Chapter 8: Digital Signal Processing

I. Overview

Digital signal processing, or DSP, is manipulation of analog and digital signals using digital electronics technology for producing digital and analog output signals. DSP is a major type of signal processing: it is used in countless electronics intensive applications such as radar signal processing, CD player operation, video and cellphone data compression, etc … Recently, FPGAs have become one of the key components of many DSP circuit designs for their inherent simplicity, speed, parallel nature, and low power.

II. Basic architecture

The basic design architecture of DSP circuit (see figure 1 below) consists of an ADC front end which converts analog input signals into digital form for processing with a microprocessor, DSP processor, or an FPGA. The processed digital signal is then either outputted in digital format or converted back to analog form with a DAC.

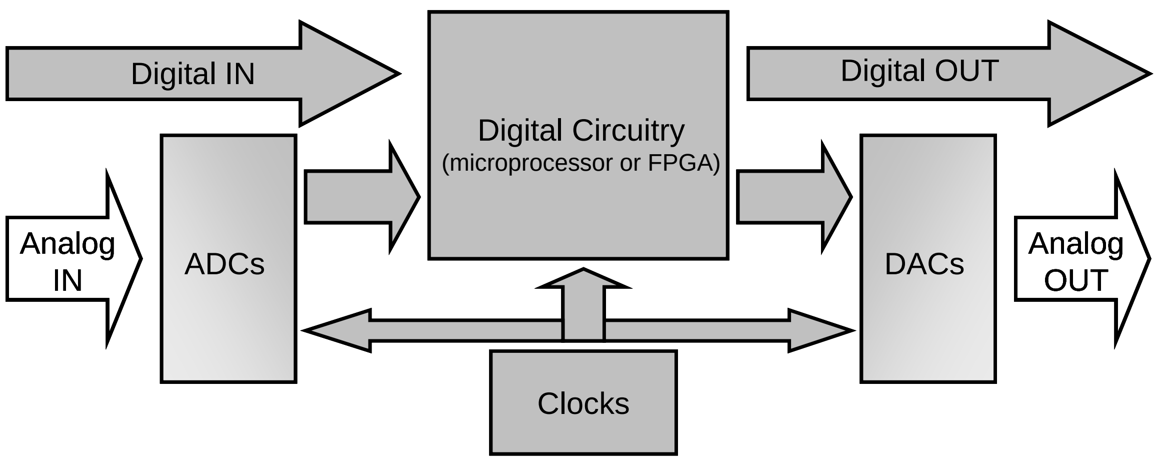

Figure 1: Basic DSP circuit architecture.

A DSP processor is microprocessor with specialized signal processing instruction circuitry. A DSP processor is used for complex tasks which can most easily accomplished with a computer program, frequently in assembly language or C, but that do not require high-speed. One of the most common DSP processors is the TMS320 (Texas Instruments) family of microprocessors.

When high speed is required the digital circuitry of choice is an FPGA. FPGAs are a little harder to program (not much though if the processing is simple) that DSP processors, but they are much faster, lower power, and easy to almost fully simulate (you do not have to worry about inopportune operating system interrupts). On the other hand, microprocessors are much harder to simulate.

The most common applications of DSP are signal filtering, Fourier signal analysis by Fast Fourier Transform (FFT), and arithmetic operations such as multiply and divide. While all of these types of transformations are conceptually simple, they are quite difficult to efficiently and accurately implement with analog circuitry. All these can be implemented very accurately with digital circuits.

III. Applications

DSP is widely used to process signals in many applications and instruments. A few common applications of DSP are the following:

a. Long-distance communication: Low-noise, long distance communication can be accomplished by transmitting a signal in digital form, which is inherently less susceptible to noise. A DSP communication module would also include compression of the digital data and the use of error correction to further reduce noise.

b. Voice recognition: Electronic telephone answering and call direction services use DSP to interpret and recognize voice commands.

c. DSP radio: An FPGA-based DSP radio can use digital multiplication to demodulate a radio signal at RF frequencies down to audio frequencies. The DSP circuit can also filter the signal or remix an audio signal for transmission. FPGA-based radio circuits are frequently used in cellphone communications.

d. DSP lock-in amplifier: DSP technology has lead to DSP-based lock-in amplifiers with very high sensitivity and accuracy.

IV. DSP vs. Analog Electronics

The choice of a DSP circuit design versus an analog circuit is essentially a choice between digital and analog electronics. You can always push the envelope further with an analog circuit than a digital circuit, but analog circuit design is much harder than digital circuit design, in particular at RF frequencies. Analog circuit design at RF frequencies requires many years of experience, sophisticated and expensive prototyping and simulation software, and considerable enthusiasm for searching and destroying various sources of noise. Digital circuit design at RF frequencies is relatively straightforward and requires only a one semester college course to write FPGA HDL circuit programs which can operate up to several hundred megahertz. Low noise, digital communication up to 100 Mbits/s can be accomplished with a twisted wire pair. Digital circuitry is inherently less sensitive to noise up to a certain threshold past which it is useless, whereas analog circuits become progressively disrupted by external and internal noise sources.

Design Exercises

Design Exercise 8-1: Construct a Quartus II Verilog project which will output the numeric 8-bit values for two sinewaves each with a amplitude of 255 and time-sampled 100 times per period, but with a variable phase with a resolution of 3.6°.

Design Exercise 8-2: Choose a periodic function and sample it 1024 times. Use Excel or MatLab to calculate the Discrete Fourier Transform (DFT) of your sampled function using an FFT algorithm (in Excel you can use Tools > Data Analysis > Fourier Analysis). Verify 5 random frequency space points to make sure that the FFT is properly computing the DFT. Plot the result with correct units on each axis. Discuss the structure of the plot.