

ECE 4624: Meeting 5

Introduction to DSP Hardware

Chris Wyatt

2025-09-10

Today we look at DSP hardware and focus on the use of general-purpose and microcontroller-based computers for implementing DSP systems. Today's lecture reviews some prerequisite material from ECE 2564 (Embedded Systems).

- ▶ DSP Hardware
- ▶ DSP Software
- ▶ Arduino Uno R4 Minima
- ▶ Lab 1

DSP Hardware Platforms

- ▶ ASICS
- ▶ FPGAs
- ▶ DSP Processors
- ▶ General-Purpose CPU and μ C
- ▶ GPUs and other accelerators

DSP Programming Models and Languages

- ▶ non-realtime
- ▶ soft realtime and audio-video processing
- ▶ hard realtime and digital control

Factors when Choosing Hardware

- ▶ flexibility
- ▶ design time
- ▶ power consumption
- ▶ performance
- ▶ development cost
- ▶ production cost

Architectures

- ▶ Harvard (most DSPs)
- ▶ von Neumann (most GPCPU and μ C)

Basic computational elements

► multiply

$$y[n] = \alpha x[n]$$

► accumulate (add in place)

$$y[n] = x_1[n] + x_2[n]$$

► delay

$$y[n] = x[n - 1]$$

While these are simple operations in theory, and in practice using floating-point, they take more care in fixed-point.

Fixed-Point Representation and Hardware

- ▶ Given a value x , the fixed-point representation is

$$x = (b_{-A}, b_{-A+1}, \dots, b_{-1}, b_0, b_1, \dots, b_B)_r$$

where r is the radix and $b_i \in [0, (r-1)]$ is a digit.

$$x = \sum_{i=-A}^B b_i r^{-i}$$

- ▶ we are interested in base 2 ($r = 2$) where digits $(0, 1)$ are called bits.
 - ▶ b_{-A} is the most-significant bit (MSB)
 - ▶ b_B is the least-significant bit (LSB)
 - ▶ the location of the point is implied
- ▶ unsigned N-bit numbers where $A = N - 1$ and $B = 0$ can store positive integers $[0, 2^N - 1]$

Fixed-Point Representation and Qm.n format

- ▶ In DSP our values are often less than 1 in magnitude, thus we generally use the fractional format where $A = 0$, $B = N - 1$

$$x = 2^A \sum_{i=0}^{N-1} b_i 2^{-i}$$

and assume $2^A = 1$. This is called the Q1.n format.

- ▶ In general the Qm.n format uses m bits for the integer part and n bits for the fractional part
- ▶ when multiplying and adding numbers the formats need to match
- ▶ to convert between formats use the shifting operators (« and ») in C

Fixed-Point representation of negative numbers

► sign magnitude

$$x \geq 0 \quad (0.b_1, b_2, \dots, b_B)_2$$

$$x < 0 \quad (1.b_1, b_2, \dots, b_B)_2$$

► ones-complement

$$x < 0 \quad (1.\bar{b}_1, \bar{b}_2, \dots, \bar{b}_B)_2$$

$$x \geq 0 \quad \text{bitwise complement}$$

► twos-complement

$$x < 0 \quad (1.\bar{b}_1, \bar{b}_2, \dots, \bar{b}_B)_2 + (0.0, 0, 0, \dots, 1)_2$$

modulo-2 and ignore the carry.

Most ALU, including those in DSPs use twos-complement.

Floating-Point Representation and Hardware

The alternative to fractional fixed-point arithmetic is floating point.

- ▶ in fixed point the resolution is fixed (distance between numbers)

$$\Delta = \frac{x_{max} - x_{min}}{2^N}$$

- ▶ floating point representations increase the dynamic range at cost of varying resolution and significant hardware complexity

$$x = M \cdot 2^E$$

where M is a sign-magnitude fraction and E is a signed integer.

- ▶ there are many floating-point formats (number of bits for M and E , how to handle $0, \infty$, NAN etc)
- ▶ IEEE 754 is a widespread standard
- ▶ using floating point is a cost/performance tradeoff

Quantization Errors

- ▶ signal quantization
- ▶ coefficient quantization

Arithmetic Errors

- ▶ roundoff or truncation
- ▶ overflow and underflow
- ▶ limit cycles

Course Development System

- ▶ board description
- ▶ PlatformIO IDE
- ▶ event loop
- ▶ ADC Setup
- ▶ DAC Setup
- ▶ resources

Lab 1