

Finally, we note that neither the MYW method nor the ML method is workable for this noisy AR(4) model described by (32), and the computational burden with the ML method is over 100 times that of the ILSD method.

VI. CONCLUDING REMARKS

The work presented in this paper provides a better way for implementing the ILSNP method, thus greatly improving its numerical efficiency. The developed ILSD method is consistently convergent. Since it involves fewer computations per iteration than the ILSNP method, the ILSD method is much more suitable for real-time applications. The good performances of the ILSD method have been illustrated by the experimental results. The important algorithmic advantages warrant that the developed ILSD method is the attractive alternative in noisy AR modeling by means of the ILS type methods. Future work will consider deriving an on-line version of the ILSD method presented in this paper in terms of the recursive LS cost function associated with the forgetting factor λ ($0 < \lambda \leq 1$). Such a new version of the algorithm could be of great interest to nonstationary signals (e.g. speech signals) analyzed by AR models in the presence of noise.

ACKNOWLEDGMENT

The author would like to thank the Associate Editor, Professor P. S. R. Diniz, and the three anonymous reviewers for their valuable comments and suggestions that have greatly helped to improve the manuscript.

REFERENCES

- [1] D. Aboutajdine, A. Adib, and A. Meziane, "Fast adaptive algorithms for AR parameter estimation using higher order statistics," *IEEE Trans. Signal Processing*, vol. 44, pp. 1998–2009, 1996.
- [2] C. Y. Chi, J. L. Hwang, and C. F. Rau, "A new cumulant based parameter estimation method for noncausal autoregressive systems," *IEEE Trans. Signal Processing*, vol. 42, pp. 2524–2527, 1994.
- [3] M. H. A. Davis and R. B. Vinter, *Stochastic Modeling and Control*, London, UK: Chapman and Hall, 1985.
- [4] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1996.
- [5] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [6] S. M. Kay, *Modern Spectral Estimation*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [7] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [8] B. D. Kovacevic, M. M. Milosavljevic, and M. D. Veinovic, "Robust recursive AR speech analysis," *Signal Processing*, vol. 44, pp. 125–138, 1995.
- [9] J. S. Lim and A. V. Oppenheim, "All-pole modeling of degraded speech," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, pp. 197–210, 1978.
- [10] A. Nehorai and P. Stoica, "Adaptive algorithms for constrained ARMA signals in the presence of noise," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 1282–1291, 1988.
- [11] H. Sakai and M. Arase, "Recursive parameter estimation of an autoregressive process disturbed by white noise," *Int. J. Control*, vol. 30, pp. 949–966, 1979.
- [12] M. D. Srinath, P. K. Rajasekaran, and R. Viswanathan, *Introduction to Statistical Signal Processing with Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [13] A. Swami and J. M. Mendel, "Identifiability of the AR parameters of an ARMA process using cumulants," *IEEE Trans. Automat. Contr.*, vol. 37, pp. 268–273, 1992.

- [14] H. Tong, "Autoregressive model fitting with noisy data by Akaike's information criterion," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 476–480, 1975.
- [15] W. X. Zheng, "Identification of autoregressive signals observed in noise," in *Proc. 1993 American Control Conf.*, vol. 2, San Francisco, CA, pp. 1229–1230.
- [16] W. X. Zheng, "An efficient algorithm for parameter estimation of noisy AR processes," in *Proc. 30th IEEE Int. Symp. Circuits and Systems (ISCAS'97)*, vol. 4, Hong Kong, pp. 2509–2512.
- [17] W. X. Zheng, "A least-squares based method for autoregressive signals in the presence of noise," *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 81–85, Jan. 1999.

A Novel Architecture of Inverse Quantization and Multichannel Processing for MPEG-2 Audio Decoding

Tsung-Han Tsai and Liang-Gee Chen

Abstract—An MPEG-2 audio decoding processor core is described with a focus on inverse quantization (IQ) and multichannel processing (MC) of Layer I and II decoding. A novel architecture that we propose can perform IQ at a high throughput. In addition, different types of dematrixing modes for MC process in the MPEG-2 standard can also be performed. The processor core is implemented and controlled with a dedicated hardware approach instead of the traditional programmable techniques. Moreover, the design has the advantages of simplicity and low cost while meeting the high-efficiency requirements with a fixed throughput.

Index Terms—Inverse quantization, MPEG-2, multichannel processing, synthesis subband.

I. INTRODUCTION

Digital audio coding has recently become an important technique in the audio industry. One of these audio-coding techniques, the ISO MPEG-2 audio standard, has developed a world-wide standard audio-coding algorithm which aims to support all the normative features listed in the MPEG-1 audio and provide extension capabilities of multichannel and multilingual audio on an extension of standard to lower sampling frequencies and lower bit rates [1]–[3]. The elementary concept behind MPEG is based on the multirate subband-based coding techniques [4]. Basically, the most computational load highly depends on the realization of the synthesis subband in the decoder, and can be reduced using the regular fast algorithm [5], [6]. As for the other important computational parts of the decoding, inverse quantization (IQ) and multichannel processing (MC) are seldom mentioned.

Some comparisons for MPEG video and audio algorithms have been described [7], [8]. Based on these references, we also present the comparison focused on IQ and MC in Table I. It shows that the IQ and MC modules make use of few computation power of the entire decoding process. However, complex controls and a relatively small amount of data reuse will be induced and complicate the design of the hardware.

Manuscript received February 26, 1998; revised September 1999. This paper was recommended by Associate Editor J. M. Dias.

T.-H. Tsai is with the Department of Electronic Engineering, Fu-Jen University, Taiwan, R.O.C.

L.-G. Chen is with the Department of Electrical Engineering, National Taiwan University, Taiwan, R.O.C.

Publisher Item Identifier S 1057-7130(00)00583-8.

TABLE I
COMPARISONS BETWEEN IQ/MC AND SYNTHESIS SUBBAND MODULES IN MPEG-2 DECODER

| Module | Computational Complexity (MOPS) | Preferred Type of Parallelism | Arithmetic : Data Transfer (data reuse quotient) | Control |
|-------------------|---------------------------------|--|--|-------------------------|
| IQ , MC | $\simeq 10\%$ | Sequential, weak instruction parallelism | 1 : 1 ~ 1 : 3 | Complex, need decisions |
| Synthesis Subband | $\simeq 90\%$ | Data parallelism | 64 : 1 ~ 32 : 1 | Simple, regular |

Thus, it is unsuitable when applying a fast algorithm based on the characteristics of complex control and irregular data flow. These inherent disadvantages can be overcome using a hardware-oriented implementation strategy for the decoding algorithm.

Referring to the architecture design, different aspects of the architecture must be utilized in the MPEG-2 audio decoder. These designs are basically applied either as general purpose DSP-based techniques such as stand-alone chip sets [9], or proposed as architecture dedicated to the individual IQ and MC function blocks [10]. Whether the architecture is DSP-based or is dedicated architecture, most processors implement the MPEG-2 decoding by programming. However, these processors suffer from considerable overheads of computation and control. In addition, some papers have only focused on the synthesis subband with a dedicated cost-effective architecture [11], [12]. In that case, they must perform the IQ and MC in the host platform, such as PC. These designs also increase the complexity in the interface and communication between the dedicated chip sets and the host.

In the brief, we propose novel architecture of IQ and MC which support the Layer I and II for the MPEG-2 audio decoding processor core. It is built using design concept different from previous works. By use of the dedicated hardware approach (ASIC), a more efficient VLSI solution can be provided than can be provided by commercial programmable and complex individual dedicated designs. Moreover, the design has the advantages of simplicity and low cost while meeting the high efficiency requirements with a fixed throughput. The processor can easily and efficiently cooperate with other dedicated synthesis subband chips.

II. IQ AND MC FOR MPEG-2 DECODING

In MPEG-2 audio decoding, emphasis in the new activity is on backward compatibility and multichannel processing [2]. With backward compatibility, it is possible to produce a multichannel audio at any time without making the two-channel MPEG-1 obsolete. In multichannel processing, five audio channels $\{L, R, C, LS, RS\}$ are mapped to five transmission channels $\{T0, T1, T2, T3, T4\}$. The $T0$ and $T1$ are equal to the MPEG-1 compatible channels $L0$ and $R0$ respectively, and the $T2$ to $T4$ channels are extended channels.

A. Inverse Quantization

IQ reconstructs the transmission channel T_x with the reconstructed sample Q'_x . It is divided into two major functional blocks: *reconstruction* (ReC) and *rescaling* (ReS). In the ReC procedure, Q'_x is applied by a linear formula to obtain the requantized sample Q_x . In the ReS procedure, Q_x is scaled by a SF_x to obtain T_x .

B. Multichannel Processing

MC reconstructs the multi-audio channels with the transmission channels. It is composed of the four functional blocks: *Dynamic crosstalk* (DC), *Dynamic transmission channel switching* (DTCS),

TABLE II
REQUIRED OPERATIONS IN IQ AND MC DECODING

| Module | Function | Arithmetic Calculations | Data Transfers | Classification |
|--------|----------|-------------------------------|----------------------------|----------------|
| IQ | ReC | $Q_x = C \cdot (Q'_x + D)$ | — | Phase I |
| | ReS | $T_x = SF_x \cdot Q_x$ | — | Phase II |
| MC | DC | — | $SF_y \leftarrow SF_x$ | Phase II |
| | DTCS | — | $A_x^w \leftarrow T_x$ | Phase II |
| | DeM | $A_x^w = M_i - A_y^w - A_z^w$ | $A_y^w \leftarrow A_x^w$ † | Phase II |
| | DeN | $A_x = A_x^w \cdot N$ | — | Phase III |

† C, D are bit-allocated coefficients; N is the combined coefficient of weighting and denormalization factors. The subscript x indicates the channel.

‡ Active "if" mode $tc_allocation = 1, 2, 6, 7$ in five-channel configuration.

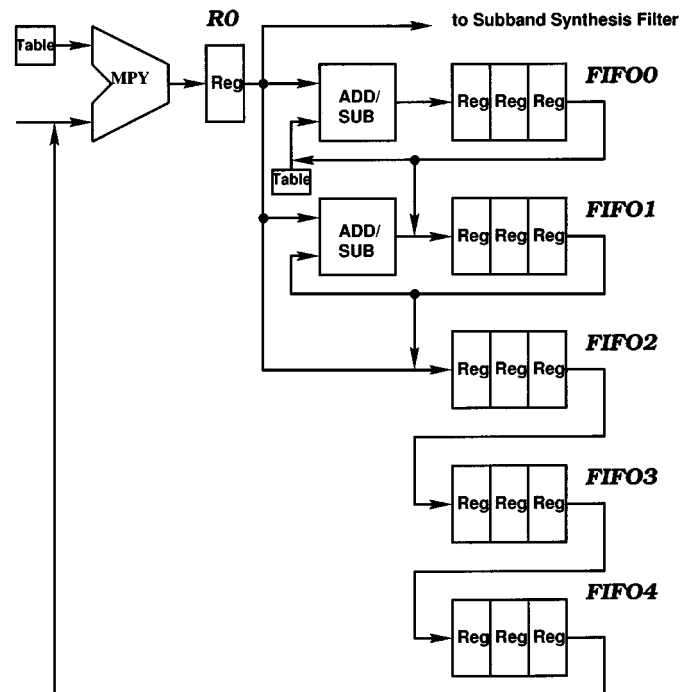
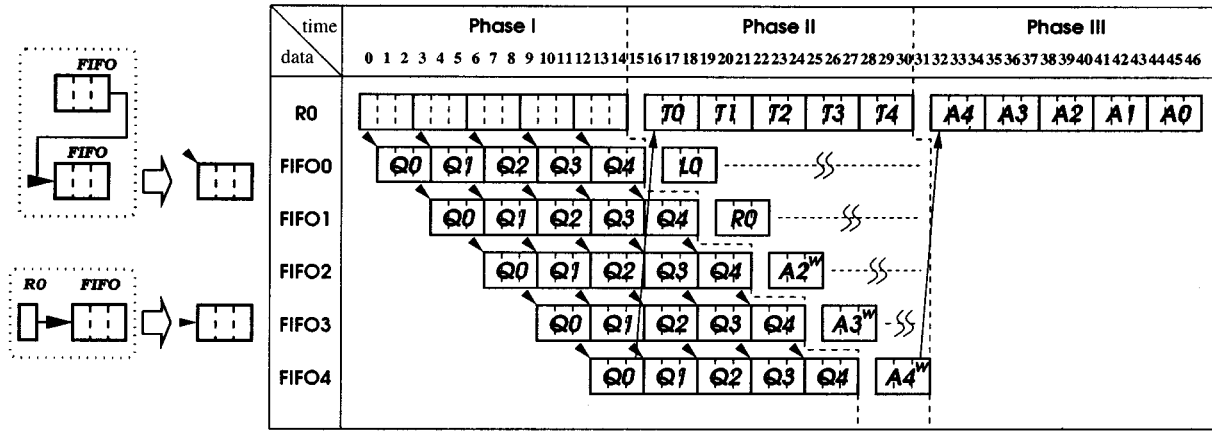
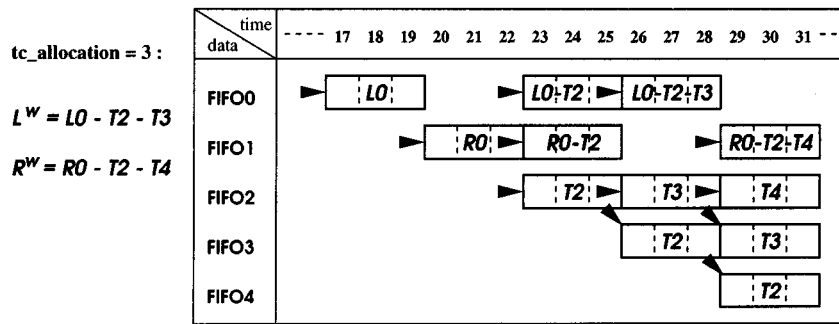


Fig. 1. Efficient architecture for processor core.

Dematrixing (DeM), and *Denormalization* (DeN). DC is a method of multichannel data reduction which allows for dynamic deletion of sample bits in specified subband of specified transmission channels. DTCS is a method of multichannel data reduction performed by allocating the most orthogonal signal components to the transmission channels. Eight allocation modes are decided by the parameter $tc_allocation$ in five-channel configuration. DeM recomputes the two coded channels to reconstruct the weight channels



(a)



(b)

 Fig. 2. Register allocation table and the related data flow for (a) overall IQ and MC decoding and (b) the flexible data allocation for DeM mode $tc_allocation = 3$.

$\{L^w, R^w, C^w, LS^w, RS^w\}$. In DeN procedure, the weight channels should be multiplied with a weighting factor and a denormalization factor to reconstruct the five audio channels.

III. IMPLEMENTATION STRATEGY AND ANALYSIS

Table II illustrates the functions needed in IQ and MC modules. A_x is the signal from one of the five audio channels, and A_x^w is the weight signal from one of the five weight audio channels. M_i is one of the signals referred to as the main signal of $L0, R0$. Firstly, it can be seen that the only arithmetic operations performed in IQ and MC are multiplication and addition. Each multiplication-and-addition pair and the related functions can be classified into each associated phase. Three phases, **Phase I** to **Phase III**, are performed to cover the whole functions in IQ and MC. Since the arithmetic operation of ReC is the addition be performed before the multiplication, it will be inconsistent with the other two phases and prohibit a regular data flow of pipelined processing. By changing the order of ReC operation

$$\begin{aligned}
 Q_x &= C \cdot Q'_x + C \cdot D \\
 &= C \cdot Q'_x + D'.
 \end{aligned} \quad (1)$$

Equation (1) allows the multiplication be performed before the addition. Based on this reordering modification, the order of operation in the three phases will be consistent and implemented using a simpler controller. Secondly, to overcome the irregular data flow and complex control in multichannel processing, distributed-registers architecture will be proposed and illustrated in the next section.

IV. ARCHITECTURE DESIGN

A proposed processor is shown in Fig. 1. Three registers as a group form a FIFO, and there are five such FIFO's to support multichannel decoding. This configuration supports up to five audio channels, including the 2-channel decoding for MPEG-1. In addition, only one multiplier and two adders/subtractors are used in a two-stage pipelined structure to achieve high performance with fully hardware utilization. The quantized, rescaled and denormalized coefficients are stored in the ROM tables. In order to achieve high quality audio decoding, the 24-bit word length of all the computation units is applied in this architecture and provides 144-dB dynamic range.

Two different data flows through the FIFO's are defined in Fig. 2(a). First, the proposed hardware unit is utilized solely to compute Q_x for all five channels in **Phase I**. This phase takes 15 clock cycles to complete. Once this is finished, the unit is reconfigured to perform the **Phase II** task, followed by the **Phase III** task. This architecture has advantages of an efficient control strategy, and a smooth data flow without the input data memory and the associated data address generator. Additionally, **Phase II** performs a flexible data allocation for the various DeM modes. For example, mode $tc_allocation = 3$ is depicted in Fig. 2(b). It shows that each channel data can be fed into any of the associated FIFO's easily and simultaneously. However, some modes have the properties that channel data are correlated with each other. One example, mode $tc_allocation = 2$, is

$$L^w = L0 - C^w - T3 \quad (2)$$

$$C^w = R0 - T2 - T4 \quad (3)$$

TABLE III
COMPARISONS BETWEEN THE PROPOSED AND THE OTHER ARCHITECTURES

| Specification | Proposed | DSP-based [9] | Function-specific [10] |
|--------------------------|-----------------------|-----------------|--|
| Architecture | a dedicated processor | a VLIW DSP core | a microprocessor (IQ) + 3 dedicated processors (MC) |
| Data Memory | register file | RAM | RAM + register file |
| Data Address Generator | no | yes | yes |
| Program Memory | no | yes | yes |
| MC optimization | yes | no | no |
| Clock cycles/15 samples | 47 | 77 ~ 81 † | 50† |
| Numbers of data accesses | 25 | 53 | 53 |

† Each of the operation, read, write, shift, add, or multiply is estimated as one clock cycle.

TABLE IV
ESTIMATED TRANSISTOR COUNT FOR PROPOSED PROCESSOR

| Functional Unit | Transistor Count |
|---------------------|------------------|
| Multiplier | 1 |
| Adder/Subtractor | 2 |
| DFF | 16 |
| MUX | 4 |
| Total Transistors † | 185280 |

† Referred to [13].

to avoid the data conflict between L^w and C^w , we substitute (3) into (2) and proceed as follows:

$$\begin{aligned} L^w &= L0 - (R0 - T2 - T4) - T3 \\ &= L0 - R0 + T2 - T3 + T4. \end{aligned} \quad (4)$$

Equation (4) implies that the required channel data used in DeM can be further decomposed into an independent sequence. Based on the distributed registers architecture and the decomposition method, any mode can be performed in order without resulting in a conflict in data.

The comparisons between the proposed and the other architectures are shown in Table III. Although some high-performance DSP structures, such as VLIW and SIMD, can perform the decoding, they also have disadvantages of a complex circuit design and no optimization in multichannel decoding. For the previous fully function-specific design, each of the MC functions is implemented in its dedicated processor, and IQ is not combined with MC, will decrease the hardware utilization and lead to cost increasing. Since the distributed register file and pipelined architecture is applied in the proposed design, the numbers of data access and clock cycles for 15 samples in Layer II application are less than others. In addition to the advantages of not requiring program ROM and low-cost design approach, the proposed architecture achieves a good synchronization with a fixed throughput, which is difficult to be realized in other processors based on straightforward implementation in MC processing. The estimated transistor count is shown in Table IV. In addition to regularity and modularity, the processor core has a small area based on the applied technology.

V. CONCLUSION

Because of the complex control and irregular data flow, IQ and MC have been traditionally implemented by software. This brief describes a novel architecture for the MPEG-2 audio decoding processor core, stressing mainly the IQ and MC. With the direct hardware implementation approach, no program ROM is needed in order to reduce the overheads of the control and chip area. Additionally, any type of dematrixing modes can be implemented efficiently with a flexible data allocation. Based on the two-stage pipelined and distributed registers architecture, the high-efficiency requirement with a fixed throughput is achieved.

REFERENCES

- [1] *Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mb/s*, MPEG-1, ISO CD 11172-3, Nov. 1991.
- [2] *Coding of moving pictures and associated audio*, MPEG-2, ISO CD 13818-3, Nov. 1994.
- [3] K. Brandenburg and M. Bosi, "Overview of MPEG audio: Current and future standards for low-bit-rate audio coding," *J. Audio Eng. Soc.*, vol. 45, pp. 4–21, Jan./Feb. 1997.
- [4] D. Scitzer, T. Sporer, K. Brandenburg, H. Gerhauser, B. Grill, and J. Herre, "Digital coding of high quality audio," *CompEuro*, pp. 148–154, 1991.
- [5] M. Iwadare, A. Sugiyama, and F. Hazu, "A 128 kb/s hi-fi audio CODEC based on adaptive transform coding with adaptive block size MDCT," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 138–144, Jan. 1992.
- [6] P. Noll, "Digital audio coding for visual communications," *Proc. IEEE*, vol. 83, pp. 925–943, June 1995.
- [7] J. Kneip, et al., "The MPEG-4 video coding standard—a VLSI point of view," in *Proc. IEEE Workshop Signal Processing Systems*, 1998, pp. 43–52.
- [8] T. H. Tsai, L. G. Chen, and Y. C. Liu, "A Novel MPEG-2 audio decoder with efficient data arrangement and memory configuration," *IEEE Trans. Consumer Electron.*, vol. 43, pp. 598–604, Aug. 1997.
- [9] L. Bergher, et al., "DOLBY AC-3 and MPEG-2 audio decoder IC with 6-channels output," *IEEE Trans. Consumer Electron.*, vol. 43, pp. 567–573, Aug. 1997.
- [10] S. C. Han and S. K. Yoo, "An ASIC implementation of the MPEG-2 audio decoder," *IEEE Trans. Consumer Electron.*, vol. 42, pp. 540–545, Aug. 1996.
- [11] Y. Jhung and S. Park, "Architecture of dual mode audio filter for AC-3 and MPEG," *IEEE Int. Conf. Consumer Electron.*, June 1997.
- [12] W. Lau, "A common transform engine for MPEG & AC-3 audio decoder," *IEEE Int. Conf. Consumer Electron.*, June 1997.
- [13] W. Lau, Ed., *Compass 0.6-Micro, 5-Volt, High-Performance*. Reading, MA: Addison-Wesley, 1993, pp. 513–589.