

Microprocessors B (17.384)

Spring 2011

Lecture Outline

Class # 11

April 12, 2011

Dohn Bowden

Today's Lecture

- Administrative
- Microcontroller Hardware and/or Interface
 - Advanced Hardware Topics
- Programming/Software
 - Programming Techniques
- Lab
- Homework

Course Admin

Administrative

- Admin for tonight ...
 - Syllabus Highlights
 - Exam #2 is next week ... April 19th
 - Lab #4 is due ... April 26th
 - Lab time ...
 - Continue/start Lab #4
 - Course Project
 - Course Project ...
 - If you finished your Labs ... start working your project

Syllabus Review



<i>Week</i>	<i>Date</i>	<i>Topics</i>	<i>Lab</i>	<i>Lab Report Due</i>
1	01/25/11	PIC pin out, C programming, Watchdog Timer, Sleep		
2	02/01/11	General-purpose IO, LED/switch IO, FSM	1	
3	02/08/11	Lab	1 con't	
4	02/15/11	Interrupts, Timers, interrupt-driven IO	2	
5	02/22/11	Lab	2 con't	1
6	03/01/11	Asynchronous and Synchronous Serial IO (UART, I²C, SPI)	3	
7	03/08/11	Examination 1		
X	03/15/11	No Class – Spring Break		
8	03/22/11	Lab	3 con't	2
9	03/29/11	Pulse Width Modulation and DC motor control	3, 4	
10	04/05/11	Lab	4 con't	3
11	04/12/11	Advanced Hardware Topics	Project	
12	04/19/11	Examination 2		
13	04/26/11	Work on Course Project	Project	4
14	05/03/11	Final Exam/Course Project Brief and Demonstration	Demo	



Microcontroller Hardware and / or Interfaces

Other Features / Peripherals ...

Direct memory Access ...

Direct memory Access (DMA)

- **Direct Memory Access ... DMA ...**
 - **A feature that allows certain hardware subsystems to access system memory for reading and/or writing independently of the CPU**
 - **Hardware assisted data transfers between memory and peripherals**
 - **Minimal CPU intervention**
 - **Reduces CPU overhead**

Direct memory Access (DMA)

- Without DMA ... using programmed input/output (PIO) mode for communication with peripheral devices ...
 - The CPU is typically fully occupied for the entire duration of the read or write operation, and is thus unavailable to perform other work
 - With DMA ... the CPU would initiate the transfer ... do other operations while the transfer is in progress ... and ...
 - Receive an interrupt from the DMA controller once the operation has been done

I²C ...

PIC24 as an I²C Slave

- **The PIC24 can also function as an I²C Slave**
 - **Useful when embedding within a product that requires an I²C interface**

Bus Arbitration for the I²C Bus

- **The I²C bus supports multiple bus masters**
 - **Any I²C device can initiate a transaction**
 - **Therefore there must be a bus arbitration mechanism that decides which device assumes control**

Controller Area Network ...

Controller Area Network (CAN)

- **Controller Area Network ... CAN ...**
 - **Is an half-duplex serial bus designed as a communication mechanism for intelligent peripherals within an automotive system**
 - **An automobile is a harsh environment**
 - **Electrical noise**
 - **Electrical systems throughout the vehicle**
 - **Communications in meters**

Controller Area Network (CAN)

- CAN buses ...
 - Uses only two wires for communications
 - Keeps cabling sizes to a minimum

Universal Serial Bus ...

Universal Serial Bus (USB)

- The Universal Serial Bus ... USB ...
 - High speed serial protocol
 - Largely replaced the RS-232
 - Some PIC24 microcontrollers has USB interface
 - PIC24FJ256GB110 ... for example

Real-Time Clocks ...

Real-Time Clocks

- External
- Internal

Analog Interface ...

Analog Interface

- **Analog to Digital Conversions**
- **Interfacing with and External ADC**
- **Comparator Module ...**
 - **Contains two independent comparators**
- **Temperature Sensors**
- **Light Sensors**
- **Accelerometer**
- **Pressure Sensors**

Other Interfaces ...

Other Interfaces

- **Liquid Crystal Displays**
- **Motors**
- **Stepper Motors**
- **Data Loggers**
- **PC interface**

Selecting Your Microcontroller ...

Product Selector Tool

- Microchip's Home Page ...
 - Product Selector Tool ...
 - Web address

<http://www.microchip.com/productselector/MCUProductSelector.html>

Mozilla Firefox (Private Browsing)

File Edit View History Bookmarks Tools Help

http://www.microchip.com/productselector/MCUProductSelector.html

Most Visited Getting Started Latest Headlines

http://www.microc...uctSelector.html

MICROCHIP Product Selector Tool

Product Selection Home Export PDF Reset

Architecture: 8 16 32

Max Speed (MHz): 1 2 4 8 16 32 64 128 256 512

Flash (KB): 0.1 0.2 0.5 1 2 4 8 16 32 64+

RAM (KB): 0 64 128 256 512 1024+

EEPROM (Bytes): 6 8 14 20 28 44 64 80 100+

Package Pins: 3.0 to 3.6

App. Voltage (V):

Low Power: Low Sleep Fast Wake Active Control XLP

SPI: 1+ 2+ 3+ 4+

I2C: 1+ 2+ 3+ 4+

UART: 1+ 2+ 3+ 4+

Other Comms: USB LIN CAN PSP Ethernet

8 Bit Timers: 1+ 2+ 3+ 4+

16 Bit Timers: 1+ 2+ 3+ 4+

32 Bit Timers: 1+ 2+ 3+ 4+

Real Time Clock: Timer H/W RTCC

PWM Channels: 1+ 2+ 3+ 4+

PWM Resolution (bits): 8+ 10+ 16+

Input Captures: 1+ 2+ 3+ 4+

ADC Channels: 0 4 8 12 16+

ADC Resolutions (bits): 8+ 10+ 12+

Comparators: 1+ 2+ 3+ 4+

Touch Channels: 0 4 8 12 16+

LCD Segments: 0 60 90 120 150 180+

Total Products: 577 To sort, click on the column header

Product Family	Architecture	5K \$ Pricing	Flash (KB)	EEPROM (Bytes)	RAM (KB)	CPU Speed (MHz, MIPS)	LowPower	Comparators	ADC Channels	ADC Bits	Total UART	SPI	I2C	USB	Ethernet	LIN	CAN	Total Timers	Input Capture	PWM Channels	Parallel Port	Segment LCD	Supply Voltage
PIC10F200	8	0.30	0.37	0	0.01	[4,1]	Yes		0			0	0					1	0	0		0	2 to 5.5
PIC10F204	8	0.33	0.37	0	0.01	[4,1]	Yes	1	0			0	0					1	0	0		0	2 to 5.5
PIC10F202	8	0.33	0.75	0	0.02	[4,1]	Yes		0			0	0					1	0	0		0	2 to 5.5
PIC10F206	8	0.36	0.75	0	0.02	[4,1]	Yes	1	0			0	0					1	0	0		0	2 to 5.5
PIC10F220	8	0.26	0.37	0	0.01	[8,2]	Yes		2	8		0	0					1	0	0		0	2 to 5.5
PIC16F54	8	0.39	0.75	0	0.02	[20,5]	Yes		0			0	0					1	0	0		0	2 to 5.5

Done

Programming/Software

Software enhancements ...

Assembly Language ...

Assembly Language

- PIC24 instruction set defines 71 distinct instructions
- Assembly language is more primitive ... operations less powerful ... than C
- It can ... however ... be used when C cannot perform the required task
- Can also embed assembly within C

C ...

C

- **Pointers and subroutines**
- **Indirect addressing**
- **Subroutines**

Math Routines ...

Math Routines

- **Multiplication**
- **Division**
- **Floating Point Numbers**
- **Solving equations**

Optimization Techniques ...

Optimization Techniques

- **Increase Code Efficiency**
- **Decreasing Code Size**
- **Reducing Memory Usage**
- **Power-Saving Techniques**

Increasing Code Efficiency ...

Increasing Code Efficiency

- **Inline functions**
- **Table lookups**
- **Hand-coded assembly**
 - **Some software modules are best written in assembly language**
 - **This gives the programmer an opportunity to make them as efficient as possible**
 - **Though most C compilers produce much better machine code than the average programmer ...**
 - **A skilled and experienced assembly programmer might do better work than the compiler for a given function**

Increasing Code Efficiency

- **Register variables**
 - The keyword `register` can be used when declaring local variables ...
 - This asks the compiler to place the variable into a general-purpose register rather than on the stack
- **Global variables**
 - It is sometimes more efficient to use a global variable than to pass a parameter to a function

Increasing Code Efficiency

- **Fixed-point arithmetic**
 - **Unless your target platform features a floating-point processor ...**
 - **You'll pay a very large penalty for manipulating float data in your program**

Increasing Code Efficiency

- Variable size
 - It is typically best to use the processor's native register width for variables whenever possible ... whether it is 8, 16, or 32 bits
 - This allows the compiler to produce code that takes advantage of the fast registers built into the processor's machine opcodes
 - For example ... if you need a count that goes from 0 to 512 ... you can't use an 8-bit variable

Decreasing Code Size ...

Decreasing Code Size

- **Avoid standard library routines**
 - **Avoid using large standard library routines**
 - **They try to handle all possible cases**
 - **It might be possible to implement a subset of the functionality yourself with significantly less code**

Decreasing Code Size

- Use goto statements
 - Good software engineering practice dictates against the use of this technique ... But ...
 - In a pinch ...goto statements ...
 - Can be used to remove complicated control structures

Power-Saving Techniques ...

Power-Saving Techniques

- A major concern in battery-powered embedded systems design is power consumption
- There are several methods to conserve power in an embedded system ... including ...
 - Clock control
 - Power-sensitive processors
 - Low-voltage ICs
- Some power-saving techniques are under software control.

Power-Saving Techniques

- It might seem ideal to select the fastest and most powerful processor available for a particular embedded system ...
 - However ... one of the tasks of the hardware designer is to use ...
 - Just enough processing power to enable the device to get its job done
 - This helps reduce the power consumed by the device

Power-Saving Techniques

- **Processor Modes**
- **Clock Frequency**
 - Reduce the clock speed to save power during the execution of noncritical tasks ... and ...
 - Increase the clock speed when processing demands are high

Lab

Peer Review of Software ...

Peer Review of Software Developed

- How did you write your code?
- What problems did you encountered?
- Any questions that you need resolved?

Next Class

Next Class Topics

- April 19th ... Exam #2
- Lab time ... projects
 - **REMINDER ... I expect to see you working in the lab**

Homework

Homework

- Labs ...
 - Lab #4
 - Lab #4 Report ... due April 26th
 - Projects
- Study for exam #2 ... next week ... April 19th

**Time to start
the lab ...**

Lab

- **Start / Continue Lab #4**
- **Start / Continue projects**

White Board ...

References

References

1. None