# COMPUTER-AIDED SYNTHESIS OF A BI-DIMENSIONAL DISCRETE COSINE TRANSFORM CHIP

*Vittorio Rampa*
*CNR - Politecnico di Milano*

*Giovanni De Micheli*
*CSL - Stanford University*

## Abstract

This paper describes the design of an integrated circuit implementing a Bi-dimensional Discrete Cosine Tranform (BDCT). Such a circuit may be used to remove redundancy of video information in low bit-rate transmission channels and to perform video compression for image storage and retrieval.

The chip architecture is motivated by the fact that the BDCT equations can be solved row-by-row and column-by-column by a simpler Monodimensional DCT (MDCT). Therefore the chip structure is partitioned into three stages: the first and the last one implement MDCTs while the second stage is a shared memory array.

The DCT design was achieved by means of the OLYMPUS synthesis system, an experimental suite of synthesis tools developed at Stanford University. A parametrized behavioral description of the monodimensional DCT operator was specified in a High-Level Description Language, HardwareC, in terms of concurrent process communicating through a shared medium. The circuit layout was synthesized automatically from this description.

## 1 Introduction.

We describe here the design of a video rate 8 × 8 Bi-dimensional Discrete Cosine Transform (BDCT) Chip achieved by means of a hardware synthesis system developed at Stanford University. Techniques based on BDCT are very important for video imaging coding. Typical applications are in video redundancy reduction, e.g. in video coding (for video telephony and teleconference) and in video compression for image storage and retrieval. For example, in a hybrid video coder, every image is divided into blocks of $N \times N$ picture elements (pixels) and it is compared, block by block, with a previously stored image. The difference between the two images is transformed by the BDCT operator, coded and broadcast to the receiving station where the image is reconstructed [2].

The most important feature of a real-time coder, is the ability to handle data at the maximum operative speed. For this reason it is appropriate to implement a dedicated VLSI chip for the BDCT [1]. Our design uses a fully pipelined structure and distributed computation to achieve maximal throughput and it is similar to the architecture proposed in [7] and [8].

In this paper, we consider a design methodology based on computer-aided synthesis tools for the BDCT chip. This methodology allows us to shorten the design cycle and to modularize the design by parametrizing the specifications. In this way, the description of chips implementing $N \times N$ BDCTs for arbitrary $N$ are readily obtained. We decided to synthesize, as an example, a BDCT chip for $N = 8$. The design of the BDCT chip has been achieved by describing the BDCT behavioral specifications in a specialized Hardware Description Language for synthesis, HardwareC, and by generating the chip layout by means of the OLYMPUS synthesis system developed at Stanford University.

## 2 The Bi-dimensional DCT and the chip architecture

### 2.1 The algorithm.

The Bi-dimensional Discrete Cosine Transform (BDCT) is a mapping:

$$BDCT \, : \, X \Rightarrow Y \tag{1}$$

where $X \epsilon \Re^{N \times N}$ and $Y \epsilon \Re^{N \times N}$ and $N$ is called the order of the transform. The BDCT [3] is defined by the orthogonal matrix $C \epsilon \Re^{N \times N}$ as:

$$Y = C^T X C \tag{2}$$

where:

$$C = \{c_{i,j}\} = \begin{cases} \sqrt{2/N} \cos [\frac{(2i-1)(j-1)\pi}{2N}] & \text{for } i = 1.....N \;\; j = 2.....N \\ \sqrt{1/N} & \text{for } j = 1 \end{cases} \tag{3}$$

Since:

$$Y = (X^T C)^T C \tag{4}$$

it is possible to divide the computation into 2 steps:

CH2692-2/89/0000-0220 $1.00 © 1989 IEEE

$$Z = X^T C \tag{5}$$

and

$$Y = Z^T C \tag{6}$$

If we consider matrix $X$ and $Y$ as the partitioned matrices:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \quad Y = \begin{bmatrix} y_1 & y_2 & \vdots & y_N \end{bmatrix} \tag{7}$$

and matrix $Z$ as:

$$Z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} = \begin{bmatrix} z_{.1} & z_{.2} & \vdots & z_{.N} \end{bmatrix} \tag{8}$$

we can write:

$$z_i = x_i^T C \tag{9}$$

and:

$$y_i = z_i^T C \tag{10}$$

Therefore the rows of $Z$ and $Y$ can be computed independently for each index $i$. The operation ( 9) and ( 10) are referred to as Monodimensional ($N$ x 1) DCT operations (MDCT$_u$, n=1,2) [3]. Unfortunately, the computation of ( 10) for any $i$ requires the $i^{th}$ column vector of $Z$ and therefore the two MDCT operations cannot be executed in parallel. This motivates the choice of the chip architecture, as described in the next section.

## 2.2 Chip architecture.

We consider here the architecture of a BDCT chip of order $N = 8$. A block diagram of the chip is shown in Fig. 1. We have partitioned our design into three stages: the first and the last compute the multiplication described in ( 9) and ( 10) and are referred to as MDCT$_n$, $n = 1,2$. The intermediate stage is a storage array for matrix $Z$ and it is implemented as a double threaded shift register (TMEM) to allow the transposition of matrix $Z$. The operations of the three stages are regulated by the control unit (CU).

In the VLSI implementation of the algorithm, each entry of the matrices of real numbers $X$, $Z$ and $Y$ is represented by IS$_1$, OS$_1$ = IS$_2$ and OS$_2$ bits respectively. For example, each input pixel $x_{h,k}$, for $h, k = 1, \ldots, N$, is represented in 2's complement by:

$$x_{h,k} = -x_{h,k}^{(IS_1-1)} 2^{IS_1-1} + \sum_{q=0}^{IS_1-2} x_{h,k}^{(q)} 2^q \tag{11}$$

The size of the bit representation of the data streams is chosen to optimize the signal to noise ratio as shown later.

The block diagram of the MDCT$_1$ is sketched in Fig. 2. The input register IR is used to convert the input pixel serial stream into the vector format. Consider for example the first MDCT. A column of input pixels, $x_{1i}$, $x_{2i}$, ..., $x_{Ni}$, $N = 8$, represented by IS$_1$ bits each, are shifted in. Then, starting from the LSB, one bit of each pixel is processed in parallel with the others by the sub-circuit called Serial Multiplier and Accumulator (SMA).

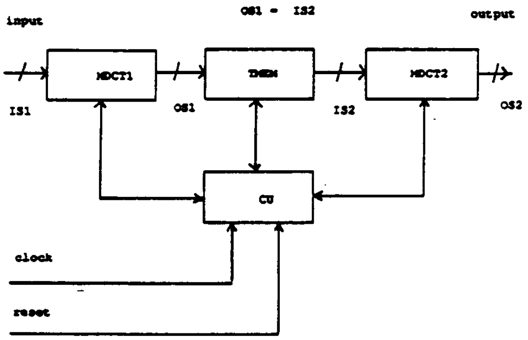

Figure 1: $N$ x $N$ BDCT decomposition into two $N$ x 1 MDCT

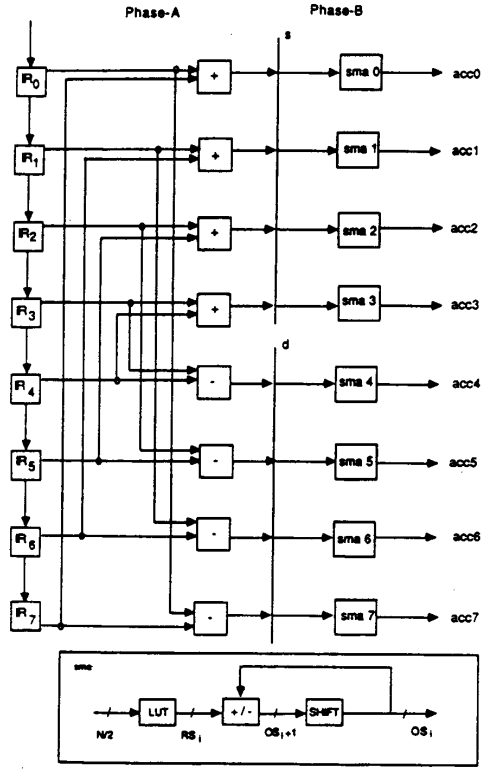

Figure 2: 8 x 1 DCT hardware structure

Since each element $z_{i,j}$ of the vector $z_{i.} = x_{.i}^T C$ is equal to:

$$z_{i,j} = \sum_{m=1}^{N} c_{m,j} x_{m,i} \tag{12}$$

then, by using the bit-wise representation ( 11):

$$\begin{aligned} z_{i,j} &= -\sum_{m=1}^{N} c_{m,j} x_{m,i}^{(IS_1-1)} 2^{IS_1-1} + \sum_{q=0}^{IS_1-2} \sum_{m=1}^{N} c_{m,j} x_{m,i}^{(q)} 2^q = \\ &= -A_j^{(IS_1-1)} 2^{IS_1-1} + \sum_{q=0}^{IS_1-2} A_j^{(q)} 2^q \end{aligned} \tag{13}$$

where $A_j^{(q)} = \sum_{m=1}^{N} c_{m,j} x_{m,i}^{(q)}$. The function represented by $A_j^{(q)}$ is memoryless and can be implemented by a multiple-level logic combinational circuit, denoted by Look-Up Table (LUT) in the circuit.

A more compact representation of the LUTs can be achieved by exploiting the symmetry of the cosine operator. Indeed we can write eq. (12) in the following form:

$$z_{i,j} = \sum_{m=1}^{N} c_{m,j} x_{m,i} = \begin{cases} \sum_{m=1}^{N/2} c_{m,j} u_{m,i} & j = 1, 3, \cdots, N-1 \\ \sum_{m=1}^{N/2} c_{m,j} v_{m,i} & j = 2, 4, \cdots, N \end{cases} \tag{14}$$

where:

$$u_{m,i} = x_{m,i} + x_{N-m+1,i} \quad i = 1, \cdots, N \quad m = 1, \cdots, N/2 \tag{15}$$

$$v_{m,i} = x_{m,i} - x_{N-m+1,i} \quad i = 1, \cdots, N \quad m = 1, \cdots, N/2 \tag{16}$$

Then the LUT equations can be written as:

$$A_j^{(q)} = \sum_{m=1}^{N} c_{m,j} x_{m,i}^{(q)} = \begin{cases} \sum_{m=1}^{N/2} c_{m,j} u_{m,i}^{(q)} & j = 1, 3, \cdots, N-1 \\ \sum_{m=1}^{N/2} c_{m,j} v_{m,i}^{(q)} & j = 2, 4, \cdots, N \end{cases} \tag{17}$$

In this case, we have reduced the number of inputs of the $N$ LUTs by a factor of two, which corresponds to a large saving of the corresponding logic, at the extra cost of $N/2$ sum and $N/2$ difference gates. The evaluation of (13) is performed by shifting and adding the quantity $A_j^{(q)}$ computed by the LUT.

The transposition of the matrix $Z$ is implemented by using a bi-directional shift-register (Fig. 3). The choice of an internal storage array versus an external RAM is mainly due to the speed of operation and to the number of pins required to interconnect the external RAM to the MDCT circuits.

The bidirectional shift-register works as follows. Every time a vector $z_{i.}$ has been computed and accumulated completely in the SMA, the vector $z_{i.}$ is shifted in the TMEM. After inserting $N = 8$ row vectors $z_{i.}$, the data is shifted in the orthogonal direction as column vectors $z_{.i}$ and passed to the second BDCT block.

The second MDCT description is similar in structure to the first one and can be obtained by replacing $z$ by $y$, $x$ by $z$ and the data stream size $IS_1$ by $IS_2$.

## 2.3 Determination of the data path parameters and circuit performances.

The determination of the size of the data path is an important aspect of the design because it affects the Signal to Noise Ratio (SNR) and it determines the chip area.

A program has been written in C to simulate the algorithm, with random input data, to determine the correct dimension of the internal registers, data-path and look-up table size. Figures 4.a and 4.b show the results of this analysis. In particular, it plots the Signal to Noise (SNR) ratio versus the size of the LUTs (RS$_1$-RS$_2$), the input (IS$_1$-IS$_2$) and output (OS$_1$-OS$_2$) data format (using rounding after each MDCT). Note that the SNR is very sensitive to the dimension (RS$_1$) of the look-up tables of the first DCT (MDCT$_1$) while the look up table dimension (RS$_2$) of the second DCT (MDCT$_2$) is less important.

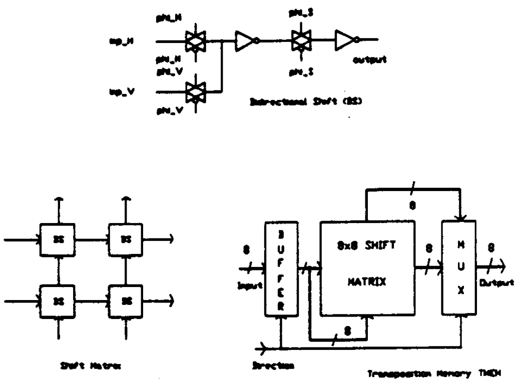

Figure 3: Bi-directional shift register

The following parameters (bit-size) have been chosen:

RS$_1$ = RS$_2$ = 11
IS$_1$ = 8
OS$_1$ = IS$_2$ = 11
OS$_2$ = 14

With this configuration, the $SNR$ is always greater than 52 dB.

The performance of this BDCT architecture can be evaluated as follows. The system clock period $T_c = 1/F_c$ is determined by the pixel period $T_p = 1/F_p$; the maximum number of cycles needed to compute a serial multiplication ($IS_2$) and the order $N$ of the DCT. Namely, the system clock frequency is:

$$F_c = F_p \lceil IS_2/N \rceil \tag{18}$$

For a pixel rate $F_p$ = 13.5 Mhz, the corresponding system clock frequency is $F_c$ = 27 Mhz, i.e. each combinational operation (e.g single add operation in the SMA circuit) is performed every $T_c$=37 ns. The complete calculations ( 9) and ( 10) are computed in:

$$T_v = T_p.N \tag{19}$$

The $T_v$ period is also the time necessary to shift a new set of data into the memory array TMEM and to retrieve the old ones. The total latency of the BDCT is equal to $10T_v$: the largest part ($8T_v$) is introduced by the memory array, while each MDCT stage adds a single $T_v$ period due to an internal pipeline register.

# 3 The DCT chip implementation.

## 3.1 Design Methodology

While most VLSI DSP chips [4] [5] [6] are designed by using predefined library macros to implement additions and multiplications, we have chosen a design methodology that merges the arithmetic logic, the control logic and the local storage registers into macro-cells. The

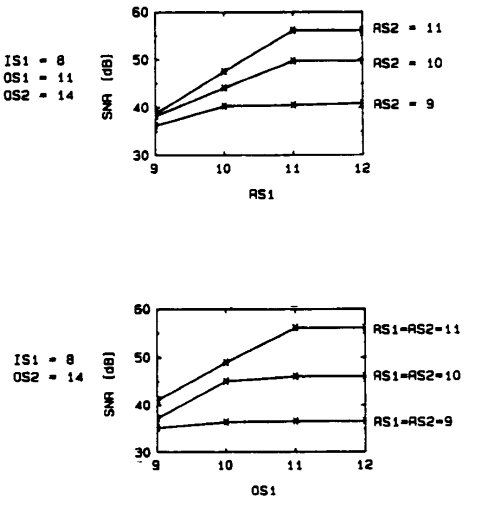

Figure 4: SNR vs. internal and external data size

macro-cells are highly optimized multiple-level logic circuits. For example, The LUTs are not implemented by logic arrays (e.g. ROMs or PLAs), but they consist of an interconnection of gates that optimizes the silicon usage under timing constraints. Their specification is derived automatically from the high-level system specifications of the BDCT chip. The design methodology is fully supported by a set of computer-aided design tools, namely:

1) a generator-of the layout of the macro-cells;
2) a logic optimizer;
3) a high-level synthesizer;
4) a set of simulators.

The physical layout of the chip was derived by interconnecting macro-cells, implementing each sub-circuit. Each macro-cell (excluding TMEM) was generated automatically by the Castor/Pollux programs [9]. The Castor program takes as input a logic description of the circuit in terms of combinational logic equations, clocked storage registers and interconnection nets and generates a symbolic layout of gates required by the implementation. The Pollux program transform the symbolic layout into a geometric layout by taking the design rules into account. It also arranges the combinational gates and the register cells into a rectangular array and interconnects them by channel routing. Pollux is designed to work in the frame of the GDT tool suite [10].

The set of logic equations describing the subcircuits was optimized by two logic synthesis tools. The former, called Minerva, manipulates hierarchically sequential-logic circuit descriptions [11]. The latter, MIS-II, determines the optimal structure of the combinational sub-circuits in as a multiple-level logic networks, where the optimality is measured is in terms of the number of transistors of the final implementation [12].

The hierarchical description of the logic circuit was derived automatically from a high-level description by program Hercules [13], which has the task of synthesizing logic structures, such as adders, from their behavioral representation. Hercules synthesizes also the control portion of the chip. The major advantages of using Hercules is the capability of entering a parametrized circuit description in terms of a hardware program, instead of a logic-level schematic and the possibility to use templates to describe parametrized functions such as the SMAs and LUTs.

The circuits was simulated, at the functional and circuit level using Lsim [10]. Macro-models were used for the functional simulation. The circuit and mixed-mode simulation was based on the description of the macro-cells in the *L* language, generated first by Pollux and then annotated with the extracted capacitance values. A logic level simulation was also done using the THOR simulator [14].

### 3.2 Hardware Description.

The BDCT circuit was described in HardwareC [13] in terms of concurrent processes that communicate through a shared medium, consisting of a set of interconnections and a shared memory (TMEM). The shared memory was not described in the high-level representation. Instead, an appropriate module generator was designed to construct the layout of TMEM. Similarly, the MDCT circuits require a shift register for serial/parallel data conversion at the input (IR) and at the output (OR). These circuits were also constructed by module generators. The BDCT circuit consists of four major building blocks: $\mathrm{MDCT}_1$, $\mathrm{MDCT}_2$, CU and TMEM. Note that the description of the two MDCT circuits is identical, except for the parameters.

A full description of the $\mathrm{MDCT}_1$ in Hardware C is reported in [15]. It consists of two concurrent processes (*phase-A* and *phase-B*). Process *phase-A* computes the additions (15) and subtractions (16) in parallel. The output of the process *phase-A* is fed to process *phase-B*, that performs in parallel the $N$ serial multiplications (SMA).

As a final remark, the high-level description allowed us to express both structural and behavioral constructs. The major partition of the MDCTs into blocks is explicitly defined in order to exploit the structure of the problem. Behavioral description of sub-modules, such as the SMAs, allows us to synthesize circuits implementing equation ( 13) as efficiently as possible, because the final implementation does not distinguish between the addition logic and the remaining logic used to store the coefficients $A_j^{(q)}$. A routine describing a 11-bit adder was explicitly used to achieve a fast adder implementation. Alternatively an implicit adder specification by means of the + sign could have been used, resulting in a 11-bit ripple-carry adder logic.

### 3.3 The chip.

The BDCT chip was designed using a $2.0\mu$ technology by AMS. The core size is 6.9 by 7.6 mm and the die size 9.5 by 9.2 mm. There are 20 macrocells, totaling to 8904 gates, excluding the I/O pads. There are 12 input, 8 output and 14 tristate bidirectional pads. There are 8 power/ground pads. The chip has been designed to operate at a frequency $F_p$ = 13.5 MHz, to be compatible with standard data communication rates. Higher rates can be sustained.

A detail of the implementation of process *phase-A* is shown in Fig. 5 as a macro-cell arranged in 12 rows of procedurally-generated cells. The final chip layout is shown in Fig. 6.

| Stage | Cell name | size | rows | transistor | cells |
|---|---|---|---|---|---|
| $\mathrm{MDCT}_1$ | *phase-A* | $900 \times 1250\mu^2$ | 12 | 832 | 352 |
| | *phase-B* | $2900 \times 3700\mu^2$ | 21 | 4634 | 2105 |
| $\mathrm{MDCT}_2$ | *phase-A* | $900 \times 1250\mu^2$ | 12 | 832 | 352 |
| | *phase-B* | $3150 \times 3850\mu^2$ | 21 | 6050 | 2213 |
| CONTROL | *control* | $1100 \times 1050\mu^2$ | 12 | 954 | 392 |

Table 1: Macro-cells layout statistics

## 4 Concluding remarks.

The implementation of a 8 x 8 BDCT VLSI chip has been described, featuring:

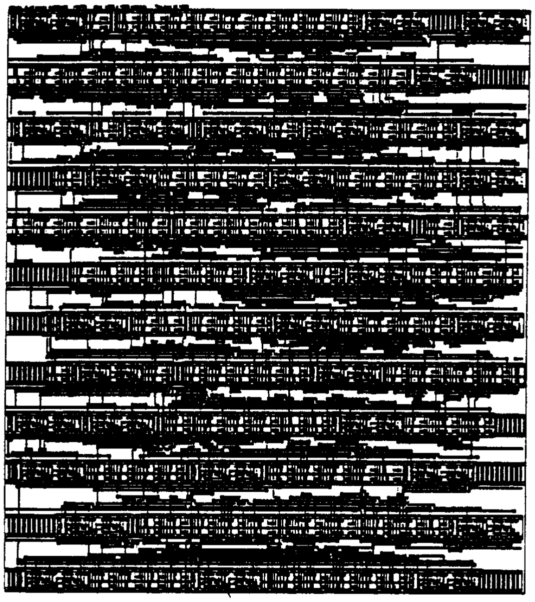

Figure 5: *phase-A* macro-cell layout

1) a VLSI distributed architecture with no multipliers but only registers, adders and shifters;
2) a macro-cell based design where regular storage arrays are constructed by module generators and sequential logic circuits are synthesized into macro-cells by means of automated tools from a parametrized high-level description;
3) a multiple-level logic implementation that is optimized in terms of chip area and delay.

The chip was built by a vertical integrated synthesis system. An important aspect is the portability of the design across technologies. The symbolic layout of the macro-cells is fairly technology independent and its mapping to geometric layout supports minor process changes and/or upgrades. In addition, the same BDCT architecture description in HardwareC can be mapped into other design styles, such as standard-cells or sea-of-gates, in a radiation-hardened process if required for satellite applications. This is of great importance in Digital Signal Processing applications, because fast prototyping and final implementations that match the fabrication technology can be derived from a single high-level description.

## 5 Acknowledgements

This research has been supperted by a NSF Presidential Investigation Award and by a seed grant of the Center for Integrated Systems at Stanford University..

## References

[1] S. Brofferio, N. Dal Degan and V. Rampa, "Requirements for Visual Communications Signal Processors", *Proceedings of GLOBECOM*, Tokyo, Vol. 1, pp. 447-452, 1987.

[2] C. Cafforio, F. Rocca, "Method for Measuring Small Displacement of Television Images", *IEEE Transaction on Information Theory*, Vol. IT-22, pp. 573-579, September 1976.

[3] N. Ahmed, T. Natarajan and K.R. Rao, "Discrete Cosine Transform", *IEEE Transactions on Computers*, Vol. C-23, pp. 90-93, January 1974.

[4] W.H. Chen, "A fast computational algorithm for the discrete Cosine Transform", *IEEE Transaction on Communication*, Vol. COM-25, pp. 1004-1009, September 1977.

[5] F.Jutland et al., "A single chip video rate 16x16 discrete cosine transform", *Proceedings of ICASSP*, Tokyo, Vol. 1, pp. 805-808, 1986..

[6] M.Vetterli and A.Ligtenberg, "A discrete Fourier-cosine transform chip", *IEEE Journal Sel. Areas in Communications*, Vol. SAC-4, pp.49-61, January 1986.

[7] M.T. Sun, L. Wu and M.L. Liou, "A concurrent Architecture for VLSI Implementation of Discrete Cosine Transform", *IEEE Transaction on Circuits and Systems*, Vol. CAS-34, August 1987.

[8] M.T.Sun, T.C. Chen and A.M. Gottlieb, "VLSI Implementation of a 16x16 DCT", *Proceedings of ICASSP*, New York, Vol. 1, pp. 1973-1976, 1988.

[9] F. Mailhot, G. De Micheli "Automatic Layout and optimization of static CMOS cells", *Proceedings of ICCD*, Rye, N.Y., pp. 180-185, 1988.

[10] Silicon Compiler System *GDT Database and Language Tools Reference*, version 3.2, May 1988.

[11] G. De Micheli, T.Klein "Algorithms for Sequential Logic Synthesis" *Proceedings of ISCAS*, Portland, May 1989.

[12] R. Brayton, R. Rudell, A. Sangiovanni-Vincentelli, A. Wang "MIS: A Multiple-Level Logic Optimization System", *IEEE Transactions on CAD*, Vol. CAD-6, No. 6, November 1987, pp. 1062-1081.

[13] G. De Micheli, D. Ku "HERCULES - A system for High- Level Synthesis", *Proceedings of DAC*, Anaheim, pp. 483-488, June 1988.

[14] R. Alverson, T. Blank, K. Choi, S. Y. Hwang, A. Saltz, L. Soule, T. Rokicki "THOR User's Manual: tutorial and commands", *Technical Report*: CSL-TR-88-348, Computer System Laboratory, Stanford University, January 1988.

[15] V.Rampa and G.De Micheli, "Computer-Aided Synthesis of a Bi-dimensional Discrete Cosine Transform Chip" *Technical Report*: CSL-TR-88-363, Stanford University, August 1988.

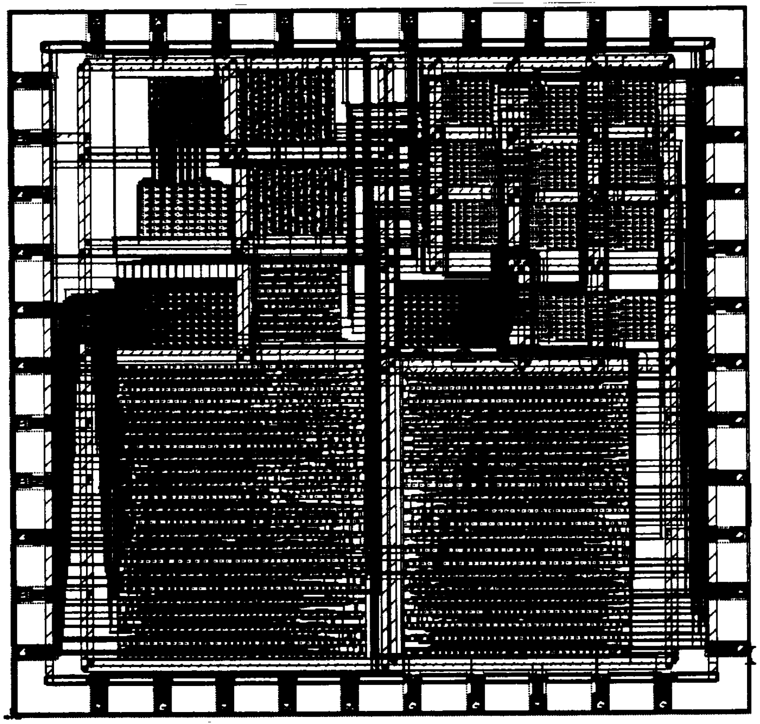

Figure 6: Layout of the BDCT chip