ECE 327: *Electronic Devices and Circuits Laboratory I*

Notes for Lab 5 (Analog-to-Digital Converter (ADC) Lab)

1. **This is *not* a lab on analog-to-digital conversion**

   - For a signal to be ***digit**al*, it must have both a discrete (i.e., *countable*) domain (e.g., time) and range (e.g., values). Additionally, its range must be *finite*. Recall that you can use the *digits* of your hands (i.e., your *fingers*) to *count* a *finite* number of items.
     - Discrete-time analog signals can be generated by *sampling* continuous-time analog signals.
       - Each sample can have a *continuous* range of values.
       - For slow signals, sampling is *error free*.
     - A digital signal can be generated by *quantizing* (or *quantumizing*) each sample of a discrete-time analog signal.
       - A *quantizer* (or *quantumizer*) can only produce a finite number of outputs (e.g., 0–255).
         - The outputs are called *codes*.
         - So a quantizer *encodes* data.
         - The *resolution* of the quantizer is the number of possible output values.
       - An *analog-to-digital converter (ADC)* is a quantizer.
         - When designing ADCs, for a single *cost*, there is a speed–resolution tradeoff.
       - A digital signal is an *estimate* of an analog signal, so it is subject to *estimation error*.
   - Nyquist–Shannon *sampling* theorem
     - An analog signal that is sampled at *constant* intervals can be *accurately reconstructed* so long as the *range* of frequencies in the signal is small enough.
     - Sampling frequency must be *twice* the *bandwidth* of the signal.
     - To reconstruct original signal, *sinc interpolation* is used to fill-in time between samples.
     - This classical result has been used by ***analog** engineers* for years to do lots of useful things.
     - You should *NOT* associate "Nyquist" and "digital" in your head.
   - Quantization noise floor
     - Quantization has two negative effects (that are really the same single effect).
       - (i) It limits the *dynamic range* (i.e., the maximum/minimum signal ratio) of signals.
         - To *encode* a signal that *can* be large, small variations (even if they are slow) will be lost *completely*.
       - (ii) It introduces *quantization (or quantumization) noise* ***and*** *distortion*.
         - It gives a continuous *range* of inputs a *single* identity (*round-off error*).
     - So quantization creates a *noise floor*. Characteristics smaller than "1 LSB" are lost.
       - "LSB" = "Least Significant Bit" = |input range|/resolution
       - Increase resolution to decrease LSB
     - Quantization noise *spectrum* can be *shaped* using *dither*.
       - When quantizing a value between two outputs, randomly choose to round up or down based on how close the value is to the nearest output.
       - For example, for outputs "0" and "1", encode "0.75" as "1" 75% of the time and "0" 25% of the time.
       - Reconstruction will involve a low-pass filter that will have an *averaging* effect.
       - Averaging effect *tilts* noise floor lower at low frequencies and higher at high frequencies.
         - Slow-changing signals have near-constant values that hit lots of sequential samples, so they get reconstructed perfectly.
         - Fast-changing signals have values that only get one sample, which is now extra noisy. So reconstruction is bad.

        - Amount of "tilt" is determined by sampling, so *oversample* too.
        - 2D visual example (replace "time" with "space"): Red-to-white color gradient
            - Discrete-time analog signal still shows "pink" in the middle.
            - 1-bit quantization (i.e., *resolve* signal into 2 values) shows red half and white half.
            - 1-bit quantization with dither (and plenty of sampling) has "pixels." Pixels in middle are interlaced 50% red and 50% white. Eye filters them and sees pink.
        - Dithered ADCs that oversample have high *signal-to-noise-and-distortion (SINAD)*, which gives them resolutions with high *effective number of bits (ENOB)*.
    - Dithering methods
        - Add 1 LSB of uniform white noise to signal and always round resulting signal *up*.
            - Values between two codes will be evenly balanced.
            - Values closer to one code will have more conversions favoring that code.
        - *Sigma-Delta (SD, ΣΔ, Delta-Sigma, or ΔΣ) conversion — pulse* ***density*** *modulation* (PDM)
            - Picture non-inverting op amp buffer (i.e., output shorted to inverting (−) input).
            - Before feeding back output, insert clocked D flip–flop that outputs op amp rails.
            - Negative feedback will cause *average* flip–flop (1-bit ADC) output to track the input.
            - Counting output *density* (i.e., fraction of high outputs in a period of time) gives *dithered* conversion.
            - Devices are **cheap** to make — a 1-bit flash ADC is a simple comparator.
    - *Precision* DMM's (i.e., > 6 digits) use Σ-Δ ADCs
        - Their inputs are slow (e.g., battery voltages).
        - For these slow inputs, they have fast conversion and very low quantization noise.
        - Precision DMM's are analog masterpieces.

2. The magic of pulses — not limited to just Σ-Δ ADCs (note: same magic is behind "Class-D" amplifiers)

    - A *pulse* is a rapid change in a signal from one value to another.
        - For signals made entirely of pulses, time between two pulses is the "*pulse width*" of first pulse.
    - *Discrete values* are easy to *generate*, *detect*, and *store*.
        - *Switches* dissipate no power ($i = 0$ or $v = 0$ so $i \times v = 0$).
            - A signal consisting of only high and low rail values can be generated with no dissipation.
            - Compare to analog amplifier that balances dissipation to move output.
        - Easy to detect "high" or "low" even with noise.
        - Easy to store (and read) "high" or "low" (e.g., "pits" and "lands" on CDs).
    - Continuous *values* can still be stored in *time* (e.g., widths of pulses).
        - Use "1-bit" resolution, detect *fast transitions*, and send continuous samples as *durations*.
            - Usually noise and distortion has a bigger impact on *value* than *time*.
        - Can convert to digital by *counting*. Resolution set by *clock*, and so it's programmable.
    - Examples: Flashlight communication from earth to mars; storing information on a CD

3. Today's lab: Transmission of analog signal via pulse *widths*
   - *Pulse width modulation (PWM)*
     - Transmitter pulses output high and low.
     - *Time* between a high and a low transition represents the *height* of a sample.
     - So each *pulse width* communicates a single *sample* of the signal.
     - Value-to-duration conversion is *modulation*. Duration-to-value conversion is *demodulation*.
     - A device that does *both* is a *modem* (i.e., a "mod"-"dem").
   - The output of this lab (the *modulator*) is an electrical signal.
     - That signal will drive an infrared transmitter so that we can communicate over the air (i.e., air as *communication channel*).
       * We will build this "IR-link" next week.
     - That signal *could* drive a CD burner so that we can communicate over CD (i.e., CD as communication channel).
     - The actual communication channel is *arbitrary*.
   - Ideal implementation with comparator
     - Place signal at one input and *comparison function* at the other
     - Comparison function converts values to durations (i.e., time)
       * Sweeps through entire input range
         · Output pulses (switches) when inputs cross
         · Intervals between samples are not constant (Nyquist implications?)
       * Several popular comparison functions
         · 555 datasheet uses decaying exponential (easy to generate from RC charging)
         · Triangle wave is harder to generate, but is demodulated cleanly with low-pass filter
         · We use sawtooth-like ramp train (tradeoff) — current source and capacitor ($i = Cv'$)
4. Infrared link, hysteresis, and design constraints
   - Infrared signals are small and noisy
   - A simple infrared receiver will see many false transitions
     - Filter the output to make the transitions smooth
     - Use Schmitt trigger to toggle output only after smooth output reaches threshold
     - *Hysteresis* effect: transfer function depends on current state/history
   - So pulses must have $6\,\mu$s between them to be detected; otherwise, trigger holds output constant
5. Our pulse-width modulator's parts:

   (i) Level-shifter amplifier with $\times 3$ gain and $5\,\mathrm{V_{DC}}$ offset
   - **Continuous-time Input**: Signal between -1 V and 1 V with $0\,\mathrm{V_{DC}}$ offset
   - Output sits between 2 V and 8 V

   (ii) Ramp generator with asynchronous reset
   - Constant current source driving capacitor load
   - Current and capacitance chosen to fix slope
   - Slope between $(2\,\mathrm{V})/(6\,\mu\mathrm{s})$ and $(8\,\mathrm{V})/(T - 6\,\mu\mathrm{s})$, where T is period of $\sim$30 kHz clock
   - Instantly resets to 0 V when reset signal asserted

   (iii) Comparator (i.e., an "ideal" PWM — does the sampling)

   (iv) $\sim$30 kHz clocked flip–flop with asynchronous reset
   - **Output**: Pulse train with widths modulated by discrete-time samples of input

6. Our PWM operation:
   - At every *rising edge* of the clock:
     (i) Modulator output (i.e., flip–flop's Q) transitions from low to high
       - Flip–flop clocks in a "1" from synchronous input
     (ii) Ramp resets to 0 V and *starts rising*
       - Its own (asynchronous) reset is tied to $\overline{Q}$, which goes low
   - Whenever comparator detects that ramp rises *above* input, asynchronously resets flip–flop
     - High output Q goes low
     - Ramp resets because $\overline{Q}$ goes high
       * Ramp doesn't rise until next clock
       * So only one sample per clock cycle
   - Shift and amplify input so that it stays within 2 V and 8 V
   - Set ramp slope to meet 6 μs hysteresis constraints
     (i) Ramp takes at least 6 μs to hit 2 V (slope $\leq (2/6)\,\text{V}/\mu\text{s} \approx 0.333\,\text{V}/\mu\text{s}$)
     (ii) Ramp hits 8 V with 6 μs or more before next clock edge (slope $\geq (8\,\text{V})/(T - 6\,\mu\text{s})$)
       - T is your clock's period
   - *Note that comparator and flip–flop could be implemented with a single 555 timer IC*
   - *Note that intervals between samples are* ***not*** *constant* (distortion? Nyquist consequences?)

7. Parts in the lab
   - Use CD4027 JK flip–flop with asynchronous set–reset (SR)
     - Clock transitions bring output high, so
       * Connect J high
       * Connect K low
     - Comparator brings output low, so
       * Connect S low
       * Connect comparator to R
   - Use LM311 voltage comparator with **open-collector** output
     - Output is the **floating collector** of a *npn* common-emitter amplifier acting as "BJT switch"
     - Output requires *pull-up resistor* (e.g., 1 kΩ) for operation (otherwise, output cannot go high)
     - Comparator pin-out *very different* from op. amp.: be sure to use the right pin-out
       * Uses all *three* V_{CC}, V_{EE}, *and* GND (tie *both* V_{EE} and GND to 0 V)
       * Inputs are still on pins 2 and 3, but they are *swapped*
       * Other similar inputs/outputs on completely different pins

8. Laboratory experience
    - Many design alternatives given
    - **Remember your ramp slope**
        - You will need to build a second ramp generator for the demodulator
        - The two ramp slopes must match
    - Make use of **bypass capacitors** at *supply pins* to reduce output noise
    - When taking plots, save as CSV or BMP
        - Saving as BMP prevents extra work, but make sure scope plots show all required information
            - Intervals between horizontal and vertical divisions should be clear
            - In most cases, channel grounds should be shown
            - Channels should be labeled in report (e.g., "top waveform is input")
        - If saving as CSV, be sure to...
            - Label axes and show units
            - Identify waveforms (e.g., "input" and "output")
    - See pin-out handout
        - Important difference between OA and comparator pin-outs
    - **Follow lab *book* procedures**
        - Handout gives detailed instructions
        - In part 2, give slopes in V/μs units
        - Keep resulting **pulse-width modulator** for future labs
9. Laboratory reports
    - Answer all questions and provide all plots from lab procedures in lab text
    - Include ALL PLOTS from procedure (even if they aren't mentioned in book)
        - USE the plots in your discussion
    - Consider answering some of the questions from the procedure