ECE 376
Test #3 Review

Test Topics

Interrupts

- Timer Interrupts
- Edge Interrupts
- Timer1 Capture
- Timer1 Compare

Digital Filters

- z-Transforms
- Programming
- Converting from s-domain to z-domain

Interrupts

- Subroutine called by hardware
- Allows you to do several things at one time

Confusing

- Someone else is calling subroutines in your program

INT Interrupts

- Trigger an interrupt on a rising or falling edge
- Can use several pins
  - RB0: INT0 interrupt
  - RB1: INT1 interrupt
  - RB2: INT2 interrupt
  - RA4: Timer0 interrupt (T0CS = 1)
  - RC0: Timer1 interrupt (TMR1CS = 1)
  - RC1: Timer3 interrupt (TMR3CS = 1)

Useful for counting events

- Voting machine
- Optical encoder
- Frequency counter

Timer2 Interrupts

N = A*B*C

- A = [1 .. 16]
- B = [1 .. 256]
- C = {1, 4, 16}

T2CON determines A and C

T2CON							
7	6	5	4	3	2	1	0
	A - 1					C	

PR2 determines B

- B = PR2 + 1

Timer0, Timer1, Timer3

- N = PS * Y

Need to set up next interrupt each time you interrupt

- Default = 65,536 counts later
- Change if you want something different

PS set up during initialization

- PS = 1, 2, 4, ..., 256 (Timer0)
- PS = {1, 2, 4, 8} (TImer1, Timer3)

Y is determined by TMRx

- Example: Y = 10,000

```
TMR0 = -10000;
```

Single Timer Interrupt

- Interrupt every N clocks

Can measure time

- Time = 1ms
  - N = 10,000
- Timer2: A*B*C = 10,000
  - A = 10, B = 250, C = 4
- Timer0: PS * Y = 10,000
  - PS = 1
  - TMR0 = -10,000

Can generate a frequency

- f = 220Hz
- $N = \left(\frac{10{,}000{,}000}{2 \cdot Hz} \right)$

Find {A, B, C} for Timer2

Multiple Timer Interrupts

Interrupts do not conflict with each other

- You can have several timer interrupts running at the same time
  - Timer0 = 100Hz
  - Timer1 = 110Hz
  - Timer2 = 113Hz
  - Timer3 = 127Hz

You can have interrupts changing interrupts

- Example: police siren
  - Timer0 = 500ms
  - Timer1 = 350Hz, 437Hz
  - Timer0 changes the frequency for Timer1

Timer1 Capture

Record the time of an event to one clock

- Capture1
  - Captures a rising / falling edge on RC2
  - Records the time of TMR1at the time of the edge
- Capture2
  - Captures a rising / falling edge on RC1
  - Records the time of TMR1 at that time

Timer1 Compare

Sets / Clears a pin at a precise time

- Compare1
  - Set or clear RC2 when TMR1 == CCPR1
- Compare 2
  - Set or clear RC1 when TMR1 == CCPR2

Useful for

- Generating a square wave with a period accurate to one clock
- Generating a pulse with a width accurate to one clock

z-Transforms

zY means y(k+1)

z-Transforms convert difference equations in to algebraic equations in z

$$y(k+2) - 1.5y(k+1) + 0.7y(k) = 2(x(k+1) - x(k))$$

becomes

$$z^2 Y - 1.5zY + 0.7Y = 2(zX + X)$$

Digital Filters

Analyze using phasors

$$Y = 2\left(\frac{(z-0.8)}{(z-0.9)}\right)X$$

$$x(t) = 3\cos(4t) + 5\sin(4t)$$

$$T = 0.1$$

Solution

$$s = j4$$

$$z = e^{sT} = e^{j0.4}$$

$$X = 3 - j5$$

$$Y = 2\left(\frac{(z-0.8)}{(z-0.9)}\right)_{z=e^{j0.4}} \cdot (3 - j5)$$

$$Y = 3.5226 - 11.6747i$$

$$y(t) = 3.5226\cos(4t) + 11.6747\sin(4t)$$

Implementing Digital Filters

Convert G(z) to a difference equation

Time shift so the output is y(k) (present output)

Convert to C code

$$Y = 2\left(\frac{z-0.8}{z-0.9}\right)X \qquad T = 0.1$$

becomes

```
while(1) {
    x1 = x0;
    x0 = A2D_Read(0);

    y1 = y0;
    y0 = 0.9*y1 + 2*(x0 - 0.8*x1);

    D2A(y0);
    Wait_ms(100);
    }
```

Converting from s-plane to z-plane

Convert as $z = e^{sT}$

- Other methods exist

Example: Assume T = 0.1

$$Y = 2\left(\frac{s+3}{s+5}\right)X$$

Convert the poles and zeros

$s = -3 \qquad z = e^{sT} = 0.7408$

$s = -5 \qquad z = e^{sT} = 0.6065$

Add a gain to match the gain somewhere (DC in this case)

$$G(z) = 1.8218\left(\frac{z-0.7408}{z-0.6065}\right)$$

Exam Overview

- Five questions
- Random order
- No back tracking
- Can use Table of Interrupts

Available on BlackBoard

- 6am - midnight
- 100 minute time limit