ECE 372A Fall 2015 - Lecture 15

Garrett Vanhoy

October 15, 2015

Outline

Inter-Integrated Circuit

Reference Material

PIC32MX Family Reference Manual Section 24
Section 18 in the PIC32MX Data Sheet

Inter-Integrated Circuit

I2C Details

- I2C is Inter-integrated Circuit, I2C, I^2C, or '2-wire'

Inter-Integrated Circuit

I2C Details

- I2C is Inter-integrated Circuit, I2C, I^2C, or '2-wire'
- Half-duplex, serial, synchronous

Inter-Integrated Circuit

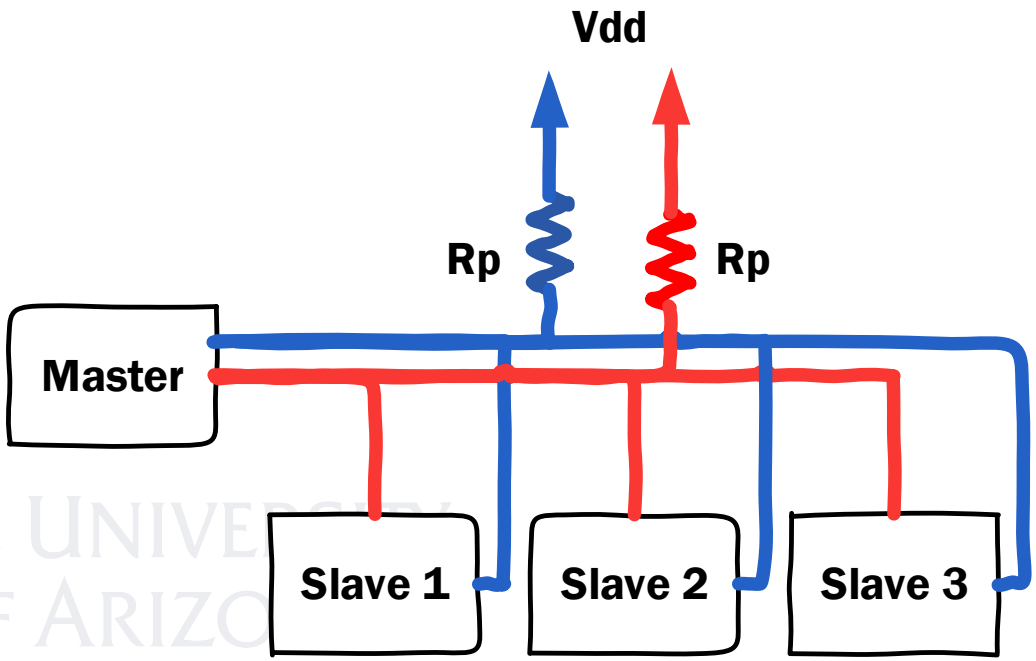
I2C Details

- I2C is Inter-integrated Circuit, I2C, I^2C, or '2-wire'
- Half-duplex, serial, synchronous
- Master-slave protocol using *acknowledgements*.

Inter-Integrated Circuit

I2C Idle State

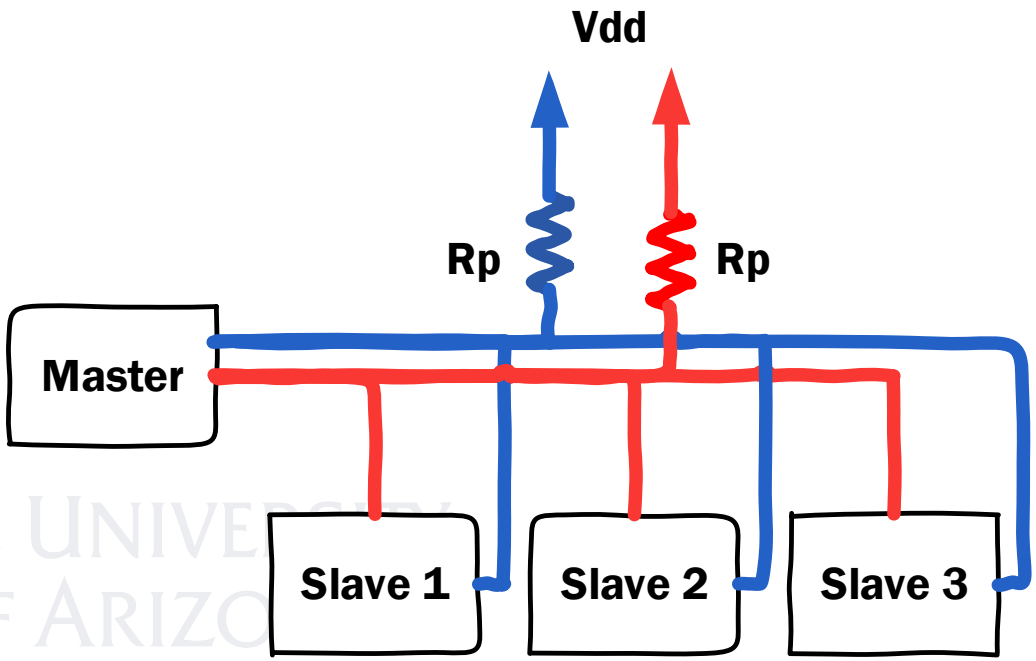
1. I2C lines are idle at logic high.



Inter-Integrated Circuit

I2C Idle State

1. I2C lines are idle at logic high.
2. If idle lines are logic low, then when one line drives the line high while the other drives it low, you can damage the devices.



Transmission Structure

First Byte

1. Start *condition*

Transmission Structure

First Byte

1. Start *condition*
2. Address bits (This can be 7 or 10 bits)

Transmission Structure

First Byte

1. Start *condition*
2. Address bits (This can be 7 or 10 bits)
3. Read/Write(not) bit
4. Acknowledgment

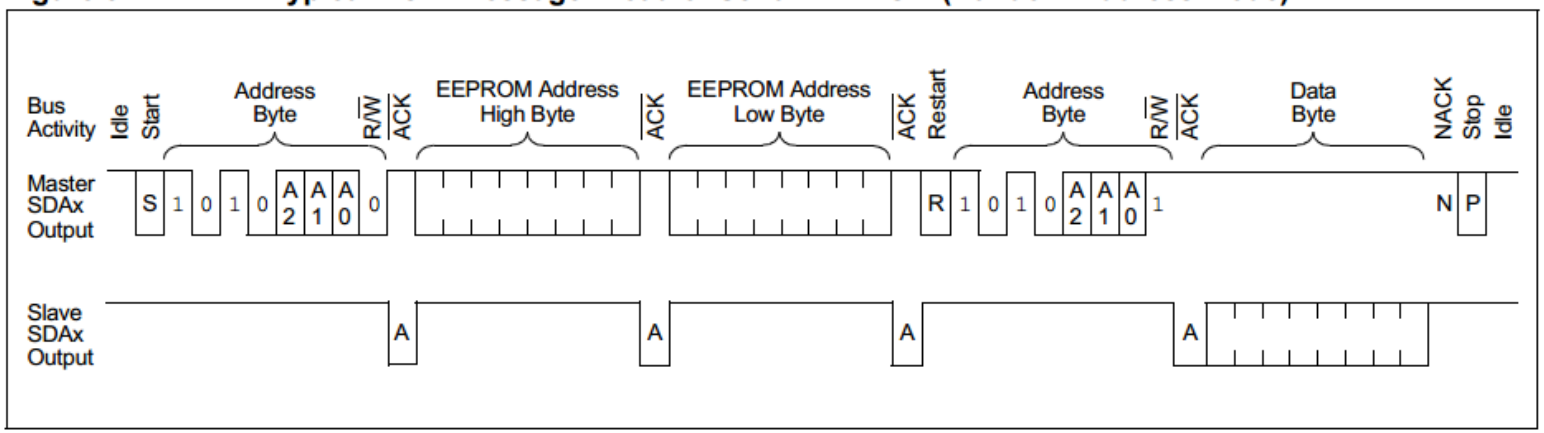
Transmission Structure

First Byte

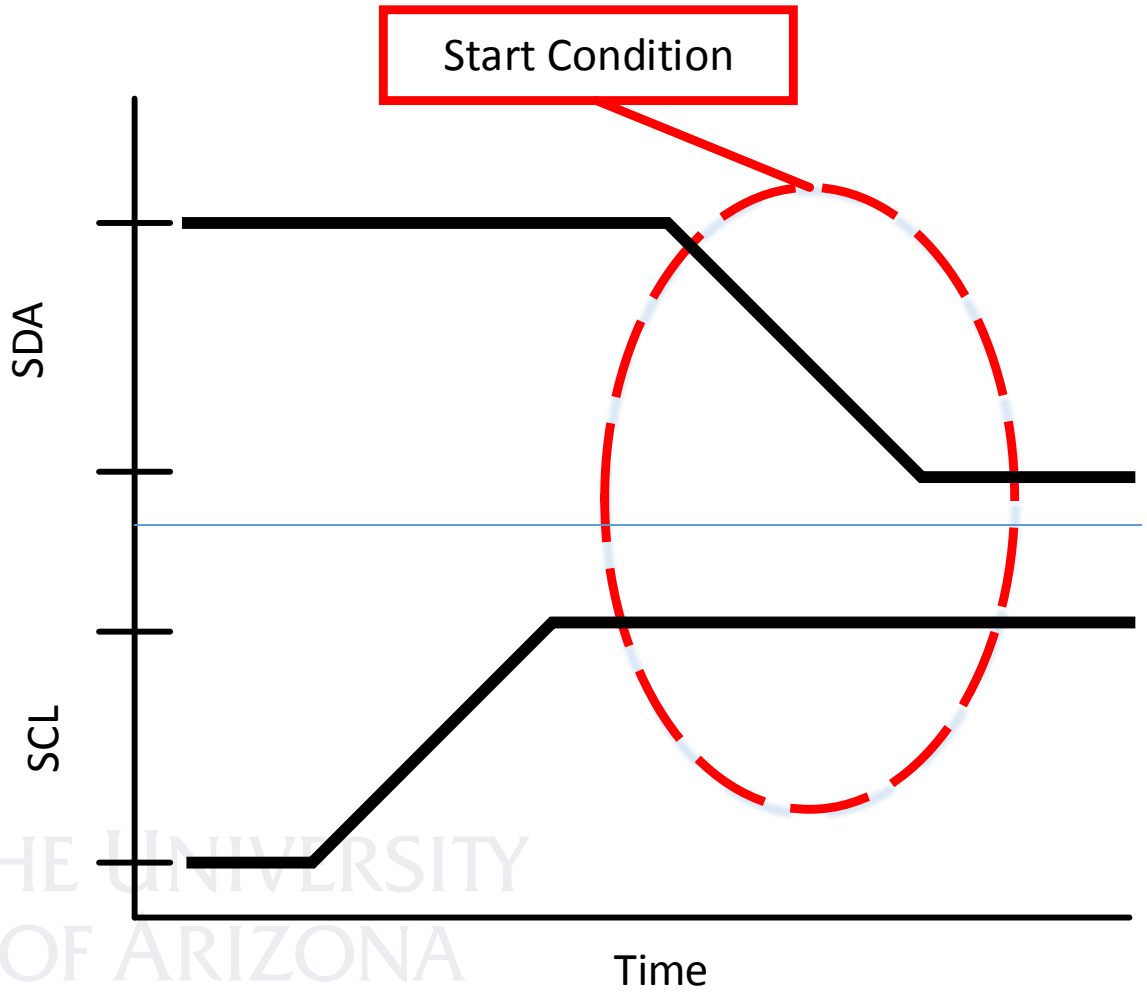
1. Start *condition*
2. Address bits (This can be 7 or 10 bits)
3. Read/Write(not) bit
4. Acknowledgment
5. Stop *condition*

Transmission Structure Example

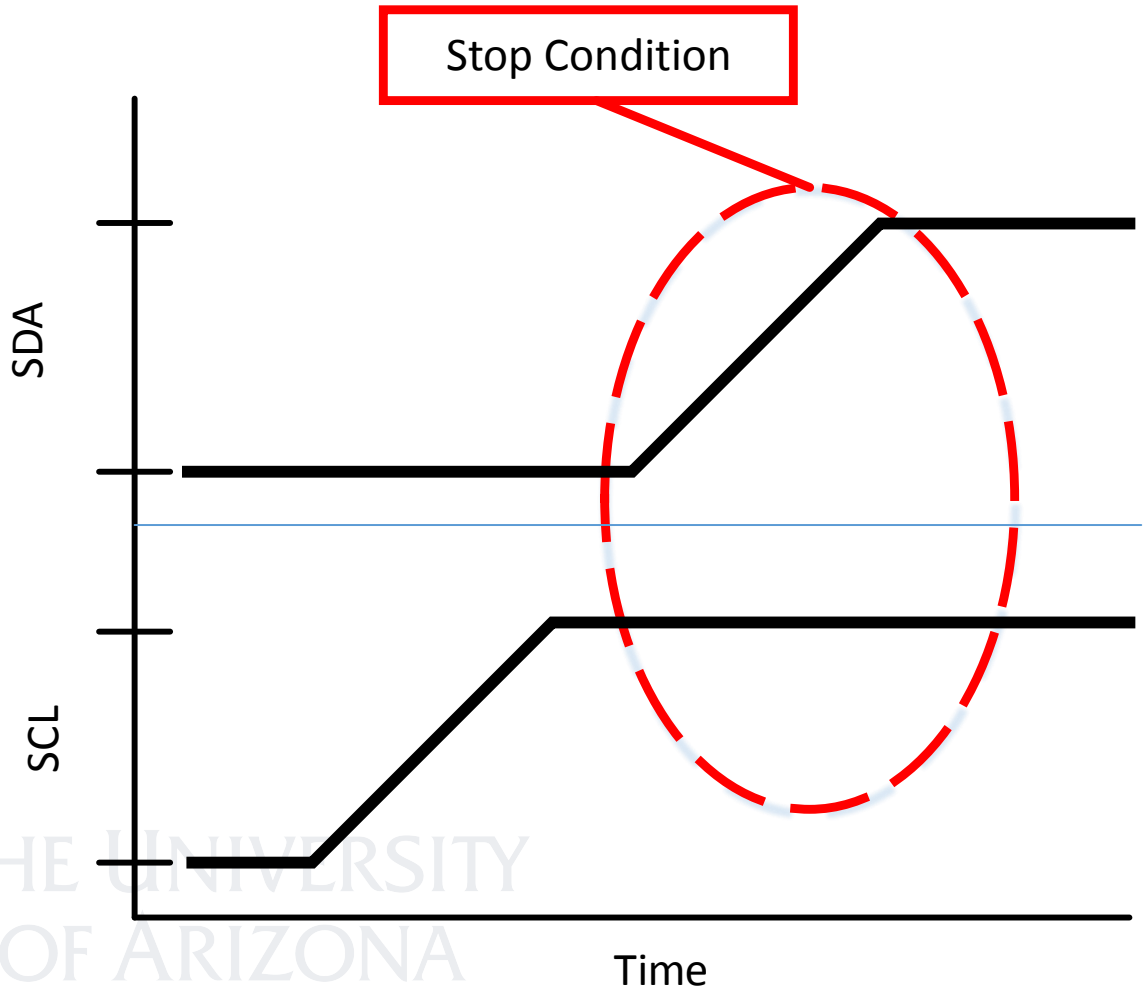
Figure 5-1: A Typical I²C™ Message: Read of Serial EEPROM (Random Address Mode)



Start Condition



Stop Condition



Repeated Start Condition

Read/Write

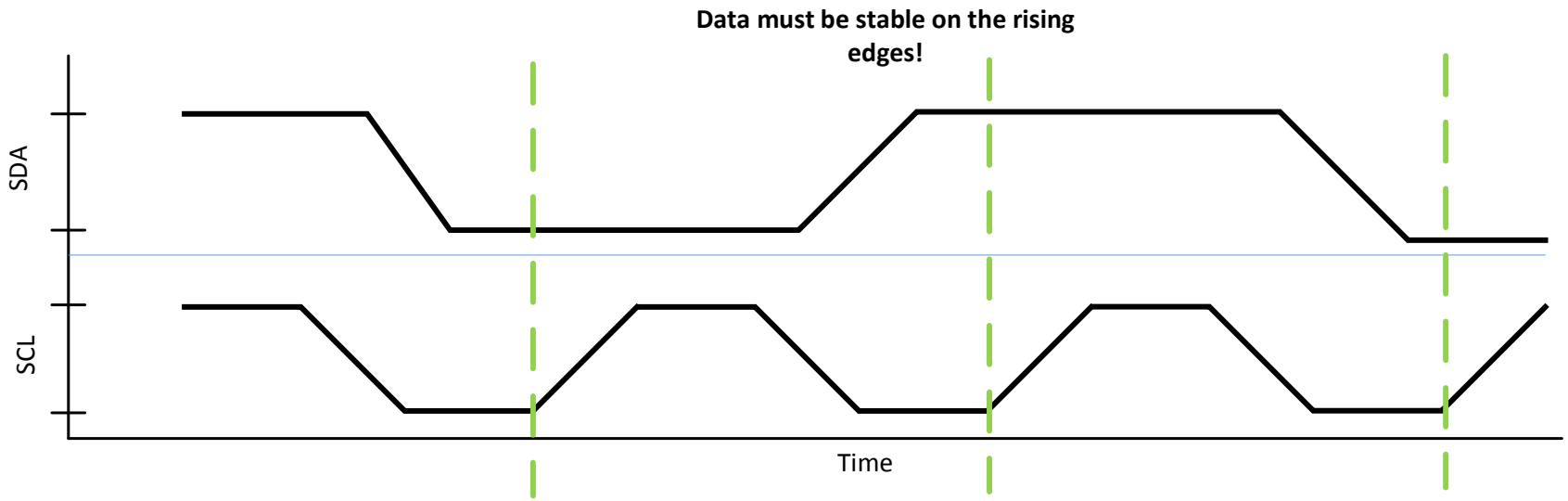
1. If a master wants to write, then read or vice versa then a repeated start will allow the master to keep control of the data line

Repeated Start Condition

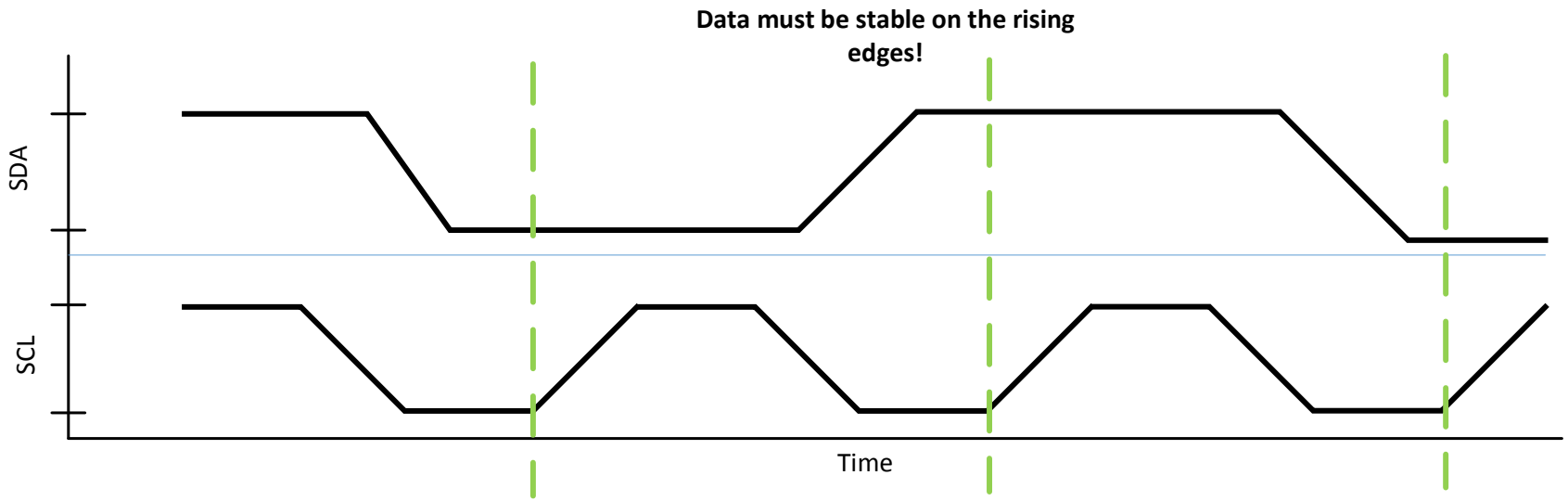
Read/Write

1. If a master wants to write, then read or vice versa then a repeated start will allow the master to keep control of the data line
2. A repeated start is a start and stop condition on the same clock cycle

Data



Data



Acknowledgements

Acknowledgements

- Acknowledgements (ACK) are when the receiver drives the line low.

Acknowledgements

Acknowledgements

- Acknowledgements (ACK) are when the receiver drives the line low.
- A NACK is when the receive drives the line high.

Acknowledgements

Acknowledgements

- Acknowledgements (ACK) are when the receiver drives the line low.
- A NACK is when the receive drives the line high.
- Acknowledgements, addresses, and read/write bits are data bits too, so they must be stable on rising edges.

Using I2C

Interrupts

- I2C is much more "manual" than other schemes.

Using I2C

Interrupts

- I2C is much more "manual" than other schemes.
- At the end of the start, transmission, reception, acknowledgement, stop, and repeated start all trigger the interrupt MI2CxIF

Generating Start Events

Start Events

1. Set the I2CxCON SEN bit to 1

Generating Start Events

Start Events

1. Set the I2CxCON SEN bit to 1
2. Wait for the interrupt to be generated

Transmitting Data

Transmiting Data

1. Load data into I2CxTRN

Transmitting Data

Transmiting Data

1. Load data into I2CxTRN
2. Wait for the interrupt again

Transmitting Data

Transmiting Data

1. Load data into I2CxTRN
2. Wait for the interrupt again
3. Check the ACK bit in the I2CxSTAT register (ACKSTAT bit)

Receiving Data

Receiving Data

1. An address is sent, read/write is set to *read*, and ack is received

Receiving Data

Receiving Data

1. An address is sent, read/write is set to *read*, and ack is received
2. Set RCEN bit in I2CxCON to 1, wait for the interrupt

Receiving Data

Receiving Data

1. An address is sent, read/write is set to *read*, and ack is received
2. Set RCEN bit in I2CxCON to 1, wait for the interrupt
3. Everything ends up in I2CxRCV

Acknowledging Data

Acknowledgements

1. ACKEN in I2CxCON is set

Acknowledging Data

Acknowledgements

1. ACKEN in I2CxCON is set
2. ACKDT in I2CxCON is determined by ACK or NACK

Stop Condition

Stop Condition

1. Enable PEN bit in I2CxCON

Stop Condition

Stop Condition

1. Enable PEN bit in I2CxCON
2. Wait for the interrupt