Journal of VLSI Signal Processing 26, 319–332, 2000

A Low Power 8 × 8 Direct 2-D DCT Chip Design

HAO-CHIEH CHANG, JIUN-YING JIU, LI-LIN CHEN AND LIANG-GEE CHEN
Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, 10764 R.O.C

Received March 25, 1999; Revised December 29, 1999

Abstract. This paper presents the design and implementation of a low power 8 × 8 2-D DCT chip based on a computation-effective algorithm. Computational complexity can be reduced by simplifying the direct 2-D algorithm. Thus, the low power consumption is achieved due to complexity reduction. Besides, the parallel distributed-arithmetic (DA) technique is used to realize constant multiplication due to the low-power consideration. Additionally, the 2 V-power supply is practiced in circuit implementation for now and future battery operated applications. By using the TSMC 0.6μm single-poly double-metal technology, 133 mW power consumption at 100 MHz and the 133 MHz maximum operation speed are achieved by critical path simulation.

Keywords: low power, DCT, distributed arithmetic, image/video coding, VLSI

I. Introduction

The discrete cosine transform (DCT) [1] is a very promising technique used for video/image coding, and widely adopted by most image and video compression standards, such as JPEG [2], MPEG [3], MPEG2 [4], H.261 [5] and H.263 [6] etc. Since increasing applications apply these standards to portable systems like hand-held videophone and multimedia terminals, it becomes imperative to develop a low power DCT chip as one key component for such applications. In this paper, the low power approach based on algorithm, architecture and circuit level optimization is derived to achieve such design goal.

There have been lots of implementations on DCT by researches and commercial products. Most of them can be classified into three major approaches: (a) row-column decomposition [7]–[12], (b) direct method [13]–[18] by polynomial transform and (c) the adaptation of other transform like DFT and DHT [19]. The row-column approach is favored in most VLSI implementations by its regularity advantage. Since direct 2-D DCT algorithm is somewhat more complex, direct method approach usually employs mathematics simplification to reduce its computational complexity [18]. Among those low power implementations for 2-D DCT, Kuroda et al. [12] had proposed a 0.9 V, 150 MHz @ 10 mV 2-D DCT chip design. Such paper reports a very good result on power reduction for a 2-D DCT design. However, it required variable threshold voltage support under 0.3 μm CMOS triple-well technology. Namely, the power optimization is derived at circuit and device level, not including architecture and algorithm level consideration in such paper.

In most low power designs, computational complexity is a very important power consumption factor at algorithm level. And, it has been proved that the computational complexity of the row-column approach is more than that of direct 2-D approach [18]. Based on this result, the low-power 2-D DCT architecture using direct method is designed. Although the direct method incurs the irregularity in realizing 2-D DCT chips, the feature of low computational complexity is attractive for low power DCT chip design. Besides, the irregularity can be reduced by employing the proposed routing techniques. Moreover, the parallel architecture is designed such that it can work at low supply voltage (2 V) to achieve low power consumption. The low-power circuits are also designed in order to reduce power consumption more. Thus, the major efforts of this paper are on effective algorithm deduction, low-complexity parallel architecture and low-power circuit designs.

This paper is organized as follows. The computation-effective direct 2-D DCT algorithm [18] will be briefly described in Section II. Based on such derivation, an efficient architecture is proposed in Section III. In the Section IV, circuit design with 2 V power supply will be shortly illustrated. The core characteristics, the simulation results and the comparisons with previous studies will be shown in Section V. A conclusion for this architecture approach, the chip design and possible further enhancements are pointed in the end.

II. The Direct 2-D DCT Algorithm

The adopted direct 2-D DCT algorithm is reviewed and its derivation from original 2-D DCT expression is briefly discussed in this section. The derivation of algorithm can be divided two major stages: (A) The mapping of input data, and (B) The direct 2-D DCT algorithm deduction.

A. The Mapping of Input Data

The 2-D DCT of an $N \times N$ real signal x_{n_1,n_2} is defined as:

$$\begin{aligned} X_{k_1,k_2} &= \frac{2}{N}c(k_1)c(k_2)\sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1} x_{n_1,n_2} \\ &\quad \cdot \cos\left[\frac{2\pi(2n_1+1)k_1}{4N}\right]\cos\left[\frac{2\pi(2n_2+1)k_2}{4N}\right] \\ &\quad n_1, n_2, k_1, k_2 = 0, 1, \ldots, N-1 \\ &\quad c(0) = \frac{1}{\sqrt{2}}, \text{ and } c(n) = 1 \text{ for } n \neq 0 \end{aligned} \tag{1}$$

For simplicity, Y_{k_1,k_2} is introduced by neglecting the kernel factor $2c(k_1)c(k_2)/N$ such that

$$\begin{aligned} Y_{k_1,k_2} &= \sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1} x_{n_1,n_2} \cdot \cos\left[\frac{2\pi(2n_1+1)k_1}{4N}\right] \\ &\quad \times \cos\left[\frac{2\pi(2n_2+1)k_2}{4N}\right] \\ &\quad n_1, n_2, k_1, k_2 = 0, 1, \ldots, N-1 \end{aligned} \tag{2a}$$

and

$$X_{k_1,k_2} = \frac{2}{N}c(k_1)c(k_2)Y_{k_1,k_2} \tag{2b}$$

By using the permutation [13], signal x_{n_1,n_2} can be permuted as:

$$\begin{aligned} y_{n_1,n_2} &= x_{2n_1,2n_2}, & n_1 &= 0,\ldots,N/2-1, \\ & & n_2 &= 0,\ldots,N/2-1 \\ &= x_{2N-2n_1-1,2n_2}, & n_1 &= N/2,\ldots,N-1, \\ & & n_2 &= 0,\ldots,N/2-1 \\ &= x_{2n_1,2N-2n_2-1}, & n_1 &= 0,\ldots,N/2-1, \\ & & n_2 &= N/2,\ldots,N-1 \\ &= x_{2N-2n_1-1,2N-2n_2-1}, & n_1 &= N/2,\ldots,N-1, \\ & & n_2 &= N/2,\ldots,N-1 \end{aligned}$$

Thus, (2a) can be rewritten as:

$$\begin{aligned} Y_{k_1,k_2} &= \sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1} y_{n_1,n_2} \cdot \cos\left[\frac{2\pi(4n_1+1)k_1}{4N}\right] \\ &\quad \times \cos\left[\frac{2\pi(4n_2+1)k_2}{4N}\right] \\ &\quad n_1, n_2, k_1, k_2 = 0, 1, \ldots, N-1 \end{aligned} \tag{3}$$

Consider the following expression:

$$\begin{aligned} U_{k_1,k_2} &= \sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1} y_{n_1,n_2} W_{4N}^{(4n_1+1)k_1+(4n_2+1)k_2}, \\ &\quad \text{where } W_{4N} = \exp\left(-j\frac{2\pi}{4N}\right) \end{aligned} \tag{4}$$

It is not difficult to find that Y_{k_1,k_2} can be computed from U_{k_1,k_2} by the following set of expressions:

$$\begin{aligned} Y_{k_1,k_2} &= \frac{1}{2}[\mathrm{Re}(U_{k_1,k_2}) - \mathrm{Im}(U_{N-k_1,k_2})] \\ Y_{k_1,N-k_2} &= \frac{1}{2}[-\mathrm{Im}(U_{k_1,k_2}) - \mathrm{Re}(U_{N-k_1,k_2})] \end{aligned} \tag{5}$$

In order to obtain all Y_{k_1,k_2}, note that U_{k_1,k_2} in (4) has to be computed for all k_1 and subset of k_2 that $\{k_2, N-k_2\}$ covers all possible values of k_2.

B. The Direct 2-D DCT Algorithm Deduction

As described in [18], the exponential term $[(4n_1+1)k_1 + (4n_2+1)k_2]$ can be treated as a rotation. In the row-column approach the term $W_{4N}^{(4n_1+1)k_1+(4n_2+1)k_2}$ is rotated twice, one for row operation $[(4n_1+1)k_1]$ and the other for column operation $[(4n_2+1)k_2]$. However, the relation between n_1 and n_2 can be employed such that the term $W_{4N}^{(4n_1+1)k_1+(4n_2+2)k_2}$ could be rotated only once so as to reduce computation. Based on this idea, let's consider the following expression,

$$4n_2 + 1 = (4t + 1)(4n_1 + 1) \bmod 4N, \quad \text{where } 0 \le t, n_1,\ n_2 \le N - 1 \tag{6}$$

By substituting (6) into (4), the latter can be rewritten as:

$$U_{k_1,k_2} = \sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1} y_{n_1,t} W_{4N}^{(4n_1+1)[k_1+(4t+1)k_2]} \tag{7a}$$

$$= \sum_{t=0}^{N-1}\left[\sum_{n_1=0}^{N-1} y_{n_1,t} W_{4N}^{(4n_1+1)[k_1+(4t+1)k_2]}\right] \tag{7b}$$

Furthermore, consider the following relation:

$$k_1 + (4t + 1)k_2 = aN + b, \quad \text{where } a \in \text{integer and } 0 \le b \le N - 1$$

By substituting the above relation into (7b), one can obtain:

$$U_{k_1,k_2} = \sum_{t=0}^{N-1}(-j)^a\left[\sum_{n_1=0}^{N-1} y_{n_1,t} W_{4N}^{(4n_1+1)b}\right] \tag{8}$$

The inner summation can be rewritten as follows:

$$\begin{aligned} U'_{t,b} &= \sum_{n_1=0}^{N-1} y_{n_1,t} \cdot W_{4N}^{(4n_1+1)b}, \quad 0 \le b \le N - 1 \\ &= \sum_{n_1=0}^{N-1} y_{n_1,t}\left[\cos\frac{2\pi(4n_1+1)b}{4N} - j\sin\frac{2\pi(4n_1+1)b}{4N}\right] \\ &= \sum_{n_1=0}^{N-1} y_{n_1,t}\left[\cos\frac{2\pi(4n_1+1)b}{4N} - j\cos\frac{2\pi(4n_1+1)(N-b)}{4N}\right] \end{aligned} \tag{9}$$

Although $U'_{t_1,b}$ is a complex number, its real part is indeed an N-point 1-D DCT, and its imaginary part can be obtained by the following equations:

$$\operatorname{Im}\{U'_{t,0}\} = 0, \tag{10a}$$

$$\operatorname{Im}\{U'_{t,b}\} = -\operatorname{Re}\{U'_{t,N-b}\}, \quad 1 \le b \le N - 1. \tag{10b}$$

By applying this algorithm, an $N \times N$ 2-D DCT can thus be realized by N N-point 1-D DCT and some extra additions. Compared with the row-column decomposition method, which needs $2N$ N-point 1-D DCT, this approach is more suitable for low-complexity implementation.

III. Low Power 2-D DCT Architecture

By reviewing (5) and (8), the data flow for computing $N \times N$ 2-D DCT from input x_{n_1,n_2} to the output Y_{k_1,k_2} is shown as follows:

$$x_{n_1,n_2} \Rightarrow y_{n_1,n_2} \Rightarrow \xrightarrow{4n_2+1=(4t+1)(4n_1+1)\%4N}$$

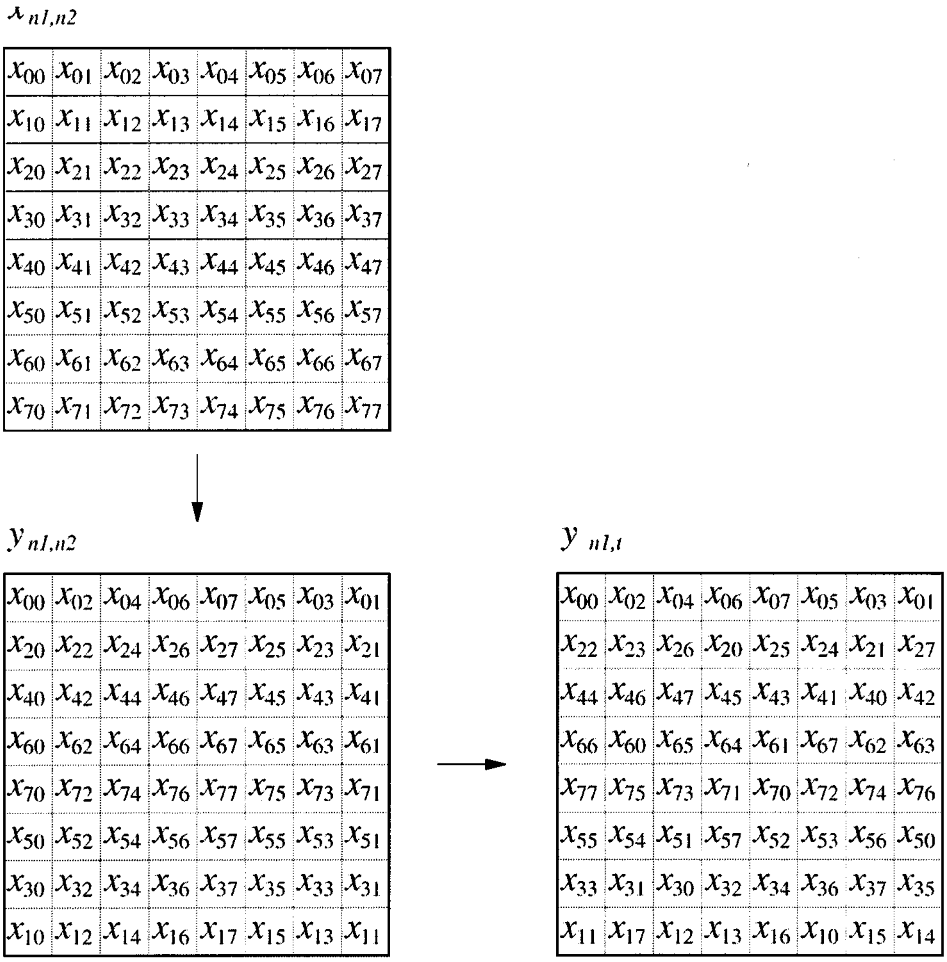
$$\Rightarrow y_{n_1,t} \Rightarrow \xrightarrow{\text{1D}-\text{DCT}} \Rightarrow U'_{t,b} \Rightarrow \underline{\sum_{t=0}^{N-1}(-j)^a \times}$$

$$\Rightarrow U_{k_1,k_2} \Rightarrow \xrightarrow{\frac{1}{2}[\pm\text{Re}(\ )\pm\text{Im}(\ )]} \Rightarrow Y_{k_1,k_2}$$

In this flow, the input data x_{n_1,n_2} are mapped to $y_{n_1,t}$ at first. Namely, the source data have to be re-ordered such that the subsequent processing can be performed correctly. This data mapping is finished by (6). After the source data mapping, $U'_{t,b}$ is then achieved by calculating the 1-D complex DCT of $y_{n_1,t}$. Then, before the output Y_{k_1,k_2} is computed by (5), U_{k_1,k_2} have to be computed by the summation with respect to t depicted in (8). In summary, this flow can be divided into three major processing stages: i) Source data re-ordering; ii) 1-D complex DCT; and iii) Post-summation;

i) *Source data re-ordering*: It is quite intuitive to realize data re-ordering by applying an input buffer and an address generator, which is used to provide the write address of input data. Since the mapping of input data is fixed, the address generator could be implemented by an address mapping table and a counter. Fig. 1 shows the mapping from x_{n_1,n_2} to $y_{n_1,t}$ for 8 × 8 block. The input buffer serves as a serial-in parallel-out convert for parallel subsequent stages. The parallel architecture is designed in order to increase the processing throughput.

ii) *1-D complex DCT*: As mentioned in Section II, 1-D complex DCT consists of two N-point 1-D DCT, one is real part and the other one is imaginary



$x_{n1,n2}$

x_{00}	x_{01}	x_{02}	x_{03}	x_{04}	x_{05}	x_{06}	x_{07}
x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}
x_{20}	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}
x_{30}	x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}	x_{37}
x_{40}	x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}	x_{47}
x_{50}	x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	x_{56}	x_{57}
x_{60}	x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	x_{66}	x_{67}
x_{70}	x_{71}	x_{72}	x_{73}	x_{74}	x_{75}	x_{76}	x_{77}

$y_{n1,n2}$

x_{00}	x_{02}	x_{04}	x_{06}	x_{07}	x_{05}	x_{03}	x_{01}
x_{20}	x_{22}	x_{24}	x_{26}	x_{27}	x_{25}	x_{23}	x_{21}
x_{40}	x_{42}	x_{44}	x_{46}	x_{47}	x_{45}	x_{43}	x_{41}
x_{60}	x_{62}	x_{64}	x_{66}	x_{67}	x_{65}	x_{63}	x_{61}
x_{70}	x_{72}	x_{74}	x_{76}	x_{77}	x_{75}	x_{73}	x_{71}
x_{50}	x_{52}	x_{54}	x_{56}	x_{57}	x_{55}	x_{53}	x_{51}
x_{30}	x_{32}	x_{34}	x_{36}	x_{37}	x_{35}	x_{33}	x_{31}
x_{10}	x_{12}	x_{14}	x_{16}	x_{17}	x_{15}	x_{13}	x_{11}

$y_{n1,t}$

x_{00}	x_{02}	x_{04}	x_{06}	x_{07}	x_{05}	x_{03}	x_{01}
x_{22}	x_{23}	x_{26}	x_{20}	x_{25}	x_{24}	x_{21}	x_{27}
x_{44}	x_{46}	x_{47}	x_{45}	x_{43}	x_{41}	x_{40}	x_{42}
x_{66}	x_{60}	x_{65}	x_{64}	x_{61}	x_{67}	x_{62}	x_{63}
x_{77}	x_{75}	x_{73}	x_{71}	x_{70}	x_{72}	x_{74}	x_{76}
x_{55}	x_{54}	x_{51}	x_{57}	x_{52}	x_{53}	x_{56}	x_{50}
x_{33}	x_{31}	x_{30}	x_{32}	x_{34}	x_{36}	x_{37}	x_{35}
x_{11}	x_{17}	x_{12}	x_{13}	x_{16}	x_{10}	x_{15}	x_{14}

Figure 1. Data mapping from x_{n_1,n_2} through y_{n_1,n_2} to $y_{n_1,t}$ for an 8 × 8 block.

part. By employing the relation between the real part and imaginary part, which is shown in (10), the 1-D complex DCT can be achieved by only calculating the real part 1-D DCT. Moreover, the symmetry of real 1-D DCT can be used to reduce the computation further.

The 8 × 8 coefficient-matrix applied to calculate the 1-D DCT can be written as:

$$A = \begin{Bmatrix} a & a & a & a & a & a & a & a \\ b & c & d & e & -b & -c & -d & -e \\ f & g & -f & g & f & g & -f & g \\ -e & -b & -c & d & e & b & -c & -d \\ h & -h & h & -h & h & -h & h & -h \\ c & -d & e & b & -c & d & -e & -b \\ -g & f & g & -f & -g & f & g & -f \\ -d & -e & b & c & d & e & -b & -c \end{Bmatrix}$$

where,

$$\begin{Bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{Bmatrix} = \begin{Bmatrix} 1 \\ \cos\dfrac{1}{16}\pi \\ \cos\dfrac{5}{16}\pi \\ \cos\dfrac{9}{16}\pi \\ \cos\dfrac{13}{16}\pi \\ \cos\dfrac{2}{16}\pi \\ \cos\dfrac{10}{16}\pi \\ \cos\dfrac{4}{16}\pi \end{Bmatrix}$$

It is observed that even rows of matrix A are even-symmetric and odd rows are odd-symmetric. Therefore, by exploiting this relationship of symmetry, the even rows and odd rows can be separated. Thus, the 8×8 1D DCT can be decomposed as

$$\begin{pmatrix} \mathrm{Re}\{U'_{t,0}\} \\ \mathrm{Re}\{U'_{t,2}\} \\ \mathrm{Re}\{U'_{t,4}\} \\ \mathrm{Re}\{U'_{t,6}\} \end{pmatrix} = \begin{pmatrix} a & a & a & a \\ f & g & f & g \\ h & -h & h & -h \\ -g & f & g & -f \end{pmatrix} \cdot \begin{pmatrix} y_{0,t} + y_{4,t} \\ y_{1,t} + y_{5,t} \\ y_{2,t} + y_{6,t} \\ y_{3,t} + y_{7,t} \end{pmatrix}$$

$$\begin{pmatrix} \mathrm{Re}\{U'_{t,1}\} \\ \mathrm{Re}\{U'_{t,3}\} \\ \mathrm{Re}\{U'_{t,5}\} \\ \mathrm{Re}\{U'_{t,7}\} \end{pmatrix} = \begin{pmatrix} b & c & d & e \\ -e & -b & -c & d \\ c & -d & e & b \\ -d & -e & b & c \end{pmatrix} \cdot \begin{pmatrix} y_{0,t} - y_{4,t} \\ y_{1,t} - y_{5,t} \\ y_{2,t} - y_{6,t} \\ y_{3,t} - y_{7,t} \end{pmatrix}$$

From the resulting decomposition, the number of multiplication is halved since 8×8 matrix-vector multiplication has been replaced by two 4×4 matrix-vector multiplications. Furthermore, these two matrix-vector multiplications can be computed in parallel. The savings in the multiplication are traded at the expense of some data ordering before these two 4×4 vector-multiplication are performed.

Based on the above deduction, basic 1-D complex DCT cell with reduced computational complexity is proposed. Since the coefficients of matrix A are constant, the implementation of multiplication can be simplified. The ROM-based Distributed Arithmetic (DA) [20] is a promising and widely used approach for implementing the constant multiplier. Therefore, ROM-based DA technique is utilized for simplifying the architecture, instead of adopting the traditional MAC implementation for the matrix-vector multiplication, The 1-D DCT basic cell composed of a ROM, an accumulator and registers, is shown in Fig. 2. The basic cell can compute one of the real part of $U'_{t,b}$ every four clock cycles. Thus, 8-point 1-D complex DCT can be realized by using eight basic 1-D DCT cells. The proposed parallel architecture is shown in Fig. 3. Note that the imaginary part of $U'_{t,b}$ is simply achieved by wire routing for the relationship presented in (10a) and (10b). The inputs for the basic cell are the addition/subtraction pairs of $y_{n_1,t}$. It should be noted that the extra shift register is used to reorder the data with parallel-in serial-out for next stage.

iii) *Post-summation*: In this stage, U_{k_1,k_2} have to be computed by the summation of $U'_{t,b}$ with respect to t first, and then all Y_{k_1,k_2} are obtained from U_{k_1,k_2}. Since the coefficient term $(-j)^a$ in (8) could have four different values, i.e., ± 1, $\pm j$, depend on exponent coefficient "a", U_{k_1,k_2} can be achieved by a serial of addition/subtraction operations on the input $U'_{t,b}$. Thus, a butterfly-like addition/subtraction network is constructed to compute U_{k_1,k_2}. Without computing the term $(-j)^a$ directly, the exponent coefficient "a" can be decomposed into several small number first such that the common terms, $(-j)^0$, $(-j)^2 \ldots$, can be extracted. Using these common terms repeatedly can thus reduce the computation. Figure 4 shows the deduction process of U_{k_1,k_2}, where $k_1 = 3, 4, 5, 6$; $k_2 = 1, 5$ as examples. In (8), as the summation index "t" goes from 0 to 7, the exponent coefficient "a" becomes 0, 1, 1, 2, 2, 3, 3, 0, respectively (Strictly speaking, the "a" should become $4N + 0, 4N + 1, \ldots, 4N + 3, 4N + 0$, respectively, where N is an integer). All these "a" coefficients are then divided into several numbers to extract the common terms. As shown in the left side of Fig. 4, these "a" are divided into three groups eventually, that is, the three columns in the rightmost dashed box. The reason for choosing $U_{3.4.5.6.,1}$ and $U_{3.4.5.6.,5}$ as an

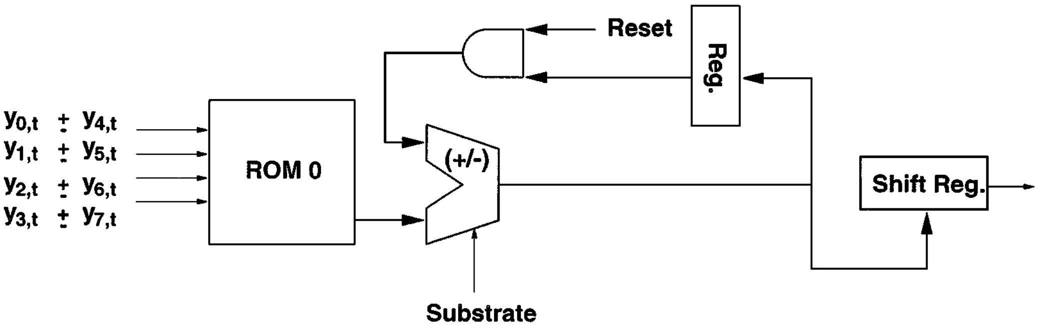


Figure 2. Architecture of 1-D DCT basic cell.

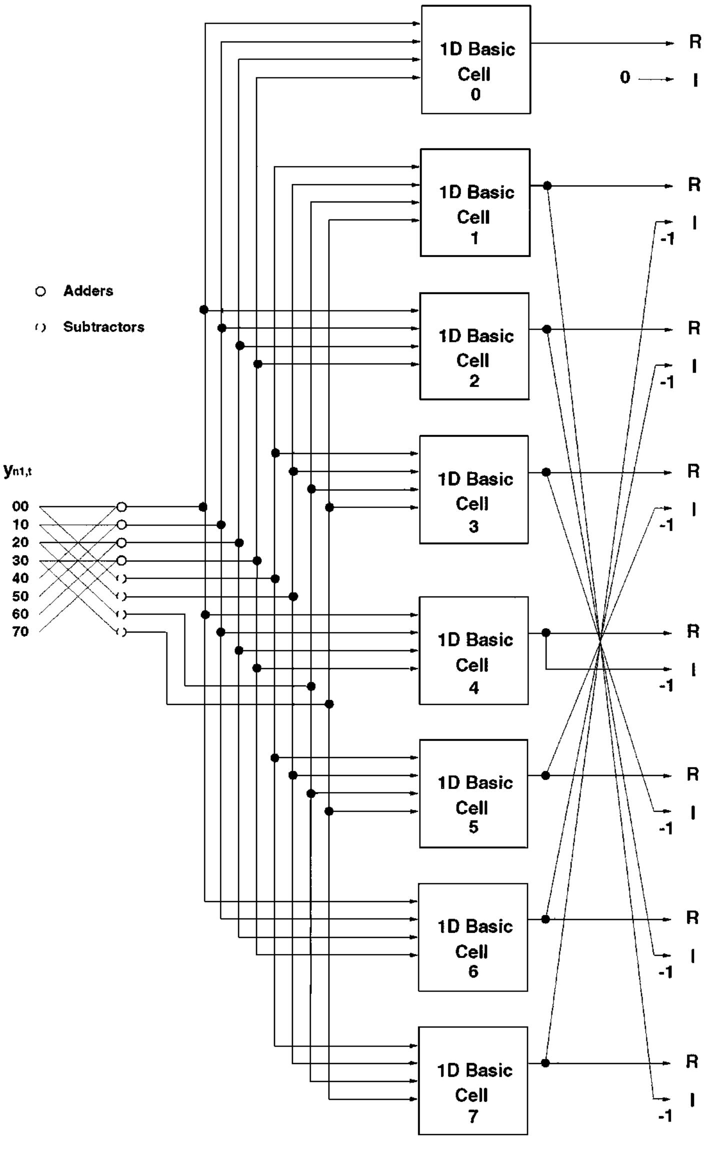
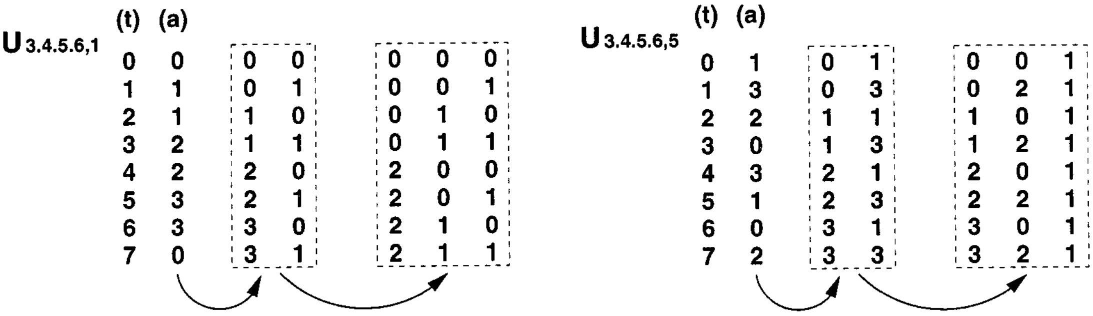


Figure 3. Architecture of the fast 1D-DCT algorithm.



$U_{3.4.5.6,1}$

(t)	(a)					
0	0	0	0	0	0	0
1	1	0	1	0	0	1
2	1	1	0	0	1	0
3	2	1	1	0	1	1
4	2	2	0	2	0	0
5	3	2	1	2	0	1
6	3	3	0	2	1	0
7	0	3	1	2	1	1

$U_{3.4.5.6,5}$

(t)	(a)					
0	1	0	1	0	0	1
1	3	0	3	0	2	1
2	2	1	1	1	0	1
3	0	1	3	1	2	1
4	3	2	1	2	0	1
5	1	2	3	2	2	1
6	0	3	1	3	0	1
7	2	3	3	3	2	1

Figure 4. Decomposition of "*a*" coefficient into several segments such that common terms can be extracted. Each column means the coefficient $(-j)^a$ multiplied by U_{k_1,k_2}.

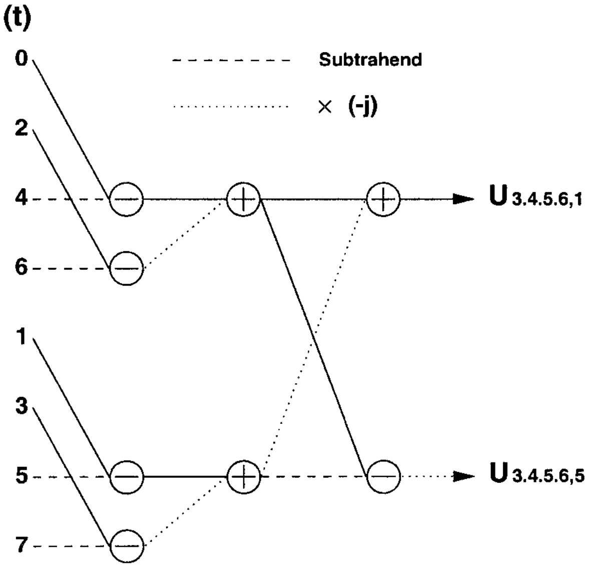


Figure 5. The example for $U_{3.4.5.6,\,1}$ and $U_{3.4.5.6,\,5}$ architecture deduction.

example is that when index "t" changes, they have the same corresponding "a" values. Accordingly, $U_{3.4.5.6.,1}$ (or $U_{3.4.5.6.,5}$) can be achieved by the same calculations. Namely, the computation amount required for achieving these four data is reduced to one fourth. In fact, the other k_1/k_2 combinations have similar results. Thus, the computation amount can be further reduced by using this summation-network deduction technique.

Based on the decomposition of "a" coefficient, an addition/subtraction network with three stages is constructed to complete the summation (8) with index from 0 to 7. It is not difficult to find that some symmetric relation exists among these decomposed coefficients. Accordingly, the common terms can be extracted to simplify the addition/subtraction network. Figure 5 shows the final network configuration for calculating $U_{3.4.5.6.,1}$ and $U_{3.4.5.6.,5.}$ Note that the dash-line in Fig. 5 represents the subtrahend in subtraction. As the depiction of (5), only a subset of k_2 in U_{k_1,k_2} is sufficient to attain all Y_{k_1,k_2}. In our design, {0, 1, 2, 4, 5} is chosen for k_2. Then applying the deduction approach for each U_{k_1,k_2}, the butterfly-like addition/subtraction network is constructed and shown in Fig. 6. After the essential U_{k_1,k_2} are computed, Y_{k_1,k_2} can be easily achieved by the final addition/subtraction stage, which is derived from (5).

The basic processing units, which are deduced in these three stages, can reduce computation complexity

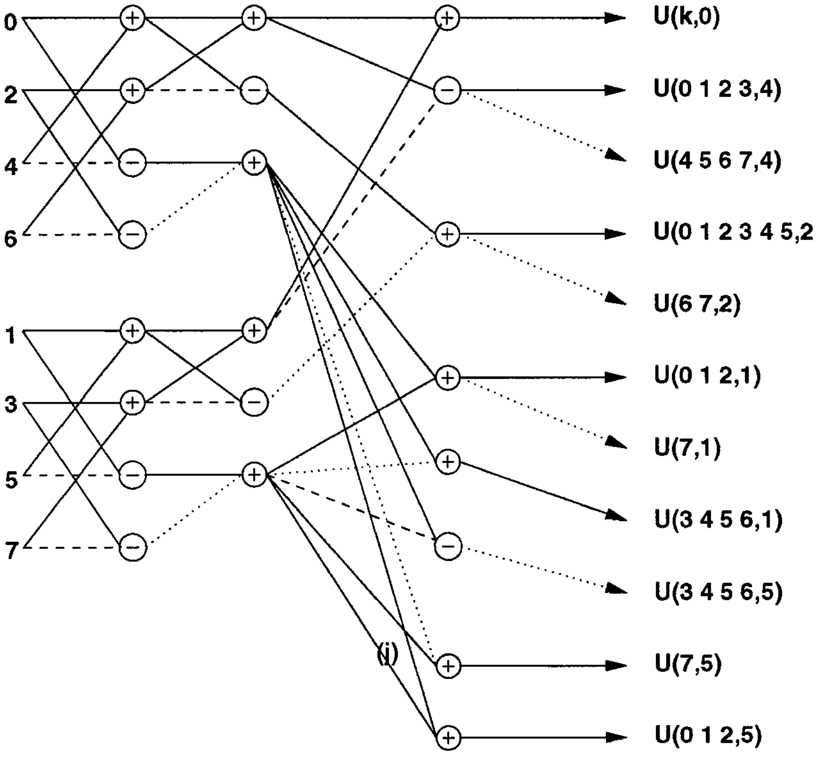


Figure 6. Butterfly-like adder-substracter routing tree.

more as we mentioned above. Less computation could lead to less power consumption essentially. Another important power factor is the supply voltage. In order to achieve a low power 2-D DCT chip at reduced supply voltage, a parallel architecture is commonly used to compensate the speed degradation due to lowering operating voltage. Figure 7 shows the core of the 2-D DCT chip. Such parallel architecture is composed of eight 1-D complex DCT cells, the butterfly-like addition/subtraction network and two routing modules. Note that the routing module is used to simplify the complexity of interconnection. Basically, the complicated routings of this parallel architecture is divided into several stages. Each stage is composed of two types of routing cell, cross-type and direct-type. As shown in Fig. 8, the complicated routing network can be implemented by locally duplicating the routing cells and cascading the stages.

By integrating I/O buffers, the complete 2-D DCT architecture can be achieved. The whole chip architecture is shown in Fig. 9. Since the ranges of input and output data is −255–255 and −2040–2040, respectively, the

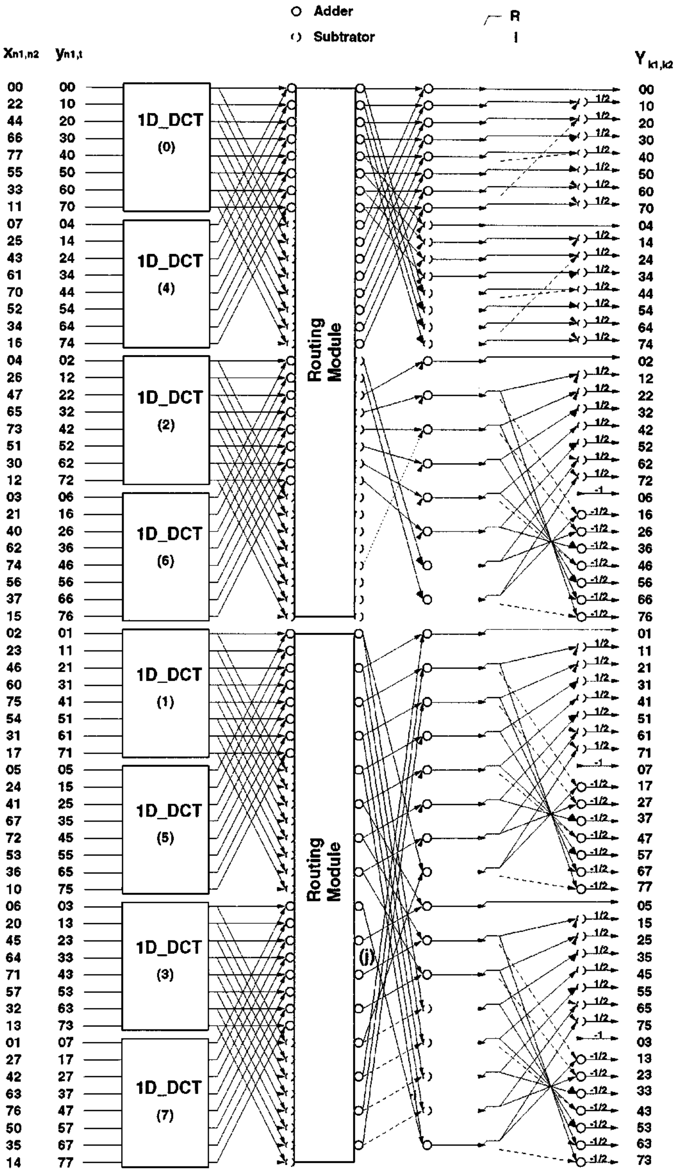


Figure 7. The proposed parallel DA 2-D DCT architecture.

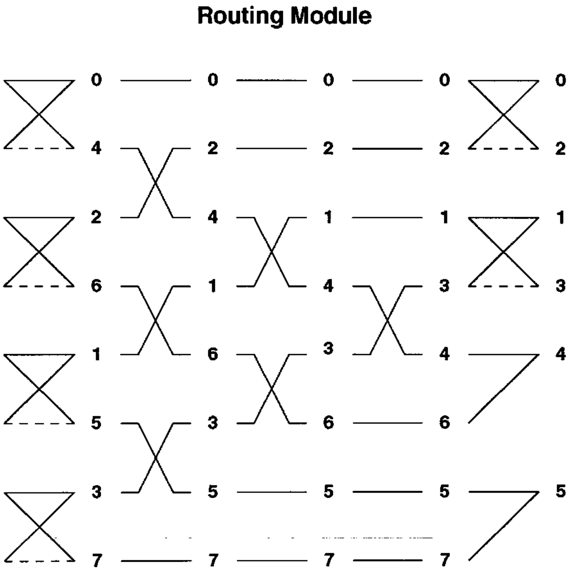


Figure 8. The details of the routing module.

word-length of input data into 9-bit and that of output data into 12-bit. For simplicity, the kernel factor $2c(n_1)c(n_2)/N$ is neglected such that the word-length of the output data turns out to 16 bits. Two-port SRAMs, which are operated in Ping-Pong mode are designed for re-ordering the input and output data of 2-D DCT core. 9-bit input data are word-serially fed to the INPUT SRAM, and are converted to 64 word-parallel bit-serial data for 2-D DCT. After 2-D DCT processing, the 64 word-parallel bit-serial resulted data are converted by the OUTPUT SRAM to 16 word-serial bit-parallel data set.

IV. Chip Implementation

The proposed low power 2-D DCT architecture has been implemented. In order to achieve 2 V power supply and 100 MHz in 0.6 μm TSMC technology, circuit-level design considerations are discussed. For speed consideration, the conventional ROM implementation is discarded. The pseudo-nMOS ROM circuit [21] can be modified for saving the power by activating the pre-charging logic only when the address line is changed. The function is achieved by integrating an ATD (address transition detection) circuit, as shown in Fig. 10. The architecture of proposed power-saving ROM is shown in Fig. 11. During the pre-charge phase, pre $=0$ and the bit-lines are pre-charged to VDD. Meanwhile,

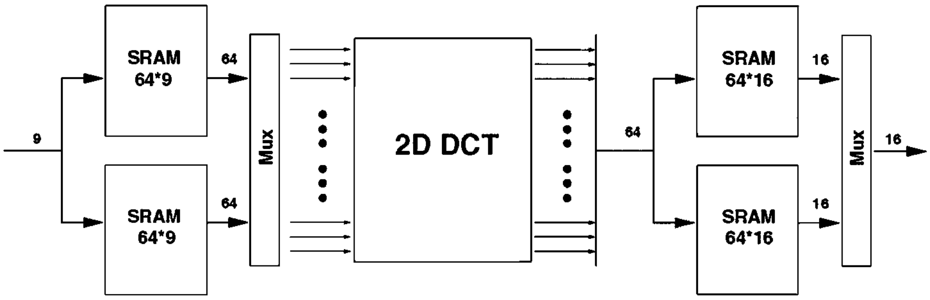


Figure 9. Low power 2D-DCT whole chip architecture.

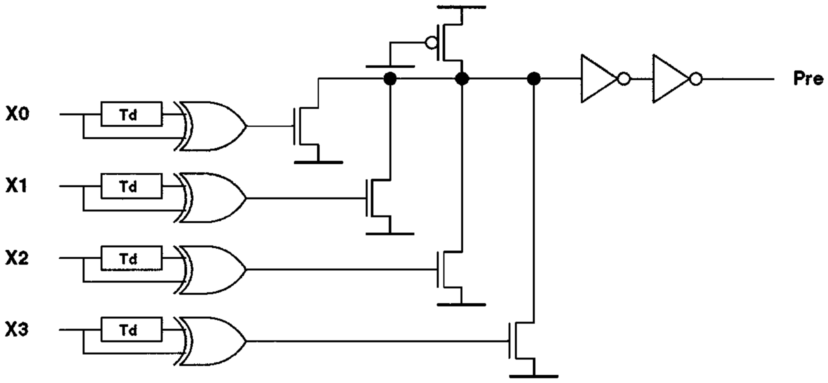


Figure 10. Address transition detector (ATD) circuit.

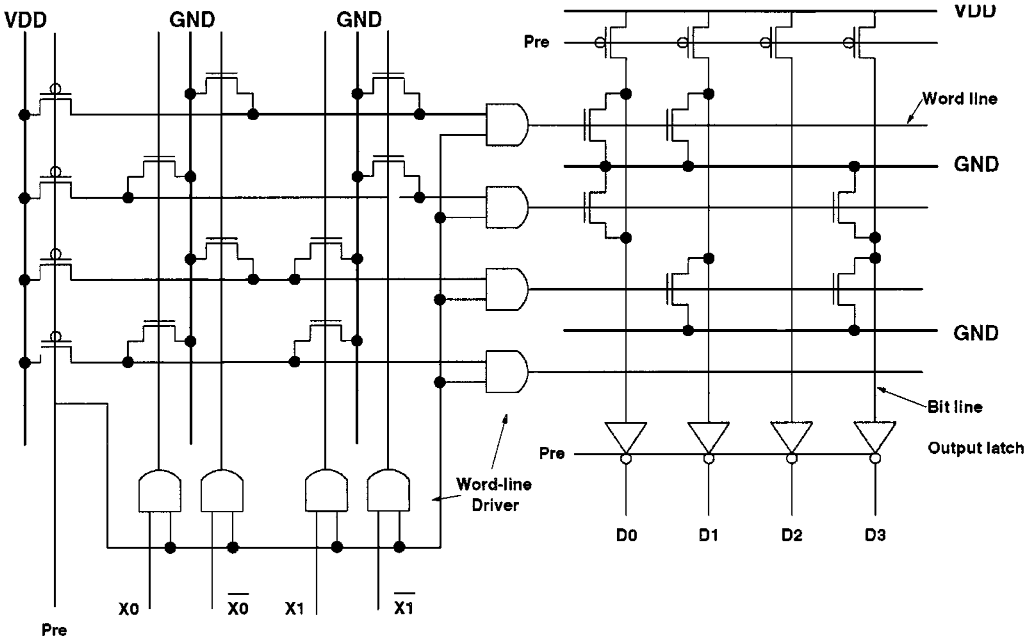


Figure 11. ROM decoder and ROM data circuit in the proposed power-saving ROM.

the AND gates in decoder ensure that all pull-down paths through the NMOS are off during pre-charging. In evaluation phase, pre = 1 and if the whole-line is activated high, the bit-line is discharged. Since the PMOS and NMOS are not turned on simultaneously during pre-charging or evaluation phase, there is no DC path from VDD to GND. Table 1 shows the comparison of the power dissipation in the conventional ROM and the proposed ROM.

Figure 12 shows the basic structure of the two-port SRAMs. They are designed for data mapping and data reordering. The INPUT-SRAM has input port size $n = 9$ and output port size $m = 64$, and the OUTPUT-SRAM has input port size $n = 64$ and output port size $m = 16$. The sense amplifier [22] consists of a cross-coupled pair of PMOS transistors and NMOS input devices. This differential pair applies the positive feedback to accelerate the sense speed. The pre-charging circuit is used for the high-speed requirement under 2 V power supply and the 0.6 μm technology.

Table 1. Power comparison of conventional ROM and proposed ROM.

ROM size = 16 × 12	Static ROM	Proposed ROM
Supply voltage	2 V	2 V
Delay	4.0 ns	7.1 ns
Psower @ 50 Mhz	2.9465 mW	0.4522 mW

In the realization of registers, TSPC [23] implementation is adopted. In addition to advantages of transistor count, TSPC-register can be integrated with extra logic functions, such as NAND, NOR and RESET. . .etc., For TSPC-registers, the power dissipated at charging and discharging phase can be eliminated if times of state-transient are reduced. Based on this consideration, the TSPC-registers are re-designed and employed for several key components, such as sequential addresser in SRAM, shift registers and counters. Actually, the simulation results also reveal that they have better performance in power and speed than that of static CMOS DFFs while applying them on those key components.

The adder design is based on the carry-select structure. In order to meet the requirement of hard specification, parallel architecture is employed to compensate the speed loss due to the low supply voltage. Large-bit adder, which is used in the 2-D DCT core, is dividing into several stages so as to make each stage operate in parallel. Each stage is implemented by using the low-power small-bit adder. In our design, Manchester adder [22] is adopted for its low power nature [24].

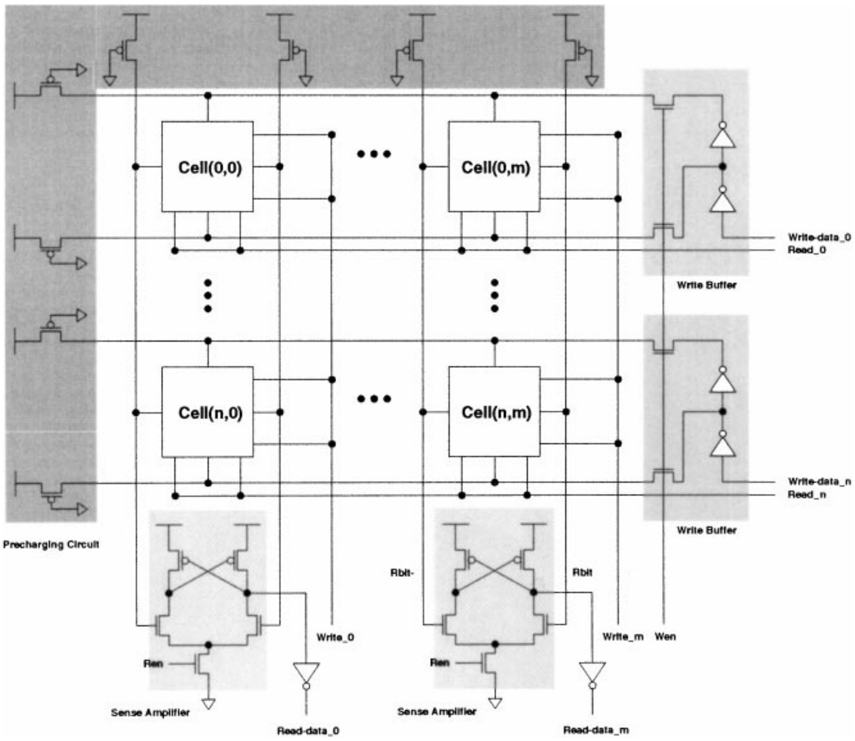


Figure 12. The core of the two-port SRAM circuits including memory cells, write buffers, sense amplifiers and pre-charging circuits.

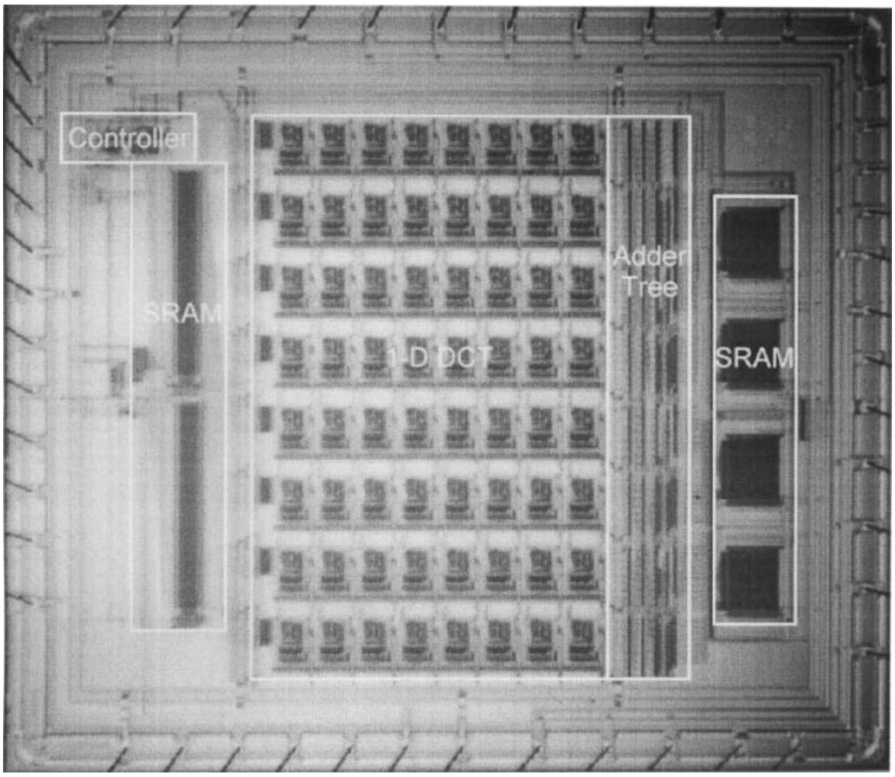


Figure 13. The photomicrograph of the whole chip including I/O pads.

Table 2. Chip characteristics.

Internal word-length	16 bits
Technology	0.6 μm CMOS SPDM
No. of transistors	152017
Core size	7.85 mm × 6.45 mm
Die size	8.98 mm × 7.79 mm
Clock rate	100 MHz
Latency	198 cycles
Block size	8 × 8
Supply voltage	2.0 V
Power	138 mW

Table 3. Simulated power dissipation by module.

Module	Counts	Power (mW)	Percentage(%)
Registers	2923	35.38	25.64
Clock buffers	1	29.35	21.27
SRAM 32×16	4	21.76	15.77
ROM	64	17.51	12.69
13-bit adder	64	15.48	11.22
SRAM 64 × 9	2	11.08	8.03
1-bit ALU	320	5.81	4.21
Controller	1	1.27	<0.92

V. Chip Performance and Specifications

By incorporating the module circuits discussed above, a low power 2-D DCT chip with direct method is designed and implemented. Figure 13 shows the photomicrograph of the whole system chip. The number of transistor is 152017, the die size is about 50 mm^2 (7.85 mm × 6.45 mm) and the latency takes 198 cycles. The core characteristics are summarized in Table 2.

In order to understand more details about the power distribution in the designed chip, a power simulation at 100 MHz for major components is shown in Table 3. It is obvious that registers consume most power. Excluding the clock buffers, the runner up are memory modules. Hence, reducing the power consumption of registers and memories can contribute more effective the proposed chip. This concludes that the circuit-level power-optimization should be mainly applied on registers and memories design to achieve the maximum power reduction.

There were several 2-D DCT implementations presented. The implementation in [7] and the product presented in [11] are not dedicated to a low power design. The chip reported by [12], which required variable threshold-voltage scheme by controlling back-bias voltage and better technology achieved a 10 mW 2-D DCT core processor. The main features of these chip implementations are summarized in Table 4. The chip reported by [12] performs better than ours in power consumption and chip area, however, it may much depend on better process technology. Another work, which reported on [25], presents a different design strategy for DCT, that is, a DCT generator. The motivation of such work is to design a reusable 1-D DCT core for different applications in points of flexibility. But, this work may not have efficient power consideration and the output latency is large, too.

The maximum frequency simulated on the chip is 133 MHz. It meets the requirement of the real-time HDTV signal processing for the chrominance format 4 : 2 : 0 and 4 : 2 : 2. The power simulated is 138 mW at 100 MHz by 0.6 μm single-poly double-metal technology.

VI. Conclusion

A low-power high-performance 2-D DCT architecture is proposed, and its circuit implementation is presented. The design features that contribute most to this result are summarized as follows. First, the usage of the direct 2-D DCT algorithm reduces the 2-D DCT into 1-D DCT and some additions. Second, a fast algorithm of

Table 4. Processor comparison.

Authors	Tech.	Core area	Trans.	Voltage	Clock rate	Power
Slawecki et al. [7]	2 μm	72.68 mm^2	67929	5 V	50 MHz	1 W
SGS-THOMSON [11]	–	–	–	5 V	20 MHz	1.5 W
Kuroda et al. [12]	0.3 μm	4 mm^2	120000	0.9 V	150 MHz	10 mW
Hunter et al. [25]	0.35 μm	2.5 mm^2	–	–	100 MHz	1.128 W([a])
Our chip	0.6 μm	50.6 mm^2	152017	2 V	100 MHz	138 mW

[a] Note: The power consumption of storage ram is not included.

1-D DCT with low computational complexity is employed. Furthermore, a parallel distributed algorithm (DA) architecture with the direct 2-D DCT approach is proposed. The address transition detection (ATD) circuit to employ self-timing approach, a power-saving ROM with dynamic circuits is implemented. It consumes power that is approximately one seventh of the power of static pseudo-nMOS. Moreover, two-port SRAMs approach for data re-ordering consumes less power than shift-registers. Though both the reductions of computation in architecture design and the circuit implementation consideration, a low power 2-D DCT chip is proposed and implemented. The simulation results shows it will be suitable for most portable video applications.

References

1. N. Ahmed, T. Natarajan, and K.R. Rao, "Discrete Cosine Transform," *IEEE Trans. Commu.*, vol. COM-23, 1974, pp. 90–93.
2. ISO/IEC, "Digital Compression and Coding of Continuous-Tone Still Images," Int. Standard DIS 10918.
3. Didier Le Gall, "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM*, vol. 34, no. 4, 1991, pp. 46–58.
4. ISO/IEC/JTC1/SC29/WG11. Draft CD 13818-2, Recommendation H.262, Committee Draft.
5. CCITT Study Group XV, TD35, "Draft Review of Recommendation H.261 Video Codec for Audiovisual Services at p×64 kbits/s," *Image Communication*, August 1990, pp. 221–239.
6. "Video Coding for Narrow Telecommunication Channels at < 64 kbits/s," Draft ITU-T Recommendation H.263, July 1995.
7. D. Slawecki and W. Li, "DCT/IDCT Processor Design for High Data Rate Image Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2,1992, pp. 135–146.
8. S. Uramoto, Y. Inoue, A. Takabetake, J. Takeda, Y. Yamashita, H. Terane, and M. Yoshimoto, "A 100 MHz 2-D Discrete Cosine Transform Core Processor," *IEEE J. Solid-State Circuits*, vol. 27, 1992, pp. 492–499.
9. T. Miyazaki, T. Nishitani, M. Edahiro, I. Ono, and K. Mitsuhashi, "DCT/IDCT Processor for HDTV Developed with DSP Silicon Compiler," *J. VLSI Signal Processing*, vol. 5, 1993, pp. 39–46.
10. A. Madisetti and A.N. Willson, "A 100 MHz 2-D 8×8 DCT/IDCT Processor for HDTV Applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, 1995, pp. 158–165.
11. SGS-THOMSON Microelectronics, "2-D Discrete Cosine Transform Image Processor," Product No. IMSA121.
12. T. Kuroda, T. Fujita, S. Mita, T. Nagamatsu, S. Yoshioka, K. Suzuki, F. Sano, M. Norishima, M. Murota, M. Kako, M. Kinugawa, M. Kakuma and T. Sakurai, "A 0.9 V, 150 MHz, 10 mV, 4 mm^2, 2-D Discrete Cosine Transform Core Processor with Variable Threshold-Voltage (VT) Scheme," *IEEE J. Solid-State Circuits*, vol. 31, no. 11, 1996, pp. 1770–1779.
13. P. Duhamel and C. Guillemot, "Polynomial transform computation of 2-D DCT," in *Proc. ICASSP'*90, April 1990, pp. 1515–1518.
14. M. Vetterli, "Fast 2-D Discrete Cosine Transform," in *Proc. ICASSP'85*, March 1995.
15. N.I. Cho and S.U. Lee, "Fast Algorithm and Implementation of 2-D Discrete Cosine Transform," *IEEE Trans. Circuits Syst.*, vol. CAS-38, 1991, pp. 297–305.
16. N.I. Cho, I.Y. Dong, and S.U. Lee, "On the Regular Structure for the Fast 2-D DCT Algorithm," *IEEE Trans. Circuits Syst. -II*, vol. CAS-40, 1993, pp. 259–266.
17. V. Srinvasan and K.J.R. Liu, "VLSI Design of High-Speed Time-Recursive 2-D DCT/IDCT Processor for Video Application," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, 1996, pp. 87–96.
18. Y.P. Lee, T.H. Chen, M.J. Chen, and C.W. Ku, "A Cost-Effective Architecture for 8×8 2-DDCT/IDCT Using Direct Method," *IEEE Trans. Circuits Syst. Video Technol.*, vol 7. no 3, 1997, pp. 459–467.
19. J.H. Hsiao, L.G. Chen, T.D. Chiueh, and C.T. Chen, "High Throughput CORDIC-Based Systolic Array Design for the Discrete Cosine Transform," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 5, 1995, pp. 218–224.
20. S.A. White, "Applications of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review," *IEEE ASSP Magazine*, vol. 6, no. 3, 1989, pp. 4–19.
21. J.M. Rabaey, *Digital Integrated Circuits—A Design Perspective*, Upper Saddle River, NJ: Prentice Hall. International Editions, 1995.
22. N.H.E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design—Systems Perspective*, 2nd edn., Reading, MA: Addison Wesley, 1993.
23. J. Yuan and C. Svensson, "High-Speed CMOS Circuit Technique," *IEEE J. Solid-State Circuits*, vol. 23, no. 1, 1989, pp. 2–70.
24. C. Nagendra, M.J. Irwin, and R.M. Owens, "Area-Time-Power Tradeoffs in Parallel Adders," *IEEE Trans. Circuits Syst. -II*, vol. 43, no. 10, 1996, pp. 689–702.
25. J. Hunter and J.V. McCanny, "Discrete Consine Transform Generator for VLSI Synthesis," in *Proc. of IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Seattle, May 1998.

Hao-Chieh Chang (S'96) was born in Taichung, Taiwan, in 1972. He received the B.S. degree in electronics engineering from National Chiao Tung University, Hsinchu, in 1994. He is currently a Ph.D. candidate in electrical engineering from National Taiwan University. His current research interests include video and image coding system, DSP architecture design, video signal processor design, and VLSI signal processing.
howard@video.ee.ntu.edu.tw

Jiun-Ying Jiu was born in Taoyuan, Taiwan, in 1972. He received the B.S. degree in electronics engineering from National Taiwan University, Taipei, in 1995. Also, he gained M.S degree in electrical engineering from National Taiwan University, in 1997. He has been a employee of Vivotek, which provides innovative internet appliance solutions, in Taiwan. His focus is on digital video processing, RTOS porting and DSP programming.
pony@vivotek.com

Li-Lin Chen was born in Tainan, Taiwan, in 1966. She received the B.S. degree from National Chiao Tung University, Taiwan and the M.S. degree from Case Western Reserve University, USA. She was an assistant engineer in OES/ITRI, Taiwan from 1987 to 1989. She has been an lecturer in Oriental Institute of Technology since 1991. Currently, she is a Ph.D. candidate of National Taiwan University. Her research interests include VLSI circuit design and video signal processor.
lilin@video.ee.ntu.edu.tw

Liang-Gee Chen received the BS, MS, and Ph.D degrees in Electrical Engineering from National Cheng Kung University, in 1979, 1981, and 1986, respectively. He was an Associate Professor (1986 – 1988) in the Department of Electrical Engineering, National Cheng Kung University. From 1988, he joined the Department of Electrical Engineering, National Taiwan University. During 1993 to 1994 he was Visiting Consultant of DSP Research Department, AT&T Bell Lab, Murray Hill. At 1997, he was the visiting scholar of the Department of Electrical Engineering, University, of Washington, Seattle. Currently, he is Professor of National Taiwan University. His current research interests are DSP Architecture Design, Video processor design, and video coding system.

Dr. Chen was the general chairman of the 7th VLSI Design/CAD Symposium. He is also the general chairman of the 1999 IEEE Workshop on Signal Processing Systems: Design and Implementation. He serves as Associate Editor of IEEE Trans. On Circuits and Systems for Video Technology from June 1996 till now and the Associate Editor of IEEE Trans. VLSI Systems from January 1999. He was the Associate Editor of the Journal of Circuits, Systems, and Signal Processing from 1999. He served as the Guest Editor of Journal of Signal Processing Systems. Dr. Chen is a Senior member of IEEE. He is also a member of the honor society Phi Tan Phi. He received the Best Paper Award from ROC Computer Society in 1990 and 1994. From 1991 to 1999, he received Long-Term (Acer) Paper Awards annually. In 1992, he received the Best Paper Award of the 1992 Asia-Pacific conference on circuits and systems in VLSI design track. In 1993, he received the Annual Paper Award of Chinese Engineer Society. In 1996, he received the Out-standing Research Award from NSC, and the Dragon Excellence Award for Acer.
lgchen@video.ee.ntu.edu.tw