



XAPP611 (v1.2.1) April 5, 2007

## Video Decompression Using IDCT

Author: Latha Pillai

### Summary

This application note describes a two-dimensional Inverse Discrete Cosine Transform (2D IDCT) function implemented on a Xilinx FPGA. The reference design file provides behavioral code for implementation on any Xilinx device. Some advantages of the module include the ability to parameterize the IDCT function and to guarantee performance. *The code can be further optimized by instantiating embedded adders and multipliers when targeting the Virtex™-II series of FPGAs.* After an initial latency of 84 clock cycles, one IDCT value is output at every clock. A "Video Compression Using DCT" application note ([XAPP610](#)) covers DCT compression.

### Decompression

Compression is the process of reducing the size of the data sent, thereby, reducing the bandwidth required for the digital representation of a signal. Compression technology can result in reduced transmission time due to less data being transmitted. It also decreases the storage requirements because there is less data. The data compressed at the transmitter needs to be decompressed at the receiver. IDCT is used to decompress DCT compressed data in the decoder. DCT/IDCT are two of the most computation intensive functions in compression. Therefore, a fast and optimized DCT/IDCT implementation is essential in improving the performance of the video coder and decoder. [Figure 1](#) is a block diagram of a compression/decompression system.

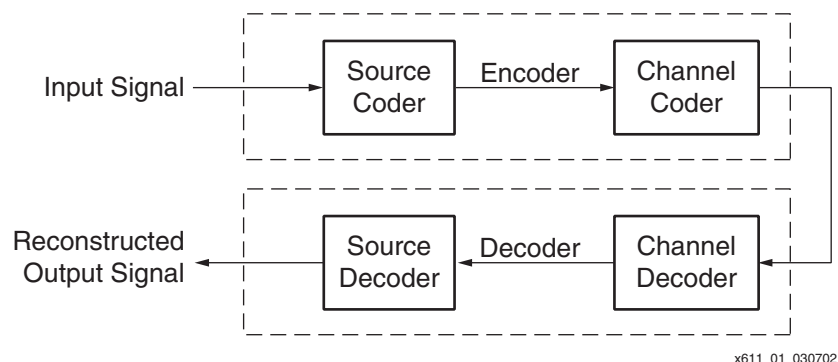


Figure 1: Block Diagram of a Compression/Decompression System<sup>[1]</sup>

### IDCT Compression

#### Technical Aspects of the Compression Algorithm

The following material provides a brief description of IDCT. Further details can be found in "Image and Video Compression Standards" by Vasudev Bhaskaran and Konstantinos Konstantinides<sup>[1]</sup>. The 16-bit signed input provides 12-bit signed out pixels for the IDCT. In the 16 bits, disregarding the sign bit, there are 15 bits of data including the fraction bit. Doing an IDCT divides these 15 bits of data by eight (represented by 3 bits) resulting in 15 – 3 or 12 bits. Since the IDCT is a perfect inverse of DCT, by starting with an integer, an integer must be received back. From the 12 bits, the one fraction bit is ignored and the sign bit is added back

© 2002-2007 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

giving a final 12-bit result. The algorithm used for the calculation of the 2D IDCT coefficients is based on the following equation:

$$XC_{pq} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} XN_{mn} \cdot \frac{c(p)c(q)}{4} \cdot \cos \frac{\pi(2m+1)p}{2M} \cdot \cos \frac{\pi(2n+1)q}{2N} \quad (\text{EQ 1})$$

First, the 1D IDCT of the rows are calculated and then the 1D DCT/IDCT of the columns are calculated. The 1D DCT coefficients for the rows and columns can be calculated by separating equation 1 into the row part and the column part.

$$C = K \cdot \cos \frac{(2 \cdot \text{col number} + 1) \cdot \text{row number} \cdot \pi}{2 \cdot M} \quad (\text{EQ 2}), \text{ where}$$

$$K = \frac{\sqrt{1}}{N} \quad \text{for row} = 0$$

$$K = \frac{\sqrt{2}}{N} \quad \text{for row} \neq 0$$

$$C^t = K \cdot \cos \frac{(2 \cdot \text{row number} + 1) \cdot \text{col number} \cdot \pi}{2 \cdot N} \quad (\text{EQ 3}), \text{ where}$$

$$K = \frac{\sqrt{1}}{M} \quad \text{for col} = 0$$

$$K = \frac{\sqrt{2}}{M} \quad \text{for col} \neq 0$$

and M = total number of columns, N = total number of rows.

The constant values for C and C<sup>t</sup> calculated from equations 2 and 3 are as follows:

$$C = \begin{bmatrix} 23170 & 23170 & 23170 & 23170 & 23170 & 23170 & 23170 & 23170 \\ 32138 & 27246 & 18205 & 6393 & -6393 & -18205 & -27246 & -32138 \\ 30274 & 12540 & -12540 & -30274 & -30274 & -12540 & 12540 & 30274 \\ 27246 & -6393 & -32138 & -18205 & 18205 & 32138 & 6393 & -27246 \\ 23170 & -23170 & -23170 & 23170 & 23170 & -23170 & -23170 & 23170 \\ 18205 & -32138 & 6393 & 27246 & -27246 & -6393 & 32138 & -18205 \\ 12540 & -30274 & 30274 & -12540 & -12540 & 30274 & -30274 & 12540 \\ 6393 & -18205 & 27246 & -32138 & 32138 & -27246 & 18205 & -6393 \end{bmatrix}$$

$$C^t = \begin{bmatrix} 23170 & 32138 & 30274 & 27246 & 23170 & 18205 & 12540 & 6393 \\ 23170 & 27246 & 12540 & -6393 & -23170 & -32138 & -30274 & -18205 \\ 23170 & 18205 & -12540 & -32138 & -23170 & 6393 & 30274 & 27246 \\ 23170 & 6393 & -30274 & -18205 & 23170 & 27246 & -12540 & -32138 \\ 23170 & -6393 & -30274 & 18205 & 23170 & -27246 & -12540 & 32138 \\ 23170 & -18205 & -12540 & 32138 & -23170 & -6393 & 30274 & -27246 \\ 23170 & -27246 & 12540 & 6393 & -23170 & 32138 & -30274 & 18205 \\ 23170 & -32138 & 30274 & -27246 & 23170 & -18205 & 12540 & -6393 \end{bmatrix}$$

A one-dimensional 8-point IDCT followed by an internal double buffer memory, followed by another one-dimensional 8-point IDCT provides the 2D IDCT architecture.

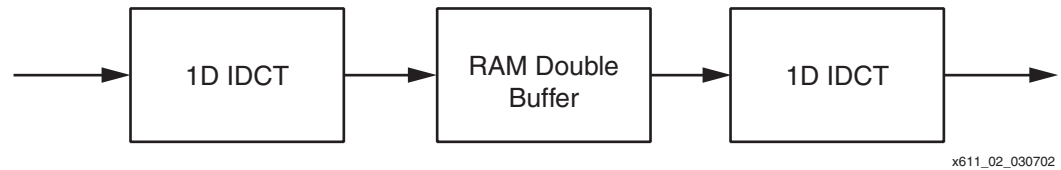


Figure 2: 2D-IDCT Using Vector Processing

Vector processing using parallel multipliers is a method used for implementation of IDCT. The advantages in the vector processing method are regular structure, simple control and interconnect, and good balance between performance and complexity of implementation.

## Behavioral Model

Using vector processing, the output  $Y$  of an  $8 \times 8$  IDCT for input  $X$  is given by  $Y = C^t \cdot X \cdot C$ , where  $C$  is the cosine coefficients and  $C^t$  are the transpose coefficients. This equation can also be written as  $Y = C^t \cdot Z$ , where  $Z = X \cdot C$ . Using the cosine  $C$  and inverse cosine  $C^t$  numbers in equation 2 and 3, the intermediate value  $Z = X \cdot C$  is calculated as follows:

$$X = \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} & x_{07} \\ x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} & x_{17} \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} & x_{27} \\ x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} & x_{37} \\ x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} & x_{47} \\ x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} & x_{57} \\ x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} & x_{67} \\ x_{70} & x_{71} & x_{72} & x_{73} & x_{74} & x_{75} & x_{76} & x_{77} \end{bmatrix} \quad C = \begin{bmatrix} 23170 & 23170 & 23170 & 23170 & 23170 & 23170 & 23170 & 23170 \\ 32138 & 27246 & 18205 & 6393 & -6393 & -18205 & -27246 & -32138 \\ 30274 & 12540 & -12540 & -30274 & -30274 & -12540 & 12540 & 30274 \\ 27246 & -6393 & -32138 & -18205 & 18205 & 32138 & 6393 & -27246 \\ 23170 & -23170 & -23170 & 23170 & 23170 & -23170 & -23170 & 23170 \\ 18205 & -32138 & 6393 & 27246 & -27246 & -6393 & 32138 & -18205 \\ 12540 & -30274 & 30274 & -12540 & -12540 & 30274 & -30274 & 12540 \\ 6393 & -18205 & 27246 & -32138 & 32138 & -27246 & 18205 & -6393 \end{bmatrix}$$

$$Z_{(0,0)} = 23170x_{00} + 32138x_{01} + 30274x_{02} + 27246x_{03} + 23170x_{04} + 18205x_{05} + 12540x_{06} + 6393x_{07}$$

$$Z_{(0,1)} = 23170x_{00} + 27246x_{01} + 12540x_{02} - 6393x_{03} - 23170x_{04} - 32138x_{05} - 30274x_{06} - 18205x_{07}$$

$$Z_{(0,2)} = 23170x_{00} + 18205x_{01} - 12540x_{02} - 32138x_{03} - 23170x_{04} + 6393x_{05} + 30274x_{06} + 27246x_{07}$$

$$Z_{(0,3)} = 23170x_{00} + 6393x_{01} - 30274x_{02} - 18205x_{03} + 23170x_{04} + 27246x_{05} - 12540x_{06} - 32138x_{07}$$

$$Z_{(0,4)} = 23170x_{00} - 6393x_{01} - 30274x_{02} + 18205x_{03} + 23170x_{04} - 27246x_{05} - 12540x_{06} + 32138x_{07}$$

$$Z_{(0,5)} = 23170x_{00} - 18205x_{01} - 12540x_{02} + 32138x_{03} - 23170x_{04} - 6393x_{05} + 30274x_{06} - 27246x_{07}$$

$$Z_{(0,6)} = 23170x_{00} - 27246x_{01} + 12540x_{02} + 6393x_{03} - 23170x_{04} + 32138x_{05} - 30274x_{06} + 18205x_{07}$$

$$Z_{(0,7)} = 23170x_{00} - 32138x_{01} + 30274x_{02} - 27246x_{03} + 23170x_{04} - 18205x_{05} + 12540x_{06} - 6393x_{07}$$

Or:

$$Z_{(k,0)} = (23170x_{k0} + 30274x_{k2} + 23170x_{k4} + 12540x_{k6}) + (32138x_{k1} + 27246x_{k3} + 18205x_{k5} + 6393x_{k7}) \\ = P01 + P02$$

$$Z_{(k,1)} = (23170x_{k0} + 12540x_{k2} - 23170x_{k4} - 30274x_{k6}) + (27246x_{k1} - 6393x_{k3} - 32138x_{k5} - 18205x_{k7}) \\ = P11 + P12$$

$$Z_{(k,2)} = (23170x_{k0} - 12540x_{k2} - 23170x_{k4} + 30274x_{k6}) + (18205x_{k1} - 32138x_{k3} + 6393x_{k5} + 27246x_{k7}) \\ = P21 + P22$$

$$Z_{(k,3)} = (23170x_{k0} - 30274x_{k2} + 23170x_{k4} - 12540x_{k6}) + (6393x_{k1} - 18205x_{k3} + 27246x_{k5} - 32138x_{k7}) \\ = P31 + P32$$

$$Z_{(k,4)} = P31 - P32$$

$$Z_{(k,5)} = P21 - P22$$

$$Z_{(k,6)} = P11 - P12$$

$$Z_{(k,7)} = P01 - P02$$

where  $k = 0, 2, \dots, 7$ .

Each element in the first row of the input matrix X is multiplied by each element in the first column of matrix C and added together to get the first value  $Z_{00}$  of the intermediate matrix Z. To get  $Z_{01}$ , each element of row zero in X is multiplied by each element in the first column of C and added and so on. As shown, input  $X_{00}$  gets multiplied by all the coefficients in the first row of C, and input  $X_{01}$  gets multiplied by all the coefficients in the second row of C, and so on.

$$\begin{array}{l}
 X_{00} \rightarrow \\
 X_{01} \rightarrow \\
 X_{02} \rightarrow \\
 X_{03} \rightarrow \\
 X_{04} \rightarrow \\
 X_{05} \rightarrow \\
 X_{06} \rightarrow \\
 X_{07} \rightarrow
 \end{array}
 \left[ \begin{array}{cccccccc}
 23170 & 23170 & 23170 & 23170 & 23170 & 23170 & 23170 & 23170 \\
 32138 & 27246 & 18205 & 6393 & -6393 & -18205 & -27246 & -32138 \\
 30274 & 12540 & -12540 & -30274 & -30274 & -12540 & 12540 & 30274 \\
 27246 & -6393 & -32138 & -18205 & 18205 & 32138 & 6393 & -27246 \\
 23170 & -23170 & -23170 & 23170 & 23170 & -23170 & -23170 & 23170 \\
 18205 & -32138 & 6393 & 27246 & -27246 & -6393 & 32138 & -18205 \\
 12540 & -30274 & 30274 & -12540 & -12540 & 30274 & -30274 & 12540 \\
 6393 & -18205 & 27246 & -32138 & 32138 & -27246 & 18205 & -6393
 \end{array} \right]$$

The matrix above is implemented by using eight multipliers and storing the coefficients in ROMs. At the first clock, the eight inputs  $x_{00}$  to  $x_{07}$  are multiplied by the eight values in column one, resulting in eight products ( $P_{00_0}$  to  $P_{00_7}$ ). At the eighth clock, the eight inputs are multiplied by the eight values in column eight resulting in eight products ( $P_{07_0}$  to  $P_{07_7}$ ). From the equations for Z, the intermediate values for the first row of Z are computed:

$$\begin{aligned}
 Z_{00} &= P_{00_0} + P_{00_1} + P_{00_2} + P_{00_3} + P_{00_4} + P_{00_5} + P_{00_6} + P_{00_7} \\
 Z_{01} &= P_{01_0} + P_{01_1} + P_{01_2} + P_{01_3} + P_{01_4} + P_{01_5} + P_{01_6} + P_{01_7} \\
 Z_{ij} &= P_{ij_0} + P_{ij_1} + P_{ij_2} + P_{ij_3} + P_{ij_4} + P_{ij_5} + P_{ij_6} + P_{ij_7}
 \end{aligned}$$

Where the matrix X row number  $i = 0$  to  $7$ , and the matrix C column number  $j = 0$  to  $7$ . The intermediate values for the second row of Z,  $Z_{1K}$ , are computed by using  $P_{1K}$  values. The  $8 \times 8$  Z matrix can be calculated using these equations. The block diagram for the implementation of the 1D-IDCT is shown in [Figure 3](#).

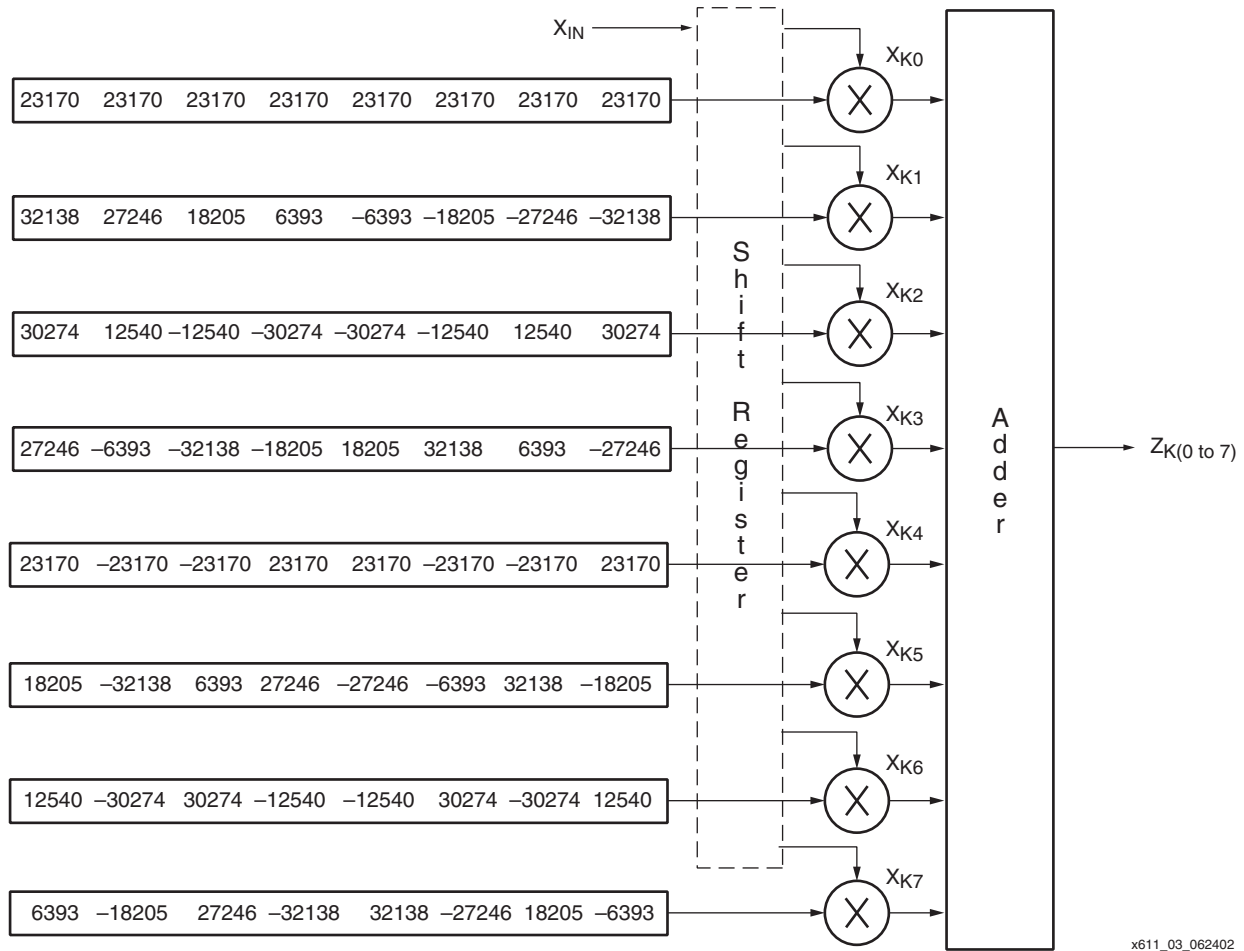


Figure 3: 1D-IDCT Implementation (K is 0 - 7)

The values for  $Z_{0(0-7)}$  can be calculated in eight clock cycles. If the adder output is also registered,  $Z_{07}$  is calculated at the ninth clock cycle. All 64 values of Z are calculated in 64 clock cycles and then the process is repeated. The values of Z correspond to the 1D-IDCT of the input X. Once the Z values are calculated, the 2D-IDCT function Y can be calculated from  $Y = C^t Z$ .

$$C^t = \begin{bmatrix} 23170 & 32138 & 30274 & 27246 & 23170 & 18205 & 12540 & 6393 \\ 23170 & 27246 & 12540 & -6393 & -23170 & -32138 & -30274 & -18205 \\ 23170 & 18205 & -12540 & -32138 & -23170 & 6393 & 30274 & 27246 \\ 23170 & 6393 & -30274 & -18205 & 23170 & 27246 & -12540 & -32138 \\ 23170 & -6393 & -30274 & 18205 & 23170 & -27246 & -12540 & 32138 \\ 23170 & -18205 & -12540 & 32138 & -23170 & -6393 & 30274 & -27246 \\ 23170 & -27246 & 12540 & 6393 & -23170 & 32138 & -30274 & 18205 \\ 23170 & -32138 & 30274 & -27246 & 23170 & -18205 & 12540 & -6393 \end{bmatrix}$$

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

$$Y = C^t Z$$

$$Y_{0X} = 23170Z_{0X} + 32138Z_{1X} + 30274Z_{2X} + 27246Z_{3X} + 23170Z_{4X} + 18205Z_{5X} + 12540Z_{6X} + 6393Z_{7X}$$

$$Y_{1X} = 23170Z_{0X} + 27246Z_{1X} + 12540Z_{2X} - 6393Z_{3X} - 23170Z_{4X} - 32138Z_{5X} - 30274Z_{6X} - 18205Z_{7X}$$

$$Y_{2X} = 23170Z_{0X} + 18205Z_{1X} - 12540Z_{2X} - 32138Z_{3X} - 23170Z_{4X} + 6393Z_{5X} + 30274Z_{6X} + 27246Z_{7X}$$

$$Y_{3X} = 23170Z_{0X} + 6393Z_{1X} - 30274Z_{2X} - 18205Z_{3X} + 23170Z_{4X} + 27246Z_{5X} - 12540Z_{6X} - 32138Z_{7X}$$

$$Y_{4X} = 23170Z_{0X} - 6393Z_{1X} - 30274Z_{2X} + 18205Z_{3X} + 23170Z_{4X} - 27246Z_{5X} - 12540Z_{6X} + 32138Z_{7X}$$

$$Y_{5X} = 23170Z_{0X} - 18205Z_{1X} - 12540Z_{2X} + 32138Z_{3X} - 23170Z_{4X} - 6393Z_{5X} + 30274Z_{6X} - 27246Z_{7X}$$

$$Y_{6X} = 23170Z_{0X} - 27246Z_{1X} + 12540Z_{2X} + 6393Z_{3X} - 23170Z_{4X} + 32138Z_{5X} - 30274Z_{6X} + 18205Z_{7X}$$

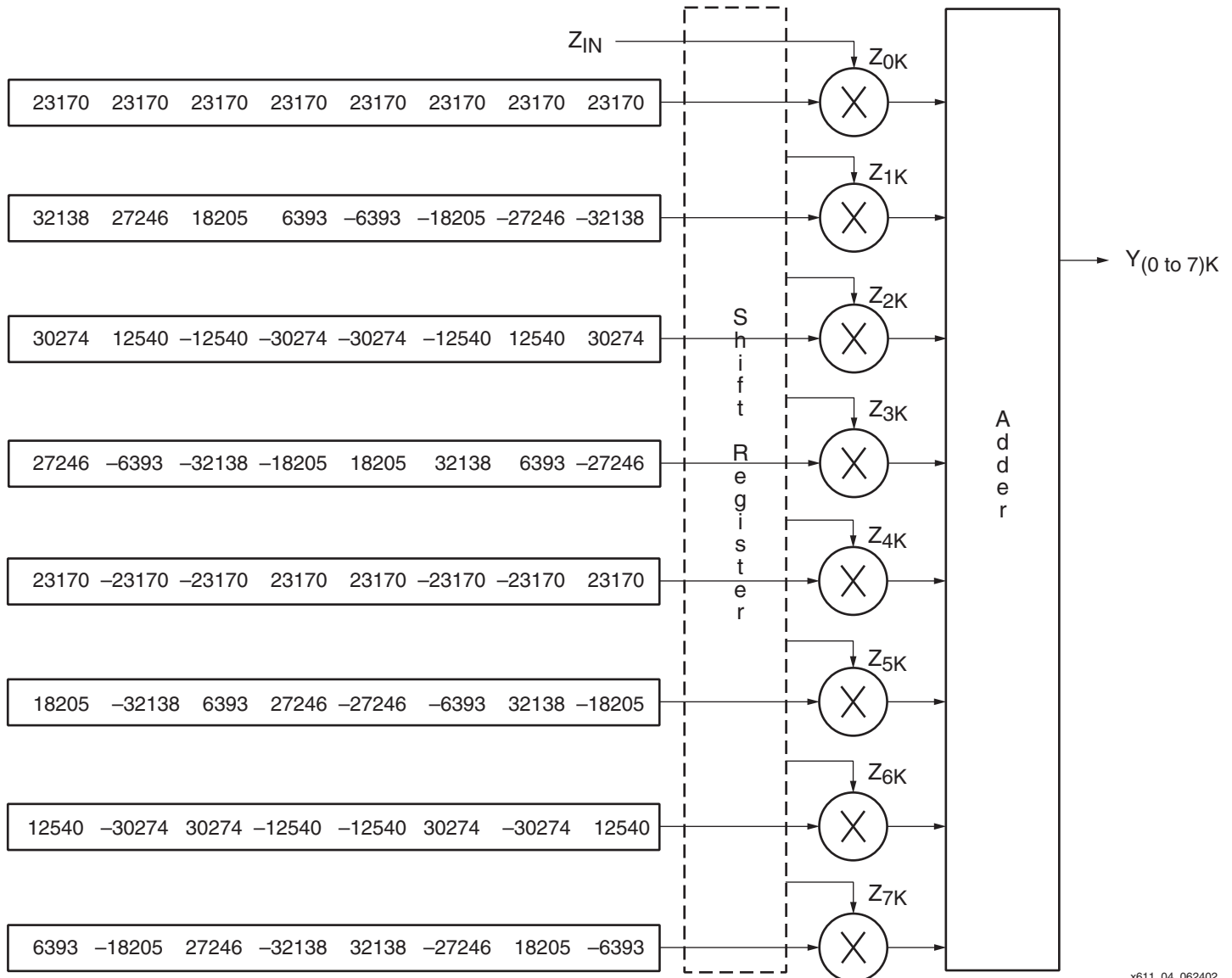
$$Y_{7X} = 23170Z_{0X} - 32138Z_{1X} + 30274Z_{2X} - 27246Z_{3X} + 23170Z_{4X} - 18205Z_{5X} + 12540Z_{6X} - 6393Z_{7X}$$

Each element in the first row in the coefficient matrix  $C^t$  is multiplied by each element in the first column of matrix  $Z$  and added together to get the first value  $Y_{00}$  of the output matrix  $Y$ . To get  $Y_{01}$ , each element of row zero in  $C^t$  is multiplied by each element in the first column of  $Z$  and is added as before. Multiplying row  $k$  of  $C^t$  by column zero of  $Z$  results in the coefficient  $Y_{k0}$ . All the elements in the first row of matrix  $Z$  are multiplied by all the elements in the first column of matrix  $C^t$ .

$$\begin{array}{l}
 23170 \rightarrow \\
 32138 \rightarrow \\
 30274 \rightarrow \\
 27246 \rightarrow \\
 23170 \rightarrow \\
 18205 \rightarrow \\
 12540 \rightarrow \\
 6393 \rightarrow
 \end{array}
 \left[ \begin{array}{cccccccc}
 z_{00} & z_{01} & z_{02} & z_{03} & z_{04} & z_{05} & z_{06} & z_{07} \\
 z_{10} & z_{11} & z_{12} & z_{13} & z_{14} & z_{15} & z_{16} & z_{17} \\
 z_{20} & z_{21} & z_{22} & z_{23} & z_{24} & z_{25} & z_{26} & z_{27} \\
 z_{30} & z_{31} & z_{32} & z_{33} & z_{34} & z_{35} & z_{36} & z_{37} \\
 z_{40} & z_{41} & z_{42} & z_{43} & z_{44} & z_{45} & z_{46} & z_{47} \\
 z_{50} & z_{51} & z_{52} & z_{53} & z_{54} & z_{55} & z_{56} & z_{57} \\
 z_{60} & z_{61} & z_{62} & z_{63} & z_{64} & z_{65} & z_{66} & z_{67} \\
 z_{70} & z_{71} & z_{72} & z_{73} & z_{74} & z_{75} & z_{76} & z_{77}
 \end{array} \right]$$

The calculation is implemented by using eight multipliers and storing the coefficients in ROMs. At the first clock, the eight coefficients in row zero of  $C^t$  are multiplied by the eight values in the first column of  $Z$  and are added together to result in  $Y_{00}$ . At the eighth clock, the eight values of row zero of  $C^t$  are multiplied by the eight values in column eight of  $Z$ . The resultant is added to give  $Y_{07}$ .

The block diagram for the implementation of the 2D-IDCT is shown in [Figure 4](#)



x611\_04\_062402

Figure 4: 2D-IDCT Implementation (K is 0-7)

## Implementation Description

Figure 3 and Figure 4 show the block diagrams for a reference design implementation. The 1D-DCT values are first calculated and stored in a RAM. The second 1D-DCT is done on the values stored in the RAM. For each 1D implementation, input data are loaded into the first input of a multiplier. The constant coefficient multiplication values are stored in a ROM and fed into the second input of the multiplier. At each clock, eight input values are multiplied with eight constant coefficients. The output of the eight multipliers are added together giving the 1D coefficient values stored in the RAM. The values stored in the intermediate RAM are read out one column at a time (i.e., every eighth value is read out every clock). This is the input for the second DCT. The RDY\_IN signal is used as a hand shake signal between the DCT and the IDCT functions. When High, the RDY\_IN signal indicates valid DCT outputs from the DCT function to the IDCT function.

## Reference Design

### Resource Utilization

This application note and reference design includes a basic explanation of IDCT. The Verilog and VHDL code shows the implementation of a 2D IDCT. The code implements a 2D IDCT with 9-bit input data and 8-bit output data. The number of input/output data bits can be changed as needed. The code is available on the Xilinx web site at:

[www.xilinx.com/bvdocs/desfiles/xapp611.zip](http://www.xilinx.com/bvdocs/desfiles/xapp611.zip).

The code has been tested and simulated against the vectors given in **"Image and Video Compression Standards," second edition, by Vasudev Bhaskaran and Konstantinos Konstantinides, ISBN 0-7923-9952-8** [1], on page 60. This code has not been tested for IEEE compliance.

Table 1: Resource Utilization

Device	Speed Grade	Package	Pre-Map (Synthesis Constraint)	Post-Route	Slices
XCV300E	-8	BG352	83.96 MHz (120 MHz)	57.89 MHz	1343 (43%)
XC2V250	-6	FG456	176.99 MHz (180 MHz)	120.95 MHz	944 (61%)
XCV300	-6	PQ240	104.29 MHz (100 MHz)	46.97 MHz	1341 (43%)
XC2S200	-6	FG256	93.56 MHz (80 MHz)	53.49 MHz	1372 (58%)

## References and Recommended Links

- "Image and Video Compression Standards," second edition, by Vasudev Bhaskaran and Konstantinos Konstantinides, ISBN 0-7923-9952-8
- A variety of cores developed by Xilinx or by partners are available on the Xilinx web site:
  - [http://www.support.xilinx.com/ipcenter/catalog/logicore/docs/dct\\_1d.pdf](http://www.support.xilinx.com/ipcenter/catalog/logicore/docs/dct_1d.pdf)
  - [http://www.xilinx.com/products/logicore/alliance/xentec\\_spotlight.htm](http://www.xilinx.com/products/logicore/alliance/xentec_spotlight.htm)
  - [http://www.support.xilinx.com/ipcenter/dct\\_lounge/index.htm](http://www.support.xilinx.com/ipcenter/dct_lounge/index.htm)
- Application note XAPP208: IDCT Implementation in Virtex Devices for MPEG Applications, v 1.1 (12/99) at: <http://www.xilinx.com/xapps/xapp208.pdf>

## Conclusion

The IDCT design files demonstrate an efficient implementation of the IDCT algorithm using Virtex devices. The IDCT reference design files can be used to target any Xilinx device. Optimize the code by instantiating the adder/subtractor and multiplier units when targeting Virtex devices.

## Revision History

The following table shows the revision history for this document.

Date	Version	Revision
03/20/02	1.0	Initial Xilinx release.
06/25/02	1.1	Revised <b>Figure 3</b> and <b>Figure 4</b> .
06/03/05	1.2	Updated matrix, equation, and text on Page 6.
04/05/07	1.2.1	Deleted reference to WP113 (obsolete).