EE2361 - Lecture 16

10/14/16

New HW 3 —

Return Exam 1 - Monday

Interrupts

Some more interrupts —

An interrupt is a hardware event that send execution to the interrupt vector (IVT)

PIC24 - each peripheral interrupt has a flag bit, enable bit, priority

An interrupt occurs at any time!

⇒ you need to save context

Context is the registers which will allow the resumption of the code you were executing when the interrupt occured

Why save context?

```
        sub w0, w1, w0
        bra N, target
next:
```

suppose an interrupt occurs here

What is the context saved when an interrupt occurs?

saved by hardware (onto the stack)

⇒ the current PC (return address)

⇒ low byte of the status register (SRL)
OV, N, C, Z and IPL<2:0>

⇒ save IPL3 (CORCON<3>)

These are restored with the RETFIE

- Stack for interrupt
- RETFIE

How long does this take

⇒ 4 cycles (for the PIC24F devices

- details

What about priorities?

16 priority levels

- levels 8-15 are traps
  one trap for each priority level
  these are hardwired
- levels 0-7 are for peripherals
  you can change these

When an interrupt occurs the priority level of the interrupt is indicated by IPL3, IPL<2:0>

4-bits

Only interrupts with a priority level greater than this are acknowledge

An interrupt with priority 0 is essentially disabled

If you set this to 7 (set the IPL<2:0> bits, you disable all peripheral interrupts

IPL3 can be cleared but not set by the user

NSTDIS bit which can be set to cause interrupts not to be nested (in INTCON<15>)

Instruction DISI which can disable interrupts at priority level 7 or less for up to 16384 cycles

Two types of traps

Hard Traps — PIC24F Oscillator failure
Address Error

Soft Traps — Stack Error &
arithmetic error

See the family reference guide

Return to the timer example and look at the software

The hardware saves some registers automatically

Software saves the rest of the context

Example