An Optimal Transform Architecture for H.264/AVC

A K Prasoon

School of Electrical Sciences

Vellore Institute of Technology

Prof. K Rajan

Department of Physics

Indian Institute of Science

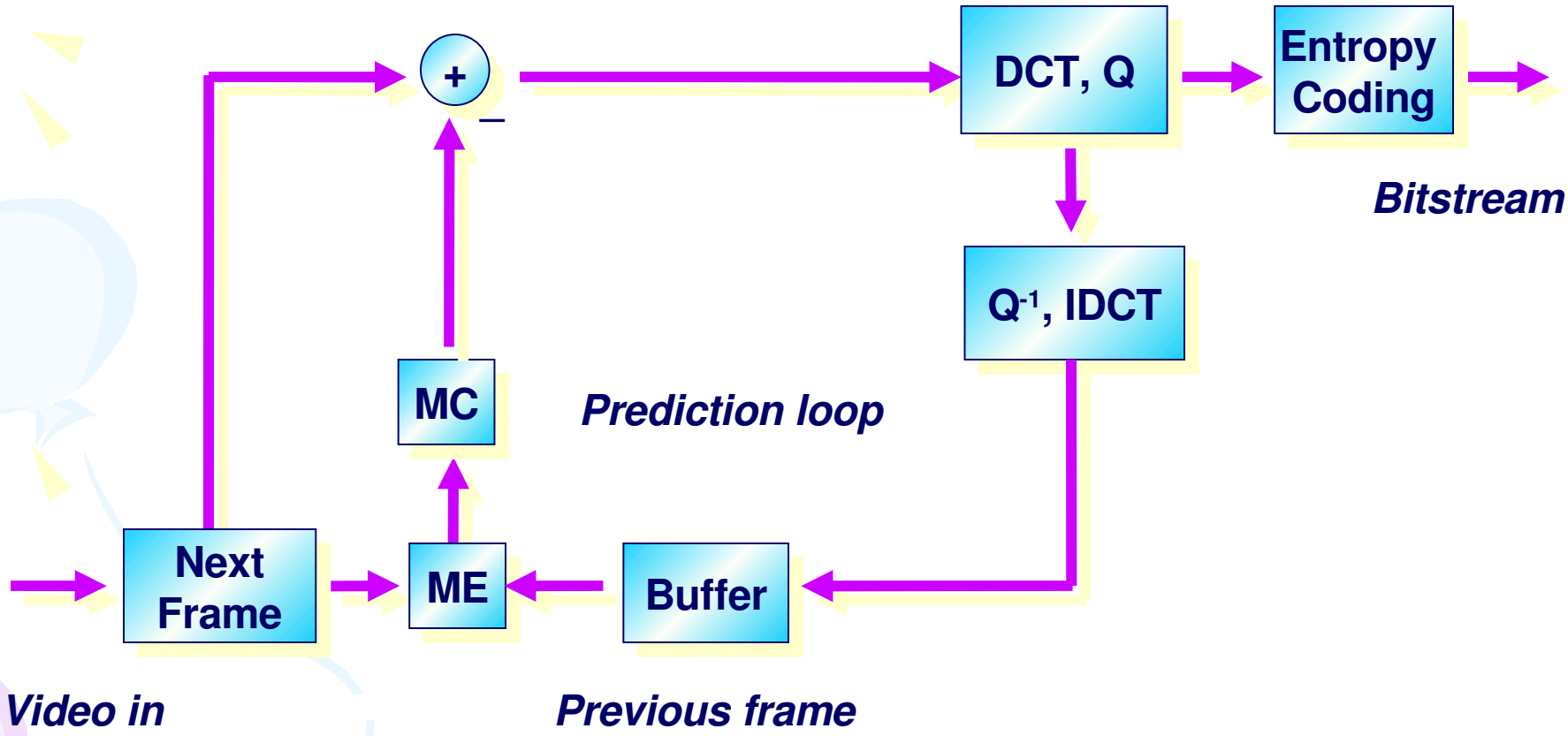
Outline

- Introduction to Video Coding
- History of Video Coding Standards
- Generic Framework
- H.264/AVC Overview
- Integer Transforms
- Architecture
- Simulation Results
- Synthesis Results
- Performance Comparisons

History of Video Coding Standards

- ITU-T Video Coding Experts Group (VCEG)
  - H.261 (1990) -> H.263 (1995) -> H.263+ (1998) -> H.26L
- ISO Motion Picture Experts Group (MPEG)
  - MPEG1(1991) -> MPEG2 (1994) -> MPEG4 (1999)
- Joint Video Team (JVT) (VCEG/MPEG) 2001
  - H.264/MPEG4 part 10. Official title: Advanced Video Coding (AVC)
- International standard: December 2002

Generic Framework*



* H.261, 263, 263+, MPEG-1/2/4

H.264/AVC Overview

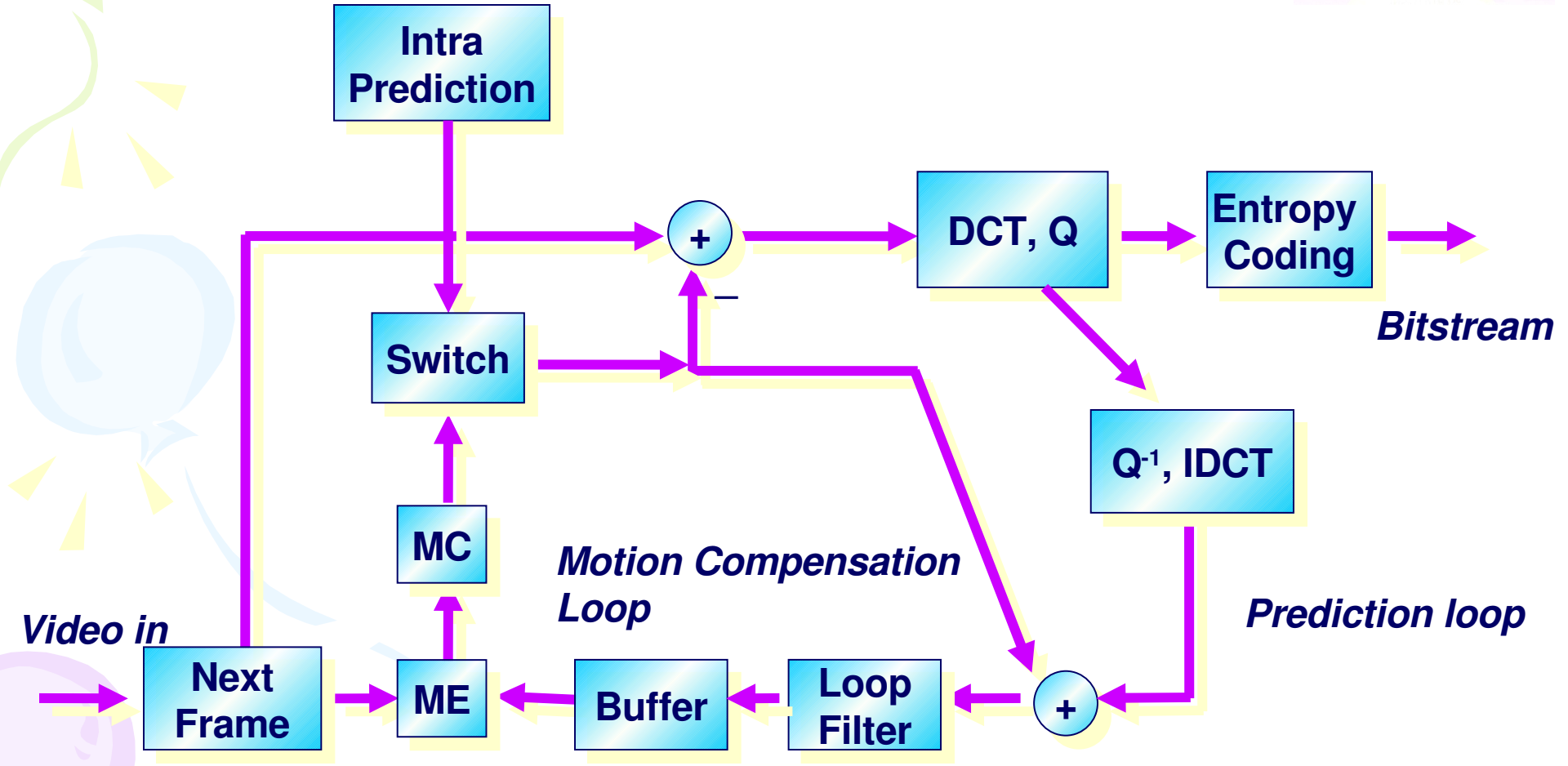
- Video coding is similar in spirit to other standards but with important differences

- Key new features:
  - Enhanced motion estimation with variable block size
  - Integer block transform
  - Improved in-loop deblocking filter
  - Enhanced entropy coding

- Average bit rate reduction of 50% given fixed fidelity compared to any other standard

- Complexity vs. Coding efficiency

Block Diagram:H.264/AVC Encoder



Integer Transforms

- Motivation
  - Quantization reduces precision
  - Small block size (4x4) reduces the performance loss
  - Reduced computational complexity

- Discrete Cosine Transform (DCT) has been used widely in standards
  - Good energy compact property
  - Orthogonal bases

- The final results of the integer transform are an approx of the real DCT results, but this integer conversion causes a minimal loss of accuracy in this calculation.

Integer Transforms (contd)

The transform computation

$$\theta = C_f X C_f^T$$

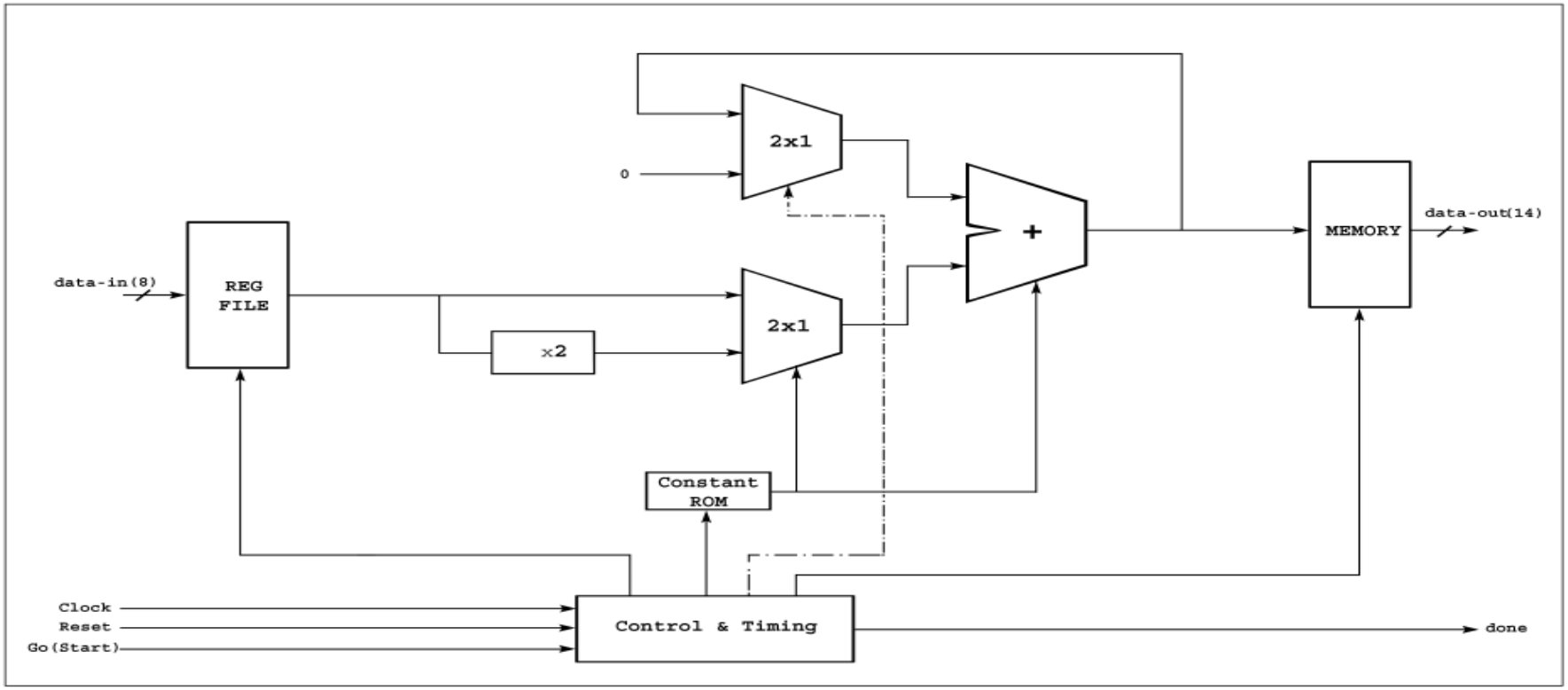
Where C_f is given by

$$C_f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

X is the 4x4 residual data block formatted in 2's complement representation and is 8-bit wide because the value of each pixel ranges from 0 to 255

$$X = \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix}$$

Architecture



In order to compute the transform (which consists of 2 matrix-multiplication), the above shown architecture is used. The serial architecture calculates each of the transform elements individually.

State Diagram

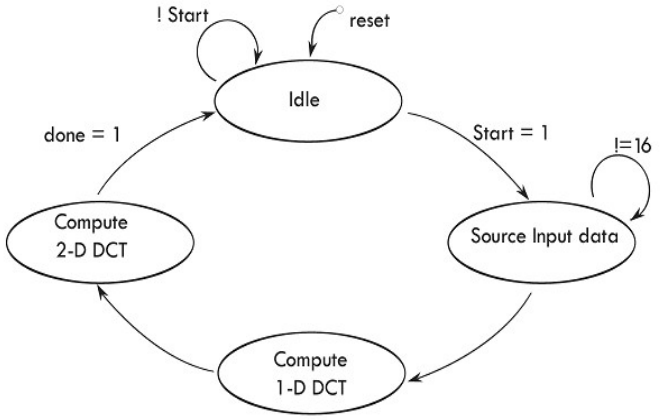


Fig.1

- In figure 1 the four primary states to compute the 2-D transform is shown.

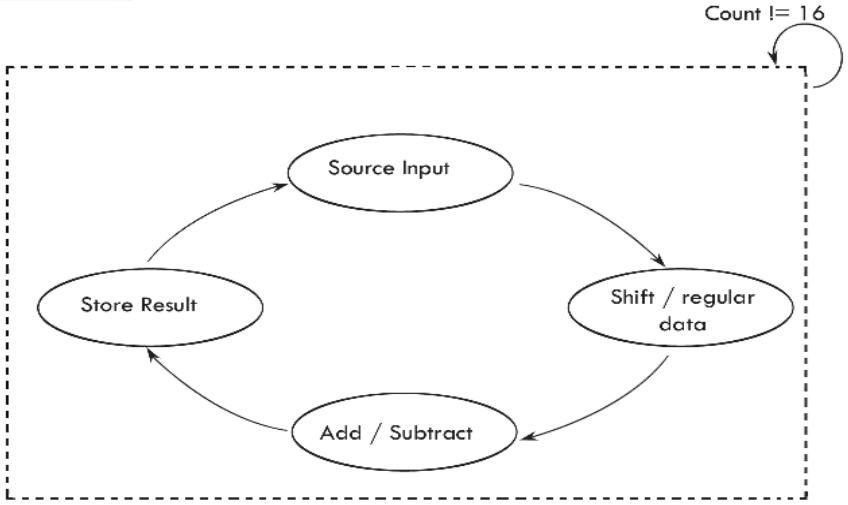


Fig.2

- Figure 2 shows the detail state diagram used to compute 1-D transform. These four states are repeated till each of the four states are repeated till each of the 16 transform elements are computed.

Bit-Width Analysis

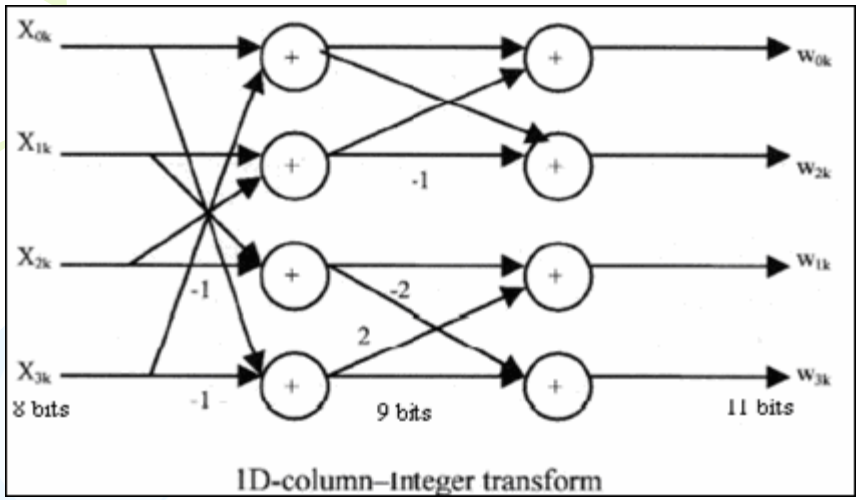


Fig.1

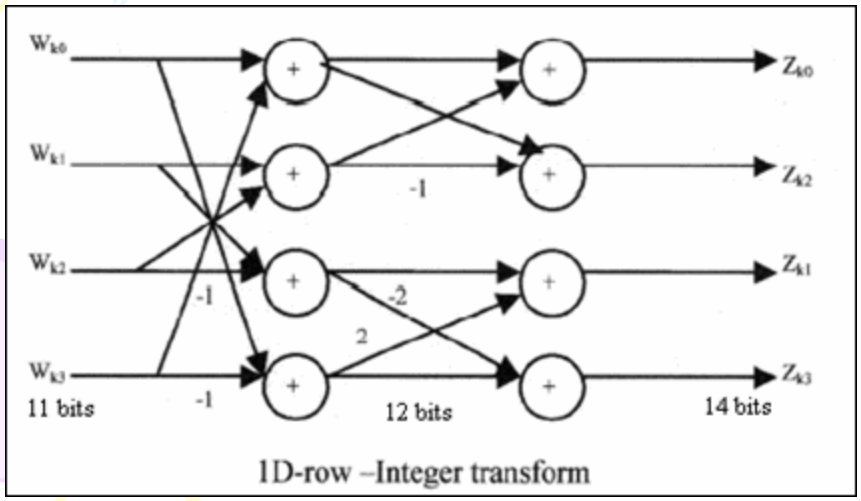


Fig.2

- *The input of the forward transform is the residual block. The dynamic range of the residual is from 0 to +255, i.e. 8 bits.*

- *From Fig.1, the bit width of the first column adder outputs should be 9 bits since addition and subtraction operations increase the dynamic range by 1 bit.*

- *The 1-D transform outputs, w_{ok}-w_{3k} , should be 11bits because there are some components doubled before additions.*

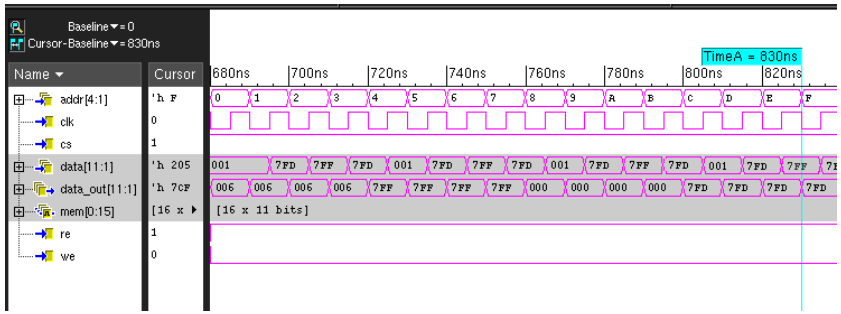
- *The dynamic range of the 1-D forward transform increases 3 bits from the discussion above. For same reasons the output of 2-D transform is 14 bits, as shown in fig.2*

Simulation Results

The transform computation $\theta = C_f X C_f^T$

computed 1-D transform is:

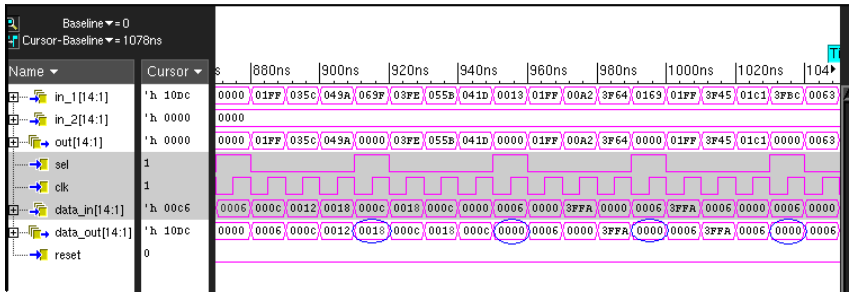
$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 6 & 6 & 6 & 6 \\ -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 \\ -3 & -3 & -3 & -3 \end{bmatrix}$$



Waveform 1-D transform

computed 2-D transform

$$\begin{bmatrix} 6 & 6 & 6 & 6 \\ -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 \\ -3 & -3 & -3 & -3 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 24 & 0 & 0 & 0 \\ -4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -12 & 0 & 0 & 0 \end{bmatrix}$$



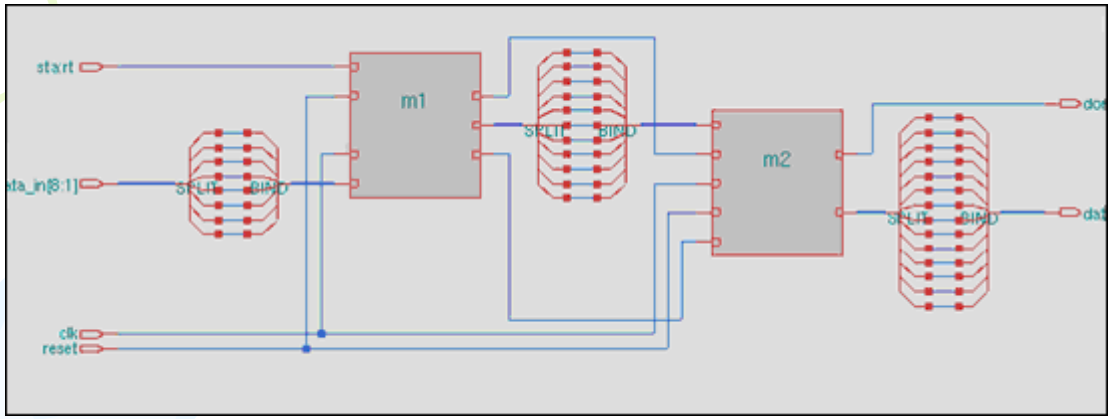
Waveform 2-D transform

Synthesis Results

Gates	1189
Area	0.07 mm^2
Power	17.535 mW
Throughput	150 M samples/sec

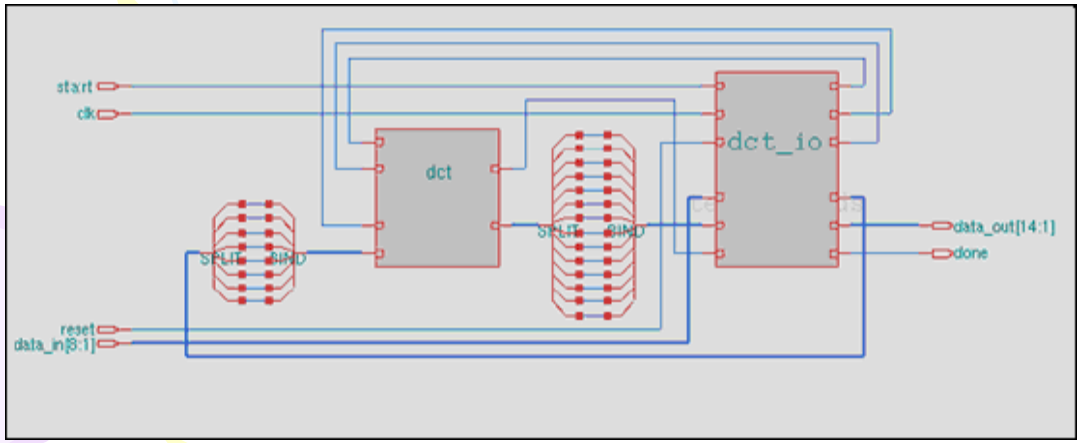
- The forward integer 2-D DCT architecture has been coded using Verilog HDL and synthesized using 0.18µ TSMC technology.
- The architecture has been optimized to process one input data per clock cycle.
- The processing speed can achieve 150 M-samples/sec for the transform computation.

Synthesis Results



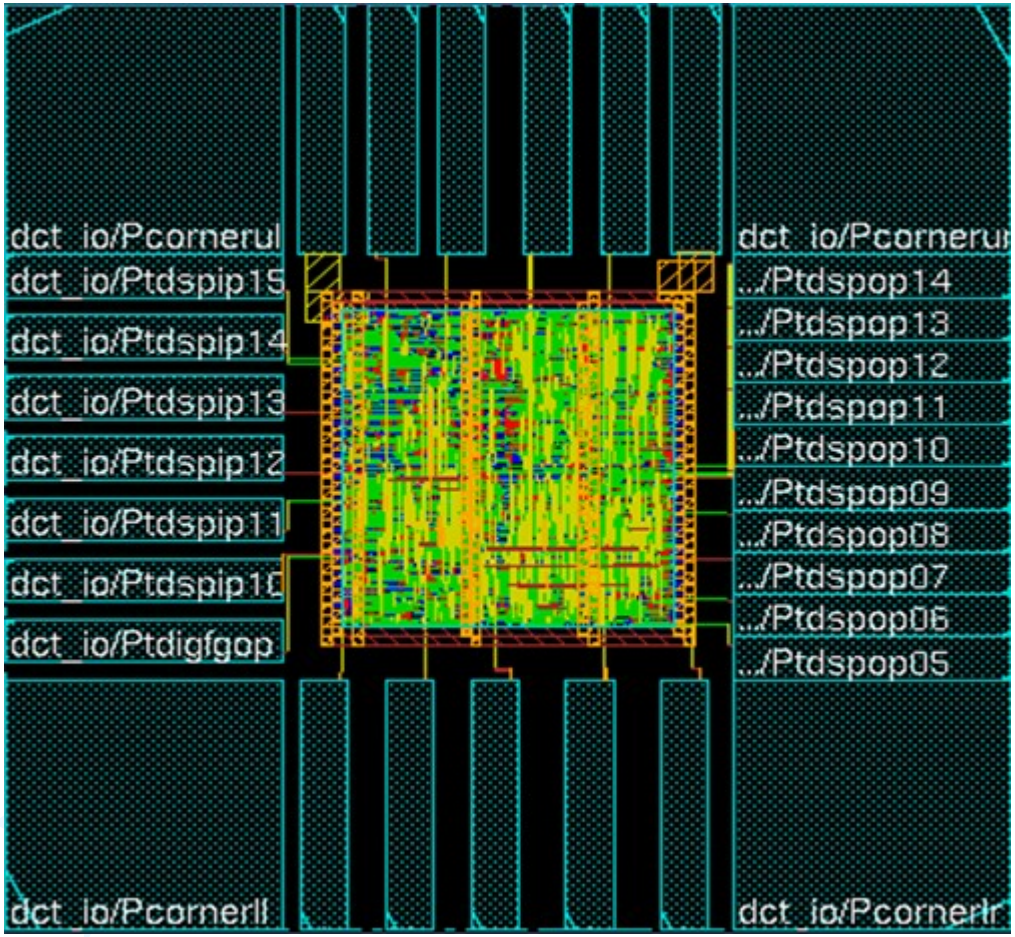
Synthesised 2D DCT-Top level module

M1-1D column-integer trans -form
M2- 1D row-integer transform



Synthesised DCT module with I/O pads

Chip Layout



Standard Cell Routing - DCT module

Performance Comparison

- Timing results (Serial Architecture)

Transform	Optimal Transform	[2]
1-D DCT*	82	112
2-D DCT	82	112
Total	164	224

*No of clock cycles required to compute the transform

- Area (Gate Count) and Throughput*

Transform	Architecture	No. of Gates	Throughput*
Optimal	Serial	1189	150 M
[2]	Serial	294	11 M
[7]	Parallel	3737	500 M

*Throughput takes into account number of clock cycles required to process a 4 × 4 block of data

Conclusions

- The aim of this paper has been to implement a H.264 transform block, that has been implemented with 1189 gates, and occupying 0.0708 mm^2 area.

- This architecture can be applied to a dedicated hardware design for H.264/AVC video codec.

References

1. K. Sayood, “Introduction to Data Compression”, Morgan Kaufman, San- Mateo, CA, 2000.
2. Roman Kordasiewicz and Shahram Shirani “On Hardware Implementations of DCT and Quantization Blocks for H.264/AVC”, Journal of VLSI Signal Processing 2007
3. H. S. Malvar, A. Hallapuro, M. Karaczewicz, and L. Kerofsky, “Low-Complexity Transform and Quantization in H.264/AVC,” IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, July 2003.
4. I. E. G. Richardson, H.264/MPEG-4 Part 10: Transform & Quantization. [Online]. Available: http://www.vcodex.com, 2002.
5. M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, “H.264/AVC Baseline Profile Decoder Complexity Analysis,”, IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, July 2003.
6. A. Hallapuro and M. Karczewicz, “Low Complexity Transform and Quantization-Part 1: Basic Implementation”, February 2001, JVT Document JVT-B038.
7. T.-C. Wang, Y.-W. Huang, H.-C. Fang, and L.-G. Chen, ”Parallel 4X4 2D transform and inverse transform architecture for MPEG-4 AVC/H.264”, in Proceedings of the 2003 International Symposium on Circuits and Systems, vol. 2, May 2003, pp. II-800-II-803.
8. A. M. Patino, M. Peiro, F. Ballester, and G. Paya, “2D-DCT on FPGA by Polynomial Transformation in Two-Dimensions”, in Proceedings of the 2004 International Symposium on Circuits and Systems, vol. 3, May 2004, pp. III-365-III-368.
9. M. Fu, G. Jullien, V. Dimitrov, and M. Ahmadi, “A Low-Power DCT IP Core Based on 2D Algebraic Integer Encoding”, in Proceedings of the 2004 International Symposium on Circuits and Systems, vol. 2, May 2004, pp. II-765-II-768.
10. T.Wiegand, G.J.Sullivan, Ajay Luthra, “Overview of the H.264/AVC Video coding Standard”, IEEE Trans on Circuits and Systems for Video Tech Vol.13 No.7 ‘July 2003.