Cost-effective Low-Power Architectures of Video Coding Systems

Jie Chen* and K. J. Ray Liu

*Bell Laboratories, Lucent Technologies
Murray Hill, NJ 07974, USA

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

Abstract

A new low-power design technique, multirate, has been used along with other methods such as look-ahead, pipelining in designing the cost-effective low-power architectures of video coding system. We demonstrate both low-power and high-speed can be accomplished at algorithm/architecture level. Based on the calculation and simulation results, the design can achieve significant power saving in the range of 60% – 80% or speedup factor of two at the needs of users.

I. Introduction

In recent years, the need for personal mobile communications – "anytime, anywhere" access to multimedia and communication services – has become increasingly clear. Digital cellular telephony, such as the U.S. third generation code-division multiple access PCS and the European GSM systems, has seen rapid acceptance and growth in the marketplace. Due to the limited power-supply capability of current battery technology, low-power design to prolong the operating time of those mobile handsets becomes vital to success. However, the development of low-power video coding systems is still in its infancy. In this paper, we focus on the combined low-power design of DCT and motion estimation units, which serve as the computing engine in video coding system. The current low-power video coding systems are achieved at *device/process level* such as low-power video coder design in [1], which use 0.5 μm VLSI fabrication technology. Nevertheless, the cost of those approaches is the most expensive among all low-power techniques, namely, from system, algorithm/architecture down to circuit/logic, device/process level design [2].

In this paper, we extend our video coding architectures in [3], [4] for low-lower applications. Our low-power design is achieved at the *algorithm/architecture* level, which provides the most leveraged way to achieve low-power consumption when both effectiveness and cost are taken into consideration. In principle, The algorithm/architecture low-power design is achieved by reformulating the algorithms and mapping them to efficient low-power VLSI architectures to compensate for the speed loss caused by lowered supply voltage. As a result, we trade silicon area for power consumption under the current technology, without invoking new expensive devices and advanced VLSI fabrication technology. Compared with other low-power techniques, our algorithmic/architectural approach is one of the most cost-effective ways to save power.

Unlike the conventional video coder design in MPEG standards, the motion estimation in our low-power video coder is achieved in DCT instead of spatial domain. As a result, we can naturally accommodate both DCT and motion estimation processors into one processing unit, which saves silicon area drastically and also enables the combined low-power design. In addition, all advantages mentioned in [3], [4] i.e. high throughput, numerical stability, multiplier-free, modular and solely local connected properties are also inherited in our low-power design. Furthermore, it is important to recognize that our low-power design can smartly conquer both low-power and high-speed requirements, which are often considered to be the problems of opposite natures, at the needs of users. Based on the calculation and simulation results, the proposed design can be readily applied to high-speed video communication with the speedup factor of two under normal supply voltage ($5V$). Or, the same design

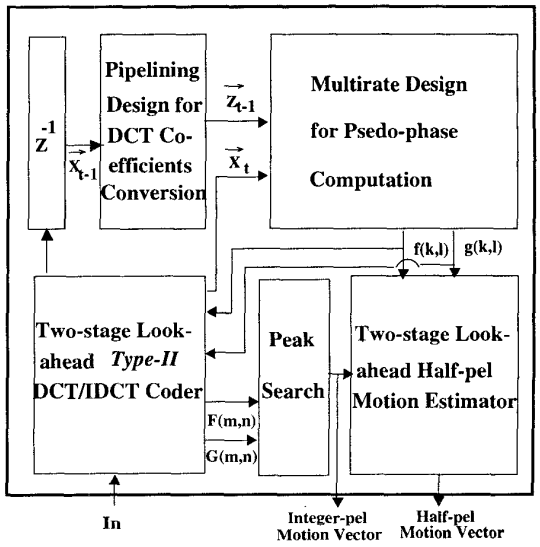

Figure 1: Low-power architecture for video coding.

can be operated at two-time slower operating frequency under lowered supply voltage ($3.08V$) while retaining

*This work is supported in part by the NSF NYI award MIP9457397 and the ONR grant N00014-93-10566.

the original data throughput rate. This feature enables us to achieve significant power saving in the range of 60% – 80% without sacrificing system performance (refer to the detail later).

The proposed low-power design has fully pipelined parallel architecture as shown in Fig. 1. A new low-power design technique, multirate [5], [6], has been used along with other methods such as look-ahead, pipelining in our design to achieve low-power/high-speed performance. In what follows, we explain the architectures of each building block in Fig. 1 in detail. Then we present the simulation results in Section 3 to demonstrate the performance of our design. Finally the paper is concluded in Section 4.

II. Low-power/High-speed Architectures

A. Two-stage Look-ahead Type-II DCT/IDCT Coder:

Unlike the conventional DCT coder design using matrix factorization, we adopt the time-recursive DCT [7], [8] which is able to simultaneously generate *type-II* DCT and DST coefficients – X_t^c and X_t^s needed by pseudo-phase computation module which we will discuss later. Due to the inherent time-recursive characteristic, we use *look-ahead* method to reduce the power consumption. In principle, the speed-up provided by look-ahead compensates the speed loss caused by reduced supply voltage at the cost of increasing hardware complexity.

The two-stage look-ahead time-recursive updating of DCT and DST coefficients is given by:

$$\begin{cases} \begin{bmatrix} X_{t+2}^c(k) \\ X_{t+2}^s(k) \end{bmatrix} = \begin{bmatrix} cos\frac{2k\pi}{N} & sin\frac{2k\pi}{N} \\ -sin\frac{2k\pi}{N} & cos\frac{2k\pi}{N} \end{bmatrix} \left[\begin{bmatrix} X_t^c(k) \\ X_t^s(k) \end{bmatrix} + \begin{bmatrix} \overline{X}_t^c(k) \\ \overline{X}_t^s(k) \end{bmatrix} \right] \\ \begin{bmatrix} \overline{X}_t^c(k) \\ \overline{X}_t^s(k) \end{bmatrix} = D(k) \begin{bmatrix} cos\frac{k\pi}{2N} & cos\frac{3k\pi}{2N} \\ sin\frac{k\pi}{2N} & sin\frac{3k\pi}{2N} \end{bmatrix} \begin{bmatrix} -x(t) + (-1)^k x(t+N) \\ -x(t+1) + (-1)^k x(t+N+1) \end{bmatrix} \end{cases} \quad (1)$$

where t is time index, $X_t^c(k)$ and $X_t^s(k)$ are defined as:

$$X_t^c(k) = D(k) \sum_{n=t}^{t+N-1} x(n) \cos[\frac{k\pi}{N}[(n-t)+\frac{1}{2}]]; \; k \in \{0, \ldots, N-1\},$$

$$X_t^s(k) = D(k) \sum_{n=t}^{t+N-1} x(n) \sin[\frac{k\pi}{N}[(n-t)+\frac{1}{2}]]; \; k \in \{1, \ldots, N\},$$

$$\text{where } D(k) = \begin{cases} \frac{2}{\sqrt{2N}}, & \text{for } k = 0 \text{ or } N, \\ \frac{2}{N}, & \text{otherwise.} \end{cases}$$

Both two-stage look-ahead DCT and its inverse counterpart, inverse IDCT, undergo the similar computing procedure in (1) except for minor differences in the data inputs and rotation angles. In order to save chip area, we interleave them into a unified structure which contains 3 CORDIC (COordinate Rotation Digital Computer [9]) processors as shown in Fig. 2.

Clearly, the look-ahead system can be clocked at two-time faster rate than the original system for high-speed application. Or, by reducing the supply voltage from $5V$ to $3.08V$, we increase the propagation delay of look-ahead system until it equals to that of the original system. In other words, we achieve low-power design while still keep the same system throughput. The ratio of the power consumption of 2-stage look-ahead design, $P_{2-stage}$, to the power of original design, P_{orig}, can be written as:

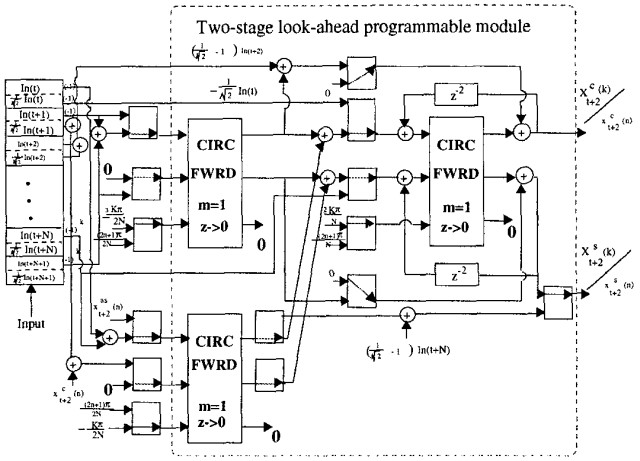

Figure 2: Two-stage look-ahead type-II DCT/IDCT coder, here the switch setting is for DCT ($x_{t+2}^c(k), x_{t+2}^s(k)$) and the complementary setting is for inverse IDCT ($x_{t+2}^c(n), x_{t+2}^s(n)$).

$$\frac{P_{2-stage}}{P_{orig}} = \frac{C_{2-stage}}{C_{orig}} (\frac{3.08V}{5V})^2 \frac{\frac{1}{2}f}{f} = 0.28,$$

where f is the original operating frequency, $C_{2-stage}$ and C_{orig} represent the total switching capacitances of look-ahead and original implementation. Provided that the capacitances due to CORDICs are dominant in the circuit and are roughly proportional to the number of CORDICs, we get $C_{2-stage} \approx \frac{3}{2} C_{orig}$ because the low-power design requires 3 CORDICs while the original design only needs 2 CORDICs. Overall the look-ahead design results in 72% power saving without sacrificing the system throughput at the expense of an increasing output latency and 50% hardware overhead. In essence, we trade silicon area for low-power consumption.

Because two-dimensional DCT can be decomposed into 2-stage pipelined one-dimensional computation, we therefore adopt the same approach as in [3] to extend our low-power DCT design to two-dimensional design.

B. Pipelining Design for DCT Coefficients Conversion:

Note that the *type-I* DCT coefficients, $\vec{\mathbf{Z}}_{t-1}$, required by the pseudo-phase computation in Fig. 1 can actually be obtained by the plane rotation of its counterpart *type-II* DCT coefficients, $\vec{\mathbf{X}}_{t-1}$ [3]. To achieve high-speed design, we can insert flip-flop (**D**) across the feed-forward cut-set as shown in Fig. 3. Now the pipelining design can run two-time faster than the original design because the critical path has been halved. Or, we can reduce the power supply voltage from $5V$ to $3.08V$ while still maintain the same system throughput. The ratio of the power consumption of pipelining design, P_{pipe}, to the power of original design, $P_{rotator}$, is given by:

$$\frac{P_{pipe}}{P_{rotator}} = \frac{4}{4} (\frac{3.08V}{5V})^2 \frac{\frac{1}{2}f}{f} = 0.19,$$

which leads to 81% power saving at the cost of increased system latency.

C. Multirate Design for Pseudo-phase Computation:

Traditionally, multirate technique is widely used in sub-band coding [5]. Our interest, on the other hand, is to

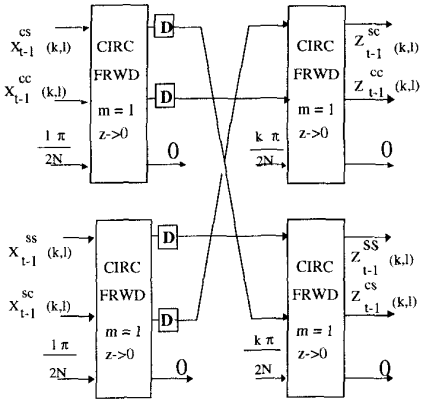

Figure 3: Pipelining design for *type-II* to *type-I* DCT coefficients conversion

apply this technique to compensate the speed loss due to lowered supply voltage or to simply speed-up the design under normal condition. For the pseudo-phase computation module in the original design as shown in Fig. 4 (a), the processing rate of the operator has to be as fast as the input data rate. By employing *multirate* low-power design, the pseudo-phases are computed from the reformulated circuit using the decimated sequences ($M = 2$) as shown in Fig. 4 (b). Now the multirate design oper-

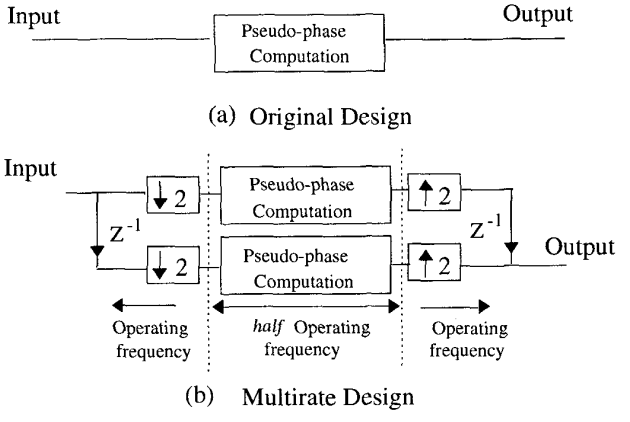

Figure 4: Multirate design for pseudo-phase computation.

ates at two different rates. Because the operating frequency of pseudo-phase computation is reduced to half of the input data rate while the overall throughput rate is still remained the same, the speed penalty therefore is compensated at the architectural level. As stated previously, we can keep the overall throughput rate while reduce the power supply voltage from $5V$ to $3.08V$. The multirate design needs 20 CORDICs, which is twice the number of CORDICs in original design. The ratio of the power consumption of multirate design, P_{multi}, to the power of original design, P_{phase}, can be obtained as:

$$\frac{P_{multi}}{P_{phase}} = \frac{20}{10}\left(\frac{3.08V}{5V}\right)^2 \frac{\frac{1}{2}f}{f} = 0.38.$$

Overall we can achieve the power saving of 62% or the speed-up factor of two at the cost of doubled hardware complexity.

D. Two-stage Look-ahead Half-pel Motion Estimator:

To obtain motion at half-pel accuracy, we first compute the integer-pel motion vectors (m, n) then use "half-pel motion estimator" in Fig. 1 to compute the half-pel motion vectors. With such an approach, we can avoid conventional interpolation procedure [10] because we can determine the half-pel motion vectors by only considering the nine positions $u \in \{m - 0.5, m, m + 0.5\}$ and $v \in \{n - 0.5, n, n + 0.5\}$ surrounding integer-pel motion vectors (m, n) as illustrated at the upper right corner of Fig. 5. In other words, the peak position among nine $\overline{DCS}(u, v)$ and $\overline{DSC}(u, v)$,

$$\overline{DCS}(u,v) = \sum_{k=0}^{N-1}\sum_{l=1}^{N} C(k)C(l)f(k,l)\cos\frac{k\pi}{N}(u+\frac{1}{2})\sin\frac{l\pi}{N}(v+\frac{1}{2})$$

$$\overline{DSC}(u,v) = \sum_{k=1}^{N}\sum_{l=0}^{N-1} C(k)C(l)g(k,l)\sin\frac{k\pi}{N}(u+\frac{1}{2})\cos\frac{l\pi}{N}(v+\frac{1}{2}),$$

indicates the half-pel motion. In order to figure out both $\overline{DCS}(u, v)$ and $\overline{DSC}(u, v)$, we can decompose the computations into hierarchic one-dimensional calculations of *type-II* inverse IDCT as illustrated in Fig. 5 (Here we use $\overline{DSC}(u, v)$ as an example). By taking a close look at

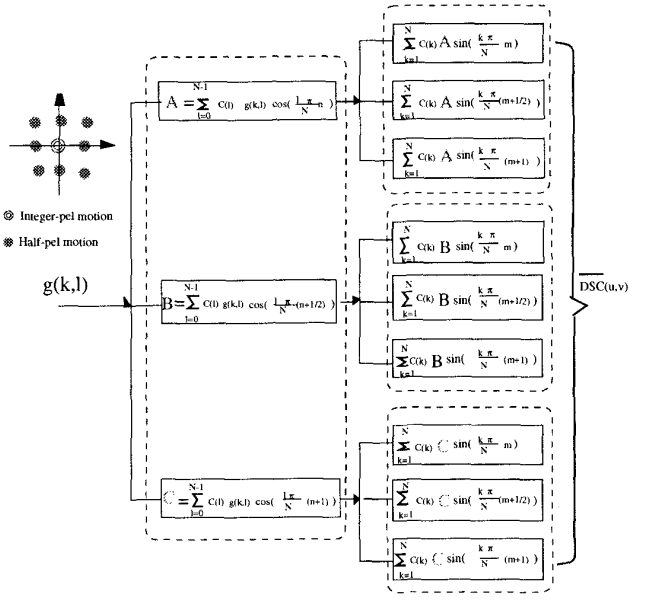

Figure 5: Schematic diagram of decomposing $\overline{DSC}(u, v)$.

those computations in Fig. 5, we find that they are similar to that of two-dimensional inverse IDCT calculation. Therefore, it enables us to adopt look-ahead approach mentioned previously to achieve the low-power design. Based on the same argument as in two-stage look-ahead DCT coder design, we can achieve 72% power saving at the expense of an increasing output latency and 50% hardware overhead.

III. Simulation Results and Hardware Cost

Cadence circuit design tool, $Verilog^{TM}$, has been used in simulating the performance of our design. We use

"Miss America" in QCIF format as the test sequence. The original frame 91 and reconstructed frame using our

(a) original frame

(b) reconstructed frame

Figure 6: "Miss America" frame 91

proposed low-power design are shown in Fig. 6 (a) and (b), respectively. The simulation results demonstrate that our low-power design can achieve comparable video quality as the original one.

To compare the speed of original [4] and our low-power/high-speed design for each module in Fig. 1, we use the synthesis tool to check the static timing of each block. The resulted speed-up factors are listed in Table 1. Based on the simulation results, we observe that our low-power/high-speed design can operate at about two-time faster clock rate than the original design, which is corresponding to our previous derivations.

Unit	Type-II DCT/ IDCT Coder	DCT Coeff-icients Conversion	Pseudo-phase Computation	Half-pel Estimator
Speedup factor	1.87	1.95	1.81	1.85

Table 1: Simulation result of speed-up

To process the video sequence, each frame is divided into non-overlapped macroblock which contains $N \times N$ pixels as the input to our low-power/high-speed design. The hardware cost and data throughput rate of the building blocks in Fig. 1 to process each macroblock are summarized in Table 2. Overall our design is flexible and scalable because it uses $33N$ CORDIC processors,

Component	CORDICs	Adders	Registers	Through-put
Type-II DCT/IDCT	9N	27N+12	$N+6N^2$	$O(N)$
Type Conversion	4N	0	0	$O(N)$
Pseudo Phase	20N	2N	0	$O(N)$
Peak Searching	0	0	$2N^2$	$O(N)$
Half-pel Motion Estimator	3N+9	7N+33	$3N+N^2$	$O(N)$
Total	36N+9	36N+45	$4N+9N^2$	$O(N)$

Table 2: Hardware cost and data throughput rate

$29N + 12$ adders to achieve integer-pel accuracy and requires additional $3N + 9$ CORDICs, $7N + 33$ adders to achieve half-pel accuracy.

IV. Conclusion

Anticipating the future trend of running video applications on the portable personal devices, we propose cost-effective low-power/high-speed architectures for video coding system. Unlike the existing low-power video codec design using the costly $0.5\mu m$ fabrication technology, our low-power/high-speed design is achieved at the algorithmic/architectural levels. Basically, we only trade more silicon area or system latency for low-power consumption or high-speed performance under current technology, without invoking dedicated circuit design, new expensive devices and advanced VLSI fabrication technology. Compared with other approaches, our algorithmic/architectural low-power approach is one of the most economic ways to save power. Techniques such as look-ahead, multirate, pipelining have been used in our design. Based on the simulation results, our low-power/high-speed design can achieve comparable performance as the original system at the speed-up factor of two (equivalent to the power saving in the range of $60\% - 80\%$).

References

[1] K. Hasegawa, K. Ohara, and *et al.*, "Low-power video encoder/decoder chip set for digital VCR's", *IEEE J. Solid-State Circuits*, vol. 31, no. 11, pp. 1780–1788, Nov. 1996.

[2] Anantha P. Chandrakasan and Robert W. Brodersen, *Low Power Digital CMOS Design*, Kluwer Academic Publishers, 1995.

[3] J. Chen and K. J. R. Liu, "A complete pipelined parallel CORDIC architecture for motion estimation", *IEEE Trans. Circuits Syst. II*, vol. 45, no. 5, pp. 653–660, May 1998.

[4] Jie Chen and K. J. R. Liu, "A fully pipelined parallel CORDIC architecture for half-pel motion estimation", in *Proc. IEEE Int. Conf. on Image Processing*, Santa Barbara, CA, Oct. 1997, vol. 2, pp. 574–577, Also submitted to *IEEE Trans. Circuits and Systems for Video Technology.*

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

[6] K. J. Ray Liu, A.-Y. Wu, A. Raghupathy, and J. Chen, "Algorithm-based low-power and high-performance multimeida signal processing", *Proceedings of the IEEE*, vol. 86, no. 6, pp. 1155–1202, June 1998.

[7] K. J. R. Liu and C. T. Chiu, "Unified parallel lattice structures for time-recursive Discrete Cosine/Sine/Hartley transforms", *IEEE Trans. Signal Processing*, vol. 30, no. 3, pp. 1357–1377, Mar. 1993.

[8] C. T. Chiu and K. J. R. Liu, "Real-time parallel and fully pipelined two-dimensional DCT lattice structures with applications to HDTV systems", *IEEE Trans. Circuits and Systems for Video Technology*, vol. 2, no. 1, pp. 25–37, Mar. 1992.

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

[10] U. V. Koc and K. J. R. Liu, "Interpolation-free subpixel motion estimation techniques in dct domain", *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, no. 4, pp. 460–487, Aug. 1998.