A Mini Stereo Digital Audio Processor (MSDAP)

Zhangnong Jiang

Electrical Engineering Department, UCLA

I. Motivation

Digital audio signal processing has become increasingly popular in multimedia systems, largely thanks to the advent of general digital signal processor (DSP) chips and high-precision oversampling A/D and D/A converts [1, 2]. For portable and home entertainment audio systems, however, the general-DSP-chip-based solution might be too costly and consume too much power, at least in near years. Then a low-cost and low-power solution for this purpose should be sought. By carefully examining digital audio processing applications, we believe there is a possibility to develop a tiny digital audio processor chip that is capable of performing several basic audio processing functions with much improved performance/(cost.power) ratios over a general DSP chip. The fundamental function for this mini stereo digital audio processor (MSDAP) in our mind is a programmable finite impulse response (FIR) digital filter that has various audio applications [1, 2].

The performance of the MSDAP is expected to be the same level as that of two general DSP chips for implementing two-channel FIR digital filtering in audio applications. Its cost might be reduced by a factor of five to ten, compared to a general DSP chip and, of course, depending on the production volume. Power dissipation of the MSDAP chip will fall on the level of 100 mW so that it can also be used in portable systems. (A general DSP chip usually consumes five to ten times more power using the same IC technology and having comparable performance.)

II. Algorithm Description

Although there exist a great number of audio processing algorithms [1, 2], the MSDAP will focus on the FIR digital filtering – a fundamental and computation-intensive task. The FIR digital filtering involves the following linear convolution:

$$y(n) = \sum_{k=0}^{N} h(k) \times x(n-k) \qquad (1)$$

where $x(n)$ and $y(n)$ are input and output audio sequences, and $h(k)$ are filter coefficients with the filter order N. (An FIR filter structure is chosen because its simplicity will facilitate hardware implementations and make computing errors negligible. Also, an FIR filter can well mimic some primitive infinite impulse response (II) filter transfer functions. See Section V for more details.)

Obviously, the computation in (1) calls for $N+1$ multiplications and N additions. For linear-phase FIR filters, the symmetry (anti-symmetry) of filter coefficients can save multiplications by half [3]. A hardware multiplier, either a fixed-point or a floating-point version, is invariantly integrated on a general DSP chip so that it can accommodate a variety of application requirements. Since with the MSDAP only the linear convolution (1) will be performed, a much simpler hardware architecture than that of a general DSP chip will be employed.

One way to reduce the hardware complexity of an FIR filter is to efficiently code the filter coefficients. When the coefficients $h(k)$ is coded by using the minimum-number power-of-two (POT) digits, multiplications can thus be performed at a very high efficiency. For example, suppose that $h(k)=0.1172$ and a good approximation, $h(k) \approx 2^{-3}-2^{-7}+2^{-16}$, is available, the multiplication $h(k)x(n-k)$ shows that the computation procedure $h(k)x(n-k)=2^{-3}x(n-k)-2^{-7}x(n-k)+2^{-16}x(n-k)$ uses only three shifts and three addition/subtractions, and it is several times more efficient than using a general multiplier with the same precision. Optimal POT coefficients for an FIR filter are usually obtained by using integer optimization techniques that have been widely researched at least for a decade [4-8].

One problem involved in hardware implementation of programmable FIR filters with POT coefficients is how to perform shift operations. Due to high-precision requirements in digital audio systems, filter coefficients might need a big POT digit such as $\pm 2^{-16}$. Therefore, a programmable shifter from one to sixteen bits may be required. Such a programmable shifter is not attractive in hardware implementation because it has a huge size and operates slowly. An excellent programmable FIR filtering structure using only a one-bit shifter was proposed in [6]. This structure transforms the computation in (1) into a series of addition/subtraction operations that can easily be expressed as:

$$y(n) = 2^{-1}(\cdots 2^{-1}(2^{-1}(2^{-1}u_1 + u_2) + u_3) + \ldots) + u_{16}) \tag{2}$$

and

$$u_j = x_j(1) + x_j(2) + \cdots + x_j(r_j) \qquad 1 \le j \le 16 \tag{3}$$

where $x_j(l) \in \{\pm x(n-k)\}, 1 \le l \le r_j$, and r_j is the total number of the POT digits $\pm 2^{-j}$ occurred among all filter POT coefficients. By way of illustrating the equivalence between (1) and (2), we consider a design example that is a 7th-order FIR filter performing $y(n) = \sum_{k=0}^{7} h(k) \times x(n-k)$. Suppose that its POT coefficients are:

$$h(0) = 2^{-1} - 2^{-3} + 2^{-10} + 2^{-13} - 2^{-15}; \quad h(1) = 2^{-2} + 2^{-4} - 2^{-8} - 2^{-12}; \quad h(2) = -2^{-3} + 2^{-5} - 2^{-11};$$
$$h(3) = 2^{-4} + 2^{-9} - 2^{-14}; \quad h(4) = -2^{-5} - 2^{-9} + 2^{-13}; \quad h(5) = 2^{-6} + 2^{-10} + 2^{-15};$$
$$h(6) = -2^{-8} - 2^{-12} + 2^{-16}; \quad h(7) = 2^{-9} - 2^{-14} - 2^{-16}.$$

Based on the above POT coefficients, it is readily to obtain $u_j (1 \le j \le 16)$:

$$u_1 = x(n-6) - x(n-7); \quad u_2 = -x(n) + x(n-5); \quad u_3 = -x(n-3) - x(n-7); \quad u_4 = x(n) + x(n-4);$$
$$u_5 = -x(n-1) - x(n-6); \quad u_6 = -x(n-2); \quad u_7 = x(n) + x(n-5); \quad u_8 = x(n-3) - x(n-4) + x(n-7);$$
$$u_9 = -x(n-1) - x(n-6); \quad u_{10} = 0; \quad u_{11} = x(n-5); \quad u_{12} = x(n-2) - x(n-4);$$
$$u_{13} = x(n-1) + x(n-3); \quad u_{14} = -x(n) - x(n-2); \quad u_{15} = x(n-1); \quad u_{16} = x(n).$$

The MSDAP will be able to implement two FIR filters with order up to *N=255*, and the total number *R* of addition/subtraction operation involved in computing (2) and (3) is chosen no more than 512, i.e., $R = \sum_{j=1}^{16} r_j < 512$. We will see shortly that those considerations give a good balance between system complexity, filter performance and power dissipation. (It should be pointed out that there exists an optimal selection for the filter order vs. the coefficient precision so as to achieve the best FIR filter under the desired specifications.)

III. Processor Architecture

As stated in the previous section, the MSDAP focuses on computing the linear convolution characterized by (1) – (3). At time point *n*, suppose that *N* data samples, *x(n), x(n-1), ...x(n-N+1)*, are stored in a data RAM. Then, the virtual addresses $a_j(l)$, corresponding to $x_j(l)$ with $1 \le l \le r_j$ and $1 \le j \le 16$, can be created based on the filter POT coefficients when they are available. (The physical address of the data RAM is the sum of the virtual address and a base address, and the base address is incremented when a new input sample replaces the oldest sample in the data RAM.) It should also be mentioned that two auxiliary bits will be attached to each $a_j(l)$. The first bit indicates whether *+x(n-k)* or *–x(n-k)* will be processed, while the second determines one-bit shift operations. In summary, the MSDAP should have two data RAMs with a total capacity of $2 \times 256 \times 16$ bits. (We have assumed that two-channel audio signals are sampled with 16-bit precision.) A program RAM consisting of $a_j(l)$, $1 \le l \le r_j$ and $1 \le j \le 16$, thus has a capacity of 512×10 bits: two bits of each word indicate the sign and shift information and the rest eight bits represent a virtual address of $x_j(l)$. Therefore, only about *1.8 kbytes* data and program memory circuits will be required in our MSDAP. (If only one program RAM is used, this implies that two FIR filters perform the identical transfer functions.)

Based on the above analysis, the block diagram of the MSDAP is readily depicted in Fig. 1 where only two accumulators and two one-bit shifters are needed in addition to memories and control circuitry.

Let us now turn to the determination of the system clock rates. It is well known that three industry standards exist concerning the sampling rates of digital audio signals: *48 kHz* for studio use, *44.1 KHz* for CD players and *32 kHz* likely for digital audio broadcasting. Thus, the highest sampling rate, namely *48 kHz*, requires that each output sample be processed within *1000/48=20.8μs*. Taking *512* cycles needed for processing each output sample in the MSDAP into account, the shortest cycle time then *20.8/512=40.7 ns*. In other words, the highest clock rate of the MSDAP will be *24.58 MHz*. (*22.58 MHz* and *16.39 MHz* are the system clock rates for sampling rates *44.1 kHz* and *32 kHz* situations, respectively.)

VI. Circuit Design

To aid understanding of the circuit design aspects of the MSDAP, we use a digital filter chip [9] as a reference. This is a multi-rate (decimation or interpolation) digital filter specifically designated for audio oversampling A/D or D/A converters. It has a die area of only 2.37mm^2 in 1.0-μ CMOS technology and with a power dissipation of 18.8 mW from a 5-V supply and 6.5 mW from a 3-V supply. We now describe the circuit design aspects of the MSDAP in some detail.

Static RAM may be used to store data samples and programs to avoid the required refresh procedure for a dynamic RAM.

When two 40-bit accumulators are employed, the computation in the MSDAP is free from errors. Round-off noise, of course, will be introduced if output samples are rounded or truncated to 16-bit precision. Moreover, if ripple-carry adders [10] were used, the critical path would be longer than 40.7 ns. This problem can easily be solved by introducing a pipeline stage in accumulators.

The registers required in the MSDAP may employ NMOS only transmission gates to avoid the need of complementary clocks [9].

We need a control unit that can set up the program from an external memory such as an EPROM to the program RAM. For this purpose we can integrate a small ROM containing a bootstrap program that is often included in a general DSP chip such as Motorola 56000. It is expected that when all processing units and input/output signals are synchronized by an external clock.

Circuitry for test purpose is another significant issue. Also, when *R* is small, i.e., there are a lot of empty cycles, the MSDAP should be shut down to save power.

We expect the MSDAP will have an active area about 8 mm^2 in 1.0-u CMOS technology and its power dissipation is less than 100 mW from a 5-V supply.

V. FIR Filters for Digital Audio Signal Processing

A. Frequency-Selective Digital Filters

Low-pass, high-pass, band-pass and band-stop FIR filters are typical frequency-selective digital filters whose functions are to pass useful signals but attenuate unwanted signals or noise in some specified frequency band(s). One special advantage of using FIR filters is tat they can have exactly linear-phase responses [3].

B. Digital Tone Control

Although there are various methods for the design of digital tone control systems, here we merely describe a tunable digital filter based on a second-order all-pass function

$$A(z) = \frac{r^2 + 2r\cos\theta z^{-1} + z^{-2}}{1 + 2r\cos\theta z^{-1} + r^2 z^{-2}} \text{ [11-13].}$$

A good approximation for the recursive part is

$$\frac{1}{1 + 2r\cos\theta z^{-1} + r^2 z^{-2}} = \prod_{k=0}^{P-1} [1 + 2r^{2^k} \cos 2^k \theta z^{-2^k} + r^{2^{k+1}} z^{-2^{k+1}}] \tag{4}$$

where $r^P \rightarrow 0$. This approximation is easily extended to an anti-causal all-pass function that has been found useful for phase equalization. (Linear-phase FIR filters for tone control are studies in [14-16].)

C. Digital Equalization

The use of digital equalization is audio systems may be one of the most fascinating applications of digital audio signal processing. The principles of digital equalization have been established for the past two decades [17-25], although many more application-oriented algorithms need yet to be investigated. In short, the digital equalization technique attempts to "correct" many kinds of distortion introduced by audio system components such as amplifier, speakers or even the listening room. For perfect equalization, we need to implement an inverse digital filter that is characterized by E(z) = 1/D(z) where D(z) is the distortion function of an audio system and may be obtained by either using an acoustic theory model or acoustic measurement. In practice, we might use an FIR filter to approximate H(z) as much as possible. (As described in Part B, first- or second-order IIR filter sections, either causal or anti-causal, may well be approximated by an FIR filter.)

References

[1] J. Watkinson, The Art of Digital Audio, Boston, MA:Focal Press 1994.

[2] Digital Audio, IEEE ASSP Magazine, Special issue, vol.2, Oct, 1985.

[3] A. V. Oppenheim and R. W. Schafer, Discrete-Time Signal Processing, Englewood Cliffs, NJ:Prentice Hall, 1989.

[4] Y. C. Lim and S. R. Parker, "FIR filter design over a discrete powers-of-two coefficient space," IEEE Trans. Acoust., Speech, Signal Processing, vol.ASSP-31, pp.583-591, June 1983.

[5] H. Samueli, "A simple design of FIR filters with powers-of-two coefficients," IEEE Trans. Circuits Syst., vol.CAS-36, pp.1044-1047, July 1989.

[6] Zhongnong Jiang, "FIR filter design with powers-of-two coefficients," Proc. ICASSP'89 pp.1239-1242.

[7] H. Shaffeu et al., "Improved design procedure for multiplierless FIR digital filters," Electron. Lett., 20th June 1991, vol.27, pp.1142-1144.

[8] D. Li et al.,, "A polynomial-time algorithm for designing digital filters with powers-of-two coefficients," Proc. ISCAS'93, pp.84-87.

[9] B. P. Brandt and B. A. Wooley, "A low-poer, area-efficient digital filter for decimation and interpolation," IEEE J. Solid-State Circuit, vol.SC-29, pp.679-687, June 1994.

[10] N. Weste and K. Eshraghian, Principle of CMOS VLSI Design. Addison-Wesley 1992.

[11] P. A. Regalia and S. K. Mitra, "Tunable digital frequency response equalization filters," IEEE Trans. Accoust., Speech, Signal Processing, vol.ASSP-35, pp.118-120, Jan. 1987.

[12] E. H. J. Persoon and C. J. B. Vandenbulcke, "Digital audio examples of the application of the ASP integrated signal processor," Philips Tech. Rev., vol.42, pp.201-216, Apr. 1986.

[13] J. K. J. Van Ginderduen et al., "A high quality digital audio filter set designed by silicon complier CATHEDRAL-I," IEEE J. Solid-State Circuit, vol.SC-21, pp.1067-1075, Dec. 1986.

[14] Y. C. Lim, "Linear-phase digital audio tone control," J. Audio Engl. Soc., vol.35, pp.38-40, Jan./Feb. 1987.

[15] R. H. Yang, "Linear-phase digital audio tone control using dual RRS structure," Electron. Lett., vol.25, pp.360-362, March 1989.

[16] Y. Lian and Y. Lim, "Linear-phase digital audio tone control using multiplication-free FI filter," J. Audio Eng. Soc., vol.41, pp.791-794, Oct. 1993.

[17] D. Preis, "Linear distortion," J. Audio Eng. Soc., vol.24, pp.346-367, June 1976.

[18] J. A. Jensen, "A new principle for an all-digital preamplifier and equalizer," J. Audio Eng. Soc., vol.35, pp.994-1003, Dec. 1987.

[19] R. A. Greiner and M. Schoessow, "Electronic equalization of closed-box loudspeakers," J. Audio Eng. Soc., vol.31, pp.125-134, March 1983.

[20] D. Preis, "Audio signal processing with transversal filters," in Proc. ICASSP'79, pp.310-313.

[21] C. Bunks and D. Preis, "Minimax time-domain deconvolution for transversal filter equalizers," Proc. ICASSP'80, pp.943-946.

[22] D. Preis, "Phase distortion and phase equalization in audio signal processing – A tutorial review," J. Audio Eng. Soc., vol.30, pp.774-794, Nov. 1982.

[23] J. A. Henriquez et al., "A phase-linear audio equalizer: design and implementation," J. Audio Eng. Soc., vol.38, pp.653-666, Sept. 1990.

[24] R. Greenfield and M. O. Hawksford, "Efficient filter design for loudspeaker equalization," J. Audio Eng. Soc., vol.39, pp.739-751, Oct. 1991.

[25] J. N. Mourjopoulos, "Digital equalization of room acoustics," J. Audio Eng. Soc., vol.42, pp.884-900, Nov. 1994.