EE2361 - Lecture 40

12/14/16

- Review
- Evaluations

Post Lectures 39, 40

+ other odds & ends,

+ discussion 13 solutions

General Plan

Part 1. Intro to μC + assembly and architecture → Exam 1

Part 2. C coding for μC and basic peripherals (I/O, PPS, Timers)

Part 3. Advanced peripherals and other things → Exam 2

Final Exam

20 December 2016, 1:30p - 3:30p

In this room

Same format as midterms

5 or 6 problems

(weighted with regard to material)

Review

1. Basic Computer Systems and microcontrollers

2. Instruction Sets and some computer architecture (how do this fit together, how are instructions executed)

Basic to μC is memory map

1. Harvard or von Neuman Architecture

Instructions + data in separate memorys

PIC24F is a Harvard Arch

Inst + data in same memory

PIC24F

- Program Memory (Flash)
  24-bit locations, these have additional "dummy" 8-bits
- Data memory (SRAM)
  16-bit words
- SFRs (Special Function Registers)
  (at bottom of data memory)

- Machine instructions
  24-bits (main reference is the Programmers Ref Manual)
- Assembly language
  1-1 with machine code
- Higher level language (like C)
  (XC16 User Manual)

all access the Programmers Model

Assembly programming

Several Classes of instructions

- Arithmetic and logic, etc. operations (ADD, SUB, MUL...)
- Data Move instructions (NOV...)
- Control (BRA, JUMP...)

- Directives and pseudo-ops which tell the assembler what to do

- 6 addressing modes

  #value

  wn

  [wn]

  ⋮

The programmers model ⇒ registers to worry about

PC - prog counter

SR - status register

W0 - W15 - working registers

· Resets and Interrupts

Need to understand how an interrupt works.

Interrupts (PIC24F)

vectored interrupts

Interrupt Vector Table

3 ~~bits~~ registers for each interrupt

- → Flag bit IF_ _ 1-bit
- → Enable bit IE_ _ 1-bit
- → Interrupt Priority IP - 3 bits

difference between interrupt and trap

Data Structures

- FIFO queue
  (buffer)
- FILO queue
  (stack)

Peripherals

Control Registers which are in the section of data memory where the SFRs are located.

macros used in C to acess and manipulate bits

ex: PORTBbits.RB0
_RB0

Basic Peripherals

- I/O : Ports, Latch, TRIS registers
- PPS - peripheral pin select
- Timers : Timer1, Timers 2/3, 4/5

  TxCON, PRx

interrupts with peripherals

Advanced Peripherals

- Communication

  Asynchronous UART (no clock signal)

  Synchronous SPI, I2C (have a clock)

associated registers ⇒ control, data
baud rate

A/D peripheral

- How to configure pin's
- scanning /noscanning
- Sampling
- Converting

⇒ under stend the timing (Tad, Tcy)

Capture, Compare, and PWM

- Input ~~compare~~ capture
- Output compare
  - ↳ OC mode (1 or 2 timers)
  - ↳ PWM

Misc stuff

- Low-Power
- WDT