

## Dynamic Partial Reconfiguration for Discrete Cosine Transform Computation

**Bhavesh Jaiswal**

*Shankersinh Vaghela Bapu Institute of Technology, Gandhinagar, Gujarat, India  
Affiliated to Gujarat Technological University  
E-mail: bhavesh.jaiswal@bapugkv.ac.in*

**Nagendra Gajjar**

*Institute of Technology, Nirma University  
Ahmedabad, Gujarat, India  
E-mail: nagendra.gajjar@nirmauni.ac.in*

### Abstract

Discrete Cosine Transform is one of the most important building blocks for the emerging video coding standard, H.264. In this paper, architecture for DCT computation, a hardware intensive operation, using dynamic partial reconfiguration is proposed. Field programmable gate array (FPGA), due to inherent parallelism, is obvious choice for the task. The architecture can perform DCT computations for different zones and change the configuration of processing elements. The unused elements can be used for other processing. Moreover the functions being performed on the parts of the FPGA that are not being reconfigured will not be interrupted during reconfiguration.

*Keywords:* Partial Reconfiguration, Discrete Cosine Transform (DCT), Field Programmable Gate Array (FPGA)

### 1. Introduction

There has been considerable interest in implementing DSP algorithms using FPGA technology. Examples include the Altera Mega-function Partners Program [1] and examples from Xilinx [2]. However, FPGA solutions are typified by poor hardware utilization and solutions dominated by routing. This is due to the fact that little account is given to developing a suitable architectural description of the algorithm for the specific FPGA technology.

DCT is a computation intensive operation. DCT is used in most of the digital processing application because of its energy compaction characteristics. The development of efficient algorithms for the computation of DCT began soon after Ahmed et al [3] reported their work on DCT. It was natural for initial attempts to focus on the computation of DCT by using Fast Fourier Transform

(FFT) algorithms. Many algorithms for fast computation of DCT are reported in the literature [4, 5, 6, 7]. General approach used in DCT is converting the image pixels of block 8x8 into series of coefficients that define spectral composition of the block. Its direct implementation requires large number of adders and multipliers.

In this paper we present FPGA based architecture for two dimensional DCT computations using dynamic partial reconfiguration. DCT computations from 1x1 DCT to 8x8 DCT is possible which can reduce power considerably. Dynamic or Run-time reconfiguration reduces the hardware resources and can also be used for circuit specialization based on the information known only during run-time. The architecture can change the configuration of processing elements to trade off the precisions of DCT coefficients with computational complexity.

## 2. Previous Work

A low power DA-based DCT core using adaptive bit width and arithmetic activity considering signal correlations and quantization was proposed in [8]. First, they used a MSB rejection module to reduce the number of arithmetic operations required in the presence of correlated inputs. Second, they used a row-column classification module to reduce the overall signal activity by introducing a small error in the arithmetic computation. Their experimental results show that both MSBR and RCC achieve about 40% power savings for still images.

A low-complexity scalable DCT image compression scheme was proposed by [9]. It eliminates entropy coding and quantization and achieves quality scalability by encoding the DCT coefficients bit plane by bit plane. They used a rectangular zone for DCT operations. Their experimental results show that their scheme has about the same compression performance as the baseline JPEG and has lower complexity.

An energy efficient hardware architecture for variable N-point 1D-DCT was given in [10], which can be used to implement the shape adaptive DCT. They used a new distributed arithmetic architecture for the 1D-DCT implementation. They used clock gating to shut down the redundant logic based on the value N.

Dynamic partial reconfiguration has drawn many attentions these days. A new FPGA-based reconfigurable computer called the Erlangen slot machine (ESM) developed in [12]. It uses slot-based architecture, which allows the slots to be reconfigured independently of each other during runtime. Reconfigurable hardware architecture for video-based driver assistance applications in future automotive systems was proposed in [13]. Different operations such as shape engine, tunnel engine, and taillight engine can be dynamically reconfigured during runtime.

A waveform-like reconfiguration for dynamic partial reconfiguration was presented in [14]. It decreases the overhead of reconfiguration by dividing the reconfiguration modules according to the specific data graph. Therefore, some parts of the data graph start processing while the following parts of the data graph are still being reconfigured.

## 3. DCT Computation

The forward discrete cosine transform (DCT) processes 64 spatial samples, arranged as an 8x8 block, and

converts them to 64 similarly arranged frequency coefficients. These 64 coefficients are the scale factors which correspond to the 64 respective cosine waveforms.

The cosine basis functions are orthogonal, and hence, independent. Any block of 64 samples can be represented by first scaling the 64 cosine basis functions by the corresponding 64 DCT-computed coefficients and then progressively summing the results. As a result of the energy compaction properties of the DCT on natural images, the most significant contributions to the reconstructed image are from the low-order frequency coefficients. Therefore, the majority of an image can be captured from just the summation of the lower order values.

Implementation of the 2-D DCT directly from the theoretical equation results in 1024 multiplications and 896 additions. Fast algorithms exploit the symmetry within the DCT to achieve dramatic computational savings.

The first category of 2-D DCT implementation is indirect computation through other transforms, most commonly, the Discrete Hartley Transform (DHT) and the Discrete Fourier Transform (DFT). The DHT-based algorithm of [16] shows increased performance in throughput, latency, and turnaround time. Optimization with respect to these parameters is not the focus of the proposed project. A DFT approach [17] calculates the odd-length DCT.

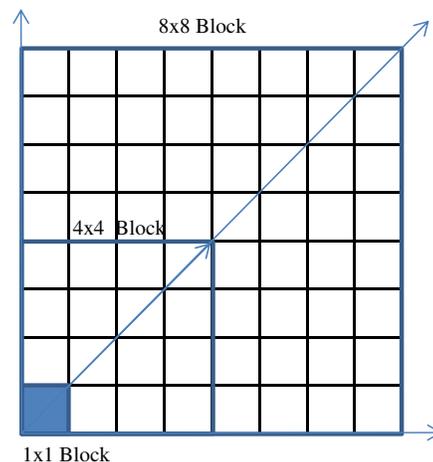


Fig. 1. DCT Computation Blocks from 1x1 to 8x8

The second style of algorithms computes the 2-D DCT by row-column decomposition. In this approach, the

separability property of the DCT is exploited. An 8-point, 1-D DCT is applied to each of the eight rows, and then again to each of the eight columns. The 1-D algorithm that is applied to both the rows and columns is the same. Therefore, it could be possible to use identical pieces of hardware to do the row computation as well as the column computation.

The third approach to computation of the 2-D DCT is by a direct method using the results of a polynomial transform. Computational complexity is greatly reduced, but regularity is sacrificed. Instead of the sixteen 1-D DCTs used in the conventional row-column decomposition, [18] uses all real arithmetic including eight 1-D DCTs, and stages of pre-adds and post-adds (a total of 234 additions) to compute the 2-D DCT. Thus, the number of multiplications for most implementations should be halved as multiplication only appears within the 1-D DCT.

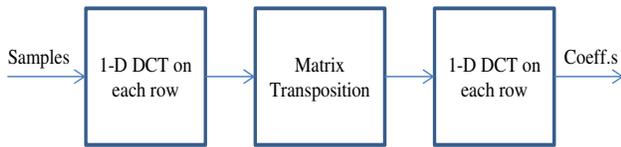


Fig. 2. Block representation of Row-column decomposition

In this paper we have used the reconfigurable DCT block based on row-column decomposition. In this approach, the computation is in the 1-D DCT block, which can potentially be reused. The transposition matrix would separate the two DCT blocks as shown in Figure 2.

#### 4. System Architecture

In our approach, the computation of two dimensional DCT can be done over different block sizes i.e. from 1x1 DCT to 8x8 DCT as shown in Figure 1. These could be interpreted as the precision required for DCT computation. This is possible due to dynamic partial reconfiguration. Partial reconfiguration allows flexibility in selecting the quality of precision of a specific processing element. The configuration can be adjusted during runtime depending upon the data provided. Moreover the DCT computations are not interrupted when switching from different blocks. The schematic diagram of the reconfigurable processing elements for DCT computation along with controller block is shown in the Figure 3.

In DCT implementation, DCT 0 -7 are the computation blocks and the unused ones can be reconfigured to perform other similar operations. Reconfiguration is stored in the configuration memory block. Controller is used to generate the address and control signals for data fetching and data assigning. Dynamic reconfiguration can be used in different ways to enhance different characteristics of the circuit based on the information known only at the run-time. By using Partial reconfiguration one can use specific circuits based on the run time data and consequently accelerating the computation process [19].

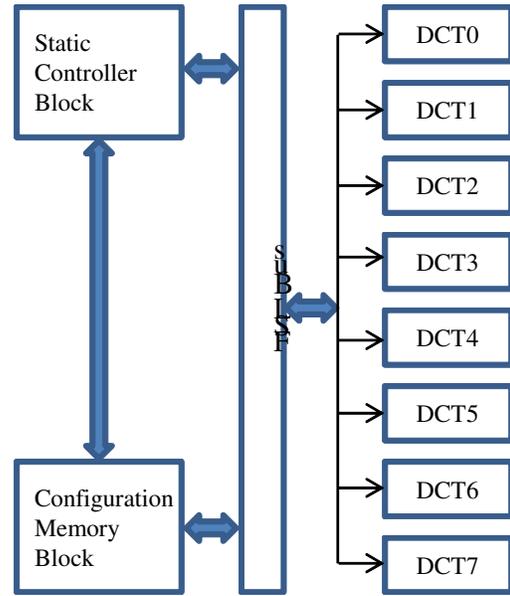


Fig. 3. Block representing different areas for architecture

#### 5. FPGA Implementation

The architecture uses eight blocks for DCT computation. Computation of inner product is parallelized and 1-D DCT can be performed in one clock cycle. For the implementation of the architecture with eight separate reconfigurable areas, the DCT logic is arranged on the periphery and the static controller area is kept in the middle, so that each of the processing element have easy access to the static part.

$$T_{config} \approx \frac{Bytes}{f_{clock}} \quad (1)$$

Implementation of the said DCT unit was done using Verilog coding and the Altera Quartus-II tool. The row-column decomposition approach was used for the DCT implementation. The target device chosen was Cyclone-II. A cut-out section depicting the RTL view of one of the DCT unit is shown in Fig. 4. The configuration time can be estimated based on [21]

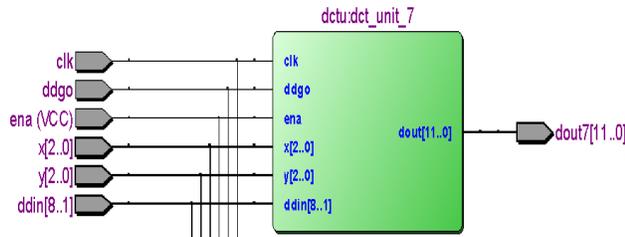


Fig. 4. Screenshot of RTL view of DCT Block

A bit-stream file is generated for the DCT block and downloaded on Cyclone-II. The downloaded DCT module was verified for the proper operation. Aside from the two extremes of the fully bit serial approach and the fully parallel approach, a partially paralleled structure is adopted. The extent of parallelization can be varied to make the number of clock cycles converge.

## 6. Conclusion

The proposed DCT unit can be used for upcoming video standard H.264. The DCT computation core is found to be working on the Cyclone-II. All the DCT blocks implementation and Configurability was not possible on Altera DE-II board used, so the design has to be ported to Xilinx Design flow. Xilinx Virtex-3 has capability for the reconfiguration. Dynamic partial reconfiguration can be done using the Xilinx Early Access Partial Reconfiguration design flow [22]. This will save considerable amount of area and the unused elements can be used for other computation purpose.

In addition, our algorithm implementation is not yet fully optimized and control logic is implemented in a very naive manner. The power saving for the core can be computed using much powerful tools. The trade-off between power reduction capability and the speed of the core during run-time is to be studied in further detail. The flow would be completed after configurability check has been done and this would show the actual working benefits of the core.

## References

1. Altera Meg-function Partners Program: [http://www.altera.com/html/ programs/ampp.html](http://www.altera.com/html/programs/ampp.html).
2. G.R. Goslin "A Guide to using FPGAs for Application-Specific Digital Signal Processing Performance", Xilinx Corporate Applications Group Report, 1995.
3. N. Ahmed, T. Natarajan, and K.R. Rao, "Discrete Cosine Transform", IEEE Trans. Comm., COM-23, pp. 90-93, Jan. 1974.
4. B.G. Lee, "A new algorithm to compute the discrete cosine transform", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-32, pp. 1243-1245, Dec. 1984.
5. H.S Hou, "A fast recursive algorithms for computing the discrete cosine transform", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-35, pp. 1455-1461, Oct. 1987.
6. N. I. Cho and S. U. Lee, "DCT algorithms for VLSI parallel implementation", IEEE Trans. Acoust., Speech, Signal Processing, vol. 38, pp. 121-127, Jan. 1990.
7. Nam Ik Cho, Sang Uk Lee, "Fast Algorithm and Implementation of 2-D Discrete Cosine Transform", IEEE Transaction on Circuits and Systems, Vol.38, No.3, March 1991.
8. Xanthopoulos, T. and Chandrakasan, A. P. "A low power DCT core using adaptive bit-width and arithmetic activity exploiting signal correlations and quantization". IEEE J. Solid State Circuits 35, 2000.
9. Vander Vleuten et al., "Low-complexity scalable DCT image compression", In Proceedings of International Conference on Image Processing. vol. 3. IEEE, Los Alamitos, CA, 837-840, 2000.
10. Kinane et al., "Energy-efficient hard-ware architecture for variable N-point 1D DCT", In Proceedings of International Workshop on Power and Timing Modeling, Optimization and Simulation. IEEE, Los Alamitos, CA, 780-788, 2004.
11. Shams et al., "A low-power high-performance distributed DCT architecture", In Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'02). IEEE, Los Alamitos, CA, 26.
12. Majer et al., "The Erlangen slot machine: A dynamically reconfigurable FPGA-based computer", Journal of VLSI Signal Process. System, 47, 15-31, 2007.
13. Claus et al., "Using partial-run-time reconfigurable hardware to accelerate video processing in driver assistance system", In Proceedings of the Conference on Design, Automation and Test in Europe (DATE'07). ACM, New York, 498-503, 2007.
14. Braun et al., "Data path driven waveform-like reconfiguration", In Proceedings of International Conference on Field-Programmable Logic and Applications. IEEE, Los Alamitos, CA, 607-610, 2008.
15. Mitchell, Joan and Pennebaker, William "JPEG: Still Image Data Compression Standard", Van Nostrand Reinhold, New York. 1993. pp. 1-96, 335-626.

16. J.H. Hsiao, L.G. Chen, T.D. Chiueh, C.T. Chen, "High Throughput CORDIC-Based Systolic Array Design for the Discrete Cosine Transform", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 3, pp. 218-225, June 1995.
17. M.T. Heideman, "Computation of an odd-length DCT from a real -valued DFT of the same length", *IEEE Trans. Signal Process.*, vol. 40, no. 1, pp. 54-61, Jan. 1992.
18. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces (Translation Journals style)," *IEEE Transl. J. Magn.Jpn.*, vol. 2, Aug. 1987, pp. 740-741 [*Dig. 9<sup>th</sup> Annu. Conf. Magnetism* Japan, 1982, p. 301].
19. N. McKay, T. Melham, and Kong Woei Susanto, "Dynamic specialization of XC6200 FPGAs by Partial Evaluation," in proceedings, *IEEE Symposium on FPGAs for Custom Computing Machines*, 1998, pp. 308-309.
20. M.J. Wirthlin and B.L. Hutchings "Improving functional density through run-time constant propagation," in Proceedings, *ACM Fifth International Symposium on Field Programmable Gate Arrays*, 1997, pp. 86-92.
21. XAPP138 – Virtex FPGA Series configuration and read-back. Xilinx Inc., 2005, San Jose, CA.
22. Early Access Partial Reconfiguration User Guide, Xilinx Inc., 2006, San Jose, CA.
23. Altera DE-II Development and Education Board, User Manual, 2006.
24. Altera Quartus-II Design Guide.
25. J. Huang, M. Parris, J. Lee, R.F. Demara, "Scalable FPGA-based Architecture for DCT Computation Using Dynamic Partial Reconfiguration", *ACM Transactions on Embedded Computing Systems*, Vol. 9, No. 1, Article 9, October 2009.
26. M. Sun, C. Ting., and M. Albert, "VLSI Implementation of a 16x16 Discrete Cosine Transform", *IEEE Transactions on Circuits and Systems*, Vol. 36, No. 4, April 1989.
27. T.T. Trang, P. Binh, "A High Accuracy and High Speed 2-D 8x8 Discrete Cosine Transform Design", *Proceedings of ICGCRCICT 2010*, Vol. 1, 2010, pp. 135-138.