What is Digital Signal Processing?

• Generically, Digital Signal Processing refers to manipulating analog signals using digital operations.

• There are special purpose Digital Signal Processors, or DSP chips that are essentially microprocessors that are customized for implementing DSP algorithms.

• DSP chips, however, *are not* necessary for performing signal processing in the digital domain.
Types of DSP

• Many things can be called digital signal processing.

• Often, DSP programs are applications of discrete-time signal processing to do:
  – Filtering (high-pass, low-pass, band-pass, etc)
  – Modulation / Demodulation
  – DFT / Correlation

• Generically, any operation on a digital signal is DSP.
Generic System Model

- Analog Input
  - A/D Conversion
  - Digital Processing
  - D/A Conversion
  - Analog Output
- Digital Input
  - Protocol Conversion
  - Digital Processing
  - Protocol Conversion
  - Digital Output
- Digital Input
Focus on Analog Input

Analog Input → A/D Conversion → Digital Signal

Analog Filter → Analog Gain → Analog Bias → A/D Converter → Digital Output

Conversion Clock

Sampling Clock

Power Source(s)
Focus on Analog Output

A block diagram illustrating the process of converting a digital signal to an analog output. The diagram includes steps such as D/A conversion, analog signal, output, and additional elements like digital input, conversion clock, sampling clock, analog filter, analog bias, analog gain, and power source(s).
A Simple Example System

• Let’s suppose we want to design a simple digital voice recorder.
• Some “obvious” requirements:
  – a microphone that collects an analog signal,
  – a converter that turns the analog signal into a digital signal, and
  – a processor that stores the digitized analog signal for subsequent playback.
Ok, Where do we go Next?

• The absolute **WRONG** question to ask at this point is:

  “Is _________ OK to use for this project?”
We Need REAL Specifications!

• High level specifications need to be broken down into lower level specifications.
• We know from our earlier block diagram that we probably need some kind of signal conditioning between the user and the microprocessor.
• How do we figure out what we need?
Start with the Inputs

• So, what is our input signal?
• Human voice characteristics:
  – “Normal” speech has an amplitude of 40 to 60 dB SPL.
  – Intelligibility of speech increases with bandwidth.
  – “Telephonic” bandwidths of < 4kHz guarantee ambiguity.
  – Bandwidths around 7kHz significantly reduce ambiguity.
  – “Telephonic” dynamic range is about 48dB
What Next?

• Well, we need a microphone with a frequency response of at least 7kHz.
• A quick look at the DigiKey website finds the Knowles MB3015USB-4:

<table>
<thead>
<tr>
<th>ITEM</th>
<th>SYMBOL</th>
<th>TEST CONDITION</th>
<th>MINIMUM</th>
<th>STANDARD</th>
<th>MAXIMUM</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>S</td>
<td>f=1kHz, Pin=1Pa</td>
<td>-55</td>
<td>-51</td>
<td>-47</td>
<td>dB</td>
</tr>
<tr>
<td>Impedance</td>
<td>Zout</td>
<td>f=1kHz, Pin=1Pa</td>
<td></td>
<td></td>
<td>22</td>
<td>k Ω</td>
</tr>
<tr>
<td>Directivity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UNIDIRECTIONAL</td>
</tr>
<tr>
<td>Current Consumption</td>
<td>I</td>
<td></td>
<td></td>
<td>0.5</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>S/N Ratio</td>
<td>S/N (A)</td>
<td>f=1kHz, Pin=1Pa</td>
<td></td>
<td>55</td>
<td></td>
<td>dB</td>
</tr>
<tr>
<td>Sensitivity Reduction</td>
<td>ΔS</td>
<td>f=1kHz, Pin=1Pa</td>
<td></td>
<td>-3</td>
<td></td>
<td>dB</td>
</tr>
<tr>
<td>Frequency Range</td>
<td></td>
<td></td>
<td>100-10,000</td>
<td></td>
<td></td>
<td>Hz</td>
</tr>
</tbody>
</table>

Microphone Output

• The specification sheet for the microphone tells us exactly what we can expect as output from the microphone.
• Unfortunately, the information we want is “hidden” in the specifications.
• Specifically, the “sensitivity” tells us how many volts the microphone outputs per “Pascal.”
• That makes things much clearer – yeah, right.
A Bit More Research

- After a bit of research, we learn that:
  \[ S = 20 \log(V_{OC}) \bigg|_{1V/Pa} \]

- So, for the minimum and maximum sensitivities of this microphone, we get:
  \[ S = -55dB \Rightarrow 10^{-55/20} = 1.78 \times 10^{-3} \text{ V/Pa} = 1.78 \times 10^{-3} \text{ mV/Pa} \]
  \[ S = -47dB \Rightarrow 10^{-47/20} = 4.47 \times 10^{-3} \text{ V/Pa} = 4.47 \times 10^{-3} \text{ mV/Pa} \]

- Let’s pick a design target of about 3mV/Pa – approximately in the middle.
Still More Research

• “Normal” speech has a range of 40-60 dB SPL.
  • 40 dB SPL = 2 x 10^{-3} \text{ Pa}
  • 60 dB SPL = 2 x 10^{-2} \text{ Pa}

• So, for the microphone we selected, the microphone output will be:
  • At 40 dB SPL it will output 6 x 10^{-6} \text{ V} = 6\mu\text{V}
  • At 60 dB SPL it will output 6 x 10^{-5} \text{ V} = 60\mu\text{V}

• Pretty small voltages!
Characteristics of A/D Converters

- An A/D converter translates an analog input voltage into a digital representation.
- Here we see a ±5V signal being mapped to a 16-bit 2’s complement integer:

![Diagram of A/D conversion](http://www.audiodesignline.com/showArticle.jhtml?articleID=192200610)
Word Size and Dynamic Range

• The number of bits in an A/D converter determines the “dynamic range” of the converter.
• This “dynamic range” is the ratio of the smallest and largest values that can be represented.
• DR dB = 20 log (V_{max}/V_{min})
• Different applications require different dynamic ranges.

<table>
<thead>
<tr>
<th>Audio Device/Application</th>
<th>Dynamic Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM Radio</td>
<td>48 dB</td>
</tr>
<tr>
<td>Analog Broadcast TV</td>
<td>60 dB</td>
</tr>
<tr>
<td>FM Radio</td>
<td>70 dB</td>
</tr>
<tr>
<td>Analog Cassette Player</td>
<td>73 dB</td>
</tr>
<tr>
<td>Video Camcorder</td>
<td>75 dB</td>
</tr>
<tr>
<td>ADI SoundPort Codecs</td>
<td>80 dB</td>
</tr>
<tr>
<td>16-bit Audio Converters</td>
<td>90 to 95 dB</td>
</tr>
<tr>
<td>Digital Broadcast TV</td>
<td>85 dB</td>
</tr>
<tr>
<td>Mini-Disk Player</td>
<td>90 dB</td>
</tr>
<tr>
<td>CD Player</td>
<td>92 to 96 dB</td>
</tr>
<tr>
<td>18-bit Audio Converters</td>
<td>104 dB</td>
</tr>
<tr>
<td>Digital Audio Tape (DAT)</td>
<td>110 dB</td>
</tr>
<tr>
<td>20-bit Audio Converters</td>
<td>110 dB</td>
</tr>
<tr>
<td>24-bit Audio Converters</td>
<td>110 to 120 dB</td>
</tr>
<tr>
<td>Analog Microphone</td>
<td>120 dB</td>
</tr>
</tbody>
</table>

Source: http://www.audiodesignline.com/showArticle.jhtml?articleID=192200610
Consider an 8-bit Converter

• Assume 0-3.3V maps to 00-FF
  – 256 steps means 0.0129 Volts/step
  – DR dB = 20 \log\left(\frac{3.3}{0.0129}\right) = 48 \text{ dB}

• So, comparing this to the previous slide, we see:
  – 8-bits is about “AM Radio” quality
  – 12-bits is about “FM Radio” quality
  – 16-bits is about “CD” quality

• But wait, there’s more!
Other Issues Selecting a Converter

- It’s generally good to leave some “headroom” when designing so the system won’t distort.
- In our example, let’s assume a 0 to 3.3V converter.
- Let’s also assume some more specifications:
  - 12dB headroom

Source: http://www.audiodesignline.com/showArticle.jhtml?articleID=192200610
Making Space

• If we want “AM Radio” quality in our recorder, we need 48 dB of dynamic range for our “nominal” signal.

• If we want an additional 12 dB of headroom, we need 60 dB of dynamic range in the converter.

• This means:
  – 10-bit converter
  – A 60μV input signal must drive the converter with 0.825 Volts
Setting Gains and Levels

• So, now we know that the voltage gain of the analog stage is:
  \[ \frac{0.825\text{V}}{60\mu\text{V}} = 13750. \]

• We also know that since the converter is 0-3.3V, we will need to make sure that the output of the gain stage is centered at 1.65 Volts.
What About Anti-Aliasing?

• We know that in general we need a band-limited signal as an input to the converter.
• It doesn’t matter how the signal is band-limited, it just matters that it is band-limited.
• We have two choices:
  – Sample at 20+ kHz and let the microphone characteristics band-limit the signal.
  – Sample at 14+ kHz and build a separate low-pass filter prior to the converter.
• One choice might require a faster processor, the other requires more parts.
Anything Else?

• Now let’s take a look at the microphone data sheet a bit more carefully.
• We see there are 2 components we need to add.
• We also see that the microphone needs voltage!

Finding a Converter

• A good first step is to go to DigiKey and search for 10-bit converters in DIP packages. Here’s what I found:

Source: http://www.digikey.com
Finding a Converter

- Next, I see that both serial and parallel converters are available, the choice would be based on the pins available on the microcontroller and how much software you’re willing to write.

<table>
<thead>
<tr>
<th>Data Interface</th>
<th>Features</th>
<th>Manufacturer</th>
<th>Package / Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel</td>
<td>25KHz Sampling Rate</td>
<td>Analog Devices Inc</td>
<td>8-DIP</td>
</tr>
<tr>
<td></td>
<td>26KHz Sampling Rate</td>
<td>Linear Technology</td>
<td>14-DIP</td>
</tr>
<tr>
<td></td>
<td>31kHz Sampling Rate</td>
<td>Maxim Integrated Products</td>
<td>16-DIP</td>
</tr>
<tr>
<td></td>
<td>38KHz Sampling Rate</td>
<td>Microchip Technology</td>
<td></td>
</tr>
<tr>
<td></td>
<td>73kHz Sampling Rate</td>
<td>Texas Instruments</td>
<td></td>
</tr>
<tr>
<td></td>
<td>108kHz Sampling Rate</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>133kHz Sampling Rate</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>200kHz Sampling Rate</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>350kHz Sampling Rate</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>400kHz Sampling Rate</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: http://www.digikey.com
ADC Component Selection

- I picked a Microchip MCP3001 (actually, it was my 3rd choice).
  - Serial Converter
  - Single Channel
  - 200 ksp sample rate (75 ksp @ 2.7V)
  - 8-pin DIP package
  - Very low cost -- $2.18

- Higher than needed sample rate, but that’s not a problem!
Analog Input Model

• To drive the ADC, we’ll need to know its input impedance.
• An analog input model is provided in the data sheet:

![Analog Input Model Diagram]

Digital Output

- We would also need to confirm that the digital I/O is compatible with the logic family that is used by the microcontroller:

<table>
<thead>
<tr>
<th>Digital Input/Output:</th>
<th></th>
<th>Straight Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Coding Format</td>
<td>V_{IH}</td>
<td>0.7 V_{DD}</td>
</tr>
<tr>
<td>High Level Input Voltage</td>
<td>V_{IL}</td>
<td>—</td>
</tr>
<tr>
<td>Low Level Input Voltage</td>
<td>V_{OH}</td>
<td>4.1</td>
</tr>
<tr>
<td>High Level Output Voltage</td>
<td>V_{OL}</td>
<td>—</td>
</tr>
<tr>
<td>Low Level Output Voltage</td>
<td>I_{IH}</td>
<td>-10</td>
</tr>
<tr>
<td>Input Leakage Current</td>
<td>I_{IL}</td>
<td>—</td>
</tr>
<tr>
<td>Output Leakage Current</td>
<td>I_{OL}</td>
<td>-10</td>
</tr>
<tr>
<td>Pin Capacitance</td>
<td>C_{IN}, C_{OUT}</td>
<td>—</td>
</tr>
</tbody>
</table>

And We Need the Timing

- The timing diagrams for the device show how a microcontroller controls the sampling and receives data from the device.
- This device is compatible with SPI interfaces.
We’re Done! (with the input)

Analog Input

Microphone MB3015

Gain 13,750

Analog Bias

ADC MCP3001

SPI Bus

0-0.825V
100-10kHz
Single-Supply

0-60μV
100-10kHz

1.65-2.475V
100-10kHz

CS - 20 kHz rate
CLK - 260 kHz rate

Power Source(s)

1.65V
3.3V

See, it’s EASY!!!