Analog-Digital Interface

Computer Organization
Architectures for Embedded Computing

Tuesday, 26 November 13
Summary

• Previous Class
  – Dependability
    • Redundancy
    • Error Correcting Codes

• Today:
  – Analog-Digital Interface
    • Converters, Sensors / Actuators
    • Sampling
    • DSP
    • Frequency spectrum of signals
• Everything in the physical world is an analog signal
  – Sound
  – Light
  – Temperature
  – Gravitational force
Small Computers Rule the Marketplace
Diversity of Devices
An Analog World

• Need to convert into electrical signals

• **Transducers**: device that converts a primary form of energy into a corresponding signal with a different energy form
  – Primary Energy Forms: mechanical, thermal, electromagnetic, optical, chemical, etc.
  – take form of a sensor or an actuator

• **Sensor** (e.g., thermometer)
  – a device that detects/measures a signal or stimulus
  – acquires information from the “real world”

• **Actuator** (e.g., heater)
  – a device that generates a signal or stimulus
Transducers

- Microphone/speakers
- Valve Control
- Motor Control
- Microaccelerometer
  - cantilever beam
  - suspended mass
- Pressure
- Gyroscope (rotation)
Sensor Calibration

- Sensors can exhibit non-ideal effects
  - offset: nominal output ≠ nominal parameter value
  - nonlinearity: output not linear with parameter changes
  - cross parameter sensitivity: secondary output variation with, e.g., temperature

- Calibration = adjusting output to match parameter
  - analog signal conditioning
  - look-up table
  - digital calibration
    - $T = a + bV + cV^2$, $T=$ temperature; $V=$ sensor voltage;
      - $a, b, c =$ calibration coefficients

- Compensation
  - remove secondary sensitivities
  - must have sensitivities characterized
  - can remove with polynomial evaluation
    - $P = a + bV + cT + dVT + e V^2$, where $P=$ pressure, $T=$ temperature
Analog-to-Digital Converter (ADC, A/D, or A to D): a device that converts continuous signals to discrete digital numbers.

- Quantizing - breaking down analog value in a set of finite states
- Encoding - assigning a digital word or number to each state and matching it to the input signal
ADC reads periodic samples of the input and generates a binary value.
Workings of an A/D

Clock signal

Input
analog signal

Sample and hold

analog signal segment

A/D Conversion

Output
Equally spaced Digital signal

Diagram shows the process of an A/D converter, including sampling and holding, analog-to-digital conversion, and the output as an equally spaced digital signal.
Sampling Resolution

• Resolution
  – Number of discrete values that represent a range of analog values
    • Eg, 3-bit ADC, 8 values
    • Range / 8 = Step

• Quantization Error
  – How far off discrete value is from actual
    – $\frac{1}{2}$ LSB $\rightarrow$ Range / 16
For an n-bit ADC, the number of possible states that the converter can output is:

\[ N = 2^n \]

Analog quantization size:
\[ Q = \frac{V_{\text{max}} - V_{\text{min}}}{N} \]

Example: For a 0-10V signals and a 3-bit A/D converter.
\[ N = 2^3 = 8. \]

Analog quantization size:
\[ Q = \frac{(10V - 0V)}{8} = 1.25V \]
Each bit is weighted with an analog value, such that a 1 in that bit position adds its analog value to the total analog value represented by the digital encoding.

<table>
<thead>
<tr>
<th>Digital Bit</th>
<th>Bit Weight (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>10/2 = 5</td>
</tr>
<tr>
<td>6</td>
<td>10/4 = 2.5</td>
</tr>
<tr>
<td>5</td>
<td>10/8 = 1.25</td>
</tr>
<tr>
<td>4</td>
<td>10/16 = 0.625</td>
</tr>
<tr>
<td>3</td>
<td>10/32 = 0.313</td>
</tr>
<tr>
<td>2</td>
<td>10/64 = 0.157</td>
</tr>
<tr>
<td>1</td>
<td>10/128 = 0.078</td>
</tr>
<tr>
<td>0</td>
<td>10/256 = 0.039</td>
</tr>
</tbody>
</table>
• What range to use?

![Diagram showing sampling range](image)

- **Range Too Small**
  - Ideal Range: $V_{r\,+}$ to $V_{r\,-}$

- **Range Too Big**
  - Ideal Range: $V_{r\,+}$ to $V_{r\,-}$

- **Ideal Range**
  - Ideal Range: $V_{r\,+}$ to $V_{r\,-}$
Digital-to-Analog Converter (DAC, D/A or D to A): device for converting a digital (usually binary) code to an analog signal (current, voltage or charges).

Digital-to-Analog Converters are the interface between the abstract digital world and the analog real life.

Simple switches, a network of resistors, current sources or capacitors may implement this conversion.
Digital-to-Analog Resolution

Poor Resolution (1 bit)

Better Resolution (3 bits)

**Vout**

**Desired Analog signal**

**Approximate output**

**Digital Input**

**Vout**

**Desired Analog signal**

**Approximate output**

**Digital Input**
Signal from DAC can be smoothed by a Low-pass filter.
Analog Circuits

- Most real-world signals are analog
- They are continuous in time and amplitude
- Analog circuits process these signals using
  - Resistors
  - Capacitors
  - Inductors
  - Amplifiers
  - ...
- Analog signal processing examples
  - Audio processing in FM radios
  - Video processing in traditional TV sets
Limitations of Analog Signal Processing

- Accuracy limitations due to
  - Component tolerances
  - Undesired nonlinearities
- Limited repeatability due to
  - Tolerances
  - Changes in environmental conditions
    - Temperature
    - Vibration
- Sensitivity to electrical noise
- Limited dynamic range for voltage and currents
- Inflexibility to changes
- Difficulty of implementing certain operations
  - Nonlinear operations
  - Time-varying operations
- Difficulty of storing information
Digital Signal Processing

- Represent signals by a sequence of numbers
  - Sampling or analog-to-digital conversions
- Perform processing on these numbers with a digital processor
  - Digital signal processing
- Reconstruct analog signal from processed numbers
  - Reconstruction or digital-to-analog conversion

- Analog input – analog output
  - Digital recording of music
- Analog input – digital output
  - Touch tone phone dialing
- Digital input – analog output
  - Text to speech
- Digital input – digital output
  - Compression of a file on computer
Pros and Cons of Digital Signal Processing

• Pros
  – Accuracy can be controlled by choosing word length
  – Repeatable
  – Sensitivity to electrical noise is minimal
  – Dynamic range can be controlled using floating point numbers
  – Flexibility can be achieved with software implementations
  – Non-linear and time-varying operations are easier to implement
  – Digital storage is cheap
  – Digital information can be encrypted for security
  – Price/performance and reduced time-to-market

• Cons
  – Sampling causes loss of information
  – A/D and D/A requires mixed-signal hardware
  – Limited speed of processors
  – Quantization and round-off errors
Sampling Rate: frequency at which ADC evaluates analog signal.

What sample rate do we need?
- Too little: we can’t reconstruct the signal we care about
- Too much: waste computation, energy, resources
Aliasing: different frequencies are indistinguishable when they are sampled.

For example, a 2 kHz sine wave being sampled at 1.5 kHz would be reconstructed as a 500 Hz (the aliased signal) sine wave.
Nyquist Sampling Theorem

If a continuous-time signal contains no frequencies higher than $f_{\text{max}}$, it can be completely determined by discrete samples taken at a rate:

$$f_{\text{sample}} > 2 \times f_{\text{max}}$$

$f_{\text{sample}} = 2f_{\text{max}}$ is known as the Nyquist Sampling frequency

Example:
Humans can process audio signals 20 Hz – 20 KHz
Audio CDs: sampled at 44.1 KHz
Jean Fourier proposed a wild idea in 1807:

Any periodic function can be rewritten as a weighted sum of Sines and Cosines of different frequencies.

\[ f(t) = a_0 + \sum_{n=1}^{\infty} \left( a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t) \right) \]

- Lagrange, Laplace, Poisson and others found it hard to believe!
- Called Fourier Series
  - Odd functions only need the sine
  - Even functions only need the cosine
A Sum of Sinusoids

• Building block:

\[ A_n \sin(n \omega x + \varphi_n) \]

• Add enough of them to get any signal \( f(x) \) you want!
A Sum of Sinusoids

- **Square Wave**
- **Sawtooth Wave**
- **Triangle Wave**
- **Semicircle**
• Reparametrize the signal by $\omega$ instead of $x$:

\[
\begin{align*}
f(x) & \quad \xrightarrow{\text{Fourier Transform}} \quad F(\omega)
\end{align*}
\]

For every $\omega$ from 0 to $\infty$, $F(\omega)$ holds the amplitude $A$ and phase $\phi$ of the corresponding sine: $A \sin(\omega x + \phi)$

\[
F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi\omega x} \, dx
\]

\[
e^{ik} = \cos k + i \sin k \quad i = \sqrt{-1}
\]

Time domain ($x$) $\Rightarrow$ Frequency domain ($\omega$)

• $F(\omega)$ is the **frequency spectrum** of $f(x)$
Inverse Fourier Transform

- Using a similar, but inverse transformation, the signal in the $x$ domain can be obtained from the frequency domain $\omega$:

$$f(x) = \int_{-\infty}^{\infty} F(\omega) e^{i2\pi\omega x} \, dx$$

Many operations, specially with sound, image and video, are more easily computed in the frequency domain.
Common Transform Pairs

<table>
<thead>
<tr>
<th>Time Function</th>
<th>Frequency Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Boxcar</strong></td>
<td><strong>Sinc</strong></td>
</tr>
</tbody>
</table>
| \( G(t) = \begin{cases} 
1, & |t| < \tau/2 \\
0, & |t| > \tau/2 
\end{cases} \) | \( s(f) = \frac{1}{\tau} \text{sinc} \left( \frac{f}{\tau} \right) \) |

Sinc function graph:
- \( s(f) = \frac{1}{\tau} \text{sinc} \left( \frac{f}{\tau} \right) \)
Discrete Fourier Transform

- Fourier transform applies equally to discrete-time signals:

\[ h_k = \frac{1}{N} \sum_{n=0}^{N-1} H_n e^{-2\pi i kn/N} \quad H_n = \sum_{k=0}^{N-1} h_k e^{2\pi i kn/N} \]
Conclusions

• Physical quantities need to be converted to binary in order to be processed by computers
  – Sensors translate to electric signals
  – Actuators perform physical actions with electric commands

• To levels of discretization:
  – Amplitude
    • DACs & ADCs
  – Time
    • Sampling

• Signals can be converted to the frequency domain
  – Frequency spectrum
  – Nyquist theorem
  – Many operations easier in this domain: DSP
Analog-Digital Interface

Computer Organization
Architectures for Embedded Computing

Tuesday, 26 November 13