CS/ECE 6780/5780

Al Davis

Today's topics:

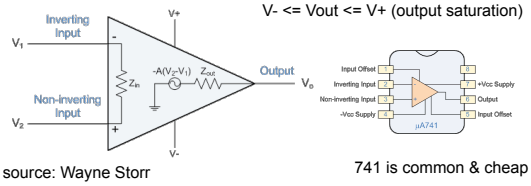
- **Midterm 2 hints**
  - **no practice midterm since it didn't help last time**
- **ADC's and DAC's**
  - **chapter 11 of your text**
  - **your kit has an A/D (Port D w/ DDR set to inputs)**
    - **handy since sensors often supply analog value**
  - **your kit doesn't have a D/A**
    - **sometimes 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**

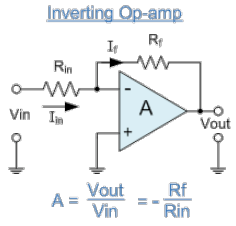


V- <= Vout <= V+ (output saturation)

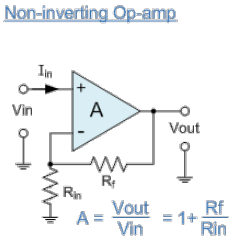
source: Wayne Storr

741 is common & cheap

Inverting & Non-inverting Circuits



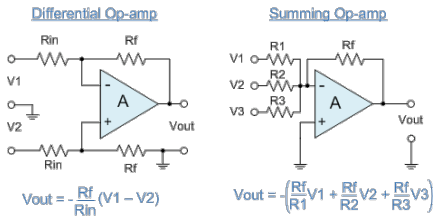
$$A = \frac{Vout}{Vin} = -\frac{Rf}{Rin}$$



$$A = \frac{Vout}{Vin} = 1 + \frac{Rf}{Rin}$$

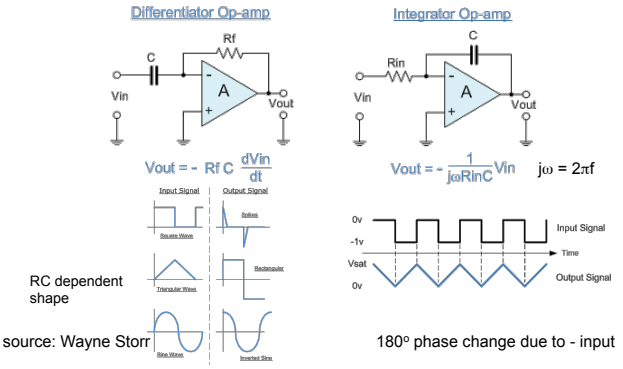
source: Wayne Storr

Differential & Summing Circuits



source: Wayne Storr

Differentiation & Integration



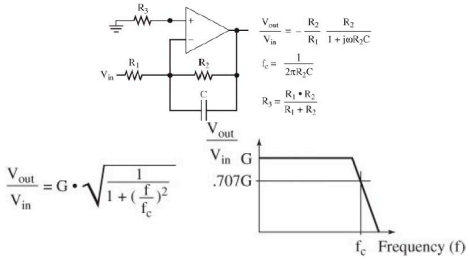
source: Wayne Storr

180° phase change due to - input

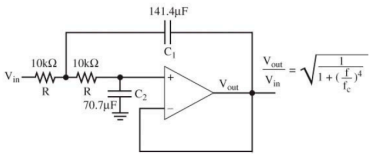
Passive Filter Review

- **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 grount**
- **R's**
  - **Impedance is not frequency dependent**
  - **but can be used in filters to aid frequency selection**
    - » **due to RC time constant**
- **Terminology**
  - **fc ::= cutoff frequency**
    - » **3db gain loss point**
      - **power is I^2V hence 3db = .707**
      - **3 db = 1/P1 where P1 = $10^{3/20}$ P0**

Simple Active Filter



2-Pole Butterworth Low-Pass Filter



Select the cutoff frequency f_c.

Divide the two capacitors by $2\pi f_c$.

$$C_{1A} = \frac{141.4\mu F}{2\pi f_c} \quad C_{2A} = \frac{70.7\mu F}{2\pi f_c}$$

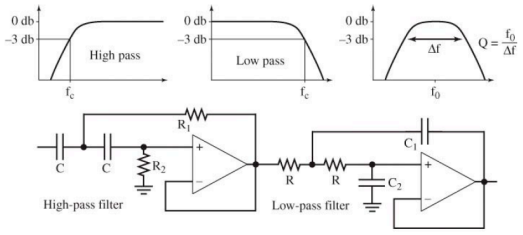
Select standard capacitors with same order of magnitude.

$$C_{1B} = \frac{C_{1A}}{x} \quad C_{2B} = \frac{C_{2A}}{x}$$

Adjust resistors to maintain f_c (i.e., $R = 10k\Omega \cdot x$).

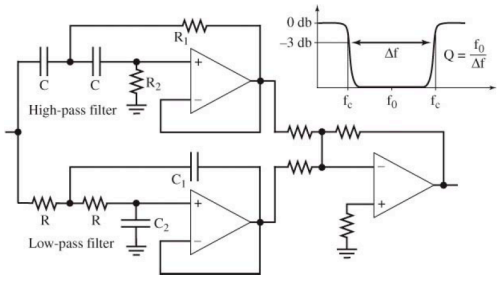
Bandpass

- **Filter highs then filter lows**



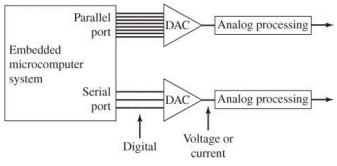
Band-Reject

- **Filter highs and lows in parallel then amplify**



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**

Range (volts) = Precision (alternatives) · Resolution (volts)

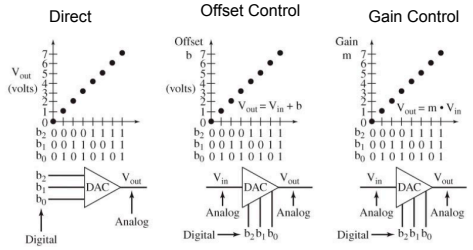
- **2 common encoding schemes 2's complement and 1's complement**

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

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

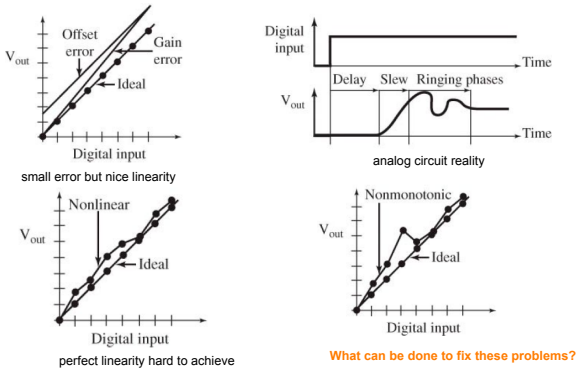
Vos = output offset voltage

DAC Flavors



All use opamps in a slightly different way

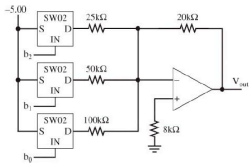
DAC Performance Measures



DAC Errors: Sources & Solutions

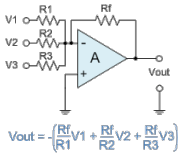
Errors can be due to	Solutions
Incorrect resistor values	Precision resistors w/low tolerances
Drift in resistor values	Precision resistors w/good temperature coefficients
White noise	Reduce BW w/low pass filter, reduce temperature
Op amp errors	Use more expensive devices w/low noise and low drift
Interference from external fields	Shielding, ground planes

DAC Using Sum OpAmp



b_2	b_1	b_0	V_{out}
0	0	1	+1
0	1	0	+2
1	0	0	+4

SW02 = switch
1 ➔ on @ XΩ
0 ➔ off
range = 7v
resolution = 1 volt



$$Vout = -\left(\frac{Rf}{R1}V1 + \frac{Rf}{R2}V2 + \frac{Rf}{R3}V3\right)$$

remember

Calculate the error if X = 75 is it linear?

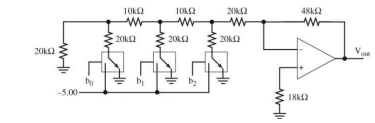
What would you use for a switch?

See any other problems?

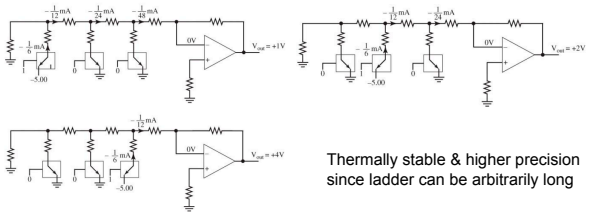
Summing Op-Amp Issues

- **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

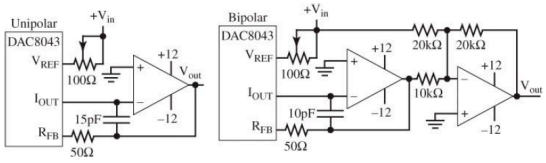


current divides by 2 at each branch point



Thermally stable & higher precision since ladder can be arbitrarily long

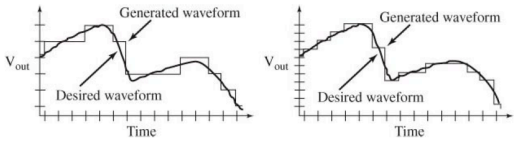
12-bit Commercial DAC8043



Digital Input	Unipolar V_{out}	Bipolar V_{out}	Unipolar gain	Bipolar gain
1111,1111,1111	-4.999	4.998	$-\frac{4095}{4096}$	$+\frac{2047}{2048}$
1000,0000,0001	-2.501	0.002	$-\frac{2049}{4096}$	$+\frac{1}{2048}$
1000,0000,0000	-2.500	0.000	$-\frac{2048}{4096}$	$+\frac{0}{2048}$
0111,1111,1111	-2.499	-0.002	$-\frac{2047}{4096}$	$-\frac{1}{2048}$
0000,0000,0001	-0.001	-4.998	$-\frac{1}{4096}$	$-\frac{2047}{2048}$
0000,0000,0000	0.000	-5.000	$-\frac{0}{4096}$	$-\frac{2048}{2048}$

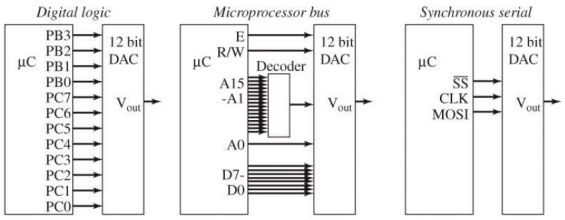
DAC Selection: Precision, Range, Resolution

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



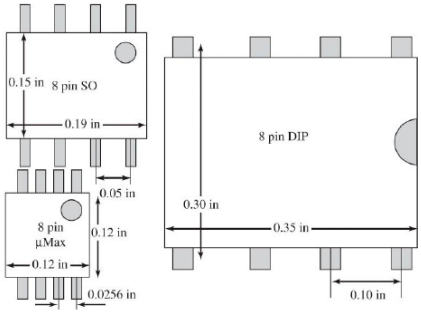
  - **smoothing can be done with RC circuits**
    - » **excellent control can be had with switched capacitive circuits**
      - **fun but somewhat hairy topic**

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**
    - » **ringing → 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**

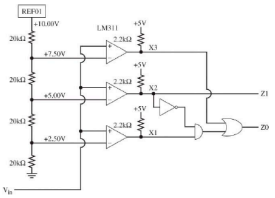
Common Encoding Schemes

Unipolar codes	Straight binary	Complementary binary
+5.00	1111,1111	0000,0000
+2.50	1000,0000	0111,1111
+0.02	0000,0001	1111,1110
+0.00	0000,0000	1111,1111

Bipolar codes	Offset binary	2s Complement binary
+5.00	1111,1111	0111,1111
+2.50	1100,0000	0100,0000
+0.04	1000,0000	0000,0001
+0.00	1000,0000	0000,0000
-2.50	0100,0000	1100,0000
-5.00	0000,0000	1000,0000

2-bit FLASH ADC

- **Use LM311 voltage comparators**



V_{in}	X3	X2	X1	Z1	Z0
$2.5 > V_{in}$	0	0	0	0	0
$5.0 > V_{in} \geq 2.5$	0	0	1	0	1
$7.5 > V_{in} \geq 5.0$	0	1	1	1	0
$V_{in} \geq 7.5$	1	1	1	1	1

High speed but low precision

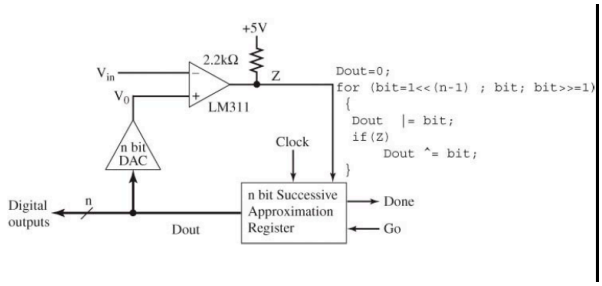
Need more bits?
extend the ladder

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

Successive Approximation ADC's

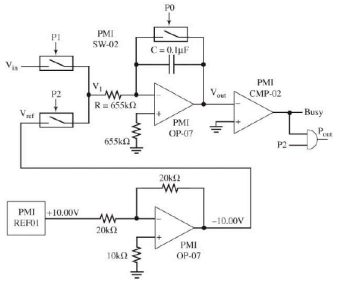
- **Most pervasive method**
- **Basic idea**
  - **n bit precision takes n clocks**
    - » **for each clock a guess is made for the current bit**
      - **starting with high order bit**
      - **set bit under test to 1**
      - **If Vout is higher than Vin then bit is reset to 0**
      - **process continues**
    - » **hence there is a Vout vs. Vin 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**
    - » **Vin also converted to a current**
    - » **current comparison keeps or flips the guess bit**
    - » **why current**
      - **more precise and faster**

Successive Approximation ADC

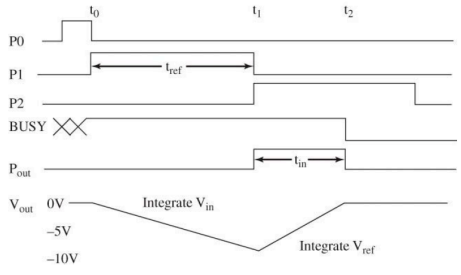


Dual Slope ADC's

- Voltage reference, 2 BIFET switches, and 2 integration stages
  - good for 16 -20 bits of precision

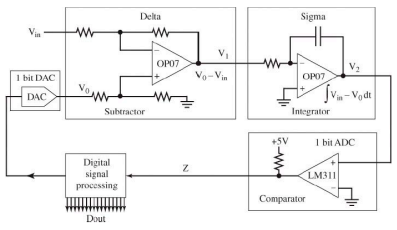


Dual Slope Waveforms



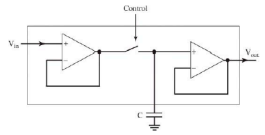
Sigma Delta ADC

- 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 Vin changes**
  - **S/H is an analog latch**
    - » **duty hold Vin constant during the current n cycle approximation phase**



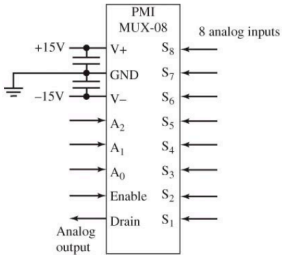
- Should use polystyrene capacitor because of its high insulation resistance and low dielectric absorption.
- A larger value of C decreases (improves) droop rate. If droop current is I_{DR}, then droop rate is:

$$\frac{dV_{out}}{dt} = \frac{I_{DR}}{C}$$

- A smaller C decreases (improves) acquisition time.

Multi-Channel ADC

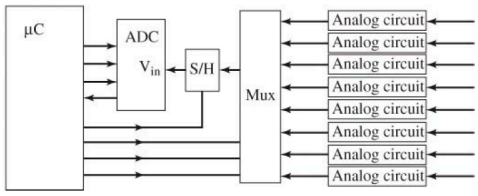
- **Need an analog MUX**
  - **uses BIFET switches with digital selection**



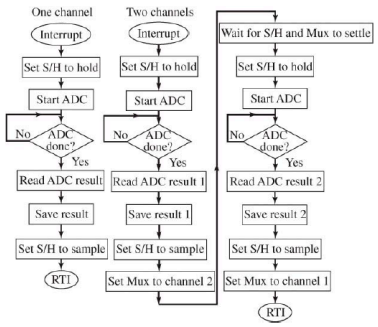
Enable	A_2	A_1	A_0	Drain
0	X	X	X	None
1	1	1	1	S_8
1	1	1	0	S_7
1	1	0	1	S_6
1	1	0	0	S_5
1	0	1	1	S_4
1	0	1	0	S_3
1	0	0	1	S_2
1	0	0	0	S_1

Maxim MAX1147

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



ADC Interrupt SW w/ S/H



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 anolog or digital input
    - » ATDDIEN register
      - 1 = digital, 0 = analog
  - If ATTDIEN 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

ETRIGLE	ETRIGP	External trigger mode
0	0	Falling edge of PAD7
0	1	Rising edge of PAD7
1	0	Convert while PAD7 is low
1	1	Convert while PAD7 is high

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

$$\text{ATD clock frequency} = \frac{1}{2}\frac{f_E}{(m+1)}$$

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

Input (V)	8-bit(u)	10-bit(ur)	10-bit (ul)	10-bit (sr)	10-bit (sl)
0.000	$00	$0000	$0000	$FE00	$8000
0.005	$00	$0001	$0040	$FE01	$8040
0.020	$01	$0004	$0100	$FE04	$8100
2.500	$80	$0200	$8000	$0000	$0000
3.750	$C0	$0300	$C000	$0100	$4000
5.000	$FF	$03FF	$FFC0	$01FF	$7FC0

ADC Software Example

- SW trigger and Gadfly loop

```
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 ATDDR0;
}
```


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