBF) and Status interrupt (INTI2CxST).

The Status interrupt has 11 factors.

Table 3.4 Interrupt Signal and Request

Interrupt signal	Interrupt request	Detection of interrupt request	Enable/disable setting of interrupt output
INTI2CxTBE	Transmission buffer empty	Always	Always
INTI2CxRBF	Reception buffer full	Always	Always
INTI2CxST	Transfer end detection	Always	Always
	Start condition detection	Always	Selectable
	Repeated Start condition detection	Always	Selectable
	Stop condition detection	Always	Selectable
	General call detection	Selectable	Selectable
	Slave address Match detection	Selectable	Selectable
	NACK detection	Selectable	Selectable
	Arbitration Lost detection	Selectable	Selectable
	Time out detection	Selectable	Selectable
	Error Start condition detection	Selectable	Selectable
	Error Stop condition detection	Selectable	Selectable

3.3.16. DMA Request Output Control

When ***[I2CxAIE]***<DMARX> is set to "1", a reception DMA request is generated at detecting a Reception buffer full. When ***[I2CxAIE]***<DMATX> is set to "1", a transmission DMA request is generated at detecting a Transmission buffer empty.

The DMA request is asserted at the next fsys timing after a Reception buffer full or a Transmission buffer empty has been detected and the corresponding flag is set in the Detection status register. It keeps "HIGH" level till the DMA clear signal is asserted. When the DMA clear signal is asserted or ***[I2CxAIE]***<DMAxx> is set to "0", the DMA request becomes "LOW" level.

If the set and clear of the DMA request are generated at the same time, the clear is prioritized.

3.3.17. I²C-bus Monitor

The I²C interface integrates the Bus monitor function which can check the current status of the I²C-bus using the ***[I2CxAPM]*** register

[I2CxAPM]<SCL> and <SDA> show the status of the corresponding input pins which are selected by ***[I2CxACR0]***<NFSEL>.

It takes 2 cycles of PRSCK to reflect the values of an external I²C-bus to these bits.

[I2CxAPM]<SCLOUT> and <SDAOUT> show the status of the output pins of this interface.

3.4. Address Match Wakeup Function

This function is effective when the wakeup block (sub) is implemented.

When the I^2C block is used in slave mode, if the frame address on the I^2C-bus matches a self-address (slave address) in Low-power consumption mode, an interrupt (INTI2CWUP) occurs to clear Low-power consumption mode.

When using this function, before shifting to the Low-power consumption mode, stop the I^2C interface (main) under bus free.

The Main block (main) of I^2C Interface keeps the no System clock status except the IDLE mode of Low-power consumption.

3.4.1. Clock Stretch Function

After the MCU confirms the match of slave addresses in slave mode, it returns an ACK and then performs the clock stretch operation that pulls the SCL to "LOW".

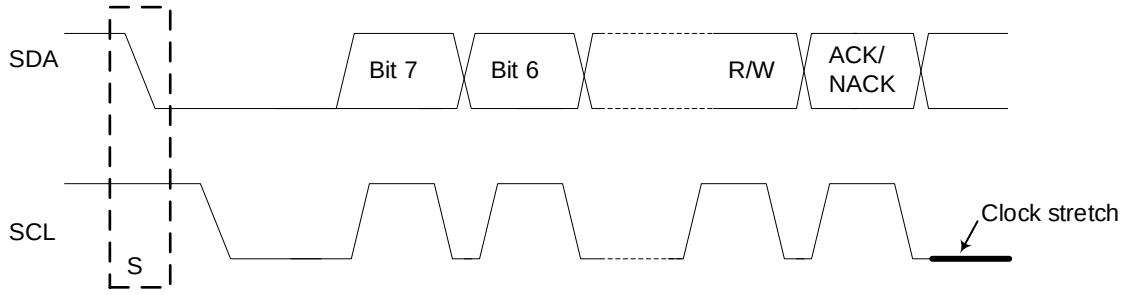


Figure 3.12 Clock Stretch Function

Note: When the MCU enters Low-power consumption mode during the clock stretch operation, the I^2C-bus maintains locked state, so that the address match wakeup function cannot be used.

3.4.2. Operation Flow of Address Match Wakeup

The flow from address Match detection to interrupt generating, the release of a Low-power consumption mode, and clock stretch release are shown in a figure.

(1) When the MCU detects an address match, the MCU returns an ACK to generate INTI2CWUP.

(2) Low-power consumption mode is released, and the MCU starts the clock stretch function.

(3) In the interrupt service routine after Low-power consumption mode is released, the interrupt request is cleared with ***[I2CSWUPCR1]*** <INTEND>.

(4) The clock stretch function is released with ***[I2CSWUPCR1]***<I2RES>.

When ***[I2CSWUPCR1]***<INTEND> is written from “1” to “0” continuously in interrupt service routine in the interrupt routine (3), the I2C information is transmitted to the main block from the wakeup block.

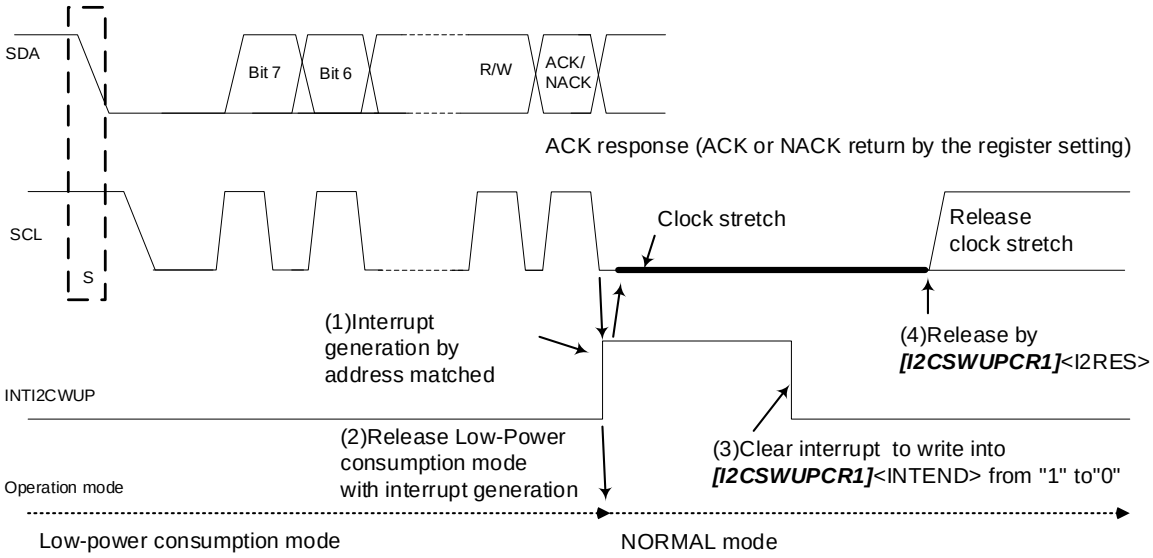


Figure 3.13 Address Match Wakeup Function

Note1: When the MCU performs the address match wakeup function, the I²C-bus is locked until the MCU returns to the Normal operation (Clock stretch).

Note2: The Master returns an ACK signal only, the slave device selects an ACK or NACK signal and then send it.

Note that the detection conditions for slave addresses and the General call must be set the same when the address match wakeup function is used.

- ***[I2CxAAR1]***<SA1[6:0]> = ***[I2CSWUPCR2]***<WUPSA1[6:0]> ; same 7bit address, detecting condition
- ***[I2CxAAR2]***<SA2[6:0]> = ***[I2CSWUPCR3]***<WUPSA2[6:0]> ; same 7bit address, detecting condition
- ***[I2CxAAR1]***<SA1[9:7]> = ***[I2CSWUPCR4]***<WUPSA1U[2:0]> ; same upper 3bit of 10bit address
- ***[I2CxAAR2]***<SA2[9:7]> = ***[I2CSWUPCR5]***<WUPSA2U[2:0]> ; same upper 3bit of 10bit address
- Setting contents of ***[I2CxCR0]***<GCE> & ***[I2CxACR1]***<ACKSEL> is as same as ***[I2CSWUPCR1]***<ACK>, <SGCDI>.

4. Register

4.1. Register List

The register and address of I2C are shown below.

Function name		Channel/unit	Base address		
			TYPE1	TYPE2	TYPE3
I²C Wakeup (Sub)	I2CS	–	0x4003E800	-	-
I²C Interface Version A (Main)	EI2C	ch0	0x400A5000	0x400D8000	0x400D1000
		ch1	0x400A6000	0x400D9000	0x400D2000
		ch2	0x400A7000	0x400DA000	0x400D3000
		ch3	0x400A8000	0x400DB000	0x400D4000
		ch4	0x400A9000	0x4000C000	0x400D5000

Note: The channel/unit and base address type are different by products. Please refer to "Product Information" of the reference manual for the details.

I²C Interface Version A (EI2C)

Register name		Address (Base+)
I2C Reset Register	***[I2CxARST]***	0x0000
I2C Enable Register	***[I2CxAEN]***	0x0004
I2C Control Register 0	***[I2CxACR0]***	0x0008
I2C Control Register 1	***[I2CxACR1]***	0x000C
I2C Transmission Date Buffer Register	***[I2CxADBRT]***	0x0010
I2C Reception Data Buffer Register	***[I2CxADBRR]***	0x0014
I2C Status Register 0	***[I2CxASR0]***	0x0018
I2C Status Register 1	***[I2CxASR1]***	0x001C
I2C Prescaler Clock Setting Register	***[I2CxAPRS]***	0x0020
I2C SCL Width Setting Register	***[I2CxASCL]***	0x0024
I2C 1st Slave Address Register	***[I2CxAAR1]***	0x0028
I2C 2nd Slave Address Register	***[I2CxAAR2]***	0x002C
I2C Interrupt /DMA Setting Register	***[I2CxAIE]***	0x0030
I2C Bus Pin Monitor Register	***[I2CxAPM]***	0x0034

x" is channel number.

The register of each channel is the same composition.

I²C Wakeup (I2CS)

Register name		Address (Base+)
I2C Wakeup Control Register 1	***[I2CSWUPCR1]***	0x0000
I2C Wakeup Control Register 2	***[I2CSWUPCR2]***	0x0001
I2C Wakeup Control Register 3	***[I2CSWUPCR3]***	0x0002
I2C Wakeup Status Register	***[I2CSWUPSL]***	0x0003
I2C Wakeup Control Register 4	***[I2CSWUPCR4]***	0x0004
I2C Wakeup Control Register 5	***[I2CSWUPCR5]***	0x0005

Note: The Above mentioned register can serve as only byte access.

4.2. Register Descriptions

4.2.1. *[I2CxARST]* (I2C Reset Register)

Bit	Bit symbol	After reset	Type	Function
31:2	-	0	R	Read as "0".
1:0	SWRES[1:0]	00	W	Software reset generation control The reset is asserted by the successive writes of 10 --> 01.

4.2.2. *[I2CxAEN]* (I2C Enable Register)

Bit	Bit symbol	After reset	Type	Function
31:1	-	0	R	Read as “0”.
0	I2CM	0	R/W	I2C operation control 0: Disable (SCL/SDA input and output are disabled.) 1: Enable (SCL/SDA input and output are enabled.)

Note: Do not rewrite during I2C communication operation.

4.2.3. *[I2CxACR0]* (I2C Control Register 0)

Bit	Bit symbol	After reset	Type	Function
31:16	-	0	R	Read as “0”.
15	-	0	R	Read as “0”.
14:12	DNF[2:0]	000	R/W	Sets the noise elimination time of the digital noise filter 000: 1 x PRSCK width 100: 5 x PRSCK width 001: 2 x PRSCK width 101: 6 x PRSCK width 010: 3 x PRSCK width 110: Same as 000 011: 4 x PRSCK width 111: Same as 000
11	NFSEL	0	R/W	Noise filter selection 0: Digital noise filter 1: Analog noise filter
10:9	TOPER[1:0]	00	R/W	Sets the timeout detection time 00: 2^13 x PRSCK 01: 2^14 x PRSCK 10: 2^15 x PRSCK 11: 2^16 x PRSCK
8	TOE	0	R/W	Sets the timeout control 0: Disable (Detection is not done.) 1: Enable (Detection is done.)
7:5	-	0	R	Read as “0”.
4	ESPE	0	R/W	Error STOP condition detection control 0: Not detected. 1: Detected.
3	ESTE	0	R/W	Error START condition detection control 0: Not detected. 1: Detected.
2	NACKE	0	R/W	Control of NACK detection at transmission 0: Not detected. 1: Detected.
1	GCE	0	R/W	General call detection control 0: Not detected. 1: Detected.
0	ALE	0	R/W	Arbitration Lost detection control 0: Not detected. 1: Detected.

Note: Do not rewrite during I2C communication operation.

4.2.4. *[I2CxACR1]* (I2C Control Register 1)

Bit	Bit symbol	After reset	Type	Function
31:16	-	0	R	Read as “0”.
15:11	-	0	R	Read as “0”.
10	OMC	0	R/W	One shot SCL output 0: No One shot SCL output is generated. 1: One shot SCL output is generated. After one shot SLC is generated, SCL becomes “0” automatically. Used in countermeasure process for an abnormal operation. If this function is used for normal communication, it causes malfunction. Write is disabled in the slave mode.
9:5	-	0	R	Read as “0”.
4	ACKWAIT	0	R/W	Selects Wait insertion position in the receiver mode. 0: Wait is not inserted. 1: After transmission of 8-bit data (before ACK/NACK)
3	ACKSEL	0	R/W	ACK/NACK output selection 0: ACK output 1: NACK output When <ACKWAIT>=1, the WAIT state is released by writing this bit. When a slave address does not match, NACK is output regardless of the setting value of this bit.
2	SP	0	R/W	STOP condition generation request 0: No request 1: Request is present. Write of ”1” is valid only when ***[I2CxASR0]***<MST> = 1. When <MST> = 0, this bit is ignored. This bit is cleared to ”0” by STOP condition detection, Arbitration Lost detection, and Error START/Error STOP condition detection.
1	RS	0	R/W	Repeated START condition generation request 0: No request 1: Request is present. Write of ”1” is valid only when ***[I2CxASR0]***<MST> = 1 and ***[I2CxAST0]***<BB> = 1. When <MST> = 0 or <BB> = 0, this bit is ignored. This bit is cleared to ”0” by Repeated START condition detection, Arbitration Lost detection, and Error START/Error STOP condition detection.
0	ST	0	R/W	START condition generation request 0: No request 1: Request is present. This bit is cleared to ”0” by START condition detection and Arbitration Lost detection.

4.2.5. *[I2CxADBRT]* (I2C Transmission Data Buffer Register)

Bit	Bit symbol	After reset	Type	Function
31:8	-	0	R	Read as “0”.
7:0	DBT[7:0]	0x00	R/W	Set transmission data

4.2.6. *[I2CxADBRR]* (I2C Reception Data Buffer Register)

Bit	Bit symbol	After reset	Type	Function
31:8	-	0	R	Read as “0”.
7	DBR[7:0]	0x00	R	Stored reception data

4.2.7. *[I2CxASR0]* (I2C Status Register 0)

Bit	Bit symbol	After reset	Type	Function
31:8	-	0	R	Read as “0”.
7:4	BC[3:0]	0000	R	Transfer bit count (Counts at the rising edge of SCL) 0000: WAIT state 0001: First bit transfer ⋮ 1000: 8th bit transfer 1001: ACK bit transfer These bits never take the value other than the above.
3	MST	0	R	Master/Slave selection monitor 0: Slave 1: Master
2	TRX	0	R	Transmission/Reception selection monitor 0: Receiver (Reception) 1: Transmitter (Transmission)
1	BB	0	R	Bus status monitor 0: Bus free 1: Bus busy This bit is set to ”1” by START condition detection. It is set to ”0” by STOP condition detection.
0	ACKF	0	R	Reception ACK bit 0: ACK 1: NACK The value for the last transfer data is kept.

4.2.8. *[I2CxASR1]* (I2C Status Register 1)

Bit	Bit symbol	After reset	Type	Function
31:14	-	0	R	Read as “0”.
13	TOERR	0	R	Timeout detection flag 0: Not detected 1: Detected
			W	Timeout detection flag clear 0: Invalid 1: Clear
12	ESP	0	R	Error STOP condition detection flag 0: Not detected 1: Detected
			W	Error STOP condition detection flag clear 0: Invalid 1: Clear
11	EST	0	R	Error START condition detection flag 0: Not detected 1: Detected
			W	Error START condition detection flag clear 0: Invalid 1: Clear
10	AAS2	0	R	Second slave address match detection flag 0: Not detected 1: Detected In the 10-bit address mode, this bit becomes ”1” when the lower 8 bits (the second byte) matches.
			W	Second slave address match detection flag clear 0: Invalid 1: Clear
9	AAS1	0	R	First slave address match detection flag 0: Not detected 1: Detected In the 10-bit address mode, this bit becomes ”1” when the lower 8 bits (the second byte) matches.
			W	First slave address match detection flag clear 0: Invalid 1: Clear
8	GC	0	R	General call match detection flag 0: Not detected 1: Detected
			W	General call match detection flag clear 0: Invalid 1: Clear
7	AL	0	R	Arbitration Lost detection flag 0: Not detected 1: Detected
			W	Arbitration Lost detection flag clear 0: Invalid 1: Clear
6	NACK	0	R	NACK detection flag 0: Not detected 1: Detected Detected only at data transmission.
			W	NACK detection flag clear 0: Invalid (Ignored) 1: Clear

Bit	Bit symbol	After reset	Type	Function
5	RBF	0	R	Reception buffer full detection flag 0: Not detected 1: Detected Not detected when a 7-bit slave address or a 10-bit slave address is received.
			W	Reception buffer full detection flag clear 0: Invalid (Ignored) 1: Clear This bit is cleared by reading ***[I2CxADBRR]***, too.
4	TBE	1	R	Transmission buffer empty detection flag 0: Not detected 1: Detected The detection is not done when a 7-bit slave address or the first byte of a10-bit slave address is transmitted. This bit is set to ”1” by START condition detection.
			W	Transmission buffer empty detection flag clear 0: Invalid (Ignored) 1: Clear This bit is cleared by writing ***[I2CxADBRT]***, too.
3	TEND	0	R	Transfer end detection flag 0: No detection 1: Detected
			W	Transfer end detection flag clear 0: Invalid (Ignored) 1: Clear
2	SPCF	0	R	STOP condition detection flag 0: Not detected 1: Detected This bit is not set to ”1” when Error STOP condition is detected.
			W	STOP condition detection flag clear 0: Invalid (Ignored) 1: Clear
1	RSCF	0	R	Repeated START condition detection flag 0: Not detected 1: Detected This bit is not set to ”1” when Error START condition is detected.
			W	Repeated START condition detection flag clear 0: Invalid (Ignored) 1: Clear When a Repeated START condition has been generated and this bit is set to ”1”, the module is in the WAIT state to set a transmission address. It is necessary that this bit is cleared and valid data exists in the transmission data buffer, to transmit the address.
0	STCF	0	R	START condition detection flag 0: Not detected 1: Detected
			W	START condition detection flag clear 0: Invalid (Ignored) 1: Clear When a START condition has been generated and this bit is set to ”1”, the module is in the WAIT state to set a transmission address. It is necessary that this bit is cleared and valid data exists in the transmission data buffer, to transmit the address.

4.2.9. *[I2CxAPRS]* (I2C Prescaler Clock Setting Register)

Bit	Bit symbol	After reset	Type	Function
31:6	-	0	R	Read as “0”.
5	PRS[5:0]	000000	R/W	Selects the frequency of the prescaler clock (PRSCK) to generate the serial clock. 000000: fsys/1 000001: fsys/2 ⋮ 111110: fsys/63 111111: fsys/64

Note: Do not rewrite during I2C communication operation.

4.2.10. *[I2CxASCL]* (I2C SCL Width Setting Register)

Bit	Bit symbol	After reset	Type	Function
31:16	-	0	R	Read as “0”.
15:8	SCLL[7:0]	0x00	R/W	SCL LOW width setting When ***[I2CxACR0]***<NFSEL> = 0: 0x00: Setting is disabled. 0x01 to 0xFF: (<SCLL[7:0]> + 2 + <DNF[2:0]>) × PRSCK period When ***[I2CxACR0]***<NFSEL> = 1: 0x00: Setting is disabled. 0x01 to 0xFF: (<SCLL[7:0]> + 2) × PRSCK period
7:0	SCLH[7:0]	0x00	R/W	SCL HIGH width setting When ***[I2CxACR0]***<NFSEL> = 0: 0x00 to 0xFF: (<SCLH[7:0> + 2 + <DNF[2:0]>) × PRSCK period When ***[I2CxACR0]***<NFSEL> = 1: 0x00 to 0xFF: (<SCLH[7:0]> + 2) × PRSCK period

Note: Do not rewrite during I2C communication operation.

4.2.11. *[I2CxAAR1]* (I2C 1st Slave Address Register)

Bit	Bit symbol	After reset	Type	Function
31:16	-	0	R	Read as “0”.
15	SAFS1	0	R/W	Selects the format of the first slave address. 0: 7-bit address format 1: 10-bit address format
14:11	-	0	R	Read as “0”.
10:1	SA1[9:0]	0x000	R/W	First slave address setting: Upper 3bits<9:7> Sets upper 3 bits of a 10-bit address.
			R/W	First slave address setting: Lower 7bits<6:0> Sets lower 7 bits of a 10-bit address or a 7-bit address.
0	SA1E	0	R/W	First slave address control 0: Invalid (Match detection is not done.) 1: Valid (Match detection is done.)

Note: Do not rewrite during I2C communication operation.

4.2.12. *[I2CxAAR2]* (I2C 2nd Slave Address Register)

Bit	Bit symbol	After reset	Type	Function
31:16	-	0	R	Read as “0”.
15	SAFS2	0	R/W	Selects the format of the second slave address. 0: 7-bit address format 1: 10-bit address format
14:11	-	0	R	Read as "0".
10:7	SA2[9:0]	0x000	R/W	Second slave address setting:Upper 3bits<9:7> Sets upper 3 bits of a 10-bit address.
			R/W	Second slave address setting: Lower 7bits<6:0> Sets lower 7 bits of a 10-bit address or a 7-bit address.
0	SA2E	0	R/W	Second slave address control 0: Invalid (Match detection is not done.) 1: Valid (Match detection is done.)

Note: Do not rewrite during I2C communication operation.

4.2.13. *[I2CxAIE]* (I2C Interrupt/DMA Setting Register)

Bit	Bit symbol	After reset	Type	Function
31:17	-	0	R	Read as “0”.
15	DMARX	0	RW	Sets reception DMA request control. 0: Invalid (DMA request is not output.) 1: Enable (DMA request is output.)
14	DMATX	0	RW	Sets transmission DMA request control. 0: Invalid (DMA request is not output.) 1: Enable (DMA request is output.)
13	INTTOE	0	RW	Sets the Status interrupt generation factor (Timeout detection) 0: Invalid (Interrupt is not output.) 1: Enable (Interrupt is output.)
12	INTESPE	0	RW	Sets the Status interrupt generation factor (Error STOP condition detection) 0: Invalid (Interrupt is not output.) 1: Enable (Interrupt is output.)
11	INTESTE	0	RW	Sets the Status interrupt generation factor (Error START condition detection) 0: Invalid (Interrupt is not output.) 1: Enable (Interrupt is output.)
10	-	0	R	Read as “0”.
9	INTASE	0	RW	Sets the Status interrupt generation factor (The first or the second slave address match detection) 0: Invalid (Interrupt is not output.) 1: Enable (Interrupt is output.)
8	INTGCE	0	RW	Sets the Status interrupt generation factor (General call detection) 0: Invalid (Interrupt is not output.) 1: Enable (Interrupt is output.)
7	INTALE	0	RW	Sets the Status interrupt generation factor (Arbitration Lost detection) 0: Invalid (Interrupt is not output.) 1: Enable (Interrupt is output.)
6	INTNACKE	0	RW	Sets the Status interrupt generation factor (NACK detection) 0: Invalid (Interrupt is not output.) 1: Enable (Interrupt is output.)
5:3	-	0	R	Read as “0”.
2	INTSPE	0	RW	Sets the Status interrupt generation factor (STOP condition detection) 0: Invalid (Interrupt is not output.) 1: Enable (Interrupt is output.)
1	INTRSE	0	RW	Sets the Status interrupt generation factor (Repeated START condition detection) 0: Invalid (Interrupt is not output.) 1: Enable (Interrupt is output.)
0	INTSTE	0	RW	Sets the Status interrupt generation factor (START condition detection) 0: Invalid (Interrupt is not output.) 1: Enable (Interrupt is output.)

Note: Do not rewrite during I2C communication operation.

4.2.14. *[I2CxAPM]* (I2C Bus Pin Monitor Register)

Bit	Bit symbol	After reset	Type	Function
31:4	-	0	R	Read as “0”.
3	SDAOUT	1	R	SDA output monitor 0: SDA pin is pulled to "LOW". 1: SDA pin is released.
2	SCLOUT	1	R	SCL output monitor 0: SCL pin is pulled to "LOW". 1: SCL pin is released.
1	SDA	Undefined	R	SDA pin monitor 0: SDA pin is "Low". 1: SDA pin is "High".
0	SCL	Undefined	R	SCL pin monitor 0: SCL pin is "Low". 1: SCL pin is "High".

4.2.15. *[I2CSWUPCR1]* (I2C Wakeup Control Register 1)

Bit	Bit symbol	After reset	Type	Function
7	BUSY	0	R	0: Stop condition detection 1: Start condition detection
6	SGCDI	0	R/W	0: General call detection on 1: General call detection off
5	ACK	0	R/W	0: ACK output ("0" outputs) 1: No ACK output ("1" outputs) (NACK setup)
4	I2RES	1	R	I^2C-bus reset 0: Reset release 1: Under reset
			W	0: Reset release 1: Reset done
3	RW	0	R	The transceiver demand monitor from a master 0: Slave reception 1: Slave transmission
2	–	0	R	Read as ”0”
1	GC	0	R	General call detection status 0: No detection 1: Detection
0	INTEND	0	R/W	Interrupt release 0: – 1: Interrupt release

Note 1: When <I2RES>=1, I^2C-bus is reset; however, the setting is not automatically returned to "0". When the reset state is released, set <I2RES>=0 prior to the reset.

Note 2: When the reset is performed with <I2RES>, all read registers of the I2C wakeup block (sub) are initialized; however write data is not initialized.

Note 3: After the reset is released with <I2RES>, the circuit operations are performed along to the preset values. Therefore, set slave addresses or ACK signals when <I2RES>=1.

Note 4: <RW> is "0" regardless of transmission/receive request from the master when the addresses do not match.

4.2.16. *[I2CSWUPCR2]* (I2C Wakeup Control Register 2)

Bit	Bit symbol	After reset	Type	Function
7:1	WUPSA1[6:0]	0x00	R/W	The 1st slave address setting
0	-	0	R	Read as "0"

4.2.17. *[I2CSWUPCR3]* (I2C Wakeup Control Register 3)

Bit	Bit symbol	After reset	Type	Function
7:1	WUPSA2[6:0]	0x00	R/W	The 2nd slave address setting
0	WUPSA2EN	0	R/W	Selection for the 2nd salve address 0: No use 1: Used

4.2.18. *[I2CSWUPSL]* (I2C Status Register)

Bit	Bit symbol	After reset	Type	Function
7:3	-	0	R	Read as “0”.
2	WUPSA2	0	R	Receiving status of the 2nd address. 0: Not matched 2nd Slave address. 1: Matched 2nd Salve address.
1	WUPSA	0	R	Receiving status of the 1st address. 0: Not matched 1 st Slave address. 1: Matched 1st Slave address.
0	-	0	R	Read as “0”.

4.2.19. *[I2CSWUPCR4]* (I2C Wakeup Control Register 4)

Bit	Bit symbol	After reset	Type	Function
7	WUPSAFS1	0	R/W	Format selection of 1st address 0: 7-bit address 1: 10-bit address
6:3	-	0	R	Read as “0”
2:0	WUPSA1U[2:0]	000	R/W	The setting of 1st slave address (upper 3bits of 10-bit)

4.2.20. *[I2CSWUPCR5]* (I2C Wakeup Control Register 5)

Bit	Bit symbol	After reset	Type	Function
7	WUPSAFS2	0	R/W	Format selection of 2nd address 0: 7-bit address 1: 10-bit address
6:3	-	0	R	Read as “0”
2:0	WUPSA2U[2:0]	000	R/W	The setting of 2nd slave address (upper 3bits of 10-bit)

5. Usage Example

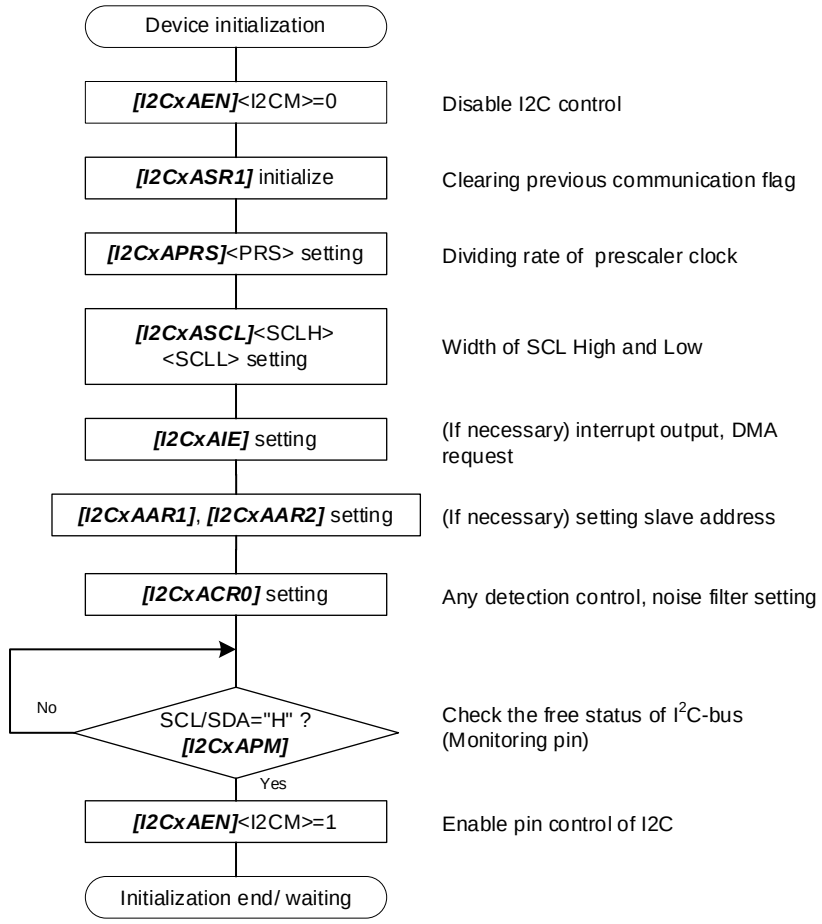
5.1. Data Transfer Procedure

5.1.1. Device Initialization

At first, it should be checked that the SDA and the SCL pins are "HIGH" (Bus free). ***[I2CxAEN]***<I2CM> = 0 is set to disable the I2C control. The communication flag is cleared by ***[I2CxASR1]***. And the prescaler clock is set. Then, the "HIGH" and "LOW" widths of the SCL are set.

If necessary, ***[I2CxAIE]***, ***[I2CxAAR1]***, and ***[I2CxAAR2]*** are set. And, using ***[I2CxACR0]***, each detection control and the noise filter settings are done.

At last, after checking that the communication is not operating, ***[I2CxAEN]***<I2CM> = 1 is set to enable the I2C control.



5.1.2. Master Transmission

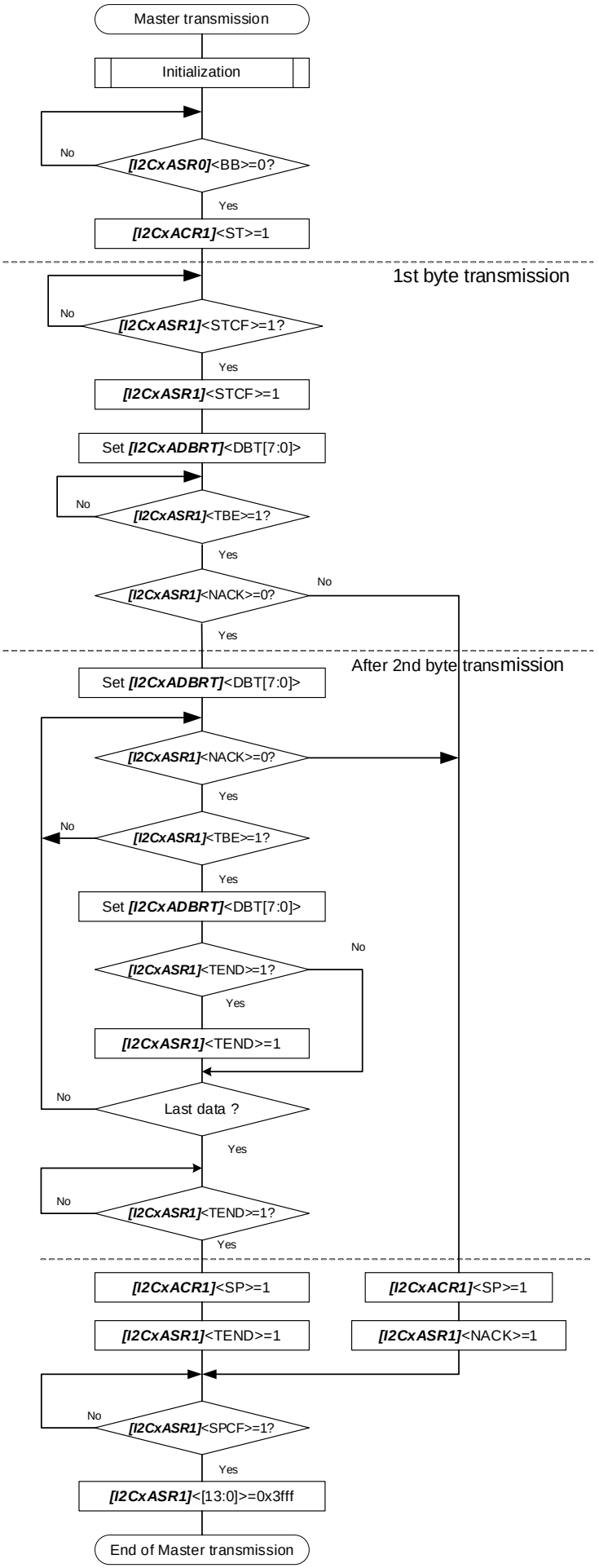
When an address has been transmitted and its ACK is returned, ***[I2CxASR1]***<TBE> is set to "1" and the INTI2CxTBE is generated. The module enters the WAIT state for communication data preparation. When data is written to the Transmission data buffer ***[I2CxADBRT]***<DBT> or ***[I2CxASR1]***<TBE> is set to "1", ***[I2CxASR1]***<TBE> becomes "0", and the data in ***[I2CxADBRT]***<DBT> is transferred to the shift register. Data transmission starts. When the data is transferred to the shift register, ***[I2CxASR1]***<TBE> is set to "1" and the INTI2CxTBE is generated. When the fall edge of the SCL is detected during ***[I2CxASR0]***<BC> = 8, the SDA is released (becomes "1") to receive an ACK/NACK.

The operation after the reception of the ACK/NACK depends on the setting of ***[I2CxACR0]***<NACKE>.

If ***[I2CxACR0]***<NACKE> = 0 and ***[I2CxASR1]***<TBE> = 0 after the reception of the ACK/NACK, the next transmission starts immediately. If ***[I2CxASR1]***<TBE> = 1, ***[I2CxASR1]***<TEND> is set to "1" and the INTI2CxST is generated. The module enters the WAIT state. ***[I2CxASR1]***<TEND> should be cleared to release the WAIT state.

If ***[I2CxACR0]***<NACKE> = 1 and the NACK is received, the NACK detection flag ***[I2CxASR1]***<NACK> is set to "1". The module enters the WAIT state regardless of the value of ***[I2CxASR1]***<TBE>. When an interrupt is enabled, the INTI2CxST is generated. The NACK detection flag is cleared by writing "1" to ***[I2CxASR1]***<NACK> to release the WAIT state. If ***[I2CxASR1]***<TEND> = 1, ***[I2CxASR1]***<TEND> should be also cleared. When the STOP condition generate or the Repeated START condition generate has been set and the NACK detection flag is cleared, each condition is generated. If no conditions are set, the next transmission starts. When the ACK is received, the same operation is done as the operation at ***[I2CxACR0]***<NACKE> = 0.

The data of the ACK/NACK is not acquired by the shift register. It is directly set to ***[I2CxASR0]***<ACKF>.



Initialization

Status check I²C-bus

Start condition output

Check start condition status

Clear start condition detection flag

Address setting (7-bit address: Slave address 7 bits+0b0)
(10-bit address: 1st 7 bits 0b11110
+ Upper 2 bits slave address+0b0)

Release clock stretch

Check transmission buffer status

Check ACK

Set transmission data (7-bit address: transmission data)
(10-bit address: lower 8 bits slave address)

Release clock stretch

Check ACK

Check transmission buffer status

Set transmission data

Check transfer end flag
(When next data cannot be set during I2C 1byte transfer)

Release clock stretch

Check end of last data transfer

Request stop condition output

Release clock stretch

Check stop condition output

For next communication, clear the flag

5.1.3. Master Reception

When an address has been transmitted and its ACK is returned, ***[I2CxASR0]***<TRX> is set to "0" and the reception operation starts. When the shift register receives 8-bit data, the value is transferred to the Reception data buffer register ***[I2CxADBRR]***<DBR>. ***[I2CxASR1]***<RBF> is set to "1", and the interrupt INTI2CxRBF is generated.

When ***[I2CxACR1]***<ACKWAIT> is set to "1", this module enters the WAIT state.

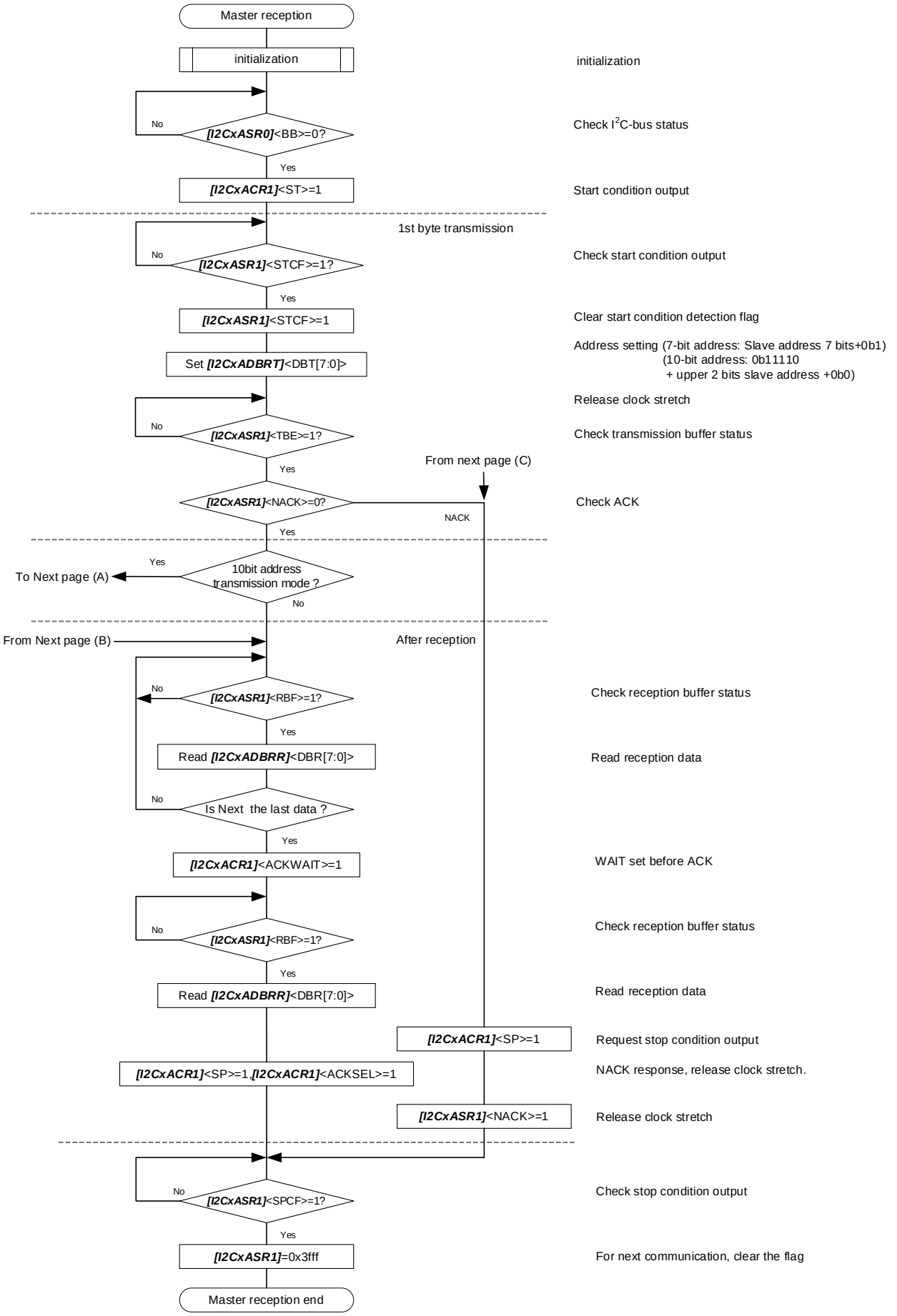
The WAIT state is released by writing ***[I2CxACR1]***<ACKSEL>. And an ACK/NACK is output according to the value of ***[I2CxACR1]***<ACKSEL>. When ***[I2CxACR1]***<ACKWAIT> = 0, the ACK/NACK is output according to the value of ***[I2CxACR1]***<ACKSEL> without entering the WAIT state. After the ACK/NACK is output, the SDA is released at the fall edge of the SCL.

The reception operation continues when ***[I2CxASR1]***<RBF> = 0 or a valid data is not present in the shift register at the transmission of the ACK. When ***[I2CxASR1]***<RBF> = 1 and a valid data is present in the shift register, ***[I2CxASR1]***<TEND> is set to "1" and the module enters the WAIT state. The WAIT state is released by setting ***[I2CxASR1]***<TEND> to "1" to clear ***[I2CxASR1]***<TEND> to "0".

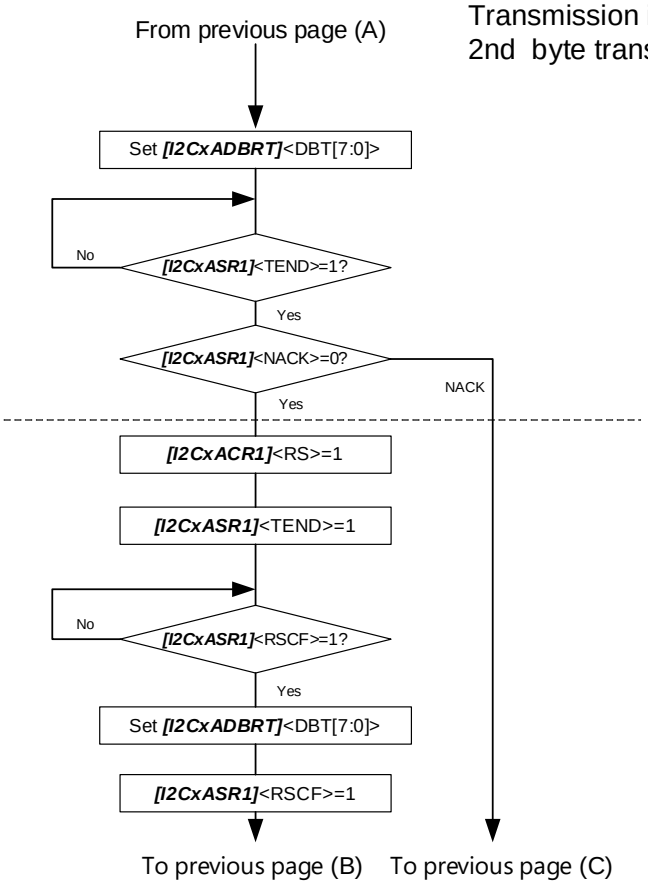
When a STOP condition coming out or a Repeated START condition coming out has been set before the NACK is transmitted, the set operation is done after the NACK is transmitted. When the set operation is not done, the reception operation continues.

When ***[I2CxACR1]***<ACKWAIT> is set to "1" before the last bit is received during data communication, this module enters the WAIT state before the ACK is returned. When it is set to "1" after the last bit is received, this module enters the WAIT state before the ACK response of the next data reception.

Master reception using with <ACKWAIT>:



Transmission in 10-bit address
2nd byte transmission



Address setting (10-bit address : Slave address 2nd byte)

Check transfer end flag

Check ACK

Request restart condition output

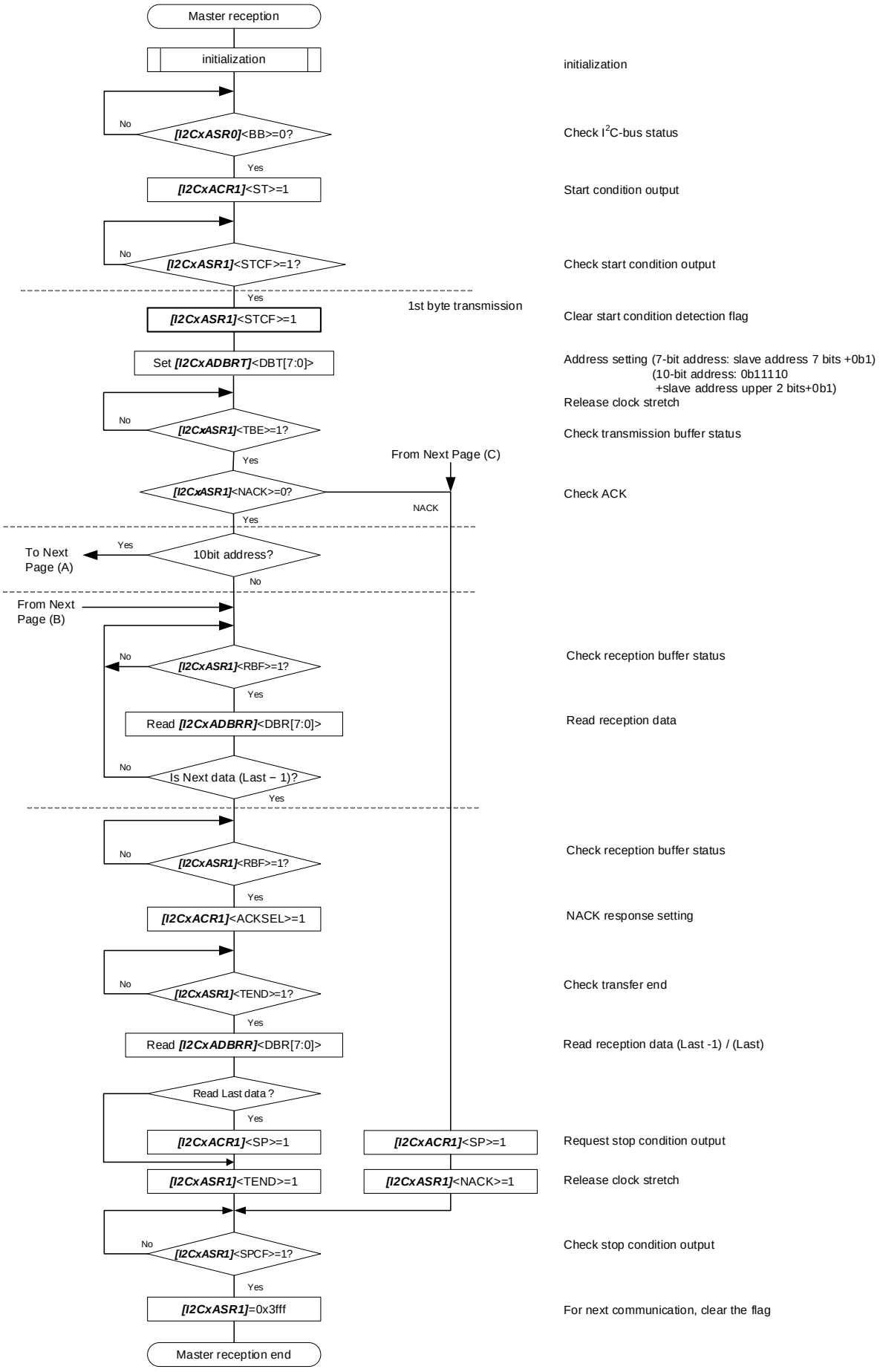
Release clock stretch

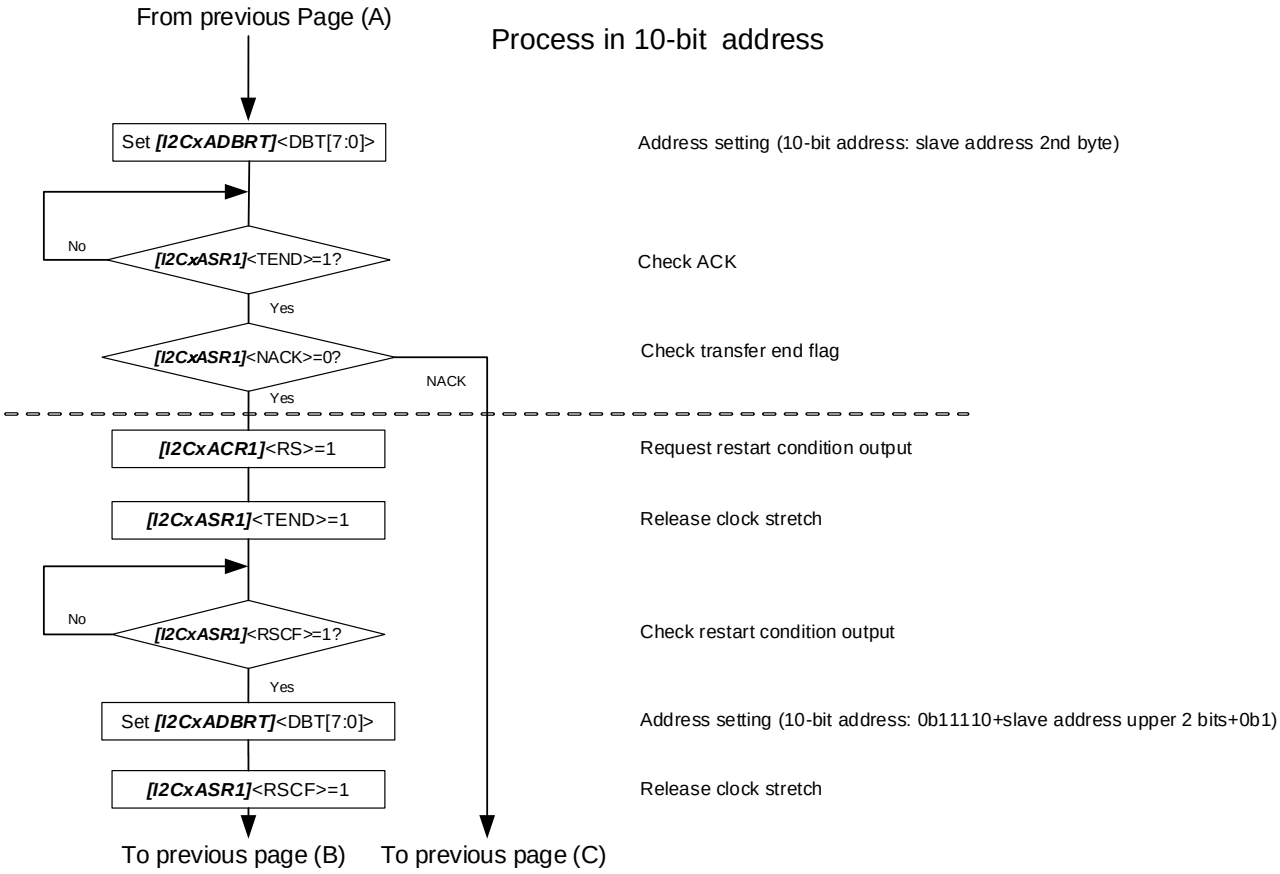
Check restart condition output

Address set (10-bit address:
Slave address 1st 7 bits (0b11110+Upper 2 bits slave address)+0b1 (read))

Release clock stretch

Master reception with transfer end flag:




From previous Page (A)
Process in 10-bit address
Set [I2CxADBRT]<DBT[7:0]>
Address setting (10-bit address: slave address 2nd byte)
No
[I2CxASR1]<TEND>=1?
Check ACK
Yes
[I2CxASR1]<NACK>=0?
Check transfer end flag
NACK
Yes
[I2CxACR1]<RS>=1
Request restart condition output
[I2CxASR1]<TEND>=1
Release clock stretch
No
[I2CxASR1]<RSCF>=1?
Check restart condition output
Yes
Set [I2CxADBRT]<DBT[7:0]>
Address setting (10-bit address: 0b11110+slave address upper 2 bits+0b1)
[I2CxASR1]<RSCF>=1
Release clock stretch
To previous page (B)
To previous page (C)

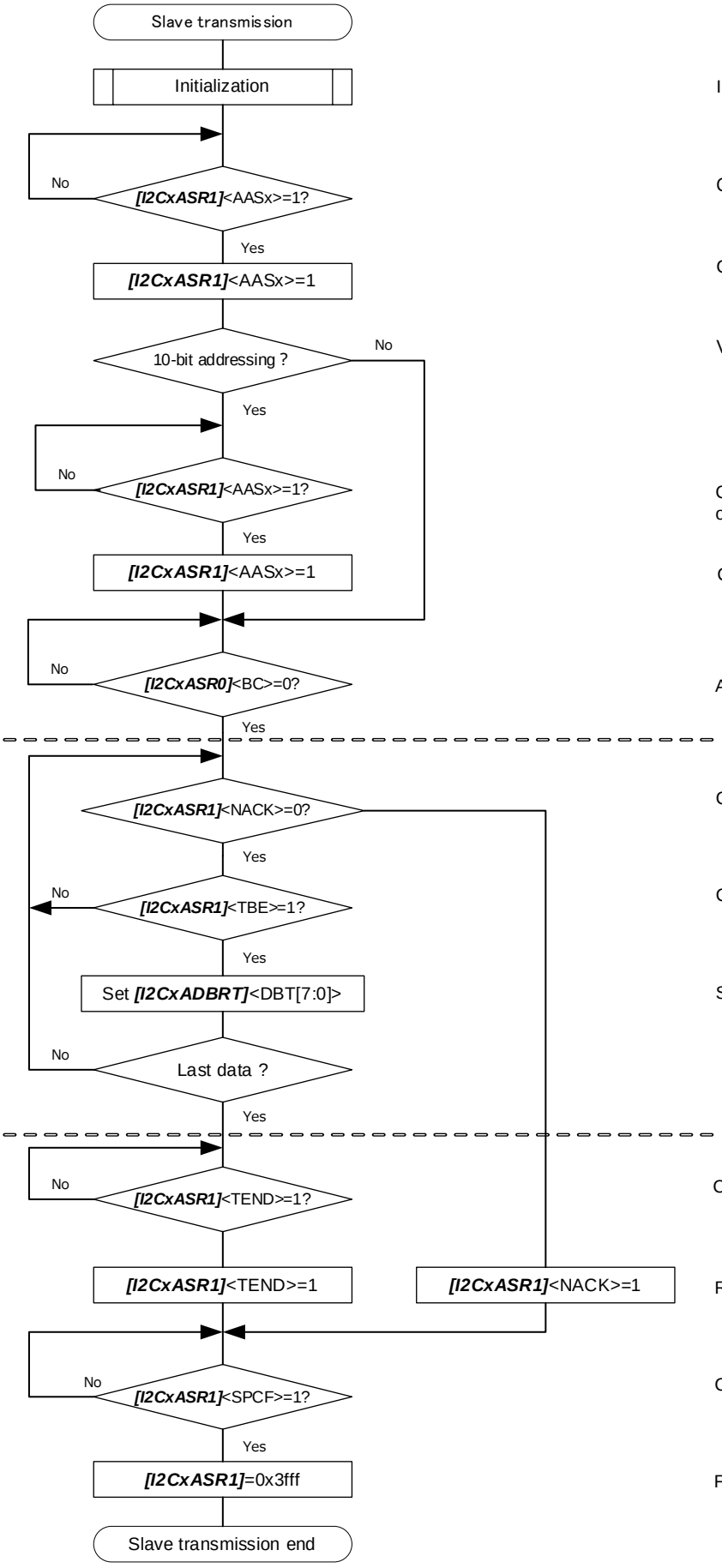
5.1.4. Slave Transmission

After the ACK which shows the slave address match is output, the clock stretch is done. ***[I2CxASR1]***<TBE> is set to "1" and the INTI2CxTBE is generated. When data is written to the Transmission data buffer ***[I2CxADBRT]***<DBT> or ***[I2CxASR1***]<TBE> is set to "1", ***[I2CxASR1]***<TBE> becomes "0". When the data in ***[I2CxADBRT]***<DBT> is transferred to the shift register, the clock stretch is released at the same time.

When a NACK is detected at ***[I2CxACR0]***<NACKE> = 1, the clock stretch is done. The clock stretch is released by clearing the NACK detection flag. When the NACK detection flag is cleared, the clock stretch finishes and the SDA is released (becomes "1"). The module is in the state where it waits for STOP condition detection or Repeated START condition detection, but the transmission continues if the master outputs the clock continuously.

When an ACK is detected, the master will continue to output clocks and transmit operation will continue.

Note that ACK / NACK is not captured in the shift register and is directly set in ***[I2CxASR0]*** <ACKF>.

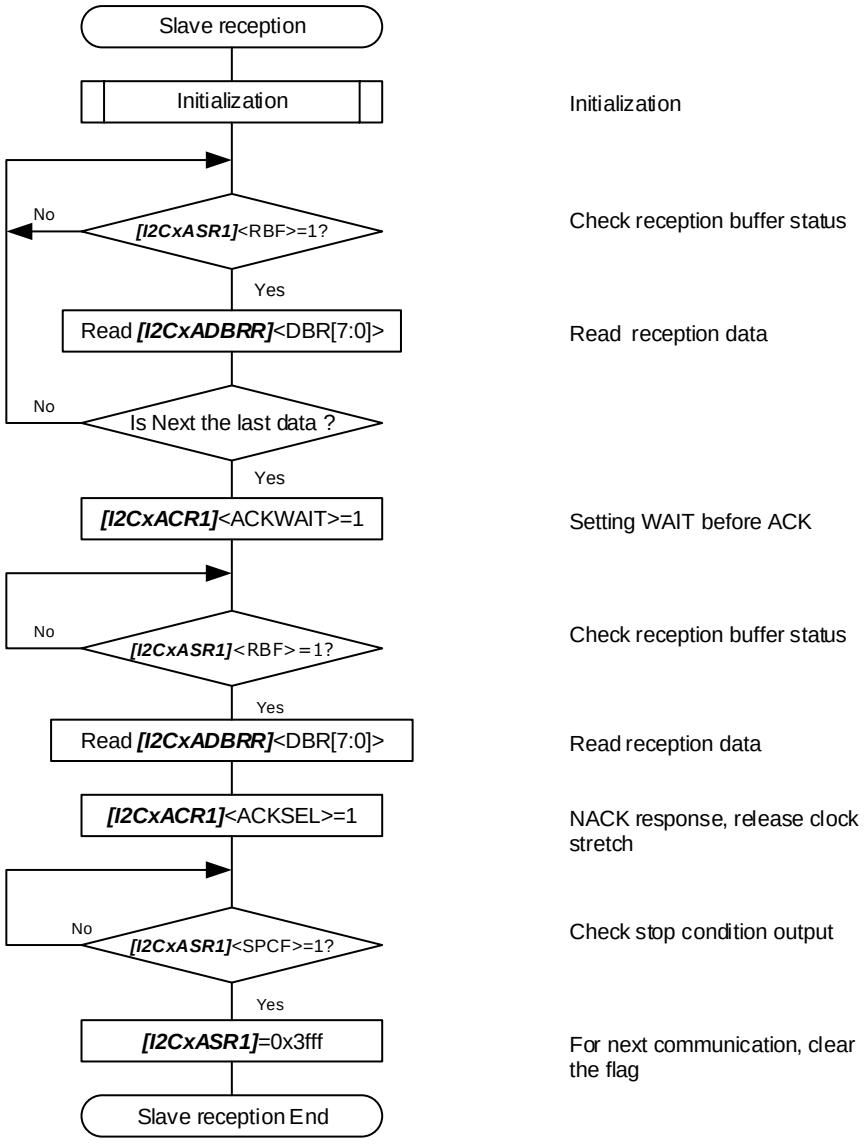

Slave transmission
Initialization
Initializtion
No
[I2CxASR1]<AASx>=1?
Check address match
Yes
[I2CxASR1]<AASx>=1
Clear address match flag
10-bit addressing ?
No
Valid 10-bit addressing
Yes
No
[I2CxASR1]<AASx>=1?
Check address match after restart condition output for changing transfer direction
Yes
[I2CxASR1]<AASx>=1
Clear address match flag
No
[I2CxASR0]<BC>=0?
Address (1byte) reception completion
Yes
[I2CxASR1]<NACK>=0?
Check ACK
Yes
No
[I2CxASR1]<TBE>=1?
Check transmission buffer status
Yes
Set [I2CxADBRT]<DBT[7:0]>
Set transmission data
No
Last data ?
Yes
No
[I2CxASR1]<TEND>=1?
Check last data transmission completion
[I2CxASR1]<TEND>=1
[I2CxASR1]<NACK>=1
Release clock stretch
No
[I2CxASR1]<SPCF>=1?
Check Stop condition output
Yes
[I2CxASR1]=0x3fff
For next communication, clear the flag
Slave transmission end

5.1.5. Slave Reception

After the ACK of a slave address match is output, the reception operation starts. When the shift register receives 8-bit data, the value is transferred to the Reception data buffer register (***[I2CxADBRR]***<DBR>). ***[I2CxASR1]***<RBF> is set to "1", and the interrupt INTI2CxRBF is generated.

When an ACK is transmitted, the operation is the same as the operation of the master reception.

When a NACK is transmitted, the SCL and the SDA are released after the NACK transmission. The module waits for STOP condition detection or Repeated STOP condition detection. But, if the master continues to transmit data, the reception operation continues.



5.1.6. Repeated Start

When ***[I2CxASR0]***<MST> and <BB> are "1", a Repeated START condition can be output by writing "1" to ***[I2CxACR1]***<RS>.

The following shows the release timing of the SDA to output the Repeated START condition.

1. Data transfer completes if an ACK is detected in ***[I2CxACR0]***<NACKE> = 0 or ***[I2CxACR0]***<NACKE> = 1 (when ***[I2CxASR0]***<BC> becomes "0").

2. ***[I2CxASR1]***<NACK> is cleared when a NACK is detected during ***[I2CxACR0]***<NACKE> = 1.
   In this case, ***[I2CxACR1]***<RS> should be set to "1" before ***[I2CxASR1]***<NACK> is cleared.

3. ***[I2CxASR1]***<TEND> is cleared after the transmission end is detected.
   [I2CxACR1]<RS> should be set to "1" before ***[I2CxASR1]***<TEND> is cleared.

When a Repeated START condition is detected, ***[I2CxACR1]***<RSCF> is set to "1" and this module enters the WAIT state.

For releasing the WAIT state, at first writing "0" to ***[I2CxACR1]***<RSCF> and then writing an address and the direction bit to ***[I2CxADBRT]***<DBT>

5.2. Wakeup Operation and Setting Flow (Example)

<Initial setting> Wakeup function setting to enter STOP1 mode.

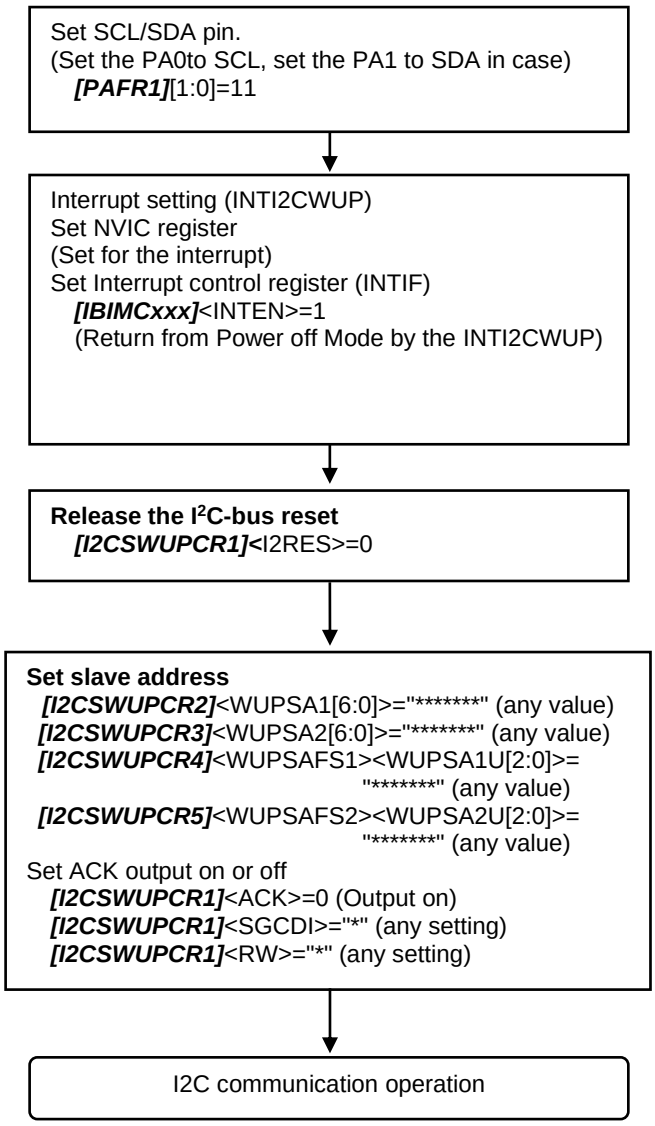


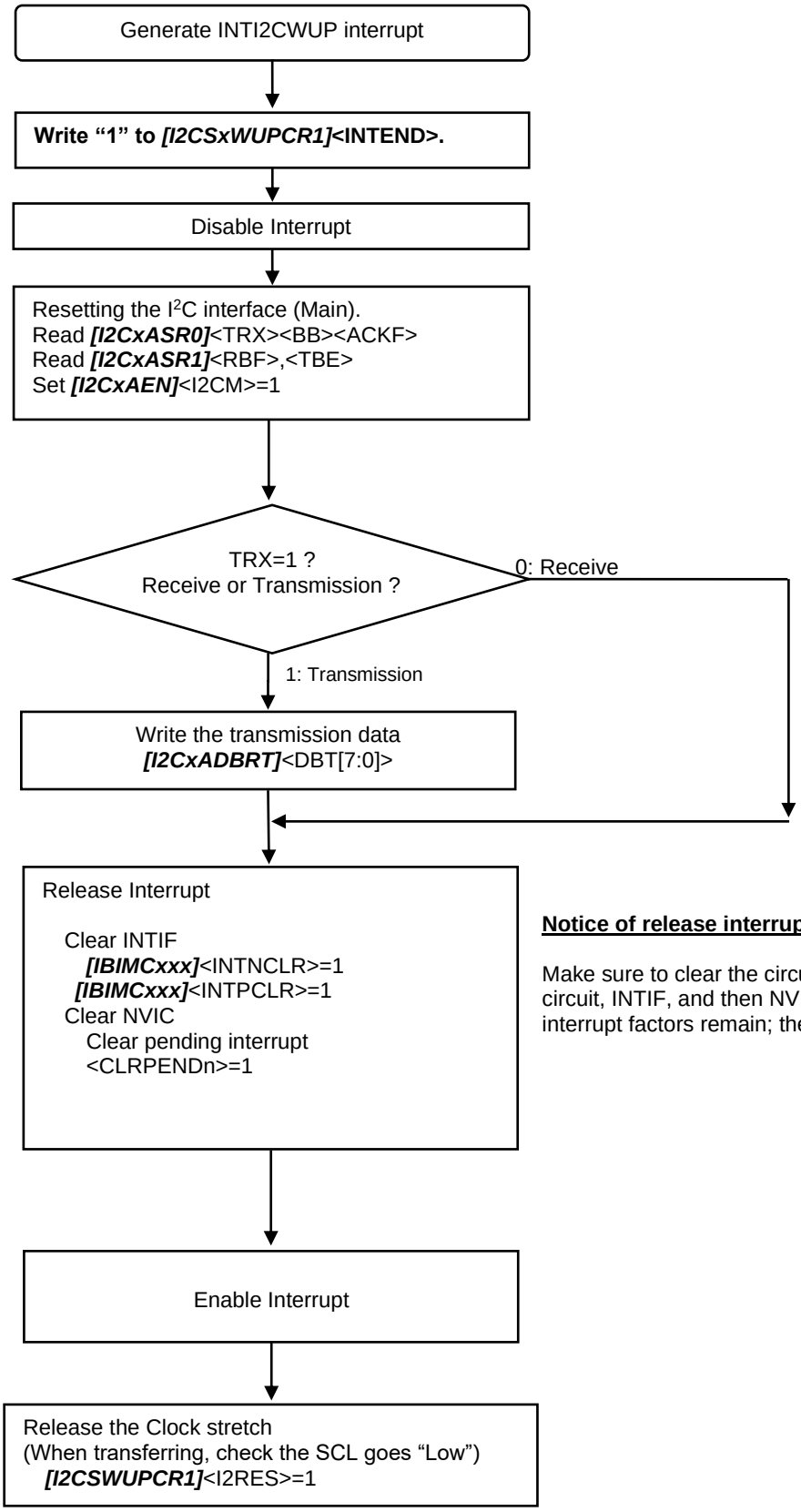
Figure 5.1 Wakeup Initialize Setting

Note: The register can be written during the reset operation caused by ***[I2CSWUPCR1]***<I2RES>=1 even when reset operation continues.

The setting for the I²C interface (main) should be completed in the release processing of the Low-power consumption mode (STOP1 or STOP2). Communications after the address match wakeup function do not work properly unless the following registers are set.

[I2CxAEN], ***[I2CxACR1]***, ***[I2CxACR0]***
[I2CxAPRS],***[I2CxSCL]***
[I2CxAAR1], ***[I2CxAAR2]*** ; any value
[I2CxAIE] ; set usage function

<When generating Interrupt>



Notice of release interrupt

Make sure to clear the circuit along the following order: I2C circuit, INTIF, and then NVIC. When the order is not correct, interrupt factors remain; therefore, the interrupts occur again.

Figure 5.2 Flow After Wakeup

6. Precaution for Usage

This product does not meet the specifications among the AC electrical characteristics defined by the I^2C standard, depending on the functions of the implemented hardware.

Some items that do not meet the specifications may need to be handled by software. Please handled by the software when using the corresponding function.

The following items need to be supported by the software.

- Bus free time between a STOP and START condition (t_{BUF})
  All modes require the bus free time by software.
  After generating a stop condition, the I^2C interface goes into a slave state and a bus free state, and the stop detection flag <SPCF> = 1.

7. Revision History

Table 7.1 Revision history

Revision	Date	Description
1.0	2020-11-17	First release
1.1	2021-02-04	- Corrected 3.3.1. Serial Clock (2) - 5.1.1. Device initialization Corrected flow chart. - 5.1.2. Master transmission Corrected flow chart. - 5.1.3. Master reception Master reception using with <ACKWAIT> Corrected flow charts. - 5.1.3. Master reception Master reception with transmission completion flag Corrected flow charts. - 5.1.4. Slave transmission Corrected flow chart. - 5.1.5. Slave reception Corrected flow chart. - 5.2. Wakeup operation and setting flow (Example) Corrected flow chart.
1.2	2023-01-17	- 3.3.4.1. START Condition Corrected from "write “0” to ***[I2CxACR1]***<STCF>" to "write “1” to ***I2CxASR1]*** <STCF>". - 4.2.2. ***[I2CxAEN]*** (I2C enable register) Corrected from "Note: Do not rewrite while I^2C is operating (***[I2CxAEN]*** <I2CM>=1)" to "Note: Do not rewrite during I2C communication operation.". - 4.2.8. [I2CxASR1] (I2C status register 1) <TBE> After reset: Corrected from "0" to "1".
1.3	2024-04-12	- 2. Configuration Changed figure 2.1 - 3.2. Data format Changed figure 3.2 - 3.3.1. Serial Clock Changed the (2) serial clock description Deleted figure - 4.1. Register List Added TYPE3 to base address field - 4.2.7. ***[I2CxASR0]*** (I2C status register 0) Changed the function column of <BC> - 4.2.10. ***[I2CxASCL]*** (I2C SCL width setting register) Changed the function column of <SCLL> and <SCLH> - 5.1.2. Master Transmission Changed figure - 5.1.3. Master Reception Changed figure

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE"). ** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, lifesaving and/or life supporting medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, and devices related to power plant. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**