

# A Low-Power and Low-Complexity DCT/IDCT VLSI Architecture Based On Backward Chebyshev Recursion

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

A low-power parallel VLSI structure for DCT/IDCT is proposed. By treating the transformations as the evaluation of the Chebyshev series, and exploiting the *Backward Chebyshev Recursion (BCR)*, we can reduce the total number of multipliers ( $N+1$  for IDCT,  $2N-2$  for DCT). The property of BCR is also used to compute the DCT/IDCT through the down-sampled even and odd sequences. Since the operation frequency for the down-sampled sequences is two times slower, the speed penalty caused by the low-voltage design can be compensated at the architectural level. The total multipliers required for the low-power design is only  $2N+1$  for IDCT and  $3N-3$  for DCT. Extension to downsampling-by-4 is also achievable at a reasonable increase in hardware complexity.

## 1. Introduction

With recent developments in personal communications services (PCS), it is possible now to integrate voice, image, and cellular phone networks in a personal communicator. Due to the limited power-supply capability of current technology, the power constraint is particularly important to the design of PCS devices. Several techniques to achieve low-power CMOS design were investigated by Chandrakasan *et al.* [1]. A simple model is used to estimate the power dissipation

$$P \approx C_{eff} \cdot V_{dd}^2 \cdot f_{clk}, \quad (1)$$

where  $C_{eff}$  is the effective loading capacity,  $V_{dd}$  is the supply voltage, and  $f_{clk}$  is the operating frequency. The simulation results show that a reduction of the supply voltage is the leveraged decision to lower the power consumption. However, a speed penalty is suffered as  $V_{dd}$  goes down. Among many approaches to "compensate" the increased delay, the architecture approach is the most promising one [1]. By imposing "parallelism" and "pipelinability" on the algorithm, the power consumption can be drastically reduced by a factor of 3-5.

In this paper we propose a parallel DCT/IDCT algorithm and architecture based on the Chebyshev polynomial. The Chebyshev polynomial derivation of DCT/IDCT algorithm was first proposed in [2]. However, the architecture in [2] needs global communication and requires  $O(N \log N)$  multipliers. In our work, we treat the transformations as the evaluation of a Chebyshev series. Then by exploiting the

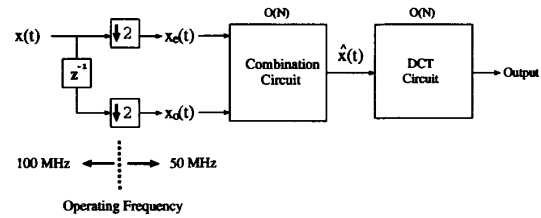


Figure 1: Low-power circuit design using the downsampling scheme.

three-term recurrence property of the Chebyshev orthogonal polynomial, we can significantly reduce the number of multipliers ( $N+1$  for IDCT,  $2N-2$  for DCT). Also, the resulting architecture is modular and regular. Thus the basic requirements for low-power DCT/IDCT design are met [3].

In order to compensate the speed penalty, we employ downsampling to reduce the operating speed. For most of the existing DCT algorithms [4][5][6], the processing rate must be as fast as the input data rate, say 100 MHz. In our low-power design, a downsampling-by-2 scheme is employed (Fig.1). The input sequence  $x(t)$  is first split into two half-rate sequences, and a new sequence  $\hat{x}(t)$  is formed in the combination circuit. If the DCT of  $\hat{x}(t)$  is the same as the DCT of  $x(t)$ , it is clear that now the data can be processed at half of the original data rate, say 50 MHz. Since the operating speed is reduced while the throughput rate is still maintained, the speed penalty is compensated. For example, suppose that the cost of the combination circuit in Fig.1 is about the same as DCT ( $O(N)$ ), the  $C_{eff}$  is approximately doubled due to the hardware overhead. However, since all the operations are at half of the original speed, the lowest possible voltage can be reduced from 5 V to 2.9 V [1]. From the simple model in (1), the overall power consumption is only

$$(2C_{eff}) \left( \frac{2.9V}{5V} \right)^2 \left( \frac{1}{2} f \right) \approx 0.34P. \quad (2)$$

Therefore, downsampling provides a direct and efficient way for the low-power design at the architectural level.

## 2. The Chebyshev Polynomial

Given the frequency mapping

$$t = \cos \omega, \quad (3)$$

\*The work is supported in part by the ONR grant N00014-93-10566 and the NSF grant MIP93-09506.

the  $n$ th order Chebyshev polynomial is defined as

$$T_n(t) = \cos(n\omega) = \cos(n \cos^{-1} t). \quad (4)$$

It satisfies the “three-term recurrence” formula [7, chap 1]

$$T_{n+1}(t) = 2tT_n(t) - T_{n-1}(t), \quad (5)$$

with the initial condition  $T_0(\hat{t}) = 1$ ,  $T_1(t) = t$ . Now consider the following Chebyshev series

$$Y_c(t) = \frac{1}{2}a_0 + \sum_{k=1}^{N-1} a_k \cos(k\omega) = \frac{1}{2}a_0 + \sum_{k=1}^{N-1} a_k T_k(t). \quad (6)$$

One efficient way to evaluate  $Y_c(t)$  for a given value  $t$  is the *Clenshaw's algorithm* [7, chap 3] [8, chap 4]. By defining the “backward recurrence sequence”

$$b_k(t) = 2t b_{k+1}(t) - b_{k+2}(t) + a_k, \text{ for } k = N-1, \dots, 1, 0 \quad (7)$$

with the initial condition  $b_N(t) = b_{N+1}(t) = 0$ , and substituting (7) in (6), we have

$$Y_c(t) = \sum_{k=0}^{N-1} [b_k(t) - 2t b_{k+1}(t) + b_{k+2}(t)] T_k(t). \quad (8)$$

Using the recurrence formula in (5), (8) becomes

$$Y_c(t) = \frac{b_0(t) - b_2(t)}{2}. \quad (9)$$

For our purpose, we need the evaluation of

$$Y'_c(t) = \sum_{k=0}^{N-1} a_k \cos(k\omega) = \sum_{k=0}^{N-1} a_k T_k(t). \quad (10)$$

It can be shown that by scaling  $a_0$  by 2 beforehand, we can evaluate  $Y'_c(t)$  through the same steps in (7)-(9). Therefore, once we scale  $a_0$  by 2 and calculate  $b_0(t)$  and  $b_2(t)$  using (7), the evaluation of  $Y'_c(t)$  can be readily obtained from (9) with one addition and one shift operation. The corresponding architecture to evaluate  $Y'_c(t)$  is shown in Fig.2. Since (7) is a “backward” recurrence formula, the input sequence must be *reversed*. The second-order recurrence structure computes the  $b'_i$ 's in (7), and (9) is performed through the adder followed by a right shift. The evaluation of  $Y'_c(t)$  can be obtained after the last input is fed into the system.

Another two interesting properties of the Chebyshev polynomial are [7, chap 3]

$$T_s(T_r(t)) = T_r(T_s(t)) = T_{rs}(t), \quad (11)$$

$$T_s(t)T_r(t) = \frac{1}{2}(T_{s+r}(t) + T_{s-r}(t)) \quad (12)$$

which will also be used in later derivations.

### 3. Parallel IDCT Architecture

In order to illustrate the relationship between the Chebyshev polynomial and the transforms, we will begin with the derivation of IDCT algorithm. Let  $X(k)$ ,  $k = 0, 1, \dots, N -$

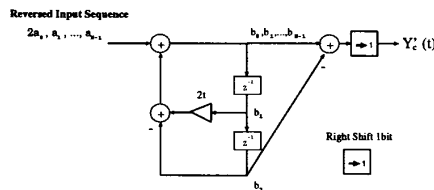


Figure 2: Systolic architecture to evaluate  $Y'_c(t)$ .

1, be a DCT-domain sequence. The block IDCT to compute the time-domain sequence  $x(n)$ ,  $n = 0, 1, \dots, N - 1$ , is

$$x(n) = \sum_{k=0}^{N-1} \epsilon(k) X(k) \cos\left(\frac{(2n+1)\pi}{2N} k\right), \quad (13)$$

where

$$\epsilon(k) = \begin{cases} 1/\sqrt{2}, & \text{if } k = 0 \\ 1, & \text{otherwise.} \end{cases} \quad (14)$$

Let

$$\omega_n \triangleq \frac{(2n+1)\pi}{2N} \quad (15)$$

and use the definition of the Chebyshev polynomial in (4), (13) can be written as

$$x(n) = \sum_{k=0}^{N-1} \epsilon(k) X(k) \cos(\omega_n k) = \sum_{k=0}^{N-1} X'(k) T_k(t_n). \quad (16)$$

Comparing (16) with (10), we see that the IDCT at index  $n$  can be treated as the evaluation of Chebyshev series at  $t_n = \cos \omega_n$  with coefficient set  $[X'(0), X'(1), \dots, X'(N-1)]$ . As a consequence, if we replace the  $t$  in Fig.2 with  $t_n$ , the recurrence structure can perform the IDCT at the center frequency  $\omega_n$ . Since  $n = 0, 1, \dots, N - 1$ , we need  $N$  second-order recurrence structures to compute IDCT in parallel.

Fig.3 shows the overall IDCT structure based on the Chebyshev recursion. It has two parts: the *Reverse Array* (RA) and the IDCT array. The RA consists of one *serial-input-parallel-output* (SIPO) register and one *parallel-input-serial-output* (PISO) register. It provides the capability of reversing the input sequence and scaling  $X(0)$  in a fully pipelined way. The IDCT array performs IDCT at different  $\omega_n$ . The whole system works in a serial-input-parallel-output way and requires only  $N + 1$  multipliers and  $3N$  adders including the scaling multiplier in RA. Compared with other schemes, our IDCT architecture uses the least multipliers to date.

### 4. Parallel DCT Architecture

As with the derivation of the IDCT algorithm, the DCT can be written as

$$X(k) = \frac{2\epsilon(k)}{N} \sum_{n=0}^{N-1} x(n) \cos((2n+1)\omega_k)$$

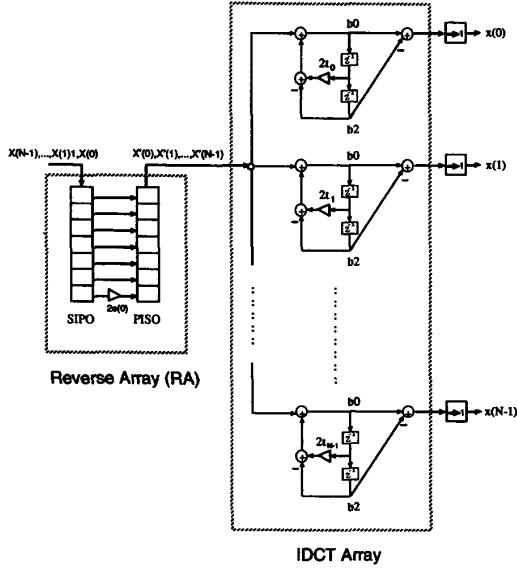


Figure 3: Systolic architecture for IDCT.

$$= \frac{2\epsilon(k)}{N} \sum_{n=0}^{N-1} x(n) T_{2n+1}(t_k) \quad (17)$$

with  $\omega_k \triangleq \frac{k\pi}{2N}$  and  $t_k \triangleq \cos^{-1} \omega_k$ . Multiplying  $T_1(t_k)$  on both sides of (17) and using the Chebyshev property in (12), we obtain

$$\begin{aligned} T_1(t_k) X(k) &= \frac{2\epsilon(k)}{N} \sum_{n=0}^{N-1} \frac{x(n)}{2} [(T_{2n}(t_k) + T_{2n+2}(t_k))] \\ &= \frac{2\epsilon(k)}{N} \sum_{n=0}^N x'(n) T_{2n}(t_k) \end{aligned} \quad (18)$$

where

$$x'(n) \triangleq \frac{x(n-1) + x(n)}{2}, \quad (19)$$

and the assumption that  $x(-1) = x(N) = 0$  in block processing is used. Note that  $T_1(t_k) = \cos \omega_k = t_k$ , and  $T_{2n}(t_k) = T_n(T_2(t_k))$  (from (11)). If we define

$$t'_k \triangleq T_2(t_k) = \cos(2\omega_k) = 2t_k^2 - 1, \quad (20)$$

we can compute  $X(k)$  from (18) by

$$X(k) = \frac{2\epsilon(k)}{N t_k} \sum_{n=0}^N x'(n) T_n(t'_k). \quad (21)$$

Therefore, the DCT at the center frequency  $\omega_k$  can be computed by first evaluating the Chebyshev series with coefficient  $x'(n)$ ,  $n = 0, 1, \dots, N$  at the value  $t'_k$ , and then scaling the evaluation by  $2\epsilon(k)/N t_k$ .

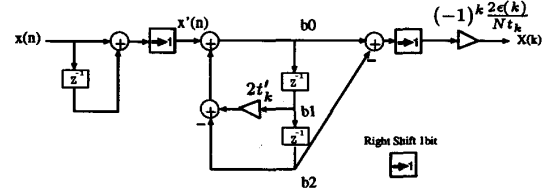


Figure 4: Systolic architecture for DCT.

Because the DCT of a reversed sequence  $\tilde{x} = [x(N-1), x(N-2), \dots, x(0)]$  can be computed as

$$\tilde{X}(k) = \frac{2\epsilon(k)}{N} \sum_{i=0}^{N-1} x(i) \cos[k\pi - \frac{(2i+1)k\pi}{2N}], \quad (22)$$

we can relate  $\tilde{X}(k)$  to  $X(k)$  by

$$\begin{cases} \tilde{X}(k) = X(k), & \text{if } k \text{ is even} \\ \tilde{X}(k) = -X(k), & \text{if } k \text{ is odd} \end{cases} \quad (23)$$

Thus the *Reverse Array*, which reverses the input sequence, can be eliminated by complementing  $X(k)$ 's with odd index while keeping  $X(k)$ 's with even index unchanged. Fig. 4 shows the architecture to implement (21) and (23) for our DCT algorithm. The overall architecture to implement the DCT needs a total of  $2N-2$  multipliers and  $3N-1$  adders.

## 5. Low-Power Design for DCT/IDCT

Consider the Chebyshev series in (10) again and split it into the even and odd series:

$$\begin{aligned} Y'_c(t) &= \sum_{i=0}^{N/2-1} a_{2i} T_{2i}(t) + \sum_{i=0}^{N/2-1} a_{2i+1} T_{2i+1}(t) \\ &= Y_e(t) + Y_o(t) \end{aligned} \quad (24)$$

By the use of (10) and (20), the even series  $Y_e(t)$  can be written as

$$Y_e(t) = \sum_{i=0}^{N/2-1} a_{2i} T_i(T_2(t)) = \sum_{i=0}^{N/2-1} a_{2i} T_i(t') \quad (25)$$

where  $t' = 2t^2 - 1$ . On the other hand, the odd series  $Y_o(t)$  can be translated into an even series by the similar derivations in (17)-(21)

$$Y_o(t) = \sum_{i=0}^{N/2} \left[ \frac{a_{2i-1} + a_{2i+1}}{2t} \right] T_i(t') \quad (26)$$

where  $t'$  has the same definition as in (25). Now combining (25) and (26) together, we have

$$Y'_c(t) = \sum_{i=0}^{N/2} \left[ a_{2i} + \frac{a_{2i-1} + a_{2i+1}}{2t} \right] T_i(t') = \sum_{i=0}^{N/2} d_i T_i(t') \quad (27)$$

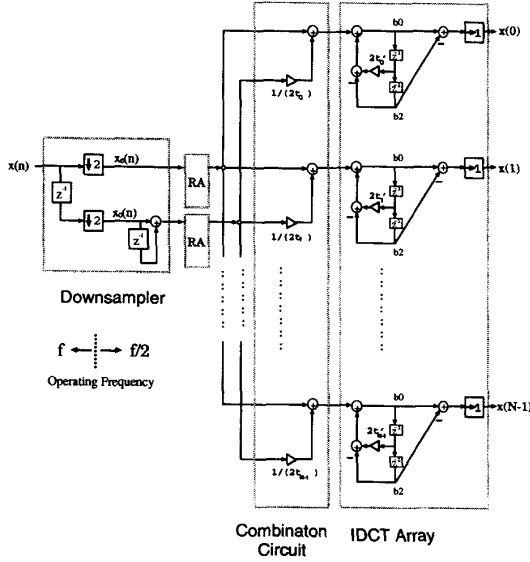


Figure 5: Implementation of the downsampling-by-2 IDCT.

where

$$d_i \triangleq \underbrace{a_{2i}}_{\text{even}} + \underbrace{\frac{a_{2i-1} + a_{2i+1}}{2t}}_{\text{odd}}. \quad (28)$$

From (27) we can see that the evaluation of  $N$ -point Chebyshev series can be reduced to a  $N/2$ -point evaluation using the new sequence  $d_i$ . The corresponding DCT/IDCT algorithms can be easily derived as in Sec.2 and Sec.3. The IDCT architecture based on (27) and (28) is depicted in Fig.5. As we can see, with some hardware overhead in the combination circuit ( $N$  extra multipliers), we can use two times slower multipliers and adders under the low voltage supply. Following the arguments for Fig.1, we know that the power consumption can be reduced to  $0.34P$ , meanwhile, the throughput rate is still retained.

To achieve downsampling by 4, we can recursively compute another new  $N/4$ -sequence  $e_i$  from  $d_i$  by using (28). Once the  $e_i$ 's are formed, the evaluation of  $Y_c'(t)$  can be computed by

$$Y_c'(t) = \sum_{i=0}^{N/4} e_i T_i(t'') \quad (29)$$

where  $t'' = 2t'^2 - 1$ . Because the operating speed has been slowed down by 4, the lowest possible voltage supply can be  $2.0V$  [1] and the total power consumption is reduced to  $0.2P$ .

The comparison of hardware complexity for our DCT/IDCT architectures is listed in Table.1. For the low-power design, the hardware overhead is reasonable for the speed compensation.

	Normal Operation		Down-sampling = 2		Down-sampling = 4	
	M	A	M	A	M	A
IDCT	$N+1$	$3N$	$2N+1$	$4N+1$	$4N+1$	$6N+4$
DCT	$2N-2$	$3N-1$	$3N-3$	$4N$	$5N-5$	$6N+3$

Table 1: Comparison of hardware cost for normal DCT/IDCT and the low-power design (M=Multiplier, A=Adder).

## 6. Conclusion

In this paper a low-complexity parallel DCT/IDCT algorithm based on the backward Chebyshev recursion is derived. A low-power design of our algorithm using the downsampling scheme is also investigated. The resulting system is modular and regular. Extension of our 1-D DCT/IDCT algorithm to 2-D operation can be easily achieved by employing the time-recursive 2-D DCT architecture proposed by Chiu and Liu [9]. Therefore, the proposed scheme will be effective for low-power signal processing systems.

## References

- [1] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, pp. 473-484, April 1992.
- [2] G. Steidl, "Chebyshev polynomial derivation of composite-length DCT algorithms," *Signal Processing*, vol. 29, pp. 17-27, Oct. 1992.
- [3] R. W. Brodersen, A. P. Chandrakasan, and S. Sheng, "Low-power signal processing systems," in *VLSI signal processing V* (K. Yao, R. Jain, W. Przytula, and J. Rabaey, eds.), pp. 3-13, IEEE Press, 1992.
- [4] M. T. Sun, T. C. Chen, and A. M. Gottlieb, "VLSI implementation of a  $16 \times 16$  discrete cosine transform," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 610-617, Apr. 1989.
- [5] 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. 41, pp. 1357-1377, March 1993.
- [6] K. J. R. Liu, C. T. Chiu, R. K. Kolagotla, and J. F. Ja'Ja', "Optimal unified IIR architectures for time-recursive discrete sinusoidal transforms," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, (Minneapolis), pp. III.73-76, 1993.
- [7] L. Fox and I. B. Parker, *Chebyshev polynomials in numerical analysis*. London: Oxford University Press, 1968.
- [8] H. R. Schwarz, *Numerical analysis: A comprehensive introduction*. John Wiley & Sons, 1989.
- [9] C.-T. Chiu and K. J. R. Liu, "Real-time parallel and fully pipelined two-dimensional DCT lattice structures with application to HDTV systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, pp. 25-37, March 1992.