M02: Basic Programming

2.2 Analog to Digital Converter

Prof. Rosa Zheng

References:

1. **Steven W. Smith**, *The Scientist and Engineer's Guide to Digital Signal Processing,* Chapter 3 ADC and DAC, Online Available at DSPguide.com.

2. J. **McClellan, R. Schafer, M. Yoder,** DSP First, companion website: http://spfirst.gatech.edu/

3. ADC, http://www.onmyphd.com/?p=analog.digital.converter

Signal Representation in Time

- **DSP First Chapter 2: Sinusoids**
  - **http://spfirst.gatech.edu/chapters/02sines/overview.html (or http://dspfirst.gatech.edu/chapters/02sines/overview.html)**
  - **Click on Clay Whistle and play the demo**
  - **Take a look at the time domain signal plots:**
    - **Identify amplitude, period, frequency, phase in a single frequency signal:** $s(t) = A\cos(2\pi f t + \alpha)$
    - **Compute signal power (assume the amplitude is voltage, and the signal is applied to a 1 ohm load)**

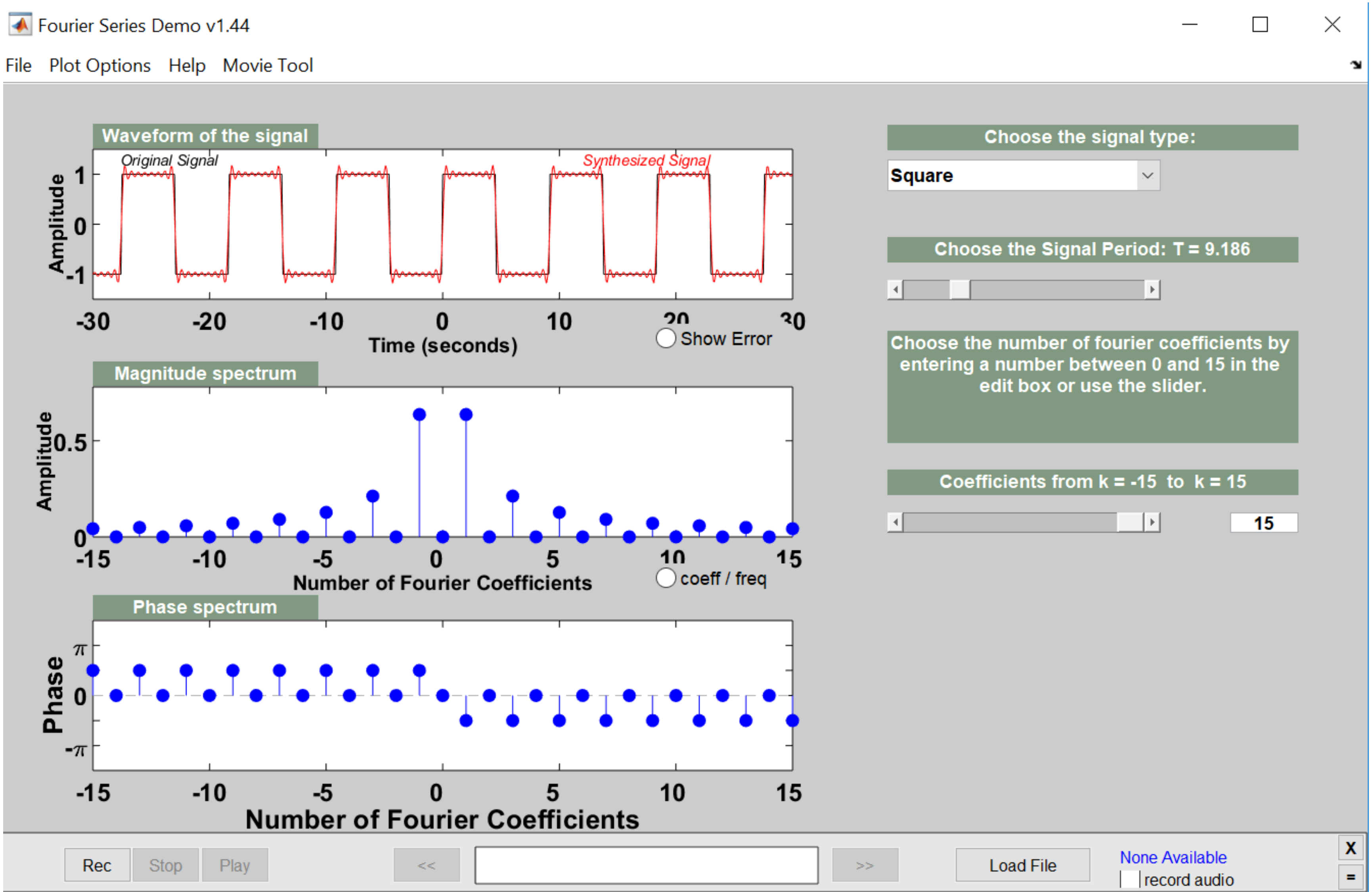
Frequency Representation

- DSP First Chapter 3: Spectrum Representation
  - http://spfirst.gatech.edu/chapters/03spect/overview.html

  Or
  http://dspfirst.gatech.edu/chapters/03spect/overview.html

  - Download the Matlab demos, extract them and run
    - FM Synthesis: shows a square wave is the sum of many sine wave components
    - Spectrograms: STFT (short time Fourier Transform) Frequency change over time

DSP First Demo: FM Synthesis



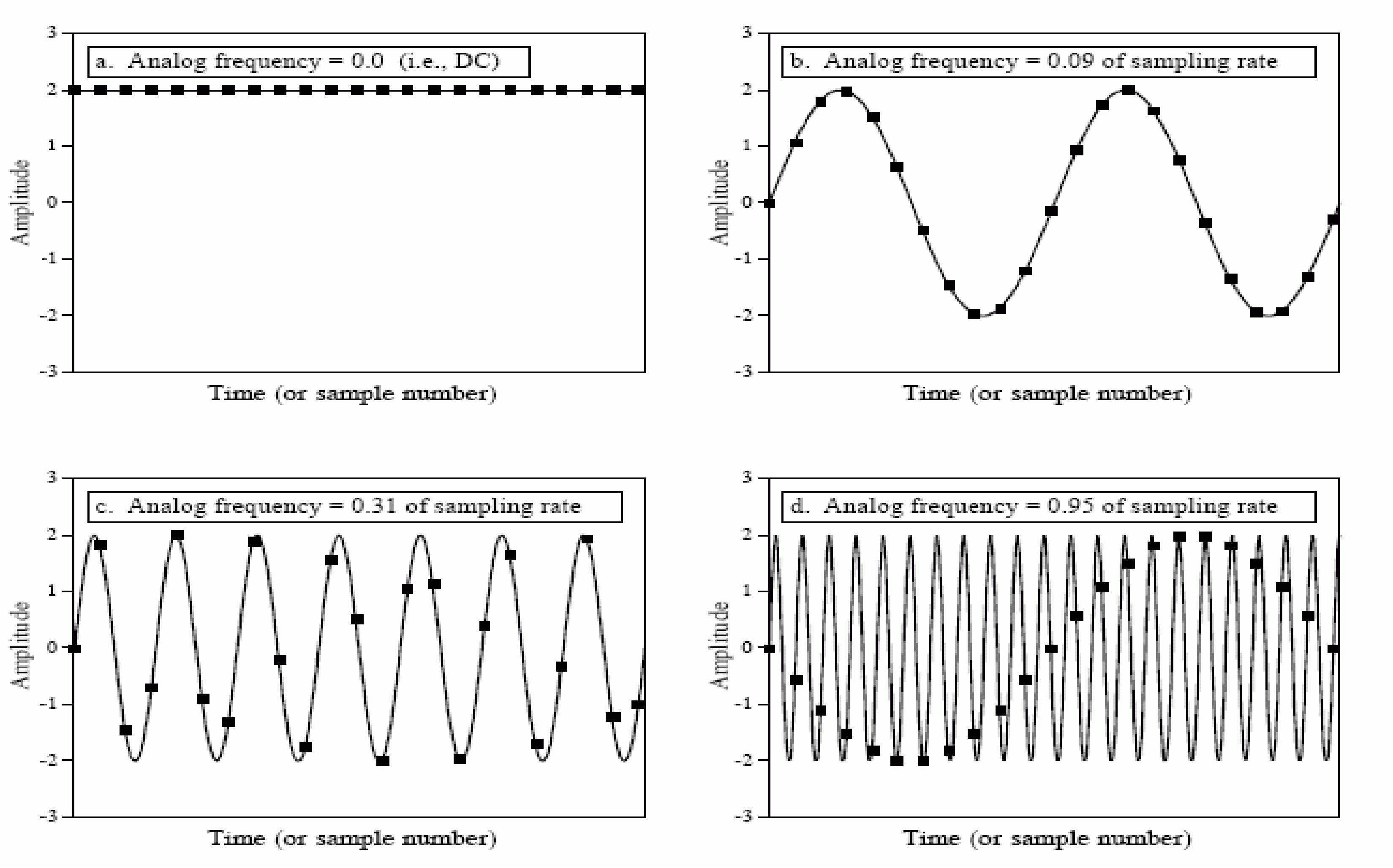
Nyquist Sampling Theorem

- Sampling frequency Fs
- Key features:
  - Uniform Sampling – equally-spaced in time
  - Use Sinc pulse to reconstruct -- Perfect reconstruction
  - Band limiting the signal – anti-aliasing filter;
  - Sinc is infinite in time -- impractical, can't wait for ever;
  - Use higher sampling frequency and finite-duration pulses

Pre-Exam problem:

Write a program to compute and plot the cosine curve $u(t)=A\cos(2\pi f t+\varphi)$, where $A=2$, $f=200$ Hz, $\varphi=90$ degree t ranges from 0 to 1 s in suitable steps (sampling period)

Aliasing



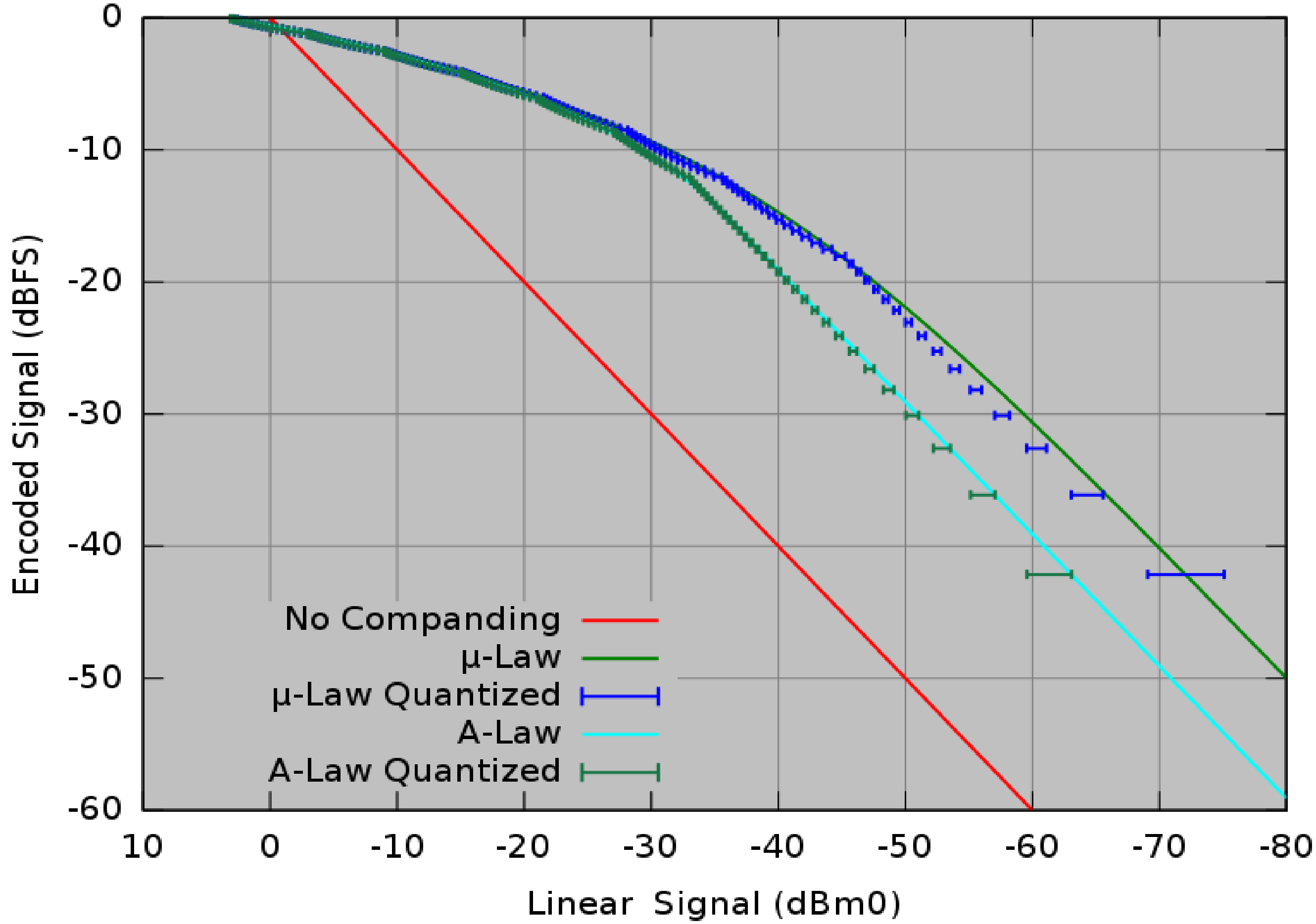
Reprint from Steven Smith, Dspguide.com, chapter 3

Quantization

- **Uniform Quantization**
  - **Dynamic range, resolution (Δv), and # of bits**
  - **Quantization error: uniformly distributed in *[-$\Delta v/2$, +$\Delta v/2$)***
- **Mu-law or A-law Compression/Companding**
  - **μ-law algorithm, https://en.wikipedia.org/wiki/%CE%9C-law_algorithm**
    - **Normalize x to [-1, 1]; set mu=255;**
    - **Apply the equation:**

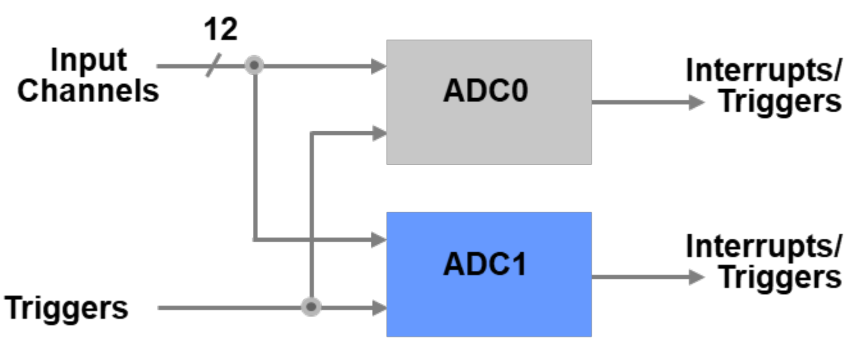
$$F(x) = \operatorname{sgn}(x)\frac{\ln(1+\mu|x|)}{\ln(1+\mu)} \quad -1 \leq x \leq 1$$

Mu-Law and A-Law Compression



TM4C123GH6PM ADC

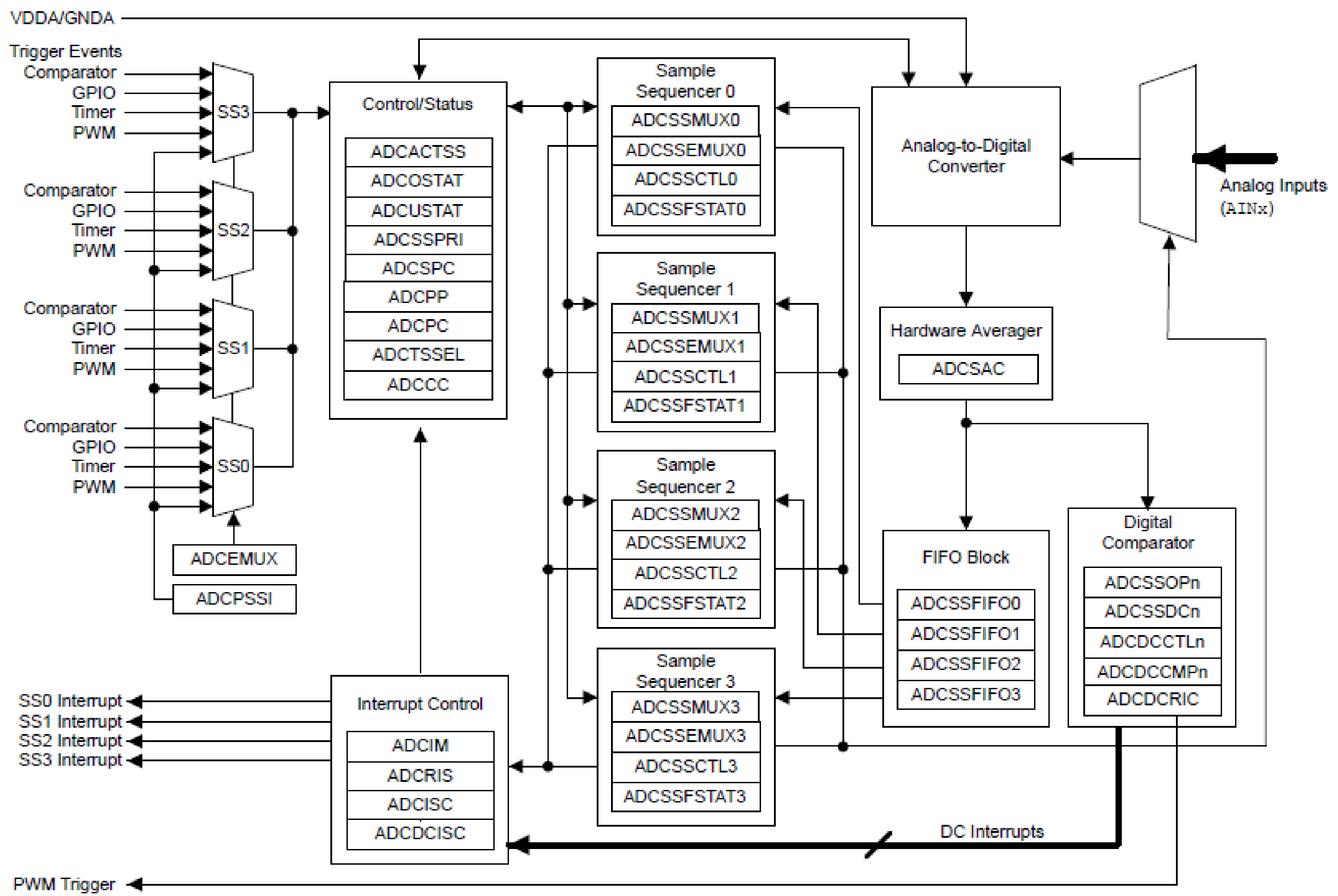
- Tiva TM4C MCUs feature two ADC modules (ADC0 and ADC1) that can be used to convert continuous analog voltages to discrete digital values by Successive Approximation Register (SAR) ADC
- Each ADC module has 12-bit resolution
- Each ADC module operates independently and can:
  - Execute different sample sequences
  - Sample any of the 12 shared analog input channels
  - Generate interrupts & triggers



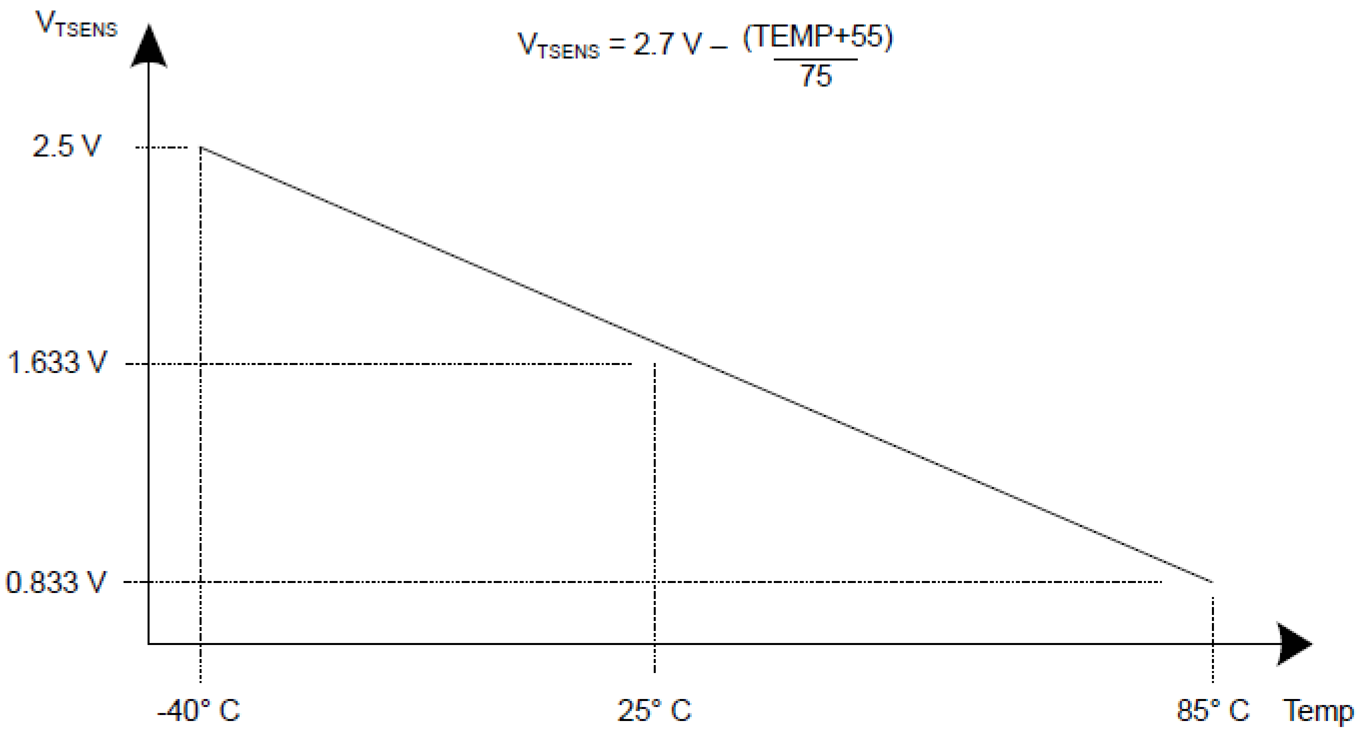
Sequencer	Number of Samples	Depth of FIFO
SS 3	1	1
SS 2	4	4
SS 1	4	4
SS 0	8	8

ADC Block in Data Sheet

Figure 13-2. ADC Module Block Diagram

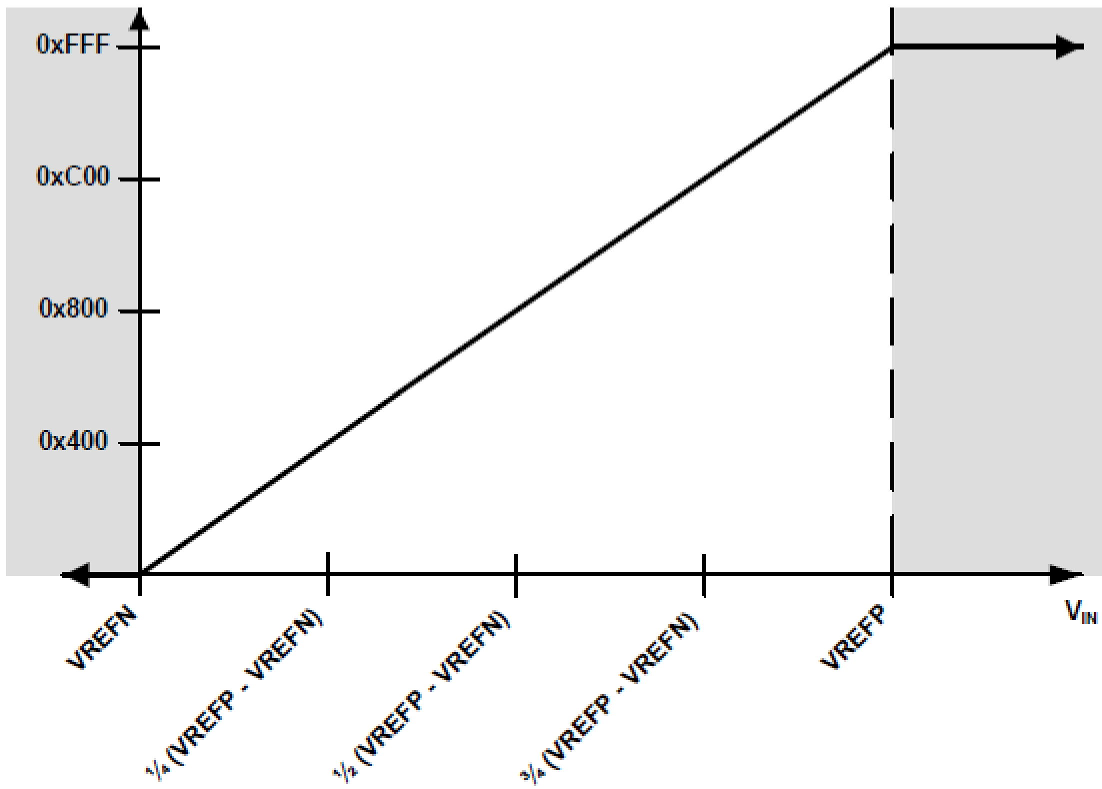


Internal Temperature Sensor



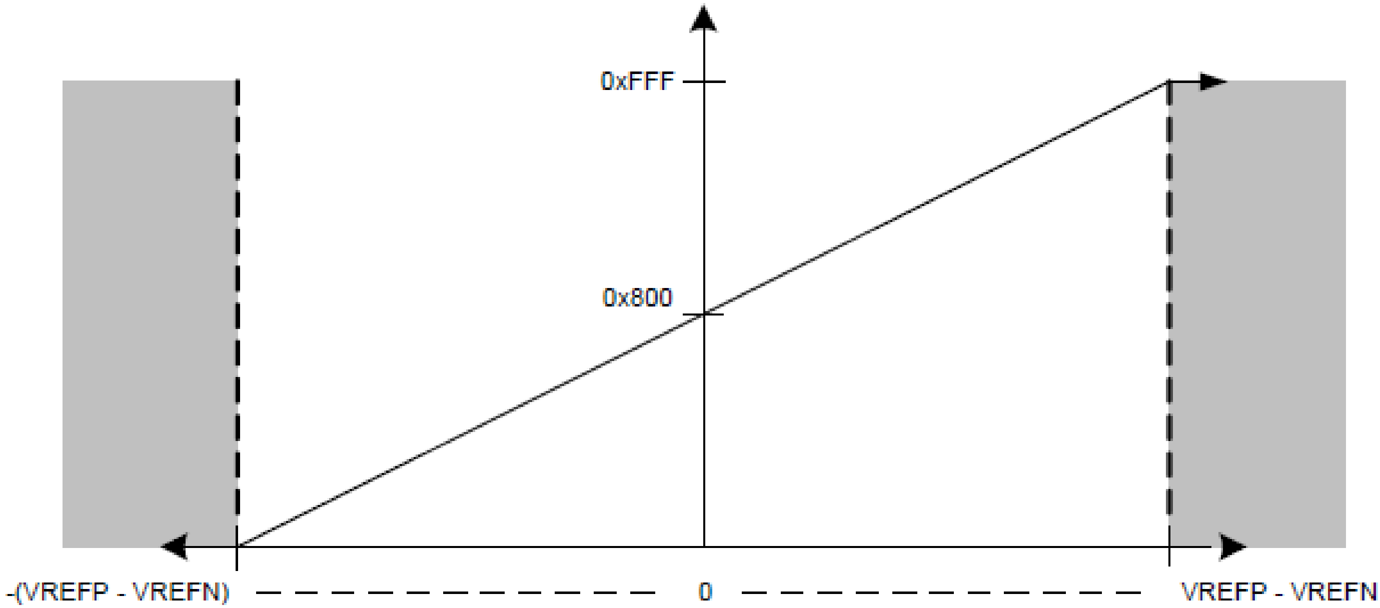
TEMP = 147.5 - ((75 * (VREFP - VREFN) × ADCCODE) / 4096)

Single-Ended Sampling



mV per ADC code = (VREFP - VREFN) / 4096

Differential Sampling



mV per ADC code = (2 *(VREFP – VREFN)) / 4096

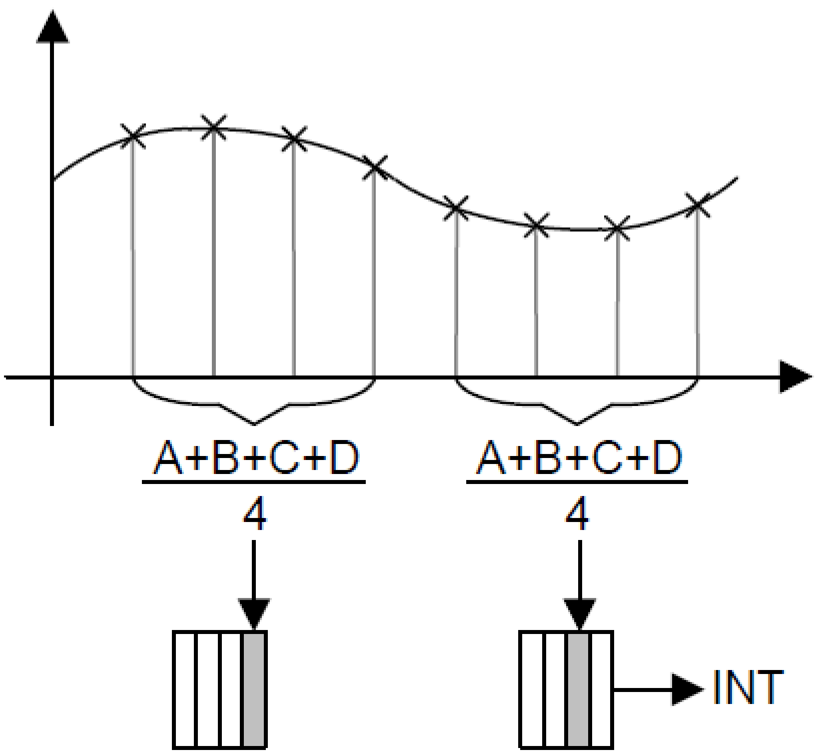
Differential Pair	Analog Inputs
0	0 and 1
1	2 and 3
…	…
5	10 and 11

Sampled: $VIN_D = VIN_+ - VIN_-$

- **Positive Voltage: VIN+ = VIN_EVEN (even channel)**
- **Negative Voltage: VIN- = VIN_ODD (odd channel)**

Sample Averaging

Hardware Sample Averaging by the API function ADChardwareOversampleConfigure()
Intermediate samples are discarded
Hardware average results are stored in FIFO



Software Sample Averaging: by sum the values stored in FIFO and dividing the # of samples: See the sample code for computing ui3TempAvg in Workshop Lab 5.