TOSHIBA

Dear Cutomers,

March, 2019

Datasheet Correction: I2C function of TX/TXZ Family

This informs you that we found the following corrections should be made to the datasheets as shown below. If you have any questions or require any further informaiton, please contact your local Toshiba representatives.

1. Products

M3H Group(1)

TMPM3H0FS	TMPM3H0FM	TMPM3H1FW	TMPM3H1FU	TMPM3H1FS
TMPM3H1FP	TMPM3H2FW	TMPM3H2FU	TMPM3H2FS	TMPM3H3FW
TMPM3H3FU	TMPM3H3FS	TMPM3H4FW	TMPM3H4FU	TMPM3H4FS
TMPM3H5FW	TMPM3H5FU	TMPM3H5FS	TMPM3H6FW	TMPM3H6FU
TMPM3H6FS				

M3H Group(2)

TMPM3HLFY	TMPM3HLFZ	TMPM3HLFD	TMPM3HMFY	TMPM3HMFZ
TMPM3HMFD	TMPM3HNFY	TMPM3HNFZ	TMPM3HNFD	TMPM3HPFY
TMPM3HPFZ	TMPM3HPFD	TMPM3HQFY	TMPM3HQFZ	TMPM3HQFD

M4G Group(1)

TMPM4G9F15	TMPM4G9F10	TMPM4G9FE	TMPM4G9FD	TMPM4G8F15
TMPM4G8F10	TMPM4G8FE	TMPM4G8FD	TMPM4G7F10	TMPM4G7FE
TMPM4G7FD	TMPM4G6F10	TMPM4G6FE	TMPM4G6FD	

M060 Group

TMPM066FW	TMPM067FW	TMPM068FW

2. Corrections

The function of I^2C use are changed as following.

Note : This information explains the correction using the M3H Group(1) product.

<Correction 1: Analog Noise Filter>

Register descriptions

[Old]

4.2.9. *[I2CxOP]* (Expanded function setting register)

Bit	Bit Symbol	After reset	Type	Function
31:8	-	0	R	Read as "0".
7	DISAL	0	R/W	Control the Arbitration Lost detected function 0: Enable Arbitration Lost detect function 1: Disable Arbitration Lost detect function
6	SA2ST	0	R	Discriminates the received slave address. (Updated when <AAS>=1.) 0: Not match <SA2> 1: Matches <SA2>
5	SAST	0	R	Discriminates the received slave address. (Updated when <AAS>=1.) 0: Not match <SA> 1: Matches <SA>
4	NFSEL	0	R/W	Select Noise cancellation (it also Changed the communication speed) 0: Digital 1: Analog
3	RSTA	0	R	Detects a Repeated START flag 0: Not detected 1: Detected
			W	0: Clear 1: Invalid
2	GCDI	0	R/W	Sets general-call detection. (Valid only when <NOACK>=0) 0: Detection is ON. 1: Detection is OFF.
1	SREN	0	R	Sets the repeated START output. (Valid only when the MCU in master mode.) 0: Repeated START has completed. 1: Repeated START is ongoing.
			W	0: Invalid 1: Repeted start requeset
0	MFACK	0	R/W	Selects an ACK Output. 0: ACK Output 1: NACK Output Be careful, It cannot be used in a free data format.

[New]

4.2.9. *[I2CxOP]* (Expanded function setting register)

Bit	Bit Symbol	After reset	Type	Function
31:8	-	0	R	Read as "0".
7	DISAL	0	R/W	Control the Arbitration Lost detected function 0: Enable Arbitration Lost detect function 1: Disable Arbitration Lost detect function
6	SA2ST	0	R	Discriminates the received slave address. 0: Not match <SA2> 1: Matches <SA2>
5	SAST	0	R	Discriminates the received slave address. 0: Not match <SA> 1: Matches <SA>
4	NFSEL	0	R/W	Select Noise cancellation 0: Digital 1: Can not be set, Only Write as "0".
3	RSTA	0	R	Detects a Repeated START flag 0: Not detected 1: Detected
			W	0: Clear (Note1) 1: Invalid
2	GCDI	0	R/W	Sets general-call detection. (Valid only when <NOACK>=0) 0: Detection is ON. 1: Detection is OFF.
1	SREN	0	R	Sets the repeated START output. (Valid only when the MCU in master mode.) 0: Repeated START has completed. 1: Repeated START is ongoing.
			W	0: Invalid 1: Repeted start requeset
0	MFACK	0	R/W	Selects an ACK Output. 0: ACK Output 1: NACK Output Be careful, It cannot be used in a free data format.

※ For more other items, deleted the contents about the Analog Noise Filter.

<Correction 2: Repeated Start detection Flag>

Functional Description

[Old]

3.3.12. Repeated start detection

When a repeated START is detected on the bus line, ***[I2CxOP]*** <RSTA> is set to "1". A repeated start is used for a master device to change the direction of transmission without terminating data transfer to a slave device.

The <RSTA> is initialized by a reset and stop condition.

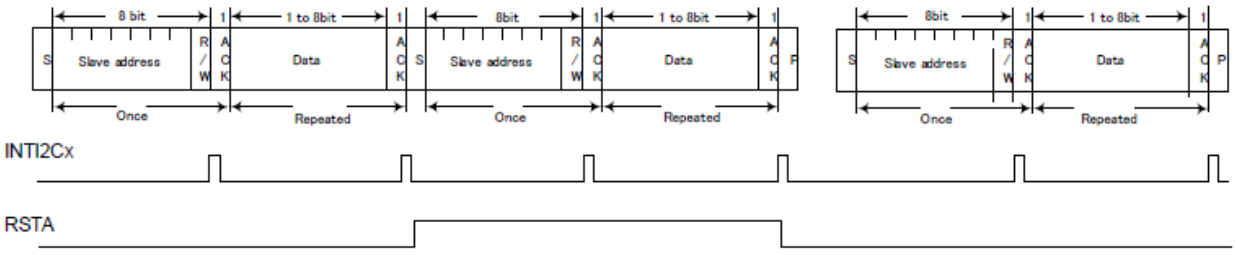

Figure 3.14 Repeated Start detection Flag

[New]

3.3.12. Repeated start detection

In slave mode, when a repeated START is detected on the bus line, ***[I2CxOP]*** <RSTA> is set to "1". A repeated start is used for a master device to change the direction of transmission without terminating data transfer to a slave device.

Since <RSTA> is only initialized by a reset, clear the flag(<RSTA>=0) when stop condition occurs, bus free,.etc. (refer to Figure 3.14)

In case ***[I2CxPRS]***<PRSCK> ≠ 1 in Master mode, <RSTA> is set to "1" after the first start condition after reset. Perform clear processing at the first INTI2Cx interrupt service routine, etc. (refer to Figure 3.15)

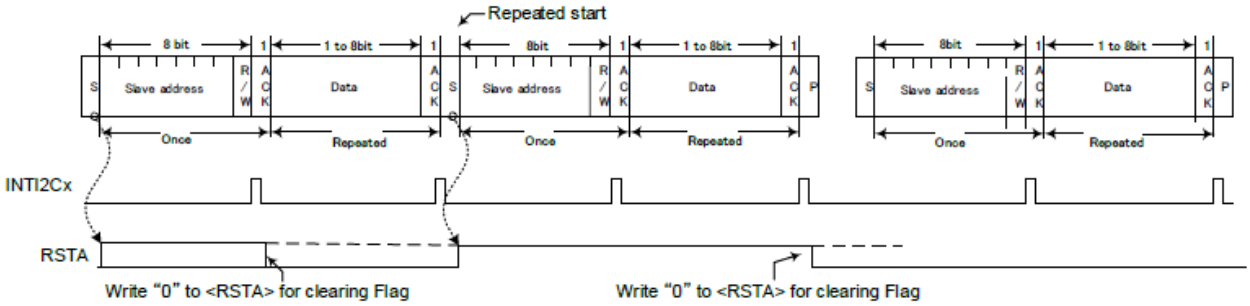

Figure 3.15 Repeated Start detection Flag (Master mode:<PRSCK>≠1)

<Correction 3: Hold time and Setup time>

Functional Description
[Old]

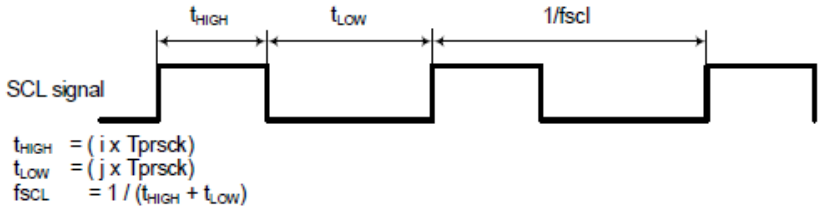

Figure 3.4 I²C SCL Output

Note: The t_{HIGH} period may differ from the specified value if the rising edge becomes blunt depending on the combination of bus load capacitance and pull-up register. If the clock synchronization function for synchronizing clocks from multiple clocks is used, the actual clock period may differ from the specified setting.

In master mode, the hold time when start conditions generated and the setup time when a stop condition is generated are defined as t_{HIGH}[s]

[New]

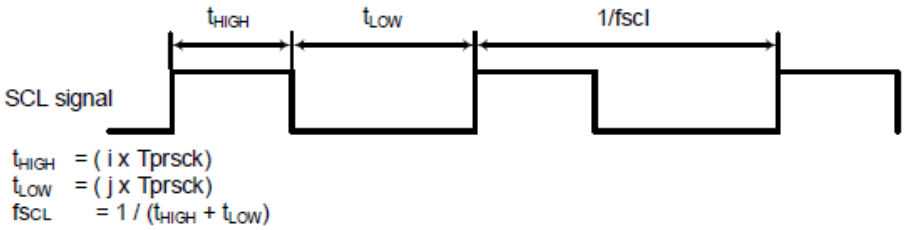

Figure 3.4 I²C SCL Output

Note: The t_{HIGH} period may differ from the specified value if the rising edge becomes blunt depending on the combination of bus load capacitance and pull-up register. If the clock synchronization function for synchronizing clocks from multiple clocks is used, the actual clock period may differ from the specified setting.

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:

[I2CxOP]<SREN>=0: t_{HIGH} [S]

[I2CxOP]<SREN>=1: 8Tprsck [S]

Setup time:

[I2CxPRS]<PRSCK>=1: t_{HIGH} [S]

[I2CxPRS]<PRSCK>≠1: t_{HIGH} - Tprsck [S]

<Correction 4: I2CxCR registger>

Registers Descriptions

[Old]

4.2.4. *[I2CxCR2]* (I^2C control register 2)

2	-	0	R	Read as "0".
1:0	SWRES[1:0]	00	W	Software reset generation Write "10" followed by "01" to generate a reset. Writing "10" followed by "01" into this 2bits generates a software reset.(reset width is 1 clock as fsys) A software reset release the SCL and SDA lines (High state) if the device is data transfer and break the communication. The I^2C bus interface is initialized except ***[I2CxCR2]***<I2CM> and ***[I2CxDBR]*** register When software reset is performed, make sure to write "0" to ***[I2CxCR2]***[7:4]

Note: the ***[I2CxCR2]***<MST>,<TRX>,<BB>,<PIN> bits are given independent functions, they are used in typical combinations, as shown below, according to the ***[I2CxSR]*** setting.
When writing to these bits, make sure that notice.

[New]

4.2.4. *[I2CxCR2]* (I^2C control register 2)

2	-	0	R	Read as "0".
1:0	SWRES[1:0]	00	W	Software reset generation Write "10" followed by "01" to generate a reset. Writing "10" followed by "01" into this 2bits generates a software reset.(reset width is 1 clock as fsys) A software reset release the SCL and SDA lines (HIGH state) if the device is data transfer and break the communication. The I^2C bus interface is initialized except ***[I2CxCR2]***<I2CM> and ***[I2CxDBR]*** register When software reset is performed, make sure to write "0" to ***[I2CxCR2]***[7:4]

Note: the ***[I2CxCR2]***<MST>,<TRX>,<BB>,<PIN> bits are given independent functions, they are used in typical combinations, as shown below, according to the ***[I2CxSR]*** setting.
When writing to these bits, make sure that notice.

Note: Don't change the contents of the registers, except ***[I2CxCR2]***<SWRST[1:0]>, when the start condition is generated, the stop condition is generated or the data transfer is in progress.
Write data to the registers before the start condition is generated or during the period from when an interrupt request is generated for stopping the data transfer until it is released.