CS/ECE 6780/5780

Al Davis

Today's topics:

• Midterm 2 hints
  • no practice midterm since it didn’t help last time
• ADC’s and DAC’s
  • chapter 11 of your text
  • your kit has an A/D (Port D w/ DDR set to inputs)
    • handy since sensors often supply analog value
  • your kit doesn't have a D/A
    • sometimes needed for analog control of external devices (e.g. VF converters)
      • which I was hoping to have as a lab (alas)

Midterm #2

• Focus
  • primarily on material covered after the first midterm
    • note I'm not a fan of the cram and forget mode
      • unhealthy attitude in a professional discipline
      • hence some (~10%) material “might” appear from pre-midterm1 material
    • style likely to be similar to midterm #1
      • focus on foundational concepts
      • “write a bunch of code” problems are good for take home exams
        • but you did this in the labs - so what's the point
    • open book and open notes
      • danger – if you have to look up every question you’ll lose
• Post midterm1 material
  • semaphores and threads
  • input capture and output compare
  • serial I/O: SCI, SPI, UART, RS232
  • relays and motors, stepper motor control
  • memory: SRAM, DRAM, NVRAM
  • ADC & DAC
• All are fair game! (book, lectures, & labs)
OpAmp Review

- Almost ubiquitous analog circuit element since ~1968
  - 3 terminal element w/ + & - voltage rails
    » acts as a differential voltage amplifier
      - Ideal opamp
        - Input impedance infinite, output impedance 0
        - Gain infinite, 0 offset voltage
      - Real opamp (varies w/ part)
        - High open-loop gain 100K to 1M
        - High Zin and low Zout

source: Wayne Storr

Inverting & Non-inverting Circuits

Inverting Op-amp

Non-inverting Op-amp

source: Wayne Storr
Differential & Summing Circuits

\[ V_{out} = -\frac{R_f}{R_{in}} (V_1 - V_2) \]

\[ V_{out} = \frac{R_f}{R_{1}} V_1 + \frac{R_f}{R_{2}} V_2 + \frac{R_f}{R_{3}} V_3 \]

source: Wayne Storr

Differentiation & Integration

\[ j\omega = 2\pi f \]

RC dependent shape

180° phase change due to - input

source: Wayne Storr
Passive Filter Review

- **Passive = RLC circuit**
  - L blocks high-\(f\) signals and pass low-\(f\) signals
  - C blocks low-\(f\) signals and pass high-\(f\) signals
- **Low pass filter**
  - signal passes through an L or C provides a path to ground
- **High pass filter**
  - signal passes through a C or L provides a path to ground
- **R's**
  - impedance is not frequency dependent
  - but can be used in filters to aid frequency selection
    » due to RC time constant
- **Terminology**
  - \(f_c := \) cutoff frequency
    » 3db gain loss point
      - power is \(IV\) hence 3db = .707
      - 3 db = \(1/P1\) where \(P1 = 10^{3/20} P0\)

Simple Active Filter

\[
\frac{V_{out}}{V_{in}} = G \cdot \sqrt{\frac{1}{1 + \left(\frac{f}{f_c}\right)^2}}
\]

\[
\begin{align*}
V_{out} &= V_{in} \cdot \frac{R_3}{R_1 + R_2} \\
V_{in} &= \frac{R_3}{R_1 + R_2} \\
R_2 &= R_1 \cdot R_2 \\
f_c &= \frac{1}{2\pi RC}
\end{align*}
\]
2-Pole Butterworth Low-Pass Filter

Select the cutoff frequency \( f_c \).
Divide the two capacitors by \( 2\pi f_c \).
\[
C_{1A} = \frac{141.4 \mu F}{2\pi f_c}, \quad C_{2A} = \frac{70.7 \mu F}{2\pi f_c}
\]
Select standard capacitors with same order of magnitude.
\[
C_{1B} = C_{1A}, \quad C_{2B} = C_{2A}
\]
Adjust resistors to maintain \( f_c \) (i.e., \( R = 10k\Omega \cdot x \)).

Bandpass

- Filter highs then filter lows

\[
Q = \frac{f_0}{\Delta f}
\]
Band-Reject

- Filter highs and lows in parallel then amplify

![Band-Reject Circuit Diagram]

DAC’s Finally

- DAC role
  - create a continuous analog waveform from discrete digital outputs
    - In practice DAC output usually put through a low-pass reconstruction filter to remove undesired high frequency components (a.k.a. ringing)
- PWM
  - DAC approximation
    - audio class D amplifiers are PWM based
DAC Parameters

- **Precision**
  - # of distinguishable DAC outputs

- **Range**
  - min to max of output values

- **Resolution**
  - smallest distinguishable change in output

\[
\text{Range (volts)} = \text{Precision (alternatives)} \cdot \text{Resolution (volts)}
\]

- **2 common encoding schemes 2’s complement and 1’s complement**

\[
V_{\text{out}} = V_0 \left( \frac{b_7}{2} + \frac{b_6}{4} + \frac{b_5}{8} + \frac{b_4}{16} + \frac{b_3}{32} + \frac{b_2}{64} + \frac{b_1}{128} + \frac{b_0}{256} \right) + V_0
\]

\[
V_{\text{out}} = V_0 \left( -\frac{b_7}{2} + \frac{b_6}{4} + \frac{b_5}{8} + \frac{b_4}{16} + \frac{b_3}{32} + \frac{b_2}{64} + \frac{b_1}{128} + \frac{b_0}{256} \right) + V_0
\]

Vos = output offset voltage

DAC Flavors

- **Direct**
  - All use opamps in a slightly different way

- **Offset Control**

- **Gain Control**
DAC Performance Measures

- **Offset error**
- **Gain error**
- **Ideal**
  - Small error but nice linearity

- **Digital input**
- **V_{out}**
- **Delay**
- **Slow**
- **Ringing phases**
- **Time**
  - Analog circuit reality

- **Nonlinear**
- **Ideal**
  - Perfect linearity hard to achieve

What can be done to fix these problems?

DAC Errors: Sources & Solutions

<table>
<thead>
<tr>
<th>Errors can be due to</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incorrect resistor values</td>
<td>Precision resistors</td>
</tr>
<tr>
<td></td>
<td>w/low tolerances</td>
</tr>
<tr>
<td>Drift in resistor values</td>
<td>Precision resistors</td>
</tr>
<tr>
<td></td>
<td>w/good temperature coefficients</td>
</tr>
<tr>
<td>White noise</td>
<td>Reduce BW w/low pass filter, reduce temperature</td>
</tr>
<tr>
<td>Op amp errors</td>
<td>Use more expensive devices</td>
</tr>
<tr>
<td>Interference from external fields</td>
<td>Shielding, ground planes</td>
</tr>
</tbody>
</table>
DAC Using Sum OpAmp

SW02 = switch
1 ➞ on @ XΩ
0 ➞ off
range = 7V
resolution = 1 volt

Calculate the error if X = 75
is it linear?

What would you use for a switch?

See any other problems?

Summing Op-Amp Issues

- Major precision problem
  - practical R values 1M to 10K
    » 1M/1K gain = 100 or approx 7 bits
  - difficult to avoid non-monotonicity problem
    » temperature changes R values
      - %/1°C common spec’d
    » in this case the gains vary
      - small change in smallest resistor (largest gain)
      - overwhelms same change in largest resistor (smallest gain)

- R-2R ladder scheme addresses this problem
  - all resistive input to a single gain
    » e.g. 1 current path to the OpAmp
      * rather than 3 additive paths
R-2R Ladder

current divides by 2 at each branch point

Thermally stable & higher precision since ladder can be arbitrarily long

12-bit Commercial DAC8043

<table>
<thead>
<tr>
<th>Digital Input</th>
<th>Unipolar $V_{out}$</th>
<th>Bipolar $V_{out}$</th>
<th>Unipolar gain</th>
<th>Bipolar gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111.1111.1111</td>
<td>-4.999</td>
<td>4.998</td>
<td>-4095</td>
<td>+2047</td>
</tr>
<tr>
<td>1000,0000,0001</td>
<td>-2.501</td>
<td>0.002</td>
<td>-3098</td>
<td>+2048</td>
</tr>
<tr>
<td>1000,0000,0000</td>
<td>-2.500</td>
<td>0.000</td>
<td>-3098</td>
<td>+2048</td>
</tr>
<tr>
<td>0111.1111.1111</td>
<td>-2.499</td>
<td>-0.002</td>
<td>-3098</td>
<td>+2048</td>
</tr>
<tr>
<td>0000,0000,0001</td>
<td>-0.001</td>
<td>-4.998</td>
<td>-4096</td>
<td>+2048</td>
</tr>
<tr>
<td>0000,0000,0000</td>
<td>0.000</td>
<td>-5.000</td>
<td>-4096</td>
<td>+2048</td>
</tr>
</tbody>
</table>
**DAC Selection: Precision, Range, Resolution**

- **Affects quality of signal that can be generated**
  - more bits means finer control and closer approximation to ideal waveform

  ![Generated waveforms](image)

- **smoothing can be done with RC circuits**
  - excellent control can be had with switched capacitive circuits
  - fun but somewhat hairy topic

---

**DAC Interfaces: the usual**

DAC’s come in lots of flavors – serial is slowest but uses the fewest pins. Other 2 are faster but more pins. Choice depends on overall system needs.
DAC Packages: several flavors

Cost varies with precision, power, accuracy, …

DAC Summary

- **Lots of commercial DAC options**
  - by themselves they usually aren't sufficient
    - need for low-pass filter
  - or amplification required to get necessary amplitude or current drive
    - opamps to the rescue
    - plus lots of other options
      - use DAC to
        - vary gain
        - vary offset
        - or just directly to specify the waveform
- **Or do it yourself with an R-2R ladder**
  - guts of the commercial versions anyway
  - although transistors are used in place of resistors to reduce thermal errors for increased accuracy
- **Next convert in the opposite direction**
  - ADC
    - common ES µC surrounded by sensors
    - hence many have an integrated ADC
      - port D in your kits
ADC Parameters

- **Precision**
  - # of distinguishable ADC inputs

- **Range**
  - max – min inputs

- **Resolution**
  - change in input causing the low order bit to flip

- **Accuracy**
  - usually a system parameter +/- %error

- **Monotonic**
  - if no missing digital codes in the range

- **Linear**
  - if resolution is constant throughout the range

- **Speed**
  - minimum time between samples
  - delay between sample and valid digital out

### Common Encoding Schemes

<table>
<thead>
<tr>
<th>Unipolar codes</th>
<th>Straight binary</th>
<th>Complementary binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>+5.00</td>
<td>1111,1111</td>
<td>0000,0000</td>
</tr>
<tr>
<td>+2.50</td>
<td>1000,0000</td>
<td>0111,1111</td>
</tr>
<tr>
<td>+0.02</td>
<td>0000,0001</td>
<td>1111,1110</td>
</tr>
<tr>
<td>+0.00</td>
<td>0000,0000</td>
<td>1111,1111</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bipolar codes</th>
<th>Offset binary</th>
<th>2s Complement binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>+5.00</td>
<td>1111,1111</td>
<td>0111,1111</td>
</tr>
<tr>
<td>+2.50</td>
<td>1100,0000</td>
<td>0100,0000</td>
</tr>
<tr>
<td>+0.04</td>
<td>1000,0000</td>
<td>0000,00001</td>
</tr>
<tr>
<td>+0.00</td>
<td>1000,0000</td>
<td>0000,0000</td>
</tr>
<tr>
<td>-2.50</td>
<td>0100,0000</td>
<td>1100,0000</td>
</tr>
<tr>
<td>-5.00</td>
<td>0000,0000</td>
<td>1000,0000</td>
</tr>
</tbody>
</table>
2-bit FLASH ADC

- Use LM311 voltage comparators

High speed but low precision

Need more bits?
extend the ladder

Need bipolar
e.g. +10 @ top, -10 @ bot
middle tap = 0V

<table>
<thead>
<tr>
<th>V_{in}</th>
<th>X3</th>
<th>X2</th>
<th>X1</th>
<th>X0</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5 &gt; V_{in}</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5.0 &gt; V_{in} ≥ 2.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7.5 &gt; V_{in} ≥ 5.0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>V_{in} ≥ 7.5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Successive Approximation ADC’s

- Most pervasive method
- Basic idea
  - n bit precision takes n clocks
    - for each clock a guess is made for the current bit
      - starting with high order bit
      - set bit under test to 1
      - if V_{out} is higher than V_{in} then bit is reset to 0
      - process continues
    - hence there is a V_{out} vs. V_{in} comparator inside the ADC
- Typical circuit
  - use a current-output DAC (rather than a Vout DAC)
    - each guess is converted to a current by the DAC
    - V_{in} also converted to a current
    - current comparison keeps or flips the guess bit
  - why current
    - more precise and faster
Successive Approximation ADC

Dual Slope ADC’s

- Voltage reference, 2 BiFET switches, and 2 integration stages
  - good for 16 -20 bits of precision
Dual Slope Waveforms

Sigma Delta ADC

- Common use is audio 44KHz sample rate (CD quality)
- Trick is to use a DSP unit to handle the successive approximation chore and a 1 bit DAC
  - Why? - It's faster - due to small digital transistors
Sample & Hold

- **Problem** – how to guess correctly while $V_{in}$ changes
  - **S/H** is an analog latch
    » duty hold $V_{in}$ constant during the current $n$ cycle approximation phase

- Should use polystyrene capacitor because of its high insulation resistance and low dielectric absorption.
- A larger value of $C$ decreases (improves) droop rate. If droop current is $I_{DR}$, then droop rate is:
  \[ \frac{dV_{out}}{dt} = \frac{I_{DR}}{C} \]
- A smaller $C$ decreases (improves) acquisition time.

---

Multi-Channel ADC

- **Need an analog MUX**
  - uses BiFET switches with digital selection

---
Maxim MAX1147

- **Discrete ADC**
  - integrates ADC, S/H, and analog mux into one component

![Diagram of Maxim MAX1147]

ADC Interrupt SW w/ S/H

```
One channel
  Interrupt
  Set S/H to hold
  Start ADC
  No ADC done?
  Read ADC result
  Yes
  Save result
  Set S/H to sample
  RTE

Two channels
  Interrupt
  Set S/H to hold
  Start ADC
  No ADC done?
  Read ADC result 1
  Yes
  Save result 1
  Set S/H to sample
  RTE

Wait for S/H and Mux to settle
  Set S/H to hold
  Start ADC
  No ADC done?
  Read ADC result 2
  Yes
  Save result 2
  Set S/H to sample
  Set Mux to channel 2
  RTE
```
6812 Internal ADC

- Eight channel operation
- 8 or 10-bit resolution
- Successive approximation technique
- Clock and charge pump to create higher voltages
- 2 operation modes
  - single sequence and stop
  - continuous
- Supports
  - multiple conversions of single channel
  - or one conversion each for a group of channels
- External reference voltages
  - Vrh – high reference
  - Vrl – low reference

6812 ADC Setup

- Port AD input configurations
  - 8 pins individually configured for analog or digital input
    - ATDDIEN register
      - 1 = digital, 0 = analog
  - If ATDDIEN indicates digital
    - then DDRAD register is used to set direction
  - SRES8 (ATDCTL4[7]) register selects resolution
    - 1 → 8-bit, 0 → 10-bit
  - ATDCTL2 register
    - [7] = ADPU – set to 1 to enable ADC system
    - [1] = ASCIE – set to 1 to enable/arm interrupts
    - [0] = ASCIF – set by ADC to 1 when sequence completes
      - only works if ASCIE is set
6812 ADC Conversions

- **When triggered**
  - 1-8 conversions are performed
    - \# = value in ATDCTL3[6:3]
    - If value >= 8 still means 8
- **Channel selection**
  - ATDCTL5[2:0] = CC, CB, CA
- **Multiple channels**
  - set ATDCTL5[4] = 1
  - sequence set by ATDCTL3[6:3] – start here and cycle
  - each channel has separate completion flag
    - ATDSTAT1 register (8 bits)
    - ATDSTAT0[2:0] – counter which shows conversion progress

6812 ADC Triggers

- **Triggered in 3 ways**
  - explicit software write to ATDCTL5 when interrupts armed
  - continuous if SCAN = ATDCTL5[5] is 1
  - external trigger if ETRIG = ATDCTL2[2] is 1
    - in this case ETRIGLE & ETRIGP controls what the trigger is

<table>
<thead>
<tr>
<th>ETRIGLE</th>
<th>ETRIGP</th>
<th>External trigger mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Falling edge of PAD7</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Rising edge of PAD7</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Convert while PAD7 is low</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Convert while PAD7 is high</td>
</tr>
</tbody>
</table>
6812 ADC Sample Period

- 2 phase sample
  - 1st phase – transfer sample to S/H
  - 2nd phase – attaches external signal to S/H
- E clock and ATDCTL4 control
  - SMP1 & SMP2 ATDCTL4[6:5]
    
    | SMP1 | SMP0 | First sample | Second sample | Total  |
    |------|------|--------------|---------------|--------|
    | 0    | 0    | 2 ADC clocks | 2 ADC clocks  | 4 ADC clocks |
    | 0    | 1    | 2 ADC clocks | 4 ADC clocks  | 6 ADC clocks |
    | 1    | 0    | 2 ADC clocks | 8 ADC clocks  | 10 ADC clocks |
    | 1    | 1    | 2 ADC clocks | 16 ADC clocks | 18 ADC clocks |

- If m is a 5 bit number ATDCTL4[4:0] & $f_E$ is E clock then

\[
\text{ATD clock frequency} = \frac{1}{2(m + 1)}
\]

6812 ADC Results

- Up to 8 samples
  - stored in 8 16-bit registers ATDDR0:ATDDR7
    - results can be signed or unsigned
      - DSGN = ATDCTL5[6]
        - 1 for signed, 0 for unsigned
    - right or left justified in the 16-bit register
      - DJM = ATDCTL5[7]
        - 1 for right justified, 0 for left

<table>
<thead>
<tr>
<th>Input (V)</th>
<th>8-bit (u)</th>
<th>10-bit (ur)</th>
<th>10-bit (ul)</th>
<th>10-bit (sr)</th>
<th>10-bit (sl)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>$0000$</td>
<td>$00000$</td>
<td>$0000$</td>
<td>$FE000$</td>
<td>$8000$</td>
</tr>
<tr>
<td>0.005</td>
<td>$0000$</td>
<td>$00001$</td>
<td>$0040$</td>
<td>$FE01$</td>
<td>$8040$</td>
</tr>
<tr>
<td>0.020</td>
<td>$001$</td>
<td>$0004$</td>
<td>$0100$</td>
<td>$FE04$</td>
<td>$8100$</td>
</tr>
<tr>
<td>2.500</td>
<td>$80$</td>
<td>$0200$</td>
<td>$8000$</td>
<td>$0000$</td>
<td>$0000$</td>
</tr>
<tr>
<td>3.750</td>
<td>$0C$</td>
<td>$0300$</td>
<td>$C000$</td>
<td>$0100$</td>
<td>$4000$</td>
</tr>
<tr>
<td>5.000</td>
<td>$FF$</td>
<td>$03FF$</td>
<td>$FFC0$</td>
<td>$01FF$</td>
<td>$7FC0$</td>
</tr>
</tbody>
</table>
ADC Software Example

• **SW trigger and Gadfly loop**

```c
void ADC_Init(void){
    ATDCTL2 = 0x80; // enable ADC
    ATDCTL3 = 0x08;
    ATDCTL4 = 0x05; // 10-bit, divide by 12
}

unsigned short ADC_In(unsigned short chan){
    ATDCTL5 = (unsigned char)chan; // start sequence
    while(((ATDSTAT1&0x01)==0){} // wait for CCF0
    return ATDDRO;
}
```

Concluding Remarks

• **Whirlwind tour for sure**
  • like everything in this course
    » learn by experimenting in the lab
    » lecture is HOPEFULLY just a conceptual start
      * can't possibly cover every detail or it would be MORE boring

• **ADC and DAC**
  • integral part of ES life
    » PWM is good for some things
    » more direct analog reading or control is required for others
  • midterm2
    » no lab on this stuff so conceptual questions only
    » you should understand the basics without having to look them up
      • look up is good for nitty gritty details
        - you'll know them by heart once you've flailed in the lab long enough

• **Midterm next Tuesday**
  • don't be late