ALGORITHM-BASED LOW-POWER TRANSFORM CODING ARCHITECTURES

An-Yeu Wu and K. J. Ray Liu

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

ABSTRACT
In most low-power VLSI designs, the supply voltage is usually reduced to lower the total power consumption. However, the device speed will be degraded as the supply voltage goes down. In this paper, we propose new algorithmic-level techniques for compensating the increased delays based on the multirate approach. We will show how to compute most of the discrete sinusoidal transforms through the decimated low-speed sequences with reasonable linear hardware overhead. For the case the decimation factor equal to two, the overall power consumption can be reduced to about one-third of the original design. The resulting multirate low-power architectures are regular, modular, and free of global communications. Such properties are very suitable for VLSI implementations. The proposed architectures can also be applied to very high-speed block transforms where only low-speed operators are required.

1. INTRODUCTION
Recent developments in personal communications services (PCS) have made it possible to integrate voice, image and cellular phone networks in a personal communicator. Due to the limited power-supply capacity of current battery technology, the power constraint becomes an important consideration in the design of PCS devices. It has been shown that a reduction of the supply voltage is the leveraged decision to lower the power consumption. However, a speed penalty is suffered for the devices (operators) as the supply voltage goes down [1]. In order to meet the low-power/high-throughput constraint, the key issue is to "compensate" the increased delay so that the device can be operated at the slowest possible speed while maintaining the same data sample rate. In [1], the techniques of "parallel processing" and "pipelining" were suggested to compensate the speed penalty, in which a simple comparator circuit was used to demonstrate how parallel independent processing of the data can achieve good compensation at the architectural level. In most digital signal processing (DSP) applications, the problems encountered are much more complex. It is almost impossible to directly decompose the problems into parallel independent tasks. Therefore, the properties of the DSP algorithms should be fully exploited in order to develop those compensation techniques to compensate the loss of performance under the low-power operation. We call such an approach the algorithm-based low-power design.

In this paper, we will show how to design algorithm-based low-power transform coding architectures using the multirate approach. To motivate the idea, let us consider the discrete cosine transform (DCT) architecture in Fig.1. For most of the existing serial-input-parallel-output (SIPO) DCT algorithms and architectures [2][3], the processing rate must be as fast as the input data rate (Fig.1(a)). In our low-power design, the DCT is computed from the reformulated circuit using the decimated sequences (Fig.1(b)). It is now a multirate system that operates at two different sample rates. Since the operating speed of the processing elements is reduced to half of the original data rate, while the data throughput rate is still maintained, the speed penalty is compensated at the architectural level. Using the CMOS power dissipation model [1], we can predict that the overall power consumption for the multirate design can be reduced to about one-third of the original system. Therefore, the downsampling scheme provides a direct and efficient way for the low-power design at the algorithm/architectural level.

2. LOW-POWER DESIGN OF THE DCT
The 1-D DCT of a series of input data starting from \( x(t-N+1) \) and ending at \( x(t) \) is defined as

\[
X_{DCT,k}(t) = C(k) \sum_{n=0}^{N-1} \cos\left(\frac{(2n+1)k\pi}{2N}\right) x(t+n-N+1),
\]

for \( k = 0, 1, 2, \ldots, N-1 \), where \( C(k) \) is the scaling factor.

An efficient IIR parallel architecture for the DCT can be derived using the transfer function approach [3]. One disadvantage of the IIR structure is that the operation speed is constrained by the recursive loops. In what follows, we will reformulate the transfer function using the multirate approach so that speed constraint can be alleviated. Splitting the input data sequence into the even and odd sequences, and taking the z-transforms, we obtain

\[
X_{DCT,k}(z) = C(k) \frac{(-1)^k - z^{-N/2}}{1 - 2\cos(\omega_k)z^{-1} + z^{-2}} \times \\
\left( \left[ X_o(z) - X_o(z)z^{-1} \right] \cos(\omega_k) + \left[ X_o(z) - X_o(z)z^{-1} \right] \cos(\omega_k) \right) \]

\[
(2)
\]

Figure 1: (a) Original SIPO DCT circuit. (b) Low-power DCT using the multirate approach.

This work was supported in part by the ONR grant N00014-93-10566 and the NSF NRI Award MIP9457397.
where \( \omega_k \triangleq \frac{\pi k}{N} \), and \( X_k(z) \) and \( X_r(z) \) are the \( z \)-transforms of the decimated inputs. The parallel architecture to realize (2) is depicted in Fig. 2, where \( M \) denotes the decimation factor. Once the last serial input \( x(t) \) is fed into the module, the DCT coefficients can be obtained at the outputs of the modules in parallel.

To achieve downsampling by the factor of four (\( M = 4 \)), we can split the input data sequence into four decimated sequences \( g_i(t,n) \triangleq x(t + (4n + i)N + 1) \), \( i = 0, 1, 2, 3 \). Then \( X_{DCT,k}(z) \) can be written as

\[
X_{DCT,k}(z) = \frac{C(k) \left[ (-1)^k - z^{-N/4} \right]}{1 - 2 \cos 8\omega_k z^{-1} + z^{-2}} \times \\
\left[ G_0(z) - G_3(z) z^{-1} \right] \cos 7\omega_k + \left[ G_1(z) - G_2(z) z^{-1} \right] \cos 5\omega_k + \\
\left[ G_3(z) - G_1(z) z^{-1} \right] \cos 3\omega_k + \left[ G_2(z) - G_0(z) z^{-1} \right] \cos \omega_k
\]

where \( G_i(z) \) is the \( z \)-transform of \( g_i(t,n) \). The corresponding multirate architecture is shown in Fig. 3. From Fig. 2 and Fig. 3, we can see that the multirate DCT architectures retain all the advantages of the original IIR structure in [3] such as modularity, regularity, and local interconnections. These features are particularly preferred for their VLSI implementations.

### 3. LOW-POWER DESIGN OF THE MLT/EILT

Next let us consider the power dissipation of the low-power architectures. The power dissipation in a well-designed digital CMOS circuit can be modeled as [4]

\[
P \approx C_{eff} \cdot V_d^2 \cdot f_{clk},
\]

where \( C_{eff} \) is the effective loading capacity, \( V_d \) is the supply voltage, and \( f_{clk} \) is the operating frequency. Also, the lowest possible supply voltage \( V_d \) can be approximated by [1]

\[
\frac{V_d}{V_r} = M \frac{V_{dd}}{(V_{dd} - V_r)^2},
\]

where \( V_r \) is the threshold voltage of the device.

Assume that \( V_{dd} = 5V \), \( V_r = 0.7V \) in the original system. For the 16-point DCT under normal operation [3], it requires 30 multipliers and 32 adders. For the low-power 16-point DCT with \( M = 2 \), 45 multipliers and 49 adders are required. From (5), it can be shown that \( V_d \) can be as low as 3.1V for the case \( M = 2 \). Provided that the capacitance due to the multipliers is dominant in the circuit and is roughly proportional to the number of multipliers, the power consumption for the low-power design can be estimated as

\[
\frac{45}{36} C_{eff} \frac{3.1V}{5V} \approx 0.29 P_0,
\]

where \( P_0 \) denotes the power consumption of the original system. Similarly, for the case \( M = 4 \), the lowest possible supply voltage can be 2.1V (from (5)) and the total power can be reduced to 0.11\( P_0 \). Therefore, we can achieve low-power consumption at the expense of reasonable complexity overhead.

The MLT [5] operates on segments of data of length \( 2N \), and can be decomposed into

\[
X_{MLT,k}(t) = -S(k) \left[ X_{C,k+1}(t) + X_{S,k}(t) \right]
\]

where \( S(k) = (-1)^{(N-1)/2} \) if \( k \) is odd, and \( S(k) = (-1)^{(N+1)/2} \) if \( k \) is even, and

\[
X_{C,k}(t) = \beta_1 \sum_{n=0}^{2N-1} \cos[(2n+1)\omega_k + \theta_k]x[k+n-2N+1],
\]

\[
X_{S,k}(t) = \beta_1 \sum_{n=0}^{2N-1} \sin[(2n+1)\omega_k + \theta_k]x[k+n-2N+1],
\]

with \( \beta_1 = \frac{\sqrt{2}}{\sqrt{2N}}, \omega_k = \frac{\pi k}{2N}, \) and \( \theta_k = \frac{\pi}{2}(k + \frac{1}{2}) \).

As with the low-power DCT, we can have a low-power MLT architecture if each MLT module can compute \( X_{C,k}(t) \) and \( X_{S,k}(t) \) using the decimated input sequences. The multirate IIR transfer functions for (8) and (9) can be computed as

\[
H_{C,k}(z) = \frac{\beta_1(1 - z^{-2N})}{1 - 2 \cos(4\omega_k) z^{-1} + z^{-2}} \times \\
\left[ \cos(3\omega_k - \theta_k) - \cos(\omega_k + \theta_k) z^{-1} \right] X_k(z) + \\
\left[ \cos(\omega_k - \theta_k) - \cos(3\omega_k + \theta_k) z^{-1} \right] X_k(z),
\]
and

\[ H_{S,k}(z) = -\beta_1 \frac{(1 - z^{-2N})}{1 - 2 \cos(4\omega_k) z^{-1} + z^{-2N}} \times \left( [\sin(3\omega_k - \theta_k) + \sin(\omega_k + \theta_k)] x_n(z) + [\sin(\omega_k - \theta_k) + \sin(3\omega_k + \theta_k)] x_{n-1}(z) \right). \] (11)

The corresponding IIR module for (10) and (11) is shown in Fig.4, where

\[
\begin{align*}
\Gamma_{1,e} &= \beta_1 \cos(3\omega_k - \theta_k), \\
\Gamma_{2,e} &= -\beta_1 \cos(\omega_k + \theta_k), \\
\Gamma_{1,o} &= -\beta_1 \cos(\omega_k - \theta_k), \\
\Gamma_{2,o} &= -\beta_1 \cos(3\omega_k + \theta_k), \\
\Gamma_3 &= -\beta_1 \sin(\omega_k - \theta_k), \\
\Gamma_{4,o} &= -\beta_1 \sin(3\omega_k + \theta_k).
\end{align*}
\] (12)

Through such manipulation, the MLT module can operate at half of the original data rate by doubling the hardware complexity. It will be used as a basic building block to implement MLT according to (7). Fig.5 illustrates the overall time-recursive MLT architecture for the case \( N = 8 \). The architecture consists of two parts: One is the IIR module array which computes \( X_{C,k}(t) \) and \( X_{S,k}(t) \) for different index \( k \) in parallel. The other is the combination circuit which selects and combines the outputs of the IIR array to generate the MLT coefficients. It can be shown that the power consumption for the low-power MLT modules is \( 0.38P_0 \) and \( 0.17P_0 \) for the case \( M = 2 \) and \( M = 4 \), respectively.

Likewise, the ELT in (6) can be represented as

\[ X_{ELT,k}(t) = -\tilde{X}_{S,k}(t) + \sqrt{2} X_{C,k}(t) + \tilde{X}_{S,k-1}(t) \] (13)

where

\[
\begin{align*}
\tilde{X}_{C,k}(t) &= \beta_2 \sum_{n=0}^{4N-1} \cos[(2n+1)\omega_n + \theta_n] x(t + n - 4N + 1), \\
\tilde{X}_{S,k}(t) &= \beta_2 \sum_{n=0}^{4N-1} \sin[(2n+1)\omega_n + \theta_n] x(t + n - 4N + 1),
\end{align*}
\] (14) (15)

with \( \beta_2 \triangleq \frac{1}{2N} \), \( \omega_n \triangleq \frac{2\pi}{2N}(k + \frac{1}{2}) \), and \( \theta_n \triangleq \frac{\pi}{2}(k + \frac{1}{2}) \).

Define the relationships in (7) and (13) as the combination functions. After comparing (7)-(9) with (13)-(15), we see that the MLT and ELT have identical mathematical structures except for the definitions of parameters and the combination functions. Therefore, the MLT architectures in Fig.4 and Fig.5 can be readily applied to the ELT by simply modifying those multiplier coefficients and setting the combination circuit according to (13).

4. UNIFIED LOW-POWER TRANSFORM MODULE DESIGN

From the transform functions described in (7)-(9) and (13)-(15), we observe that the low-power MLT module in Fig.4 can be used to realize most existing discrete sinusoidal transforms by choosing suitable parameter settings and combination functions. For example, the \( X_{C,k}(t) \) in (8) is equivalent to the DCT by setting

\[ \beta_2 = C(k), \quad \omega_k = \frac{k\pi}{2N}, \quad \theta_k = 0. \] (16)

As a result, the multirate MLT module in Fig.4 can perform the DCT at different \( \omega_k \).

The other example is the discrete Fourier transform (DFT) with real-valued inputs. With the setting

\[ \beta_1 = \frac{1}{\sqrt{N}}, \quad \omega_k = \frac{k\pi}{N}, \quad \theta_k = -\omega_k, \] (17)

(8) and (9) become

\[
\begin{align*}
X_{C,k}(t) &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \cos\left(\frac{2\pi}{N}kn\right) x(t + n - N + 1), \\
X_{S,k}(t) &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \sin\left(\frac{2\pi}{N}kn\right) x(t + n - N + 1),
\end{align*}
\] (18) (19)

which are the real part and the imaginary part of the DFT, respectively. The settings for other transforms are summarized in Table 1.

The programmable feature of the unified design is attractive in many applications. Firstly, the unified structure can be implemented as a high-performance programmable co-processor which performs various transforms for the host processor by loading the suitable parameters. Secondly, by hard-wiring the parameters to the preset values according to the transform type, we can perform any one of the transforms using the same architecture. This can significantly reduce the design cycle as well as the manufacturing cost.

6. CONCLUSIONS

In this paper, we presented an algorithm-based low-power design of the transform-coding kernels based on the multirate approach. The proposed low-power transform kernels will be effective for the low-power/high-performance signal processing systems. The other attractive application of our design is in the very high-speed signal processing. For example, if we want to perform DCT for serial data at 200 MHz, we may use the parallel architecture in Fig.3, in which only 50MHzadders and multipliers are required. Therefore, we can perform very high-speed DCT by using low-cost and low-speed processing elements.

REFERENCES


### Table 1: Parameter setting for the unified low-power IIR transform module.

<table>
<thead>
<tr>
<th>Data Length</th>
<th>$\beta_b$</th>
<th>$\gamma_b$</th>
<th>$\delta_b$</th>
<th>Combination Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCT N</td>
<td>$2N$</td>
<td>$2N$</td>
<td>$0$</td>
<td>$X_{\text{DCT}}(t) = X_{\text{C}}(t)$</td>
</tr>
<tr>
<td>IDCT N</td>
<td>$2N$</td>
<td>$2N$</td>
<td>$0$</td>
<td>$X_{\text{IDCT}}(t) = X_{\text{C}}(t)$</td>
</tr>
<tr>
<td>DST N</td>
<td>$2N$</td>
<td>$2N$</td>
<td>$0$</td>
<td>$X_{\text{DST}}(t) = X_{\text{C}}(t)$</td>
</tr>
<tr>
<td>IDST N</td>
<td>$2N$</td>
<td>$2N$</td>
<td>$0$</td>
<td>$X_{\text{IDST}}(t) = X_{\text{C}}(t)$</td>
</tr>
<tr>
<td>MLT 2N</td>
<td>$2N$</td>
<td>$2N$</td>
<td>$0$</td>
<td>$X_{\text{MLT}}(t) = X_{\text{C}}(t)$</td>
</tr>
<tr>
<td>EST 4N</td>
<td>$2N$</td>
<td>$2N$</td>
<td>$0$</td>
<td>$X_{\text{ELT}}(t) = X_{\text{C}}(t)$</td>
</tr>
<tr>
<td>DFT N</td>
<td>$2N$</td>
<td>$2N$</td>
<td>$0$</td>
<td>$X_{\text{DFT}}(t) = X_{\text{C}}(t)$</td>
</tr>
<tr>
<td>DSR T</td>
<td>$2N$</td>
<td>$2N$</td>
<td>$0$</td>
<td>$X_{\text{DSR}}(t) = X_{\text{C}}(t)$</td>
</tr>
</tbody>
</table>

### Table 2: Comparison of hardware cost for the DCT, IDCT, MLT, and ELT with their low-power designs.

<table>
<thead>
<tr>
<th>Normal Operation</th>
<th>Downsampling by 2</th>
<th>Downsampling by 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplier</td>
<td>Adder</td>
<td>Multiplier</td>
</tr>
<tr>
<td>HR DCT 2N / 2</td>
<td>$2N$</td>
<td>$2N$</td>
</tr>
<tr>
<td>HR DCT 2N / 4</td>
<td>$2N$</td>
<td>$2N$</td>
</tr>
<tr>
<td>HR MLT 4N / 2</td>
<td>$4N$</td>
<td>$4N$</td>
</tr>
<tr>
<td>HR ELT 4N / 2</td>
<td>$4N$</td>
<td>$4N$</td>
</tr>
</tbody>
</table>

### Table 3: Comparisons of different DCT architectures, where $f_s$ denotes the data sample rate, $M$ denotes the downsampling factor, and $N$ is the block size.

<table>
<thead>
<tr>
<th>Liu et. al. [8]</th>
<th>Proposed multiscale IIR DCT with $M = 4$</th>
<th>Lee [7]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data processing rate $f_s$</td>
<td>$f_s/M$</td>
<td>$f_s/N$</td>
</tr>
<tr>
<td>No. of Multipliers</td>
<td>$2N / 2$</td>
<td>$(M + 1)N$ (in order)</td>
</tr>
<tr>
<td>No. of Adders</td>
<td>$2N$</td>
<td>$(M + 1)N$ (in order)</td>
</tr>
<tr>
<td>Latency</td>
<td>$N$</td>
<td>$\log_2 N$ (in order)</td>
</tr>
<tr>
<td>Restriction on transform size $N$</td>
<td>$N$</td>
<td>$M, b \in Z^+$</td>
</tr>
<tr>
<td>Requirement for input buffer</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Index mapping</td>
<td>Local</td>
<td>Local</td>
</tr>
<tr>
<td>Communication</td>
<td>SIPO</td>
<td>SIPO</td>
</tr>
<tr>
<td>I/O operation</td>
<td>N/A</td>
<td>Good</td>
</tr>
<tr>
<td>Speed compensation capability</td>
<td>N/A</td>
<td>(at the expense of locally increased hardware overhead)</td>
</tr>
<tr>
<td>Power consumption in routing</td>
<td>Negligible</td>
<td>Negligible</td>
</tr>
<tr>
<td>Application to pruning DCT</td>
<td>Direct</td>
<td>Direct</td>
</tr>
</tbody>
</table>

### Fig.4: Low-power IIR MLT module.

### Fig.5: The time-recursive MLT architecture with $N = 8$. 3270