

7.2. TWI

7.2.1. Overview

This TWI Controller is designed to be used as an interface between CPU host and the serial TWI bus. It can support all the standard TWI transfer, including Slave and Master. The communication to the TWI bus is carried out on a byte-wise basis using interrupt or polled handshaking. This TWI Controller can be operated in standard mode (100K bps) or fast-mode, supporting data rate up to 400K bps. Multiple Masters and 10-bit addressing Mode are supported for this specified application. General Call Addressing is also supported in Slave mode.

7.2.2. Feature

The TWI Controller includes the following features:

- Software-programmable for Slave or Master
- Support Repeated START signal
- Multi-master systems supported
- Allow 10-bit addressing with TWI bus
- Performs arbitration and clock synchronization
- Own address and General Call address detection
- Interrupt on address detection
- Support speeds up to 400Kbits/s ('fast mode')
- Allow operation from a wide range of input clock frequencies

7.2.3. Pin List

Port Name	Width	Direction	Description
TWI_SCL	1	IN/OUT	TWI Clock line
TWI_SDA	1	IN/OUT	TWI Serial Data line

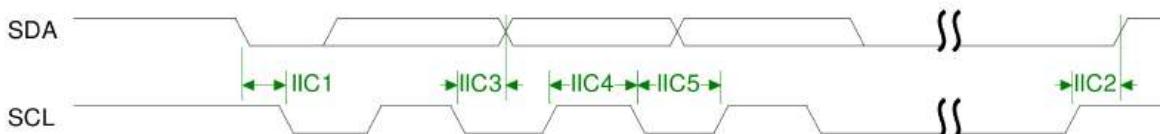
7.2.4. Timing Diagram

Data transferred are always in a unit of 8-bit (byte), followed by an acknowledge bit. The number of bytes that can be transmitted per transfer is unrestricted. Data is transferred in serial with the MSB first. Between each byte of data transfer, a receiver device will hold the clock line SCL low to force the transmitter into a wait state while waiting the response from microprocessor.

Data transfer with acknowledge is obligatory. The clock line is driven by the master all the time, including the acknowledge-related clock cycle, except for the SCL holding between each byte. After sending each byte, the transmitter releases the SDA line to allow the receiver to pull down the SDA line and send an acknowledge signal (or leave it high to send a "not acknowledge") to the transmitter.

When a slave receiver doesn't acknowledge the slave address (unable to receive because of no resource available), the data line must be left high by the slave so that the master can then generate a STOP condition to abort the transfer. Slave receiver can also indicate not to want to send more data during a transfer by leave the acknowledge signal high. And the master should generate the STOP condition to abort the transfer.

Below diagram provides an illustration the relation of SDA signal line and SCL signal line on the TWI serial bus.



TWI Timing Diagram

7.2.5. TWI Controller Operation

There are four operation modes on the TWI bus which dictates the communications method. They are Master Transmit, Master Receive, Slave Transmit and Slave Receive. In general, CPU host controls TWI by writing commands and data to its registers. The TWI interrupts the CPU host for the attention each time a byte transfer is done or a START/STOP conditions is detected. The CPU host can also poll the status register for current status if the interrupt mechanism is not disabled by the CPU host.

When the CPU host wants to start a bus transfer, it initiates a bus START to enter the master mode by setting IM_STA bit in the 2WIRE_CNTR register to high (before it must be low). The TWI will assert INT line and INT_FLAG to indicate a completion for the START condition and each consequent byte transfer. At each interrupt, the micro-processor needs to check the 2WIRE_STAT register for current status. A transfer has to be concluded with STOP condition by setting M_STP bit high.

In Slave Mode, the TWI also constantly samples the bus and look for its own slave address during addressing cycles. Once a match is found, it is addressed and interrupts the CPU host with the corresponding status. Upon request, the CPU host should read the status, read/write 2WIRE_DATA data register, and set the 2WIRE_CNTR control register. After each byte transfer, a slave device always halt the operation of remote master by holding the next low pulse on SCL line until the microprocessor responds to the status of previous byte transfer or START condition.

7.2.6. TWI Controller Register List

Module Name	Base Address
TWI0	0x01C27000
TWI1	0x01C27400
TWI2	0x01C27800

Register Name	Offset	Description
TWI_ADDR_REG	0x00	TWI Slave Address
TWI_XADDR_REG	0x04	TWI Extended Slave Address Register
TWI_DATA_REG	0x08	TWI Data Byte Register
TWI_CNTR_REG	0x0C	TWI Control Register
TWI_STAT_REG	0x10	TWI Status Register
TWI_CCR_REG	0x14	TWI Clock Control Register
TWI_SRST_REG	0x18	TWI Software Reset Register
TWI_EFR_REG	0x1C	TWI Enhance Feature Register
TWI_LCR_REG	0x20	TWI Line Control Register

7.2.7. TWI Controller Register Description

7.2.7.1. TWI Slave Address Register

Offset: 0x00			Register Name: TWI_ADDR_REG
Bit	R/W	Default/Hex	Description
31:8	/	/	/
7:1	R/W	0	SLA Slave address 7-bit addressing: SLA6, SLA5, SLA4, SLA3, SLA2, SLA1, SLA0 10-bit addressing: 1, 1, 1, 1, 0, SLAX[9:8]
0	R/W	0	GCE General call address enable 0: Disable 1: Enable

Note:

For 7-bit addressing:

SLA6 – SLA0 is the 7-bit address of the TWI when in slave mode. When the TWI receives this address after a START condition, it will generate an interrupt and enter slave mode. (SLA6 corresponds to the first bit received from the TWI bus.) If GCE is set to ‘1’, the TWI will also recognize the general call address (00h).

For 10-bit addressing:

When the address received starts with 11110b, the TWI recognizes this as the first part of a 10-bit address and if the next two bits match ADDR[2:1] (i.e. SLAX9 and SLAX8 of the device’s extended address), it sends an ACK. (The device does not generate an interrupt at this point.) If the next byte of the address matches the XADDR register (SLAX7 – SLAX0), the TWI generates an interrupt and goes into slave mode.

7.2.7.2. TWI Extend Address Register

Offset: 0x04			Register Name: TWI_XADDR_REG
Bit	R/W	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0	SLAX Extend Slave Address SLAX[7:0]

7.2.7.3. TWI Data Register

Offset: 0x08			Register Name: TWI_DATA_REG
Bit	R/W	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0	TWI_DATA Data byte for transmitting or received

7.2.7.4. TWI Control Register

Offset: 0x0C			Register Name: TWI_CNTR_REG
Bit	R/W	Default/Hex	Description

31:8	/	/	/
7	R/W	0	INT_EN Interrupt Enable 1'b0: The interrupt line always low 1'b1: The interrupt line will go high when INT_FLAG is set.
6	R/W	0	BUS_EN TWI Bus Enable 1'b0: The TWI bus inputs ISDA/ISCL are ignored and the TWI Controller will not respond to any address on the bus 1'b1: The TWI will respond to calls to its slave address – and to the general call address if the GCE bit in the ADDR register is set. Notes: In master operation mode, this bit should be set to '1'
5	R/W	0	M_STA Master Mode Start When M_STA is set to '1', TWI Controller enters master mode and will transmit a START condition on the bus when the bus is free. If the M_STA bit is set to '1' when the TWI Controller is already in master mode and one or more bytes have been transmitted, then a repeated START condition will be sent. If the M_STA bit is set to '1' when the TWI is being accessed in slave mode, the TWI will complete the data transfer in slave mode then enter master mode when the bus has been released. The M_STA bit is cleared automatically after a START condition has been sent: writing a '0' to this bit has no effect.
4	R/W	0	M_STP Master Mode Stop If M_STP is set to '1' in master mode, a STOP condition is transmitted on the TWI bus. If the M_STP bit is set to '1' in slave mode, the TWI will behave as if a STOP condition has been received, but no STOP condition will be transmitted on the TWI bus. If both M_STA and M_STP bits are set, the TWI will first transmit the STOP condition (if in master mode) then transmit the START condition. The M_STP bit is cleared automatically: writing a '0' to this bit has no effect.
3	R/W	0	INT_FLAG Interrupt Flag INT_FLAG is automatically set to '1' when any of 28 (out of the possible 29) states is entered (see 'STAT Register' below). The only state that does not set INT_FLAG is state F8h. If the INT_EN bit is set, the interrupt line goes high when IFLG is set to '1'. If the TWI is operating in slave mode, data transfer is suspended when INT_FLAG is set and the low period of the TWI bus clock line (SCL) is stretched until '0' is written to INT_FLAG. The TWI clock line is then released and the interrupt line goes low.
2	R/W	0	A_ACK Assert Acknowledge When A_ACK is set to '1', an Acknowledge (low level on SDA) will be sent during the acknowledge clock pulse on the TWI bus if: 1) Either the whole of a matching 7-bit slave address or the first or the second byte of a matching 10-bit slave address has been received.

			<p>2) The general call address has been received and the GCE bit in the ADDR register is set to '1'. 3) A data byte has been received in master or slave mode. When A_ACK is '0', a Not Acknowledge (high level on SDA) will be sent when a data byte is received in master or slave mode. If A_ACK is cleared to '0' in slave transmitter mode, the byte in the DATA register is assumed to be the 'last byte'. After this byte has been transmitted, the TWI will enter state C8h then return to the idle state (status code F8h) when INT_FLAG is cleared. The TWI will not respond as a slave unless A_ACK is set.</p>
1:0	R/W	0	/

7.2.7.5. TWI Status Register

Offset: 0x10			Register Name: TWI_STAT_REG
Bit	R/W	Default/Hex	Description
31:8	/	/	/
7:0	R	0xF8	<p>STA Status Information Byte Code Status 0x00: Bus error 0x08: START condition transmitted 0x10: Repeated START condition transmitted 0x18: Address + Write bit transmitted, ACK received 0x20: Address + Write bit transmitted, ACK not received 0x28: Data byte transmitted in master mode, ACK received 0x30: Data byte transmitted in master mode, ACK not received 0x38: Arbitration lost in address or data byte 0x40: Address + Read bit transmitted, ACK received 0x48: Address + Read bit transmitted, ACK not received 0x50: Data byte received in master mode, ACK transmitted 0x58: Data byte received in master mode, not ACK transmitted 0x60: Slave address + Write bit received, ACK transmitted 0x68: Arbitration lost in address as master, slave address + Write bit received, ACK transmitted 0x70: General Call address received, ACK transmitted 0x78: Arbitration lost in address as master, General Call address received, ACK transmitted 0x80: Data byte received after slave address received, ACK transmitted 0x88: Data byte received after slave address received, not ACK transmitted 0x90: Data byte received after General Call received, ACK transmitted 0x98: Data byte received after General Call received, not ACK transmitted 0xA0: STOP or repeated START condition received in slave mode 0xA8: Slave address + Read bit received, ACK transmitted 0xB0: Arbitration lost in address as master, slave address + Read bit received, ACK transmitted 0xB8: Data byte transmitted in slave mode, ACK received 0xC0: Data byte transmitted in slave mode, ACK not received 0xC8: Last byte transmitted in slave mode, ACK received 0xD0: Second Address byte + Write bit transmitted, ACK received 0xD8: Second Address byte + Write bit transmitted, ACK not received 0xF8: No relevant status information, INT_FLAG=0 Others: Reserved </p>

7.2.7.6. TWI Clock Register

Offset: 0x14			Register Name: TWI_CCR_REG
Bit	R/W	Default/Hex	Description
31:7	/	/	/
6:3	R/W	0	CLK_M
2:0	R/W	0	<p>CLK_N</p> <p>The TWI bus is sampled by the TWI at the frequency defined by F0: $F_{amp} = F_0 = F_{in} / 2^{CLK_N}$</p> <p>The TWI OSCL output frequency, in master mode, is $F_1 / 10$: $F_1 = F_0 / (CLK_M + 1)$</p> <p>$F_{osc} = F_1 / 10 = F_{in} / (2^{CLK_N} * (CLK_M + 1) * 10)$</p> <p>For Example: $F_{in} = 48\text{MHz}$ (APB clock input) For 400kHz full speed 2Wire, $CLK_N = 2$, $CLK_M=2$ $F_0 = 48\text{M}/2^2=12\text{MHz}$, $F_1=F_0/(10*(2+1)) = 0.4\text{MHz}$ For 100Khz standard speed 2Wire, $CLK_N=2$, $CLK_M=11$ $F_0=48\text{M}/2^2=12\text{MHz}$, $F_1=F_0/(10*(11+1)) = 0.1\text{MHz}$</p>

7.2.7.7. TWI Soft Reset Register

Offset: 0x18			Register Name: TWI_SRST_REG
Bit	R/W	Default/Hex	Description
31:1	/	/	/
0	R/W	0	<p>SOFT_RST</p> <p>Soft Reset</p> <p>Write '1' to this bit to reset the TWI and clear to '0' when completing Soft Reset operation.</p>

7.2.7.8. TWI Enhance Feature Register

Offset: 0x1C			Register Name: TWI_EFR_REG
Bit	R/W	Default/Hex	Description
31:2	/	/	/
1:0	R/W	0	<p>DBN</p> <p>Data Byte number follow Read Command Control</p> <p>0— No Data Byte to be wrote after read command</p> <p>1— Only 1 byte data to be wrote after read command</p> <p>2— 2 bytes data can be wrote after read command</p> <p>3— 3 bytes data can be wrote after read command</p>

7.2.7.9. TWI Line Control Register

Offset: 0x20			Register Name: TWI_LCR_REG
Bit	R/W	Default/Hex	Description
31:6	/	/	/
5	R	1	<p>SCL_STATE</p> <p>Current state of TWI_SCL</p> <p>0 – low</p>

			1 - high
4	R	1	<p>SDA_STATE Current state of TWI_SDA 0 – low 1 - high</p>
3	R/W	1	<p>SCL_CTL TWI_SCL line state control bit When line control mode is enabled (bit[2] set), value of this bit decide the output level of TWI_SCL 0 – output low level 1 – output high level</p>
2	R/W	0	<p>SCL_CTL_EN TWI_SCL line state control enable When this bit is set, the state of TWI_SCL is control by the value of bit[3]. 0-disable TWI_SCL line control mode 1-enable TWI_SCL line control mode</p>
1	R/W	1	<p>SDA_CTL TWI_SDA line state control bit When line control mode is enabled (bit[0] set), value of this bit decide the output level of TWI_SDA 0 – output low level 1 – output high level</p>
0	R/W	0	<p>SDA_CTL_EN TWI_SDA line state control enable When this bit is set, the state of TWI_SDA is control by the value of bit[1]. 0-disable TWI_SDA line control mode 1-enable TWI_SDA line control mode</p>