Cyber-Physical Systems

ADC / DAC

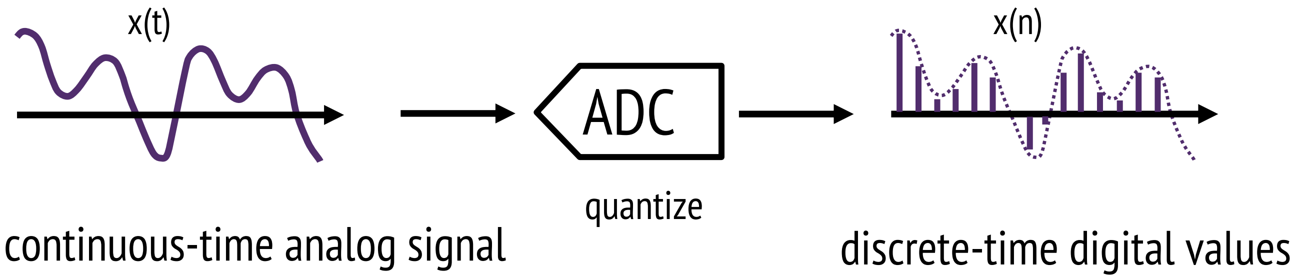
UNIVERSITY AT ALBANY
State University of New York

ICEN 553/453– Fall 2018

Prof. Dola Saha

Analog-to-Digital Converter (ADC)

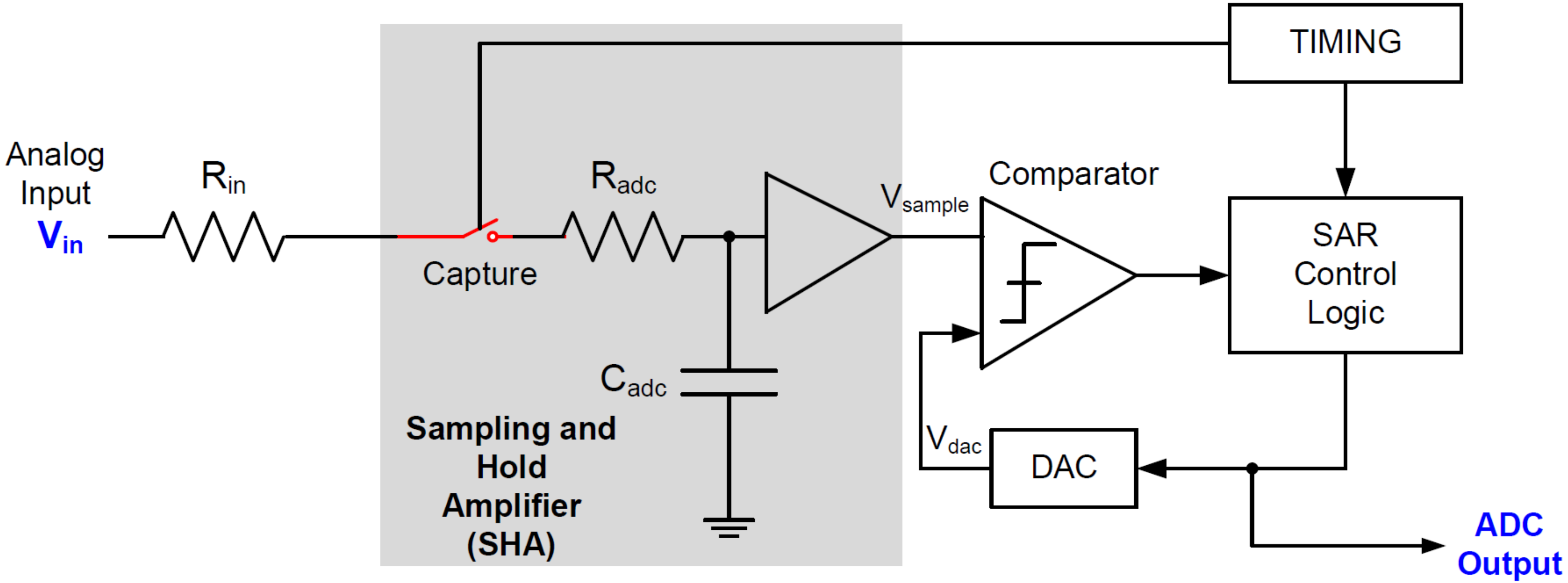
- ADC is important almost to all application fields
- Converts a continuous-time voltage signal within a given range to discrete-time digital values to quantify the voltage's amplitudes



Analog-to-Digital Converter (ADC)

- Three performance parameters:
  - sampling rate – number of conversions per unit time
  - Resolution – number of bits an ADC output
  - power dissipation – power efficiency
- Many ADC implementations:
  - sigma-delta (low sampling rate, high resolution)
  - successive-approximation (low power data acquisition)
  - Pipeline (high speed applications)

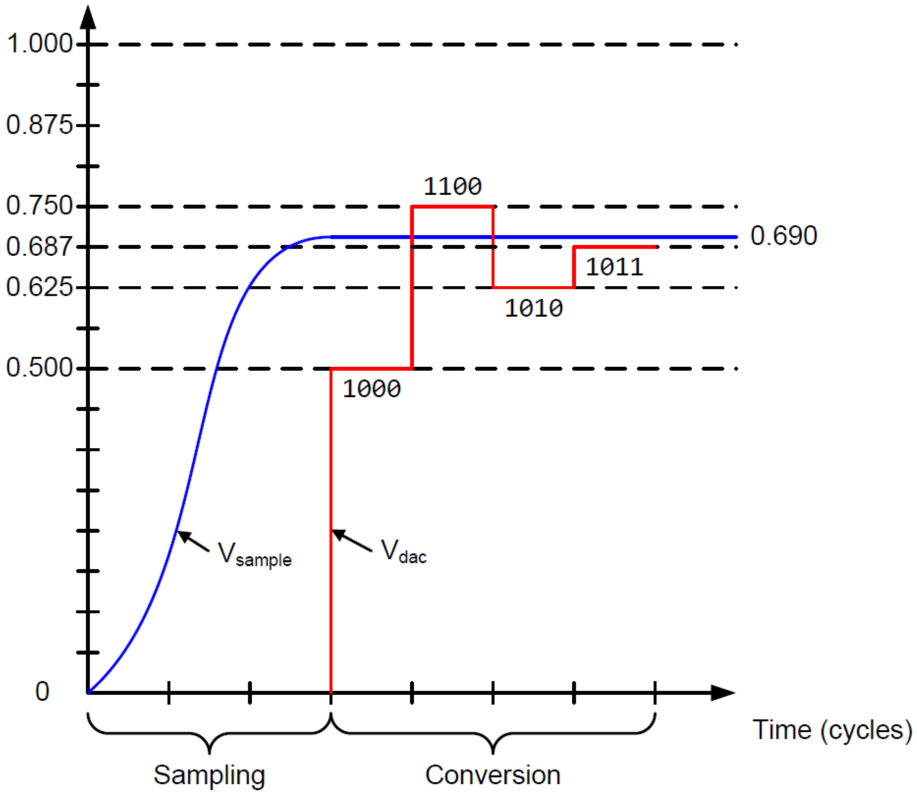
Successive-approximation (SAR) ADC



Digital Quantization

- SAR Control Logic performs Binary Search algorithm
  - DAC output is set to $1/2V_{REF}$
  - If $V_{IN} > V_{REF}$, SAR Control Logic sets the MSB of ADC, else MSB is cleared
  - V_{DAC} is set to ¾ V_{REF} or ¼ V_{REF} depending on output of previous step
  - Repeat until ADC output has been determined
- How long does it take to converge?

Successive-approximation (SAR) ADC



- Binary search algorithm to gradually approaches the input voltage
- Settle into ±½ LSB bound within the time allowed

$$T_{ADC} = T_{sampling} + T_{Conversion}$$

$$T_{Conversion} = N \times T_{ADC\_Clock}$$

$T_{sampling}$ is software configurable

ADC Conversion Time

$$\mathrm{T_{ADC}} = \mathrm{T_{sampling}} + \mathrm{T_{Conversion}}$$

- Suppose $\mathrm{ADC_{CLK}}$ = 16 MHz and Sampling time = 4 cycles

For 12-bit ADC

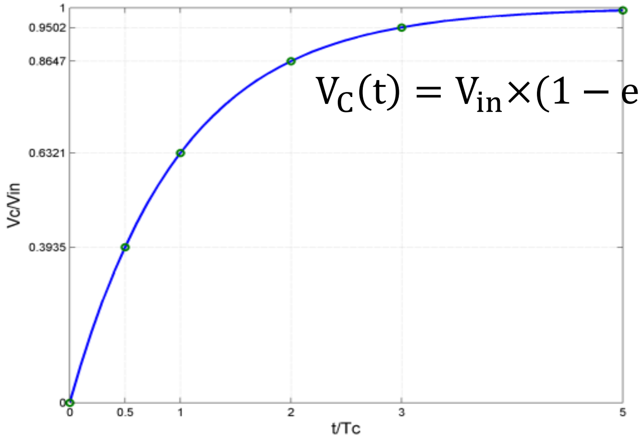
$$\mathrm{T_{ADC}} = 4 + 12 = 16 \text{ cycles} = 1\mu\mathrm{s}$$

For 6-bit ADC

$$\mathrm{T_{ADC}} = 4 + 6 = 10 \text{ cycles} = 625\mathrm{ns}$$

Determining Minimum Sampling Time

- When the switch is closed, the voltage across the capacitor increases exponentially.



$$V_C(t) = V_{in} \times (1 - e^{-\frac{t}{T_c}})$$

t= time required for the sample capacitor voltage to settle to within one-fourth of an LSB of the input voltage

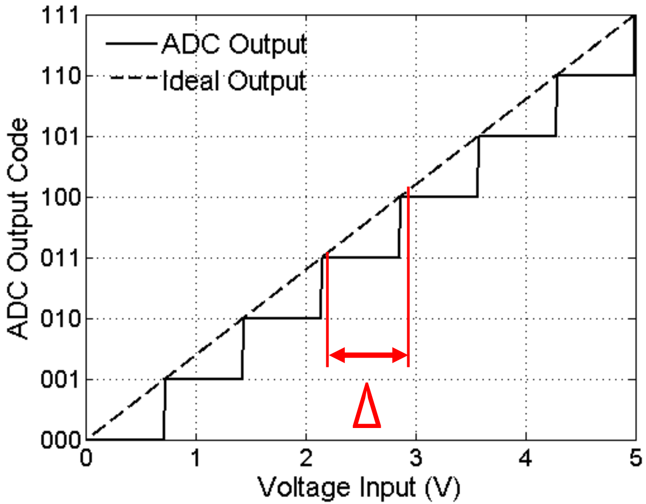
Sampling time is often software programmable!

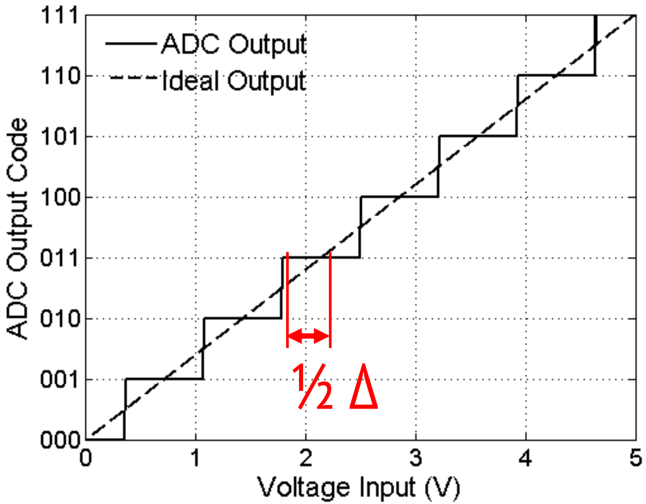
Larger sampling time → Smaller sampling error; Slower ADC speed

Tradeoff

Resolution

- Resolution is determined by number of bits (in binary) to represent an analog input.
- Example of two quantization methods (N = 3)





$$\text{Digital Result} = \text{floor}\left(2^3 \times \frac{V}{V_{REF}}\right)$$

Max quantization error = Δ = $V_{REF}/2^3$

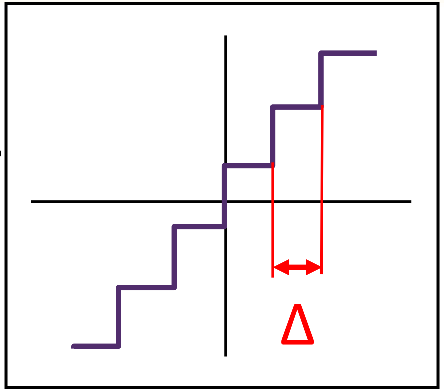
$$\text{Digital Result} = \text{round}\left(2^3 \times \frac{V}{V_{REF}}\right)$$

Max quantization error = ±½ Δ = $\pm V_{REF}/2^4$

$$\text{round}(x) = \text{floor}(x + 0.5)$$

Quantization Error

- For N-bit ADC, it is limited to ±½Δ
- Δ = is the step size of the converter.



- Example: for 12-bit ADC and input voltage range [0, 3V]

$$Max\ Quantization\ Error = \frac{1}{2}\Delta = \frac{3V}{2\times 2^{12}} = 0.367mV$$

- How to reduce error?

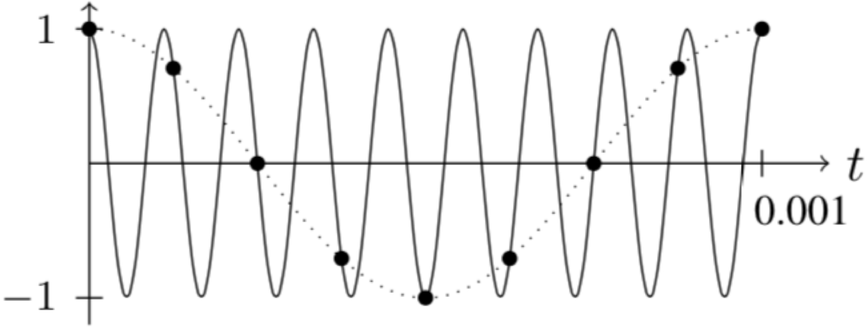
Aliasing

- Example 1:
  - Consider a sinusoidal sound signal at 1 kHz : $x(t) = \cos(2000\pi t)$
  - Sampling interval T = 1/8000
  - Samples $s(n) = f\big(x(nT)\big) = \cos(\pi n/4)$
- Example 2:
  - Consider a sinusoidal sound signal at 9 kHz : $x'(t) = \cos(18000\pi t)$
  - Sampling interval T = 1/8000
  - Samples $s'^{(n)} = f\big(x(nT)\big) = \cos\left(\frac{9\pi n}{4}\right) = \cos\left(\frac{\pi n}{4} + 2\pi n\right) = \cos\left(\frac{\pi n}{4}\right) = s(n)$
- There are many distinct functions x that when sampled will yield the same signal s.

Minimum Sampling Rate

- In order to be able to reconstruct the analog input signal, the sampling rate should be at least twice the maximum frequency component contained in the input signal
- Example of two sine waves have the same sampling values. This is called aliasing.

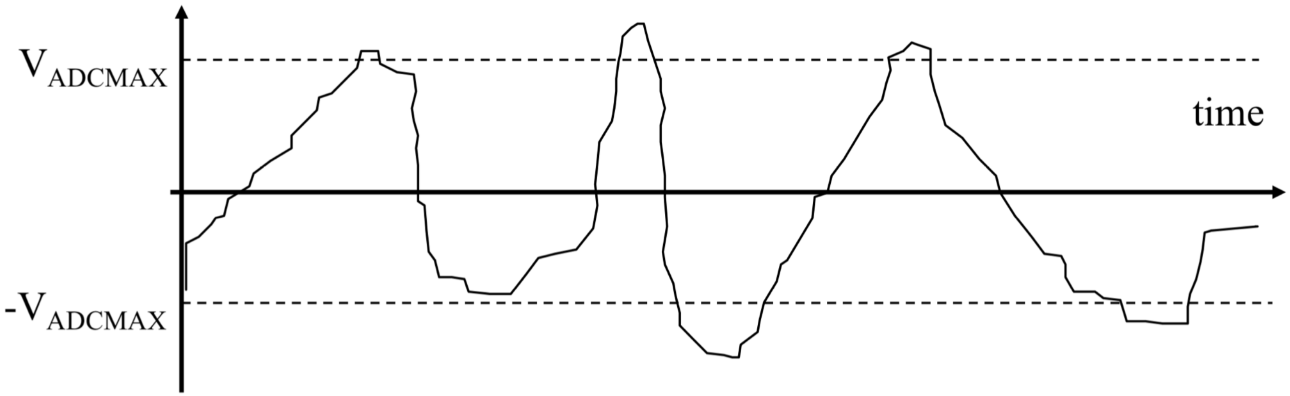
Nyquist–Shannon Sampling Theorem



- Antialiasing
  - Pre-filtering: use analog hardware to filtering out high-frequency components and only sampling the low-frequency components. The high-frequency components are ignored.
  - Post-filtering: Oversample continuous signal, then use software to filter out high-frequency components

ADC Conversion

- Input Range
  - Unipolar $(0, V_{ADCMAX})$
  - Bipolar $(-V_{ADCMAX}, +V_{ADCMAX})$
  - Clipping:
    - If $|V_{IN}| > |V_{ADCMAX}|$, then $|V_{OUT}| = |V_{ADCMAX}|$



Automatic Gain Control (AGC)

- Closed loop Feedback regulating circuit in an amplifier
- Maintains a suitable signal amplitude at its output, despite variation of the signal amplitude at the input
- The *average or peak output signal* level is used to dynamically adjust the gain of the amplifiers
- Example Use: Radio Receivers, Audio Recorders, Microphone

Power and RMS of Signal

- Average Power of a signal

$$P_x = \frac{1}{N}\sum_{n=0}^{N-1} |x_n|^2$$

- Crest Factor

$$C = \frac{|x_{PEAK}|}{x_{RMS}}$$

- Square root of the arithmetic mean of the squares of the values

$$x_{RMS} = \sqrt{\frac{1}{n}\left(x_1{}^2 + x_2{}^2 + \cdots + x_n{}^2\right)}$$

- Crest Factor
  - Sine Wave ~ 3.01dB, OFDM ~12dB

PAPR

- Crest Factor in dB

$$C_{dB} = 20log_{10} \frac{|x_{PEAK}|}{x_{RMS}}$$

- Peak to Average Power Ratio (PAPR)

$$PAPR = \frac{|x_{PEAK}|^2}{x_{RMS}{}^2}$$

$$PAPR_{dB} = 10log_{10} \frac{|x_{PEAK}|^2}{x_{RMS}{}^2} = C_{dB}$$

Example Gain Control

- AD8338

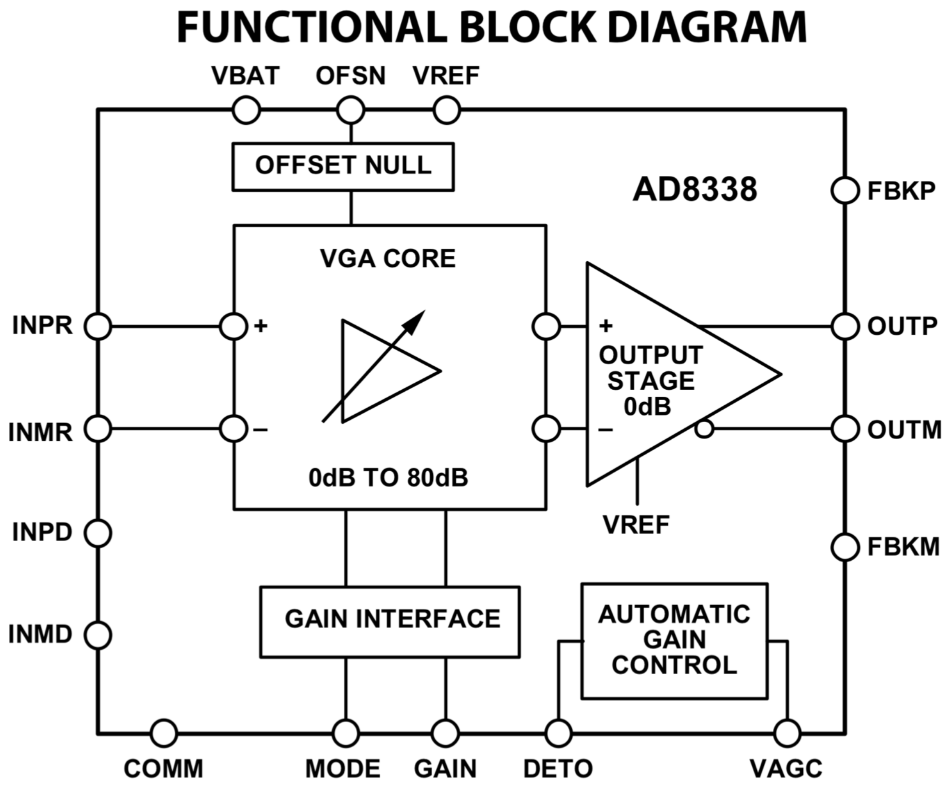


Figure 1.

Digital-to-analog converter (DAC)

- Converts digital data into a voltage signal by a N-bit DAC

$$DAC_{output} = V_{ref} \times \frac{Digital\ Value}{2^N}$$

- For 12-bit DAC

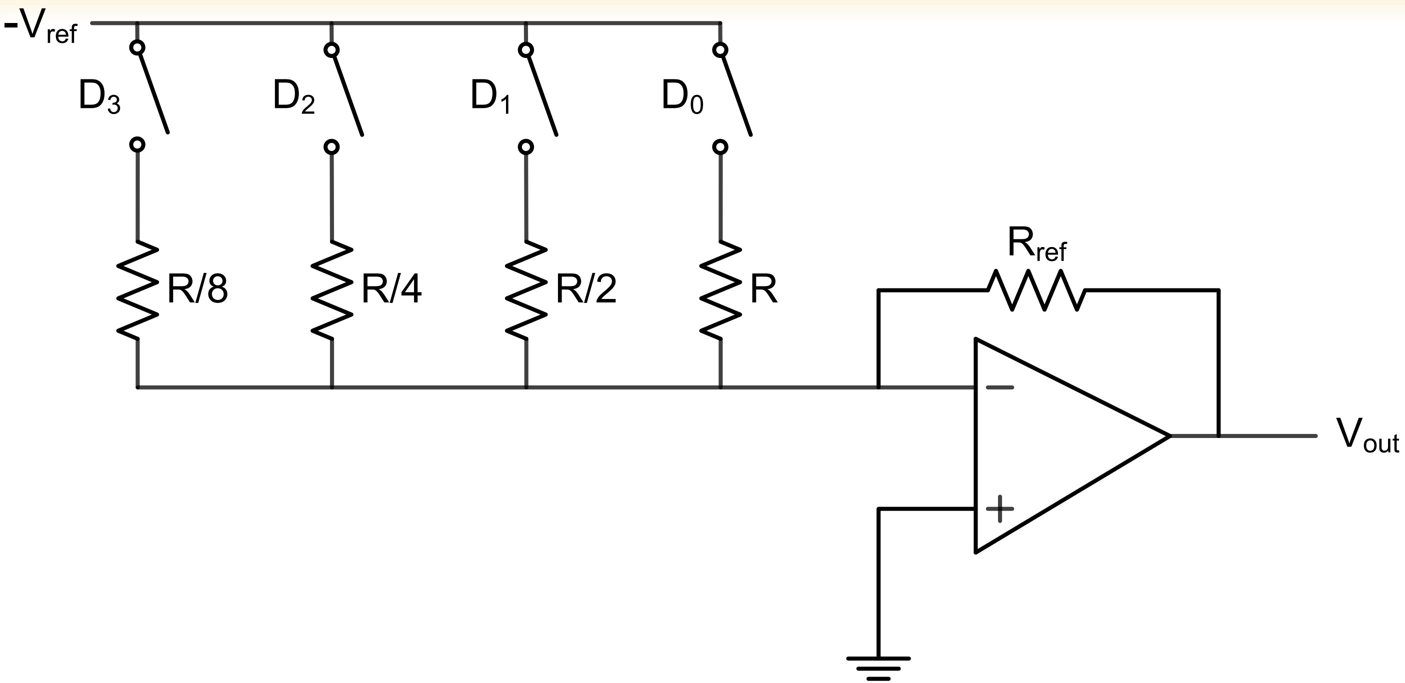
$$DAC_{output} = V_{ref} \times \frac{Digital\ Value}{4096}$$

- Many applications:
  - digital audio
  - waveform generation
- Performance parameters
  - speed
  - resolution
  - power dissipation
  - glitches

DAC Implementations

- Pulse-width modulator (PWM)
- Binary-weighted resistor (We will use this one as an example)
- R-2R ladder (A special case of binary-weighted resistor)

Binary-weighted Resistor DAC



$$V_{out} = V_{ref} \times \frac{R_{ref}}{R} \times (D_3 \times 2^3 + D_2 \times 2^2 + D_1 \times 2 + D_0)$$

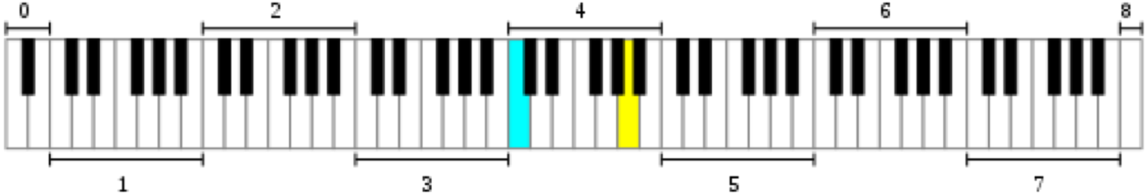
Digital Music

	0	1	2	3	4	5	6	7	8
C	16.352	32.703	65.406	130.813	261.626	523.251	1046.502	2093.005	4186.009
C#	17.324	34.648	69.296	138.591	277.183	554.365	1108.731	2217.461	4434.922
D	18.354	36.708	73.416	146.832	293.665	587.330	1174.659	2349.318	4698.636
D#	19.445	38.891	77.782	155.563	311.127	622.254	1244.508	2489.016	4978.032
E	20.602	41.203	82.407	164.814	329.628	659.255	1318.510	2637.020	5274.041
F	21.827	43.654	87.307	174.614	349.228	698.456	1396.913	2793.826	5587.652
F#	23.125	46.249	92.499	184.997	369.994	739.989	1479.978	2959.955	5919.911
G	24.500	48.999	97.999	195.998	391.995	783.991	1567.982	3135.963	6271.927
G#	25.957	51.913	103.826	207.652	415.305	830.609	1661.219	3322.438	6644.875
A	27.500	55.000	110.000	220.000	440.000	880.000	1760.000	3520.000	7040.000
A#	29.135	58.270	116.541	233.082	466.164	932.328	1864.655	3729.310	7458.620
B	30.868	61.735	123.471	246.942	493.883	987.767	1975.533	3951.066	7902.133

Musical Instrument Digital Interface (MIDI) standard assigns the note A as pitch 69.

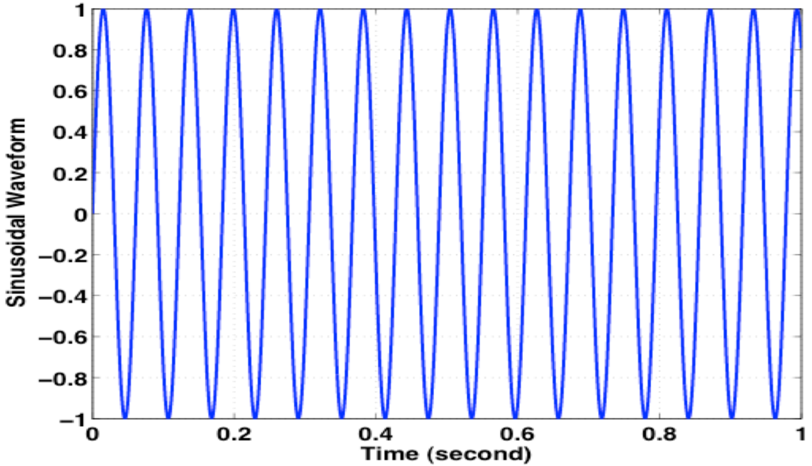
$$f = 440\times 2^{(p-69)/12} = 440$$

$$p = 69 + 12\times \log_2\left(\frac{f}{440}\right)$$



Digital Music

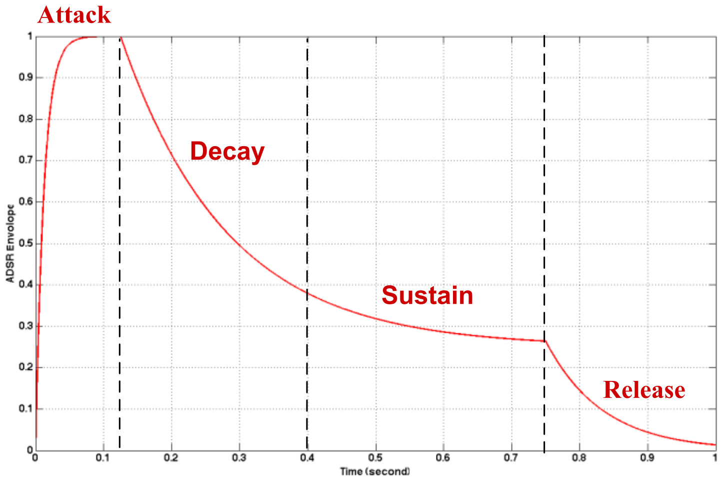
Generate Sine Wave



- No FPU available on the processor to compute sine functions
- Software FP to compute sine is slow
- Solution: Table Lookup
  - Compute sine values and store in table as fix-point format
  - Look up the table for result
  - Linear interpolation if necessary

Digital Music: Attack, Decay, Sustain, Release (ADSR)

➢ Amplitude Modulation of Tones (modulate music amplitude)



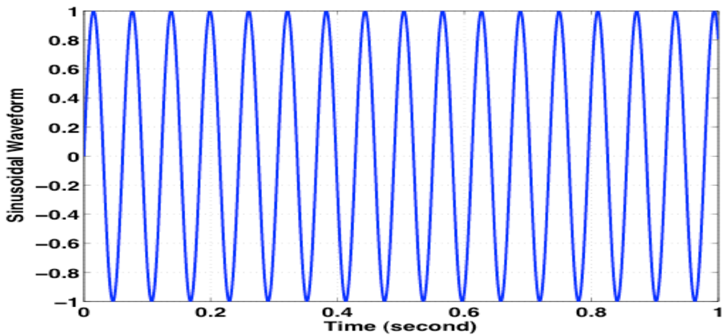
Implemented by a simple digital filter:

$$\mathrm{ADSR}(n) = g \times \overrightarrow{\mathrm{ADSR}} + (1 - g) \times \mathrm{ADSR}(n-1)$$

where $\overrightarrow{\mathrm{ADSR}}$ is the target modulated amplitude value,

g is the gain parameter.

Digital Music: ADSR Amplitude Modulation



✚