

---

**Microchip Technology Inc.**

***dsPIC***<sup>®</sup>  
**Digital Signal Controller**

Serial Communications using  
the dsPIC30F I<sup>2</sup>C™ Module

---

© 2005 Microchip Technology Incorporated. All Rights Reserved.      Serial Communications using the dsPIC30F I<sup>2</sup>C Module      1

Welcome to the “Serial Communications using the dsPIC30F I<sup>2</sup>C Module” web seminar.



## Session Agenda

- Module Overview
- I<sup>2</sup>C Message Sequence
- I<sup>2</sup>C Slave Addressing Modes
- Additional Features

In today's session, we will start by outlining some key features of the Inter-Integrated Circuit, or I<sup>2</sup>C, module in the dsPIC30F family of devices. We will then delve deeper into the processes of data transmission and reception through the I<sup>2</sup>C, as well as various slave device addressing techniques. Finally, we will study some additional features of the I<sup>2</sup>C module.



## I<sup>2</sup>C - Overview

- Serial transmission and reception of 8-bit data
  - Half-duplex communication over a 2-wire interface
  - Supports 100 kHz and 400 kHz bus specifications
  - Independent Master and Slave operation
  - 7-bit and 10-bit Addressing
  - Clock stretching

© 2005 Microchip Technology Incorporated. All Rights Reserved.

Serial Communications using the dsPIC30F I<sup>2</sup>C Module

3

The I<sup>2</sup>C module is used for transmission and reception of 8-bit data over a 2-wire interface consisting of a serial clock line, SCL, and a serial data line, SDA.

I<sup>2</sup>C communication is half-duplex, which means either a transmission or a reception can occur at a time. It is also an example of Synchronous communication, as both communicating entities use the same serial clock signal.

The I<sup>2</sup>C module may be used for communicating with external peripherals such as Analog-to-Digital or Digital-to-Analog Converters, external memory devices such as Serial EEPROMs, or other microcontrollers. It supports 100 kHz and 400 kHz baud rates as specified by the I<sup>2</sup>C standard. In general, baud rates of up to 1 MHz are supported.

The Master and Slave logic in the I<sup>2</sup>C module operate independently, rather than the user having to configure the module as either a Master or a Slave.

The I<sup>2</sup>C module supports 7-bit or 10-bit Slave address detection, besides processing of special addresses such as General Call Address and supporting the Intelligent Peripheral Management Interface, or IPMI, addressing.

In addition, the module includes integrated signal conditioning features such as Slew Rate Control and support for SMBus signal levels.



## I<sup>2</sup>C - Applications

- Interfacing with memory devices
  - Serial EEPROMs - e.g., 24XX512
- Interfacing with external ADC devices
  - MCP3221
- Interfacing with communication chips
  - Bluetooth
- Boot Loaders

© 2005 Microchip Technology Incorporated. All Rights Reserved.

Serial Communications using the dsPIC30F I<sup>2</sup>C Module

4

The I<sup>2</sup>C module can be utilized to interface and communicate with a wide variety of external peripheral devices.

For example, it can be used to program and read external memory devices, such as the 24xx512 Serial EEPROM from Microchip, or to interface with external Analog-to-Digital Converters, such as the MCP3221 from Microchip.

Another popular use of the I<sup>2</sup>C module is as a control channel for dedicated communication processors running a Bluetooth protocol stack for wireless communications.

Last but not the least, I<sup>2</sup>C is a very efficient communication channel for a Boot Loader program running on the dsPIC, to exchange command and response messages with an external Host device.



## I<sup>2</sup>C - Master / Slave

- Independent Master and Slave logic
  - No separate Master and Slave modes
  - There may be one or more Masters and one or more Slaves in a system
  - Master generates a serial clock pulse (on SCL pin) for each bit of the data
  - Master also controls the data transfer sequence and direction
    - Transmission or Reception
    - Slave active when addressed by Master

© 2005 Microchip Technology Incorporated. All Rights Reserved.

Serial Communications using the dsPIC30F I<sup>2</sup>C Module

5

The I<sup>2</sup>C module does not have separate Master and Slave modes; Master or Slave states are defined by the actions performed by the module at any given time. In a system, there may be multiple Masters and Slaves, though only one Master may control the bus at a time.

The Master is responsible for generating serial clock pulses on the SCL line that will be used to shift out data through the SDA line. The Master also determines whether the data will be transmitted by the Master and received by the Slave, or transmitted by the Slave and received by the Master.

Slave operations are activated only when the Slave logic detects that the Master has addressed it.



## I<sup>2</sup>C - Registers

- I<sup>2</sup>C module has the following memory-mapped registers:
  - I2CTRN - Data Transmit Register
  - I2CRCV - Data Receive Register
  - I2CBRG - Baud Rate Generator
  - I2CCON - I<sup>2</sup>C Control Register
  - I2CSTAT - I<sup>2</sup>C Status Register

© 2005 Microchip Technology Incorporated. All Rights Reserved.

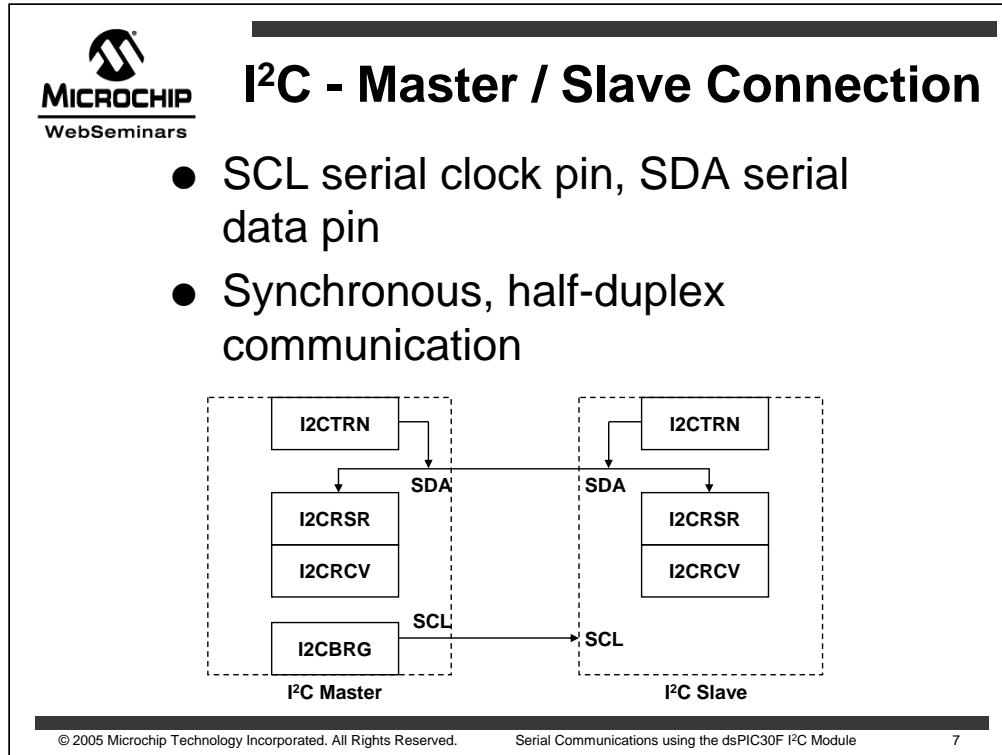
Serial Communications using the dsPIC30F I<sup>2</sup>C Module

6

The I<sup>2</sup>C module has a few control and status registers to control and monitor the functioning of the module. These include a Transmit Register and a Receive Register, as well as a Baud Rate Register.

The I<sup>2</sup>C Control Register has several bits to configure various parameters of the module such as addressing format, as well as to initiate specific real-time operating conditions such as START or STOP. All these bits can be read and written by the user.

The I<sup>2</sup>C Status Register mainly contains status flag bits that indicate various conditions that occur during a message sequence, as well as certain error conditions like overflows. Most of these bits are read-only.



As shown in this figure, two devices communicating through I<sup>2</sup>C are interconnected with their SCL and SDA pins tied together. Depending on the data transfer direction, data is either shifted out through the I<sup>2</sup>C Transmit Register, which also serves as a Transmit Shift Register, or shifted in through the I<sup>2</sup>C Receive Shift Register.

Data received in the I<sup>2</sup>C Receive Shift Register is transferred to the I<sup>2</sup>C Receive Register, which is memory-mapped and therefore user-accessible.



## I<sup>2</sup>C - Baud Rate Generator

- Dedicated 9-bit Baud Rate Generator
- Baud Rate set by I2CBRG register
  - **Baud Rate =  $F_{cy} / (I2CBRG + 1)$**   
where  $F_{cy}$  = Instruction Cycle Frequency
- Bits are transmitted and/or received at the rate defined by the Baud Rate as calculated above
  - Maximum Baud Rate is 1 MHz

© 2005 Microchip Technology Incorporated. All Rights Reserved.


Serial Communications using the dsPIC30F I<sup>2</sup>C Module

8

The I<sup>2</sup>C module contains a dedicated 9-bit Baud Rate Generator, which is used to provide timing synchronization of data transfer and all other I<sup>2</sup>C events. The Baud Rate Generator is memory-mapped, and can be configured by the user to generate serial clocks of up to 1 MHz. In I<sup>2</sup>C systems, 100 kHz and 400 kHz are the most common bit rates.

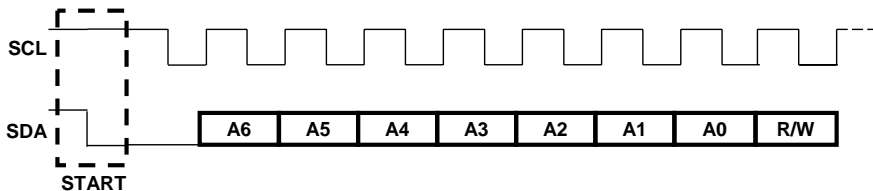
If an I<sup>2</sup>C module only performs Slave operations, then the Baud Rate Generator need not be initialized, since a Master in the system would provide the serial clock pulses required for communications.





## I<sup>2</sup>C - Message Sequence

- I<sup>2</sup>C module enabled by setting the I2CEN bit in I2CCON register
- Message begins with START condition
  - SDA pin goes high to low, SCL stays high
  - Master triggers START by setting SEN bit in I2CCON register




© 2005 Microchip Technology Incorporated. All Rights Reserved. Serial Communications using the dsPIC30F I<sup>2</sup>C Module 9

Let us take a look at the various phases of a data transfer over an I<sup>2</sup>C channel.

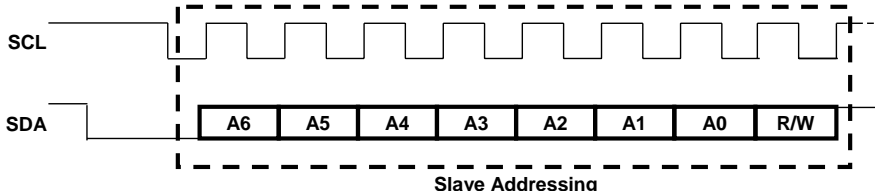
Setting the I<sup>2</sup>C Enable bit in the I<sup>2</sup>C Control Register enables the module. Subsequently, an I<sup>2</sup>C device can initiate communications on the Serial Data line by setting the START Enable bit in the I<sup>2</sup>C Control Register, thereby initiating a Master START sequence.

A START condition is characterized by the Serial Data line going to a low logic level while the Serial clock line is still high. Once a START condition has occurred, the Slave waits for the Master to send a Slave address byte over the Serial Data line.



## I<sup>2</sup>C - Message Sequence (contd.)

- Slave is activated when Master writes the 7-bit Slave address to I2CTRNL
  - Transmitted R/W bit indicates data direction
    - Slave Transmission (R/W = 1) or Slave Reception (R/W = 0)
  - The byte in I2CTRNL is transmitted



© 2005 Microchip Technology Incorporated. All Rights Reserved. Serial Communications using the dsPIC30F I<sup>2</sup>C Module 10

The Master writes the Slave address byte to the I<sup>2</sup>C Transmit Register. This byte is immediately transmitted, starting with the MSB and ending with the LSB. The address byte consists of 7 address bits, followed by a 'Read-Write' bit which signals to the Slave whether the Master wants to initiate a reception or a transmission; in other words, it signals the required data transfer direction. If the Read-Write bit is cleared, it implies that the Master wants to send data to the Slave. If the Read-Write bit is set, the Master is expecting to receive data from the Slave.



## I<sup>2</sup>C - Message Sequence (contd.)

- Slave address is defined by the value written to the I2CADD register of Slave
  - Address received by Slave in I2CRCV register only if it matches I2CADD
- Slave automatically acknowledges every byte received from Master
  - If address match occurred, Slave pulls SDA low during 9th bit time (ACK)
  - If address did not match, Slave lets SDA remain high (NACK)

© 2005 Microchip Technology Incorporated. All Rights Reserved.

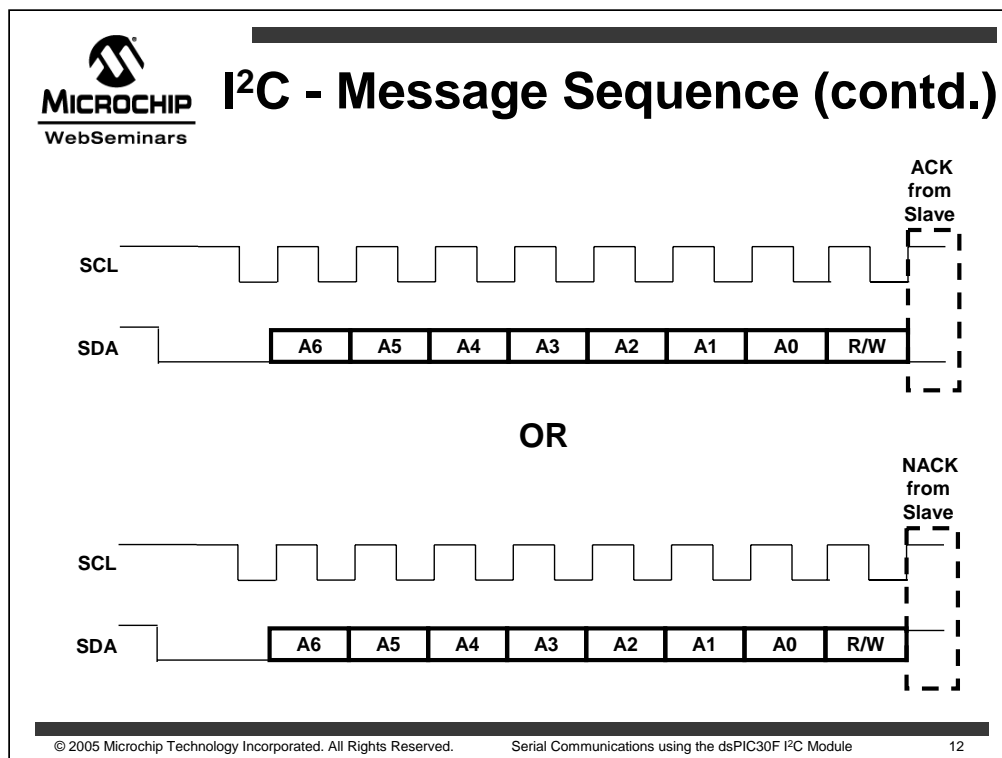
Serial Communications using the dsPIC30F I<sup>2</sup>C Module

11

Prior to any communication, the Slave software must assign a Slave address to the device by writing to the I<sup>2</sup>C Address Register. During communication, when an address byte is received from the Master, the Slave hardware compares this received address with the I<sup>2</sup>C Address Register. If the two addresses match, then the Slave hardware automatically concludes that this particular Slave device is being addressed by the Master.

Whenever an address match occurs, the received address byte is transferred to the user-accessible I<sup>2</sup>C Receive Register. During the ninth Serial Clock pulse, the Slave drives the Serial Data line low, which signifies an 'Acknowledge' condition.

However, if the received address did not match the pre-assigned I<sup>2</sup>C Address Register value, then the Slave allows the Serial Data line to float high, indicating a 'Not Acknowledge' condition. In this case, the address byte is not transferred to the I<sup>2</sup>C Receive Register, and the Slave hardware continues to wait for an address match.



These timing diagrams illustrate the sequence of events that occur during a Slave addressing sequence.



## I<sup>2</sup>C - Message Sequence (contd.)

- Master can determine ACK/NACK status by polling the ACKSTAT bit in I2CSTAT register
- Slave can determine if last byte received was address or data, by polling the D\_A bit in I2CSTAT
  - If it was an address, Slave can determine value of received R/W bit by polling the R\_W bit in I2CSTAT


© 2005 Microchip Technology Incorporated. All Rights Reserved.

Serial Communications using the dsPIC30F I<sup>2</sup>C Module

13

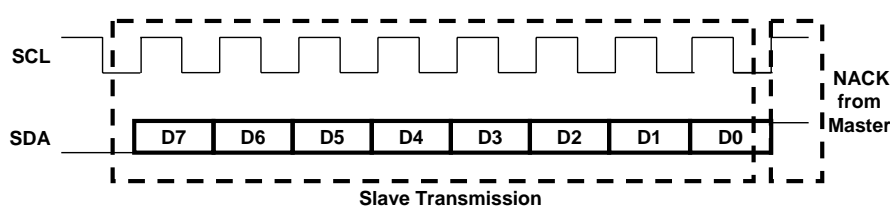
Now, how does the Master ascertain if an Acknowledge was received? It can do so by reading the value of the Acknowledge Status bit in the I<sup>2</sup>C Status Register. This bit reflects the state of the Serial Data line during the 9th clock pulse of the Address transmission sequence. Therefore, a value of zero indicates an Acknowledge received from the Slave.

Whenever a byte is transferred to the I<sup>2</sup>C Receive Register, the Slave software needs to determine if the byte received was address or data, and if it was an address, whether the Master requested a Read or a Write. This may be accomplished by first reading the Data Or Address bit in the I<sup>2</sup>C Status Register. If this bit is found to be cleared, the Slave software should then read the Read-Write bit in the same register. If the Read-Write bit is set, then the Slave prepares to send data to the Master, whereas if the bit is cleared, then the Slave waits for one or more data bytes from the Master.



## I<sup>2</sup>C - Message Sequence (contd.)

- Slave Transmission (if R/W = 1)
  - Slave writes each data byte to I2CTRN
  - Master software sets ACKEN bit in I2CCON
    - ACKDT bit in I2CCON selects ACK/NACK
  - If Slave receives ACK, it transmits next byte
  - If Slave receives NACK, it stops transmitting



© 2005 Microchip Technology Incorporated. All Rights Reserved. Serial Communications using the dsPIC30F I<sup>2</sup>C Module 14

If a Read-Write bit value of '1' was received from the Master, a Slave Transmission is initiated.


The Slave responds to a Slave Transmission request by writing a data byte to its I<sup>2</sup>C Transmit Register. The data is shifted out using a series of 8 Serial Clock pulses generated by the Master.

On the 9th Serial Clock pulse, the Master software initiates either an Acknowledge or a Not Acknowledge condition. An Acknowledge condition at this stage indicates that the Master received the data and wants the Slave software to continue transmitting data. A Not Acknowledge condition indicates that the Master wants the Slave software to stop transmitting data.

The Master software can generate an Acknowledge by first clearing the Acknowledge Data bit in the I<sup>2</sup>C Control Register and then setting the Acknowledge Sequence Enable bit in the same register.

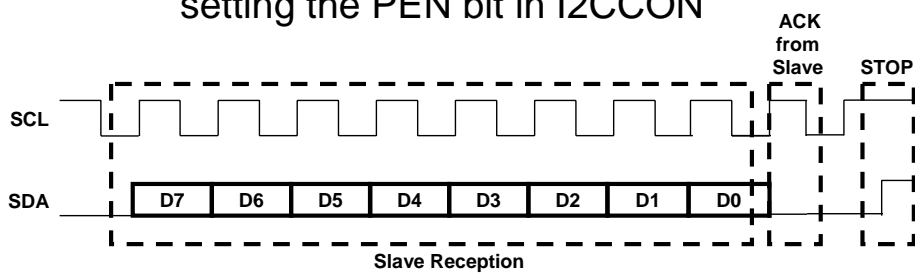
In order to generate a Not Acknowledge, the Master software must first set the Acknowledge Data bit and then set the Acknowledge Sequence Enable bit.

Note that an Acknowledge or Not Acknowledge sequence from the Master is always controlled by software, whereas an Acknowledge or Not Acknowledge sequence from the Slave is automatically generated by the hardware.



## I<sup>2</sup>C - Message Sequence (contd.)

- Slave Reception (if R/W = 0)
  - Slave reads each data byte from its I2CRCV register
  - Slave continues receiving bytes until Master initiates a STOP condition by setting the PEN bit in I2CCON



© 2005 Microchip Technology Incorporated. All Rights Reserved. Serial Communications using the dsPIC30F I<sup>2</sup>C Module 15

If a Read-Write bit value of '0' was received from the Master, a Slave Reception is initiated.

The Slave responds to a Slave Reception request by simply waiting for the Master to transmit data. The data is shifted in using a series of 8 Serial Clock pulses generated by the Master, and the Slave hardware always generates an Acknowledge condition during the 9th Serial Clock pulse. The received data can be read by the Slave software from the I<sup>2</sup>C Receive Register.

At this point, the Master can continue to send data to the Slave, and the Slave will continue receiving the data. After receiving an Acknowledge from the Slave, if the Master wants to stop transmitting data, then the Master software should initiate a STOP sequence.

A STOP condition is characterized by the Serial Data line going to a high logic level while the Serial Clock line is still high. Once a STOP condition has occurred, the Slave stops expecting data from the Master, and instead, waits for the Master to send a Slave address byte.



## I<sup>2</sup>C - Message Sequence (contd.)

- Direction of data transfer can be changed by Master initiating a 'Repeated START' condition followed by a Slave Address byte with suitable R/W bit
  - Repeated START condition is initiated by setting the RSEN bit in I2CCON

© 2005 Microchip Technology Incorporated. All Rights Reserved.


Serial Communications using the dsPIC30F I<sup>2</sup>C Module

16

At any time during the data transfer sequence, the Master can reverse the direction of data transfer by performing a Repeated Start, or RESTART, condition. This is initiated by setting the Restart Enable bit in the I2CCON register.

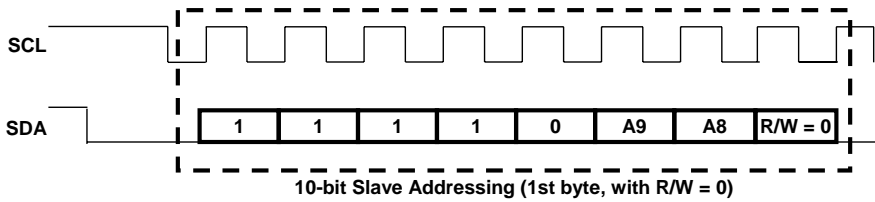
After the RESTART condition is completed, the Master would need to resend the address byte with the required Read-Write bit value.





## I<sup>2</sup>C 10-bit Addressing

- 10-bit Addressing enabled when Master sets A10M control bit in I2CCON
- Master can start address sequence by transmitting upper 2 bits of 10-bit address
- If A9 and A8 match the corresponding bits in Slave's I2CADD, it waits for next byte




© 2005 Microchip Technology Incorporated. All Rights Reserved. Serial Communications using the dsPIC30F I<sup>2</sup>C Module 17

Besides detection of 7-bit Slave Addresses, the I<sup>2</sup>C module also supports 10-bit Slave Addressing. This special mode is enabled by setting the 10-bit Slave Address Enable bit in the I<sup>2</sup>C Control Register. In this case, the Slave needs to initialize its I<sup>2</sup>C Address Register with a 10-bit value.

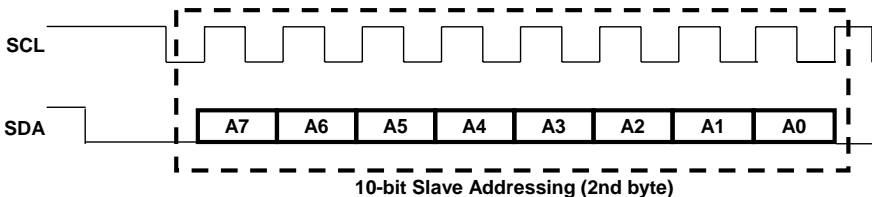
An I<sup>2</sup>C Master can initiate a 10-bit addressing sequence by transmitting a byte which consists of a pre-defined upper address preamble '11110', the upper two bits of the 10-bit Slave address, and a cleared Read-Write bit. The Read-Write bit must be clear at this stage because the address has not been completely transmitted yet.

When the Slave hardware detects this specific '11110' bit pattern, it compares the upper two bits of the received Address with the upper 2 bits of the I<sup>2</sup>C Address Register contents. If the bits matched, the Slave hardware sends an Acknowledge, otherwise it sends a Not Acknowledge.



## I<sup>2</sup>C 10-bit Addressing (contd.)

- Master then transmits lower byte of address
  - If this byte matches lower 8 bits in I2CADD, ADD10 bit in I2CSTAT is set
  - For Slave Reception, Master can start sending data bytes
  - For Slave Transmission, Master must initiate a Repeated START condition



© 2005 Microchip Technology Incorporated. All Rights Reserved. Serial Communications using the dsPIC30F I<sup>2</sup>C Module 18

Following the transmission of the first address byte, the Master transmits a second byte, which contains the lower 8 bits of the Slave address.

The Slave hardware compares these 8 bits with the lower 8 bits of the I<sup>2</sup>C Address Register, and if a match occurred, it again sends an Acknowledge. Moreover, if all 10 bits of the received address matched those of the I<sup>2</sup>C Address Register, then the 10-bit Address Status bit in the I<sup>2</sup>C Status Register is set.

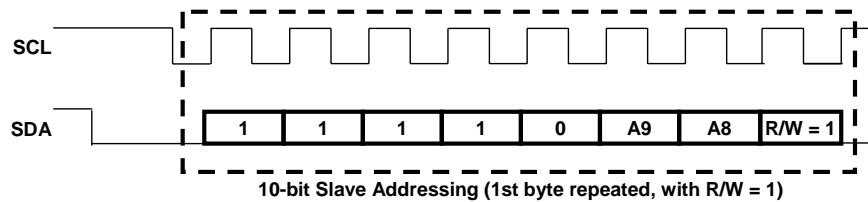
Now, if Slave Reception is required, the Master can simply start transmitting data bytes to the Slave without re-sending either of the address bytes, since a Read-Write bit value of '0' has already been sent.

However, if Slave Transmission is required, the Master is required to initiate a Repeated START sequence, which is used to invert the data transfer direction.



## I<sup>2</sup>C 10-bit Addressing (contd.)

- Following the Repeated START condition, Master must resend the 1st byte of the address
  - In this case, the R/W bit transmitted is 1



© 2005 Microchip Technology Incorporated. All Rights Reserved.

Serial Communications using the dsPIC30F I<sup>2</sup>C Module

19

After the Repeated START condition, the Master software should re-transmit only the first Address byte, as shown. However, this time the Read-Write bit should be set, not cleared, in order to request a Slave Transmission.

The transmission or reception of data bytes can now proceed, and continues until the Master generates a STOP condition.



## I<sup>2</sup>C - General Call Addressing

- General Call (GC) Addressing
  - General Call Address (0x00) can be used by Master to address all Slaves, irrespective of I2CADD values
  - Enabled for a Slave by setting the GCEN bit in I2CCON
  - Master must transmit R/W = 0 (Slave Reception) in this mode
  - Address Match status can be determined by polling GCSTAT bit in I2CSTAT


© 2005 Microchip Technology Incorporated. All Rights Reserved.

Serial Communications using the dsPIC30F I<sup>2</sup>C Module

20

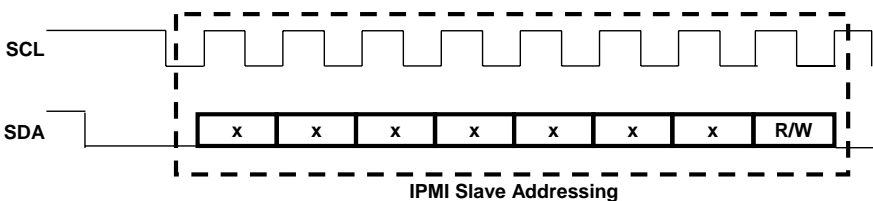
The I<sup>2</sup>C module includes support for a special address known as the General Call Address. This address, which consists of seven zeroes and a cleared Read-Write bit, signals to every Slave in the system that it is being addressed, irrespective of the value of its internal I<sup>2</sup>C Address Register.

General Call Addressing can be enabled by setting the General Call Enable bit in the I<sup>2</sup>C Control Register. When a General Call Address is received by a Slave, an Acknowledge condition is generated and the General Call Status bit in the I<sup>2</sup>C Status Register is set.



## I<sup>2</sup>C - IPMI Addressing

- Intelligent Peripheral Management Interface (IPMI) Addressing
  - Can be used to configure an I<sup>2</sup>C Slave to accept and respond to all addresses
  - Enabled for a Slave by setting the IPMIEN bit in I2CCON




© 2005 Microchip Technology Incorporated. All Rights Reserved. Serial Communications using the dsPIC30F I<sup>2</sup>C Module 21

Another special Slave Addressing feature in the I<sup>2</sup>C module is the Intelligent Peripheral Management Interface, or IPMI, mode. The IPMI bus uses I<sup>2</sup>C nodes as message repeaters in a distributed network. To allow a node to repeat all messages, the Slave module must accept all messages, regardless of the device address.

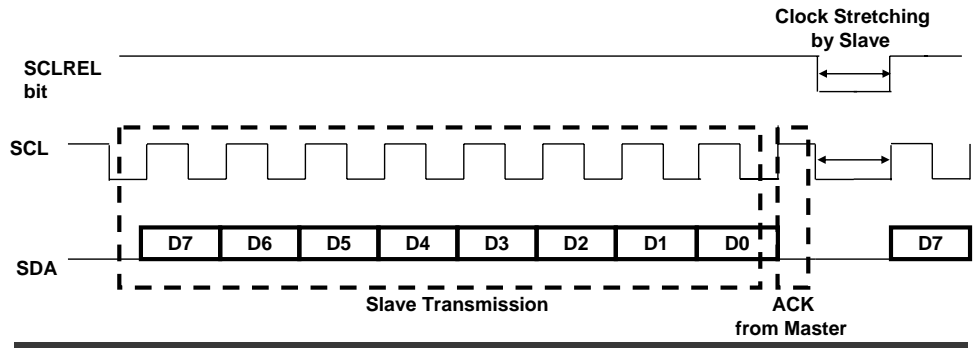
If IPMI mode is enabled by setting the IPMI Enable bit in the I<sup>2</sup>C Control Register, the Slave acknowledges and responds to any Slave address received, irrespective of the I<sup>2</sup>C Address Register value.

Besides its primary usage in an IPMI bus, this feature can also be used in any system by a device for monitoring messages intended for other recipients.



## Clock Stretching during Slave Transmission

- When Slave receives an Address Byte with R/W = 1 or an ACK, SCLREL bit in I2CCON gets cleared, which holds SCL low
  - After writing I2CTRN, Slave must set SCLREL



The diagram illustrates the clock stretching process. The SCLREL bit (top) is initially high. When the slave receives an ACK from the master, it clears the SCLREL bit, pulling the SCL line low. This period is labeled 'Clock Stretching by Slave'. The SCL line remains low until the slave has finished its transmission. The SDA line (bottom) shows a sequence of data bytes (D7 to D0) being transmitted by the slave. The master sends an ACK (low pulse) after the D0 byte. The SCL line is then released (goes high) after the slave finishes its transmission.

© 2005 Microchip Technology Incorporated. All Rights Reserved. Serial Communications using the dsPIC30F I<sup>2</sup>C Module 22

The I<sup>2</sup>C module contains a special feature called Clock Stretching. Clock Stretching is a mechanism by which a Slave can temporarily hold the Serial Clock line low, thereby allowing it sufficient time to perform certain operations expected of it.

During Slave Transmission, the Slave hardware always holds the clock line low every time it receives a Slave Transmission request from the Master, and also every time a data byte sent by the Slave is acknowledged by the Master. This indicates to the Master that the Slave is in the process of writing data to its I<sup>2</sup>C Transmit Register, and that it requires some time to do so. A Clock Stretching condition is indicated by a zero value in the Serial Clock Release bit in the I<sup>2</sup>C Control Register.

After the Slave software writes to the I<sup>2</sup>C Transmit Register, it must set the Serial Clock Release bit, thereby releasing the Serial Clock line and allowing data transmission to proceed.



## I<sup>2</sup>C - Clock Stretching during Slave Reception

- If STREN bit in I2CCON = 1, when Slave receives a byte, the SCLREL bit gets cleared, which holds SCL low
  - This prevents any further data transmission by the Master, thereby giving the Slave software time to read the received data
  - After reading the data in I2CRCV, Slave software must release SCL by setting the SCLREL bit
  - STREN = 1 also allows software-controlled stretching of the clock

© 2005 Microchip Technology Incorporated. All Rights Reserved.

Serial Communications using the dsPIC30F I<sup>2</sup>C Module

23

During Slave Reception, the Slave hardware can optionally hold the clock line low every time it receives a data byte in its I<sup>2</sup>C Receive Register. This indicates to the Master hardware that the Slave is in the process of reading the received data, and that it requires some time to do so. A Clock Stretching condition is indicated by a zero value in the Serial Clock Release bit in the I<sup>2</sup>C Control Register. This optional Clock Stretching during Slave Reception can be enabled by setting the Serial Clock Stretch Enable bit in the I<sup>2</sup>C Control Register.

After the Slave software reads the I<sup>2</sup>C Receive Register, it must set the Serial Clock Release bit, thereby releasing the Serial Clock line and allowing data transmission to proceed.

Moreover, if the Serial Clock Stretch Enable bit is set, then the Serial Clock can be manually held low at any time by the Slave software, simply by clearing the Serial Clock Release bit.



## I<sup>2</sup>C - Error Conditions

- Write Collision
  - Occurs if the user attempts to write to I2CTRN while I<sup>2</sup>C module is busy
  - Indicated by IWCOL bit in I2CSTAT
  - Avoided by writing to I2CTRN only when TBF bit in I2CSTAT is clear
- Receive Overflow
  - Occurs if a new byte has been received while I2CRCV is still holding previous byte
  - Indicated by I2COV bit in I2CSTAT
  - Avoided by reading I2CRCV as soon as RBF bit in I2CSTAT is set, before new byte is shifted in

© 2005 Microchip Technology Incorporated. All Rights Reserved.

Serial Communications using the dsPIC30F I<sup>2</sup>C Module

24

There are various error conditions that are flagged by the I<sup>2</sup>C module. These include Write Collision, Receive Overflow and Bus Collision.

A Write Collision occurs if the user writes to the I<sup>2</sup>C Transmit Register while a transmission is still going on. When this happens, the Write Collision Status bit in the I<sup>2</sup>C Status Register is set, and the attempted write is ignored.

To avoid Write Collision errors, the user should inspect the Transmit Buffer Full bit in the I<sup>2</sup>C Status Register. The I<sup>2</sup>C Transmit Register should be written to only when this bit is zero.

A Receive Overflow is generated when the I<sup>2</sup>C Receive Register contains some received data that has not been read yet, and a new data byte is completely shifted in through the I<sup>2</sup>C Receive Shift Register. In such cases, the newly received byte is discarded, and the Receive Overflow bit in the I<sup>2</sup>C Status Register is set.

Receive Overflow errors can be avoided by reading the I<sup>2</sup>C Receive Register as soon as an address or data byte is received, that is, as soon as the Receive Buffer Full bit in the I<sup>2</sup>C Status Register is set.





## I<sup>2</sup>C - Error Conditions (contd.)

- Master Bus Collision
  - Bus Collision typically occurs in a Multi-Master configuration, when one Master lets the SDA line float high and another Master drives SDA low
    - Only one Master can control the bus at a time
    - The Master that first pulls SDA low will win Bus Arbitration
    - The other Master will go into an idle state and set its BCL bit in I2CSTAT
  - Bus Arbitration and Collision Detection is performed by the module during START, Repeated START, STOP, ACK, Address and Data sequences

© 2005 Microchip Technology Incorporated. All Rights Reserved.

Serial Communications using the dsPIC30F I<sup>2</sup>C Module

25

The I<sup>2</sup>C module supports multi-master operation. A key feature that enables multi-master operation to function correctly is detection of Bus Collision errors.

A Bus Collision occurs when one Master in the system lets the Serial Data line float high, but detects a low logic level on the line. This indicates that another Master in the system has already gained control of the bus. The first Master then sets the Bus Collision Status bit in its I<sup>2</sup>C Status Register. A process known as Bus Arbitration causes the first Master to withdraw from the bus and wait for the bus to be idle before attempting any communication. This ensures robust operation, by allowing only one Master to control the bus at any given time.

Bus Arbitration is performed by the Master hardware at various stages of communication, such as START, Repeated START, STOP, Acknowledge, Slave Addressing and Data Transmission or Reception.



## I<sup>2</sup>C - Interrupts

- Master Interrupt
  - Indicated by MI2CIF bit and enabled by MI2CIE bit
  - Following events cause MI2C interrupt
    - Successful completion of START, RESTART or STOP condition
    - Transmission of ACK/NACK by Master
    - Data byte transmitted or received
    - Bus Collision

© 2005 Microchip Technology Incorporated. All Rights Reserved.

Serial Communications using the dsPIC30F I<sup>2</sup>C Module

26

The I<sup>2</sup>C module supports two interrupts, Master I<sup>2</sup>C Interrupt and Slave I<sup>2</sup>C Interrupt. As with other modules in the dsPIC30F, each of these two interrupts is serviced if it is enabled and its priority is higher than the current CPU interrupt priority.

The Master I<sup>2</sup>C Interrupt is generated for events related to Master operations. The various events that can cause a Master I<sup>2</sup>C Interrupt are listed here. These include events such as data transmission and reception, special sequences such as START, STOP and ACK, as well as Master Bus Collision.



## I<sup>2</sup>C - Interrupts (contd.)

- Slave Interrupt
  - Indicated by SI2CIF bit and enabled by SI2CIE bit
  - Following events can cause a Slave I<sup>2</sup>C interrupt
    - Detection of a valid Slave address byte
    - Reception of Slave Transmission request from Master
    - Reception of a data byte from Master

© 2005 Microchip Technology Incorporated. All Rights Reserved.

Serial Communications using the dsPIC30F I<sup>2</sup>C Module

27

The Slave I<sup>2</sup>C Interrupt is generated for events related to Slave operations. Events such as Slave address match and data reception can cause a Slave I<sup>2</sup>C Interrupt.



## I<sup>2</sup>C - Integrated Signal Conditioning

- Slew Rate Control
  - For 400 kHz operation, bus specification requires slew rate control of device pin out
  - Automatic Slew Rate Control
  - Disabled by setting DISSLW bit in I2CCON
- SMBus Support
  - For System Management Bus (SMBus), voltage levels are  $0.2 \cdot V_{DD}$  and  $0.8 \cdot V_{DD}$
  - Module automatically complies with SMBus input levels if SMEN bit in I2CCON is set

© 2005 Microchip Technology Incorporated. All Rights Reserved.

Serial Communications using the dsPIC30F I<sup>2</sup>C Module

28

The I<sup>2</sup>C hardware includes certain integrated signal conditioning features. For example, the Serial Clock and Data pins have an input filter to prevent glitches in the signal from affecting module operation.

When operating on a 400 kHz bus, the I<sup>2</sup>C specification requires slew rate control of the device pin output. This slew rate control is integrated within the dsPIC30F devices. Slew rate control is active when the Disable Slew Rate Control bit in the I<sup>2</sup>C Control Register is cleared. For communication speeds other than 400 kHz, the user should set the Disable Slew Rate Control bit, thereby disabling this feature.

A System Management Bus, or SMBus, system requires low and high voltage levels for its inputs that are different from those in a normal I<sup>2</sup>C system. The I<sup>2</sup>C module can optionally change the input signal levels so that they conform with the SMBus specifications. This signal conditioning feature is enabled by setting the SMBus Input Levels bit in the I<sup>2</sup>C Control Register.



## Key Support Documents

<u>Device Selection Reference</u>	<u>Document #</u>
● General Purpose and Sensor Family Data Sheet	DS70083
● Motor Control and Power Conv. Data Sheet	DS70082
● dsPIC30F Family Overview	DS70043
 <u>Base Design Reference</u>	 <u>Document #</u>
● dsPIC30F Family Reference Manual	DS70046
● dsPIC30F Programmer's Reference Manual	DS70030
● MPLAB® C30 C Compiler User's Guide	DS51284
● MPLAB ASM30, LINK30 & Utilities User's Guide	DS51317
● dsPIC® Language Tools Libraries User's Guide	DS51456

© 2005 Microchip Technology Incorporated. All Rights Reserved.

Serial Communications using the dsPIC30F I<sup>2</sup>C Module

29

For more information, here are references to some important documents that contain a wealth of information about the dsPIC30F family of devices.

The Family Reference Manual contains detailed information about the architecture and peripherals, whereas the Programmer's Reference Manual contains a thorough description of the instruction set.



## Key Support Documents

Device Specific Reference	Document #
● dsPIC30F2010 Data Sheet	DS70118
● dsPIC30F2011/2012/3012/3013 Data Sheet	DS70139
● dsPIC30F3010/3011 Data Sheet	DS70141
● dsPIC30F3014/4013 Data Sheet	DS70138
● dsPIC30F4011/4012 Data Sheet	DS70135
● dsPIC30F5011/5013 Data Sheet	DS70116
● dsPIC30F6010 Data Sheet	DS70119
● dsPIC30F6011/6012/6013/6014 Data Sheet	DS70117

Microchip Web Site: [www.microchip.com](http://www.microchip.com)

© 2005 Microchip Technology Incorporated. All Rights Reserved.

Serial Communications using the dsPIC30F I<sup>2</sup>C Module

30

For device-specific information such as pinout diagrams, packaging and electrical characteristics, the device datasheets listed here are the best source of information.

All these documents can be obtained from the Microchip web site shown, by clicking on the “dsPIC® Digital Signal Controllers” or “Technical Documentation” link.