PARALLEL PROGRAMMABLE VIDEO CO-PROCESSOR DESIGN

An-Yeu Wu, K. J. Ray Liu, Arun Raghupathy, Shang-Chieh Liu

Electrical Engineering Department and Institute for Systems Research
University of Maryland
College Park, MD 20742, USA

ABSTRACT

Modern video applications call for computationally intensive data processing at very high data rate. In order to meet the high-performance/low-cost constraints, the state-of-the-art video processor should be a programmable design which performs various tasks in video applications without sacrificing the computational power and the manufacturing cost in exchange for such flexibility. In this paper, we present a programmable video co-processor design that is capable of performing FIR/IIR filtering, subband filtering, and most discrete orthogonal transforms (DT), for the host processor in video applications. The computational speed of this co-processor is as fast as that of ASIC designs which are optimized for individual specific applications. We also show that the system can be easily reconfigured to perform multirate FIR/IIR/DT operations at negligible hardware overhead. Hence, we can either double the processing speed on the fly based on the same processing elements, or apply this feature to the low-power implementation of this co-processor.

1. INTRODUCTION

Modern communication services such as high-definition TV (HDTV), video-on-demand services (VOD), and PC-based multimedia applications call for computationally intensive data processing at millions of additions/multiplications per second. As a consequence, the traditional general-purpose programmable digital signal processing (DSP) processor is not sufficient under such a speed constraint. On the other hand, dedicated VLSI application-specific integrated circuit (ASIC) designs, which are optimally designed for given functions, can handle the demanding computational tasks. However, since a collection of ASIC chips are required to perform different tasks in video applications, both manufacturing cost and system area will be increased. Therefore, we are motivated to design a programmable video processor with the flexibility of a general DSP processor while meeting the stringent speed requirement like the ASIC designs.

The major goal of this paper is to integrate the FIR/IIR lattice architecture, Quadrature Mirror Filter (QMF) lattice structure [1, chap. 6], discrete transform (DT) architecture [2] into one universal programmable architecture. It will serve as a co-processor in the video system to perform those front-end computationally intensive functions for the host processor. The resulting system consists of an array of identical programmable modules and one programmable interconnection network. By simply setting parameters of the programmable modules and reconfiguring the interconnection network, we can perform FIR/IIR/QMF/DT based on the same architecture. Furthermore, since the inherent properties of each programmed function such as pipelinability and parallelism are fully exploited, the processing speed of the proposed co-processor can be as fast as dedicated ASIC designs.

The second goal of this paper is to improve the speed performance of the system using the multirate approach [1]. In video signal processing, the major constraint is the processing speed of the video processor. Such speed constraint will result in the use of expensive high-speed multiplier/adder circuits or full-custom designs. Thus, the cost and the design cycle will increase drastically. Recently, it has been shown that the multirate approach is a powerful tool for high-speed/low-power DSP applications [3]. We will show that we can map the multirate FIR/IIR/DT operations onto our video co-processor design. As a result, we can double the speed performance of the co-processor on the fly by simply reconfiguring the programmable modules and interconnection network. This feature of multirate processing can also be applied to the low-power implementation of this co-processor [3].

2. UNIFIED MODULE DESIGN

2.1. Basic Module in FIR/QMF

The finite impulse response (FIR) filter is widely used in DSP applications. In addition to the multiply-accumulate (MAC) implementation of the filtering operation, an alternate realization of the FIR filter is the lattice structure [4, chap.10]. It consists of N basic lattice sections that are connected in a cascade form. Each lattice module has one rotation-based circuit plus two scaling multipliers. In general, the rotation circuit can be implemented by the CORDIC processor in *hyperbolic mode* [5].

The Quadrature Mirror Filter (QMF) plays a key role in image compression and subband coding. Recently, the two-channel paraunitary QMF lattice was proposed [1, chap. 6]. It has been shown that every two-channel (real-coefficient, FIR) paraunitary QMF bank can be represented using the QMF lattice. The QMF lattice is very similar to the FIR lattice except that the inputs of the lattice become the decimated sequences of the input signal. Besides, the two scaling multiplers are set to one and the CORDIC processor works in the *circular mode* in the QMF lattice.

2.2. Basic Module in IIR

Next, we want to consider the basic module for infinite impulse response (IIR) filtering. Due to the opposite data flow and the irregularity in the ARMR IIR lattice structure (see [4, chap. 10]), the traditional IIR lattice module cannot be directly applied to our programmable module

This work was supported in part by the ONR grant N00014-93-10566 and the NSF NYI Award MIP9457397.

0-8186-7310-9/95 $4.00

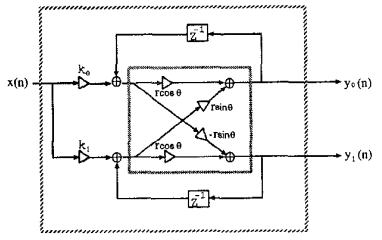

Figure 1: Second-order IIR lattice architecture.

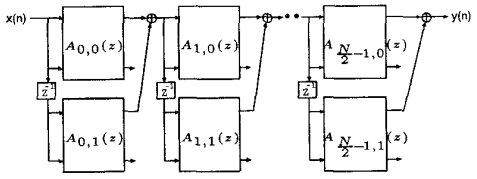

Figure 2: IIR (ARMA) structure based on the second-order IIR lattice module.

design. Therefore, we try to find an IIR module that has similar data paths as in the FIR/QMF lattice while retaining the property of modularity. In our design, we will use the lattice structure in Fig.1 as a basic module to realize a second-order IIR. The transfer functions of the two outputs are given by

$$\tilde{H}_0(z) = \frac{Y_0(z)}{X(z)} = \frac{r(k_0\cos\theta + k_1\sin\theta) - r^2k_0z^{-1}}{1 - 2r\cos\theta z^{-1} + r^2z^{-2}}, \quad (1)$$

$$\tilde{H}_1(z) = \frac{Y_1(z)}{X(z)} = \frac{r(k_1\cos\theta - k_0\sin\theta) - r^2k_1z^{-1}}{1 - 2r\cos\theta z^{-1} + r^2z^{-2}}. \quad (2)$$

Now given an even-order real-coefficient IIR (ARMA) filter $H(z)$, we can first rewrite it in cascade form:

$$H(z) = K\prod_{i=0}^{N/2-1} H_i(z), \quad (3)$$

where K is a scaling constant and each subfilter $H_i(z)$ is of the form

$$H_i(z) = \frac{1}{1 + a_iz^{-1} + b_iz^{-2}} + z^{-1}\frac{c_i + d_iz^{-1}}{1 + a_iz^{-1} + b_iz^{-2}} = A_{i,0}(z) + z^{-1}A_{i,1}(z). \quad (4)$$

Comparing (1) and (2) with (4), we see that $A_{i,0}(z)$ and $A_{i,1}(z)$ can be realized by either $\tilde{H}_0(z)$ or $\tilde{H}_1(z)$ with appropriate settings of the parameters k_0, k_1, r, and θ.

Now based on (3) and (4), we can realize $H(z)$ using the signal diagram depicted in Fig.2, in which each stage performs the filtering for $H_i(z)$. Also, $A_{i,0}$, $A_{i,1}$, $i = 0, 1, \ldots, N/2 - 1$, are realized by the second-order IIR module in Fig.1.

2.3. Basic Module in Discrete Transforms

Performing the discrete transforms (DT) based on the rotation circuit has been extensively studied in the literature [5][2]. Recently a unified IIR-based architecture for the DT was proposed [3], in which it has been shown that most orthogonal transforms can be represented as linear combinations of two functions defined by

$$X_C(k) \triangleq \beta\sum_{n=0}^{L-1}\cos[(2n+1)\omega_k + \eta_k]x(n), \quad (5)$$

$$X_S(k) \triangleq \beta\sum_{n=0}^{L-1}\sin[(2n+1)\omega_k + \eta_k]x(n), \quad (6)$$

for $k = 0, 1, \ldots, N-1$. The only differences among various DT are the setting of the parameters in (5) and (6) and how to combine $X_C(k)$ and $X_S(k)$ together (defined as *combination function*). The parameter settings and the corresponding combination functions for most DT are listed in Table 1, where $C(k)$ and $S(k)$ are the scaling constants used in the DCT/DST and MLT, respectively.

Based on the results in [2] and [3], we can have a unified rotation-based architecture for the DT. Fig.3(a) shows the module for the dual generation of $X_C(k)$ and $X_S(k)$, where the scaling multipliers are given by

$$\mathbf{f}_k = \begin{bmatrix} f_{0,k} \\ f_{1,k} \end{bmatrix} = \begin{bmatrix} \beta\cos((2L+1)\omega_k + \eta_k) \\ \beta\sin((2L+1)\omega_k + \eta_k) \end{bmatrix}. \quad (7)$$

This module will be used as a basic building block to implement the unified DT architecture. Fig.3(b) illustrates the overall time-recursive MLT architecture for the case $N = 8$. It consists of two parts: One is the *module array* which computes $X_C(k)$ and $X_S(k)$, $k = 0, 1, \ldots, N-1$, in parallel. The other is the *interconnection network* which selects and combines the array outputs to generate the desired DT coefficients according to the combination function defined in Table 1.

2.4. Unified Module Design for FIR/QMF/IIR/DT

From the basic computational modules used in the FIR/QMF /IIR/DT structures, we observe that those architectures share a common computational module with only some minor differences in the data paths, the module parameters (multiplier coefficients and rotation angle), and the way the modules are connected. We thus can integrate those basic modules into one universal programmable module as shown in Fig.4. The switch set $\mathbf{S} \triangleq [s_0s_1s_2s_3s_4s_5]$ controls the data paths inside the module. The switch pair s_0 and s_1 select the input from either in_i or in_i'. Switch s_2 and s_3 decide if the delay element is used or not. The last switch pair is s_4 and s_5. They are used to control the two feedback paths in the module. For the FIR/QMF filtering, the parameters $f_{0,i}$, $f_{1,i}$ and θ_i as well as the setting of $\mathbf{S}$ can be easily computed using the formula in the references [4][1, chap. 6]. For the IIR/DT operations, we can apply the results in Sec. 2.2 and 2.3 to compute the necessary parameters.

3. VIDEO CO-PROCESSOR DESIGN

Based on the programmable module, we are ready to design the video co-processor that is capable of performing parallel implementation for any function in the FIR/QMF/IIR/DT. Fig.5 shows the video co-processor architecture under the FIR mode. It consists of two parts: One is the *programmable module array* with P identical programmable modules. The

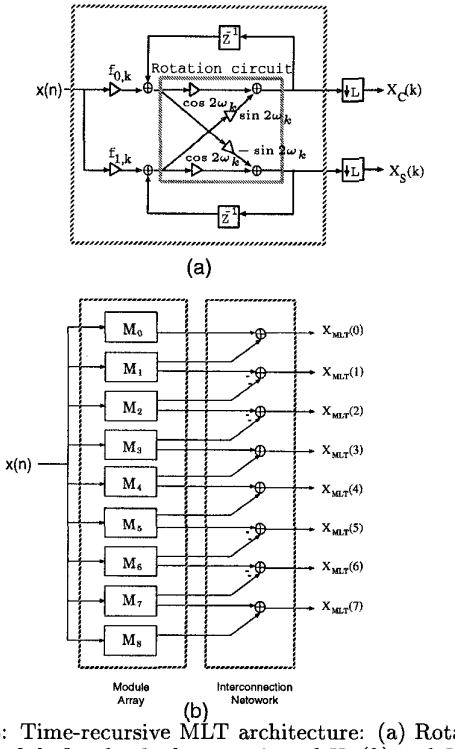

Figure 3: Time-recursive MLT architecture: (a) Rotation-based module for the dual generation of $X_C(k)$ and $X_S(k)$. (b) Overall transform architecture.

other is the *programmable interconnection network* which connects those programmable modules according to the data paths. In the FIR/QMF/IIR, the data are processed in a serial-input-serial-output way. Hence, the programmable modules need to be cascaded for those operations. For example, the FIR modules can be connected by setting the interconnection network as shown in Fig.5. Similarly, we can map the IIR structure in Fig. 2 and the DT structure in Fig. 3 to the co-processor architecture by appropriately setting the interconnection network.

The operation of the co-processor is as follows: In the *initialization mode*, the host processor computes all the necessary parameters $f_{0,i}$, $f_{1,i}$, r_i, θ_i according to the function type (FIR/QMF/IIR/DT). Once the video co-processor is initialized, it enters the *execution mode*. In the applications of FIR/IIR/QMF, the host processor continuously feeds the data sequence into the co-processor. After the first output data is ready, the processor can collect the filtering outputs in a fully pipelined way. In the block DT application, the block input data is fed into the co-processor serially. After the last datum enters the unified module array, the evaluations of $X_C(k)$ and $X_S(k)$ in Fig.3(a) are completed. Then the interconnection network will combine the module outputs according to the combination functions defined in Table 1, and the DT coefficients can be obtained in parallel at the outputs of the network.

4. SPEED-UP: THE MULTIRATE APPROACH

In video signal processing, the fundamental bottleneck is the processing speed of the processing elements. Recently,

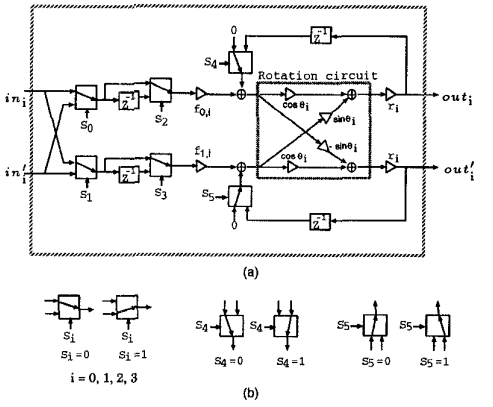

Figure 4: (a) Programmable module to perform the FIR/QMF/IIR/DT. (b) Switches used in the module.

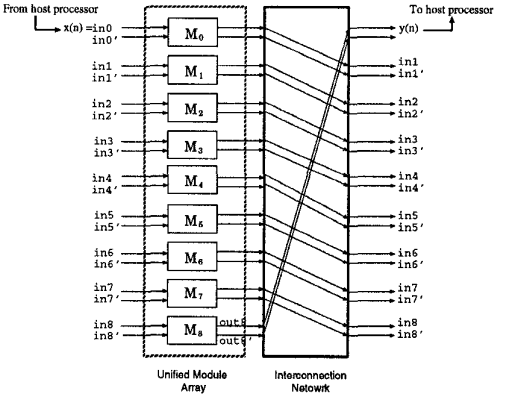

Figure 5: Overall architecture for FIR filtering.

the multirate FIR/IIR filtering architecture has been proposed [6]. Fig.6 shows the multirate architecture to realize a transfer function $H(z)$, where $H_0(z)$, $H_1(z)$ are the *polyphase components* [1] of $H(z)$, and $\hat{H}(z) \triangleq H_1(z) + H_2(z)$. As we can see, the multirate architecture can be readily applied to very high-speed filtering operation. For example, it can process data at 100 MHz rate while only 50 MHz processing elements are required. Thus, the aforementioned speed constraint can be resolved at the algorithmic/architectural level.

4.1. Multirate FIR Architecture

Given an Nth-order FIR filter, $H(z)$, we first find the three $(\frac{N}{2})$th-order FIR subfilters $H_0(z^2)$, $H_1(z^2)$, and $\hat{H}(z^2)$. Then we implement each subfilter using the FIR lattice structure as depicted in Fig.7(a), where $\tilde{R}_i$, $\hat{R}_i$, and $\bar{R}_i$ correspond to the ith basic modules used in $H_0(z)$, $\hat{H}(z)$, and $H_1(z)$, respectively. Next, we can map Fig.7(a) to our video co-processor with the mapping

$$\tilde{R}_i \longrightarrow M_{3i}, \quad \hat{R}_i \longrightarrow M_{3i+1}, \quad \bar{R}_i \longrightarrow M_{3i+2}, \qquad (8)$$

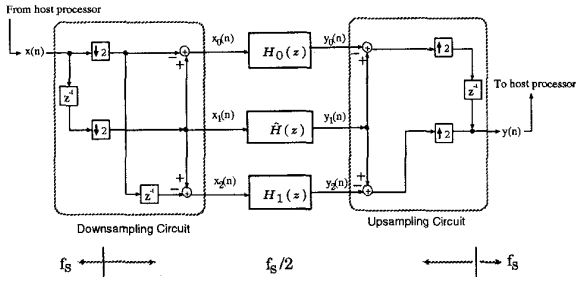

Figure 6: Multirate filtering architecture in [6], where f_s is the data sample rate.

for $i = 0, 1, \ldots, N/2 - 1$. Besides, the interconnection network is set according to data paths in Fig.7(a). Fig.7(b) illustrates the realization of a $6th$-order FIR by the use of 9 programmable modules.

Similarly, the multirate IIR filtering operation can be mapped to our co-processor design by finding the polyphase components of the IIR function and realizing each subfilter with the IIR lattice structure in Sec. 2.2. The mapping of the multirate DT architecture is also achievable based on the results presented in [3].

5. CONCLUSIONS

In this paper, we have presented a prgrammable video co-processor design that is capable of handling most of computationally intensive tasks in video applications. The co-processor can also perform adaptive filtering by mapping the QR-decomposition based recursive least-square lattice (QRD-LSL) algorithm [7] to our programmable architecture. One future application is to incorporate the DCT-based motion estimation (ME) scheme [8] into this design. The flexibility as well as the real-time processing speed of the proposed co-processor design makes it very attractive for video-rate applications.

References

[1] P. P. Vaidyanathan, *Multirate systems and filter banks.* Englewood Cliffs, NJ: Prentice Hall, 1993.

[2] E. Frantzeskakis, J. S. Baras, and K. J. R. Liu, "Time-recursive computation and real-time parallel architectures, Part I: Framework," Tech. Rep. TR 93-17r1, Institute for Systems Research, University of Maryland, 1993.

[3] A.-Y. Wu and K. J. R. Liu, "Algorithm-based low-power transform coding architectures," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, (Detroit), pp. 3267–3270, May 1995.

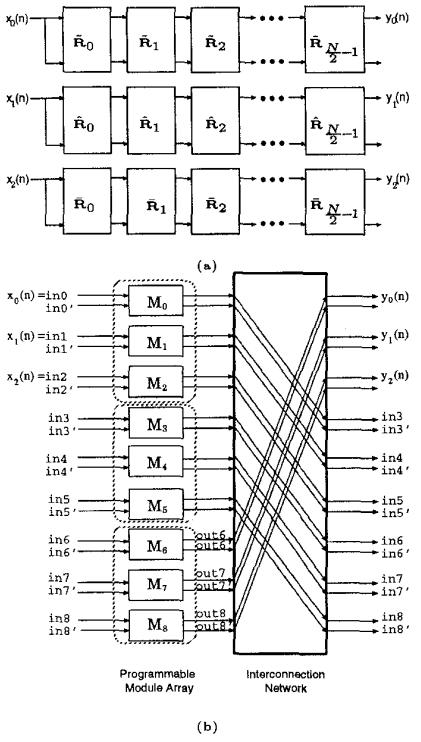

Figure 7: (a) Multirate FIR based on the lattice structure. (b) Mapping part (a) to the co-processor architecture.

[4] L. B. Jackson, *Digital Filters and Signal Processing.* Kluwer Academic Publishers, 2nd ed., 1989.

[5] Y. H. Hu, "CORDIC-based VLSI architectures for digital signal processing," *IEEE Signal Processing Magazine*, vol. 9, pp. 16–35, July 1992.

[6] Z. J. Mou and D. Duhamel, "Fast FIR filtering: Algorithm and implementations," *Signal Processing*, vol. 13, pp. 377–384, 1987.

[7] I. K. Proudler, J. G. McWhirter, and T. J. Shepherd, "Computationally efficient QR decomposition approach to least squares adaptive filtering," *IEE Proceedings-F*, vol. 138, pp. 341–353, Aug. 1991.

[8] U. V. Koc and K. J. R. Liu, "Discrete-cosine/sine-transform based motion estimation," in *Proceedings of IEEE International Conference on Image Processing (ICIP-94)*, vol. 3, (Austin, Texas), pp. 771–775, Nov. 1994.

	Data Length L	β_1	ω_k	θ_k	Combination Function
DCT	N	$C(k)$	$\frac{k\pi}{2N}$	0	$X_{\mathrm{DCT}}(k) = X_{\mathrm{C}}(k)$
IDCT	N	$C(1)$	$\frac{\pi}{2N}(k+\frac{1}{2})$	$-\omega_k$	$X_{\mathrm{IDCT}}(k) = X_{\mathrm{C}}(k) + (C(0) - C(1))\varepsilon(0)$
DST-IV	N	$C(1)$	$\frac{\pi}{2N}(k+\frac{1}{2})$	0	$X_{\mathrm{DST}}(k) = X_{\mathrm{S}}(k)$
IDST-IV	N	$C(1)$	$\frac{\pi}{2N}(k+\frac{1}{2})$	0	$X_{\mathrm{IDST}}(k) = X_{\mathrm{S}}(k)$
MLT	$2N$	$\frac{1}{\sqrt{2N}}$	$\frac{k\pi}{2N}$	$\frac{\pi}{2}(k+\frac{1}{2})$	$X_{\mathrm{MLT}}(k) = -S(k)[\,X_C(k+1) + X_S(k)\,]$
ELT	$4N$	$\frac{1}{2\sqrt{2N}}$	$\frac{\pi}{2N}(k+\frac{1}{2})$	$\frac{\pi}{2}(k+\frac{1}{2})$	$X_{\mathrm{ELT}}(k) = -X_S(k+1) + \sqrt{2}X_C(k) + X_S(k-1)$
DFT	N	$\frac{1}{\sqrt{N}}$	$-\frac{k\pi}{N}$	$-\omega_k$	$\mathrm{Re}\{X_{\mathrm{DFT}}(k)\} = X_{\mathrm{C}}(k)$, $\mathrm{Im}\{X_{\mathrm{DFT}}(k)\} = X_{\mathrm{S}}(k)$.
DHT	N	$\frac{1}{\sqrt{N}}$	$-\frac{k\pi}{N}$	$-\omega_k$	$X_{\mathrm{DHT}}(k) = X_{\mathrm{C}}(k) + X_{\mathrm{S}}(k)$.

Table 1: Parameter settings for the unified DT architecture.