An Implementation on Integer DCT Architectures for HEVC with Error Compensation

Dr.B.B.M.KrishnaKanth
Principal, Hindu College of Engineering & Technology,Amaravathi Road, Guntur,A.P,India-522002.
Mail:bbkkanth@yahoo.com

Abstract: **In this paper, we present area- and power-efficient architectures for the implementation of integer discrete cosine transform (DCT) of different lengths to be used in High Efficiency Video Coding (HEVC).We show that an efficient constant matrix multiplication scheme can be used to derive parallel architectures for 1-D integer DCT of different lengths. We also show that the proposed structure could be reusable for DCT of lengths 4, 8, 16, and 32 with a throughput of 32 DCT coefficients per cycle irrespective of the transform size. In this paper mainly we focus on the error compensation by using a adder tree. Previously for this error compensation high-throughput DCT designs have been adopted to fit the requirements of real-time application. Operating the shifting and addition in parallel, an error-compensated adder-tree (ECAT) is proposed to deal with the truncation errors and to achieve low-error and high-speed discrete cosine transforms (DCT) design. Instead of the 9 bits used in previous works, 4-bit Distributed Arithmetic was proposed. DA based DCT design with an error-compensated adder-tree (ECAT) is the proposed architecture in which, ECAT operates shifting and addition in parallel by unrolling all the words required to be computed. Furthermore, the Error-Compensated Circuit alleviates the truncation error for high accuracy design. Based on low-error ECAT, the DA-precision in this work is chosen to be 4 bits instead of the traditional 9 bits. Therefore, the hardware size and cost is reduced, and the speed is improved using the proposed ECAT.Also we reduces the time delay than previous method.**

Key Words: **Adders, DCT- Discrete Cosine Transform, DA- Distributed Arithmetic, ECAT- Error-Compensated Adder-Tree,VHDL.**

INTRODUCTION

High-Efficiency Video Coding (HEVC) achieves a 50% reduction in bit-rate over H.264/AVC at the same visual quality. A key feature in HEVC is the introduction of large 16_16 and 32_32 inverse discrete cosine transforms (IDCTs),a new 4_4 inverse discrete sine transform (IDST)and high-precision 4x4 and 8x8 IDCTs. The large transforms contribute to 6.7% - 10.1% bit-rate reduction for 1080p (1920_1080) and larger video sequences, and the increased precision in smaller transforms contribute 0.3% - 1.2%. These new features of HEVC raise several challenges for hardware implementations:

1. HEVC uses Transform Units (TUs) of size 4x4, 8x8, 16x16, and 32x32 pixels. This variety of TU sizes N multiplications per coefficient. Hence, the largest IDCT in HEVC (32x32) takes 4the number of multiplications as the largest IDCT in H.264/AVC (8x8). Further, the increased complicates the design of control logic as TUs of different sizes take different number of cycles for processing.
2. Like H.264/AVC, the 2-D transforms in HEVC are separable into 1-D transforms along the columns and rows. The N-pt 1-D IDCT used in an NxN 2-D IDCT can be viewed as the product of a N _ N transform matrix with N _ 1 input coefficients. This requires precision in HEVC transforms doubles the cost of each multiplication. Combined together, HEVC transform logic has 8x8 the computational complexity of H.264/AVC which affects both area and energy.
3. An intermediate memory is needed to store the TU between the column and row transforms operation. This memory must perform a transposition i.e. columns are written to it and rows are read out. Previous designs for H.264/AVC used register arrays due to the small TU sizes. These do not scale very well to the higher TU sizes of HEVC and one must look to denser memories such as SRAM to achieve an area-efficient implementation. However, the higher density of SRAMs comes at the cost of lesser memory throughput and lesser flexibility in read-write patterns. In this paper, we propose the following techniques to improve the throughput, energy and area of an HEVC inverse. The DISCRETE cosine transform (DCT) plays a vital role in video compression due to its near-optimal de correlation efficiency. Several variations of integer DCT have been suggested in the last two decades to reduce the computational complexity. The new H.265/High Efficiency Video Coding (HEVC) standard has been recently finalized and poised to replace H.264/AVC . Some hardware architectures for the integer DCT for HEVC have also been proposed for its real-time implementation. The DCT matrices into sparse sub-matrices where the

multiplications are avoided by using the lifting scheme. It uses the multiplier less multiple constant multiplication (MCM) approach for four-point and eight-point DCT, and have used the normal multipliers with sharing techniques for 16 and 32-point DCTs. factorization of DCT where the butterfly operation has been implemented by the processing element with only shifters. ECAT operates shifting and addition in parallel by unrolling all the words required to be computed. Furthermore, the Error-Compensated Circuit alleviates the truncation error for high accuracy design. Based on low-error ECAT, the DA-precision in this work is chosen to be 4 bits instead of the traditional 9 bits. Therefore, the hardware size and cost is reduced, and the speed is improved using the proposed ECAT.Also we reduces the time delay than previous method.

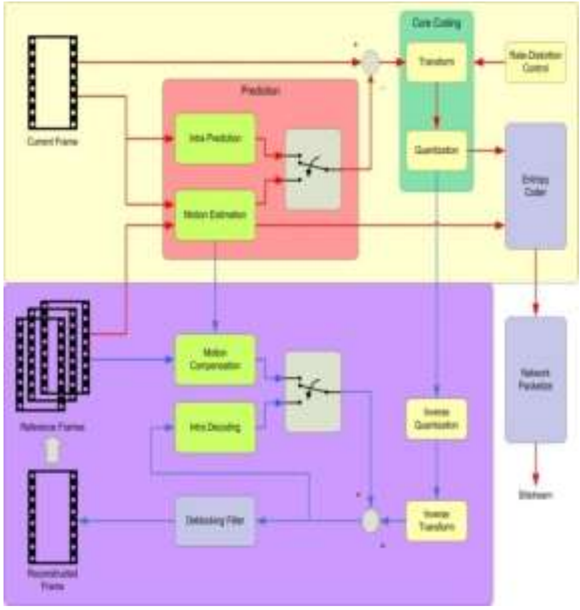


Fig 1: Basic Video Encoding Processing.

Data Compression.

Compression is used just about everywhere. All the images you get on the web are compressed, typically in the JPEG or GIF formats, most modems use compression, HDTV will be compressed using MPEG-2, and several file systems automatically compress files when stored, and the rest of us do it by hand. The neat thing about compression, as with the other topics we will cover in this course, is that the algorithms used in the real world make heavy use of a wide set of algorithmic tools, including sorting, hash tables, tries, and FFTs. Furthermore, algorithms with strong theoretical foundations play a critical role in real-world applications. In digital signal processing, data compression, source coding, or bit-rate reduction involves encoding information using fewer bits than the original representation. Compression can be either lossy or lossless. Compression reduces bits by identifying and eliminating redundancy. No information is lost in lossless compression. Lossy compression reduces bits by identifying unnecessary information and removing it.[3] The process of reducing the size of a data file is referred to as data compression. In the context of data transmission, it is called source coding (encoding done at the source of the data before it is stored or transmitted) in opposition to channel coding. Compression is useful because it helps reduce resource usage, such as data storage space or transmission capacity. Because compressed data must be decompressed to use, this extra processing imposes computational or other costs through decompression; this situation is far from being a free lunch. Data compression is subject to a complexity trade-off. For instance, a compression scheme for video may require expensive hardware for the video to be decompressed fast enough to be viewed as it is being decompressed, and the option to decompress the video in full before watching it may be inconvenient or require additional storage. The design of data compression schemes involves trade-offs among various factors, including the degree of compression, the amount of distortion introduced (when using lossy data compression), and the computational resources required to compress and uncompress the data. Basically the compression has two types of methods 1).Losyy compression method 2), Lossless Compression method. Lossless algorithms usually exploit statistical redundancy to represent data more concisely without losing information, so that the process is reversible. Lossless compression is possible because most real-world data has statistical redundancy. Lossy data compression is the converse of lossless data compression. In these schemes, some loss of information is acceptable. Dropping nonessential detail from the data source can save storage space. Lossy data compression schemes are informed by research on how people perceive the data in question. Also for video data coding there are several techniques are available like huff man coding and DCT and etc.In this paper we proposed we mainly focus on DCT compression method. The below figure shows the basic video compression block diagram.

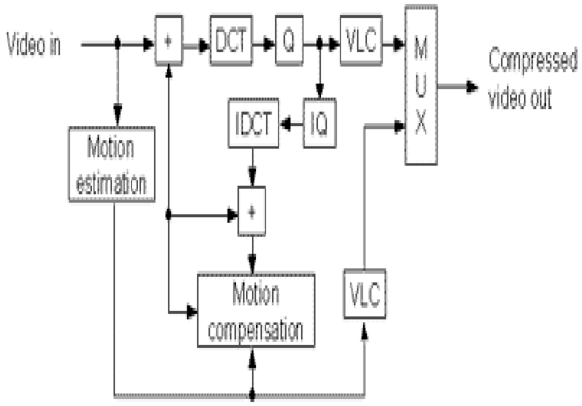


Fig 2: Basic video compression method using DCT.

DISCRETE COSINE TRANSFORM

A discrete cosine transform (DCT) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications in science and engineering, from lossy compression of audio (e.g. MP3) and images (e.g. JPEG) (where small high-frequency components can be discarded), to spectral methods for the numerical solution of partial differential euations.The use of cosine rather than sine functions is critical for compression, since it turns out (as described below) that fewer cosine functions are needed to approximate a typical signal, whereas for differential equations the cosines express a particular choice of conditions. In particular, a DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), where in some variants the input and/or output data are shifted by half a sample. There are eight standard DCT variants, of which four are common. The most common variant of discrete cosine transform is the type-II DCT, which is often called simply "the DCT",[1][2] its inverse, the type-III DCT, is correspondingly often called simply "the inverse DCT" or "the IDCT". Two related transforms are the discrete sines transform (DST), which is equivalent to a DFT of real and odd functions, and the modified discrete cosines transform (MDCT), which is based on a DCT of overlapping data.

DCT-I

$$X_k = \frac{1}{2}\left(x_0 + (-1)^k x_{N-1}\right) + \sum_{n=1}^{N-2} x_n \cos\left[\frac{\pi}{N-1} nk\right] \qquad k = 0, \ldots, N-1.$$

Some authors further multiply the x_0 and x_{N-1} terms by √2, and correspondingly multiply the X_0 and X_{N-1} terms by 1/√2. This makes the DCT-I matrix orthogonal, if one further multiplies by an overall scale factor of $\sqrt{2/(N-1)}$, but breaks the direct correspondence with a real-even DFT.The DCT-I is exactly equivalent (up to an overall scale factor of 2), to a DFT of $2N-2$ real numbers with even symmetry. For example, a DCT-I of N=5 real numbers *abcde* is exactly equivalent to a DFT of eight real numbers *abcdedcb* (even symmetry), divided by two. (In contrast, DCT types II-IV involve a half-sample shift in the equivalent DFT.)Note, however, that the DCT-I is not defined for N less than 2. (All other DCT types are defined for any positive N.)Thus, the DCT-I corresponds to the boundary conditions: x_n is even around n=0 and even around n=N-1; similarly for X_k.

DCT-II

The DCT-II is probably the most commonly used form, and is often simply referred to as "the DCT".This transform is exactly equivalent (up to an overall scale factor of 2) to a DFT of $4N$ real inputs of even symmetry where the even-indexed elements are zero.

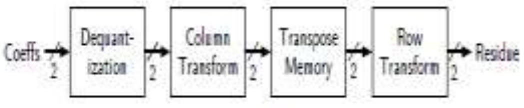
$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \qquad k = 0, \ldots, N-1.$$

That is, it is half of the DFT of the $4N$ inputs y_n ,where $y_{2n} = 0$, $y_{2n+1} = x_n$ for $0 \leq n < N$, $y_{2N} = 0$ and $y_{4N-n} = y_n$ for $0 < n < 2N$.

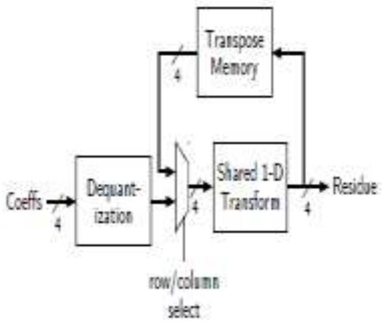
Some authors further multiply the X_0 term by 1/√2 and multiply the resulting matrix by an overall scale factor of $\sqrt{2/N}$ (see below for the corresponding change in DCT-III). This makes the DCT-II matrix orthogonal, but breaks the direct correspondence with a real-even DFT of half-shifted input. This is the normalization used by Mat lab, for example. In many applications, such as JPEG, the scaling is arbitrary because scale factors can be combined with a subsequent computational step (e.g. the quantization step in JPEG$^{)}$, and a scaling that can be chosen that allows the DCT to be computed with fewer multiplications.The DCT-II implies the boundary conditions: x_n is even around n=-1/2 and even around n=N-1/2; X_k is even around k=0 and odd around k=N.

Applications of DCT

The DCT, and in particular the DCT-II, is often used in signal and image processing, especially for lossy compression, because it has a strong "energy compaction" property in typical applications, most of the signal information tends to be concentrated in a few low-frequency components of the DCT. For strongly correlated Markov processes, the DCT can approach the compaction efficiency of the Karhunen-Loève transform (which is optimal in the de correlation sense). As explained below, this stems from the boundary conditions implicit in the cosine functions. A related transform, the *modified* discrete cosine transform, or MDCT (based on the DCT-IV), is used in AAC, Vorbis, WMA, and MP3 audio compression.DCTs are also widely employed in solving partial differential equations by spectral methods, where the different variants of the DCT correspond to slightly different even/odd boundary conditions at the two ends of the array. The DCT is used in JPEG image compression, MJPEG,MPEG, DV, Daala.



(a) Separate row and column transforms



(b) Shared transform block

Fig. 1: Possible high-level architectures for transform(bus-widths are in co efficients/residue pixels). To achieve an overall 2 pixel/cycle throughput, the shared 1-D transform block must be designed for 4 pixel/cycle.4x4 TUs. This is to avoid a stall when a 4x4 TU immediately follows a 32x32 TU. After the last row of the 32x32 TU is computed, it takes 8 cycles to write it out. During those cycles, the partial 1-D transform block computes the column transforms of the 4_4 TU and is ready to write the first row out after 4 cycles itself. To avoid stalling the pipeline while the last row of the 32x32 TU is being written out, the residues of the 4_4 TU are saved in the separate residue FIFO. The proposed design implements zero-column skipping based on IDCT pruning in which the 1-D transform is skipped for columns that have all zero coefficients. This reduces cycle-count by 27% to 66%. TUs with larger sizes and higher quantization benefit more from zero-column skipping since they have a higher proportion of all-zero columns. Zero-column skipping also improves energy/pixel by avoiding any switching to zero. For example, when processing 2000 TUs (173360 pixels with a mixture of all TU sizes) from the Park Scene test sequence at QP 32, zero-column skipping reduces the cycle count by 38% and energy/pixel by 29%.

DIGITAL VIDEO PROCESSING

Digital video processing has been developed to handle each of the techniques of broadcast, interlacing, and recording as well as Mr. Kell's technique of predictive compression. So many standards for digital video compression are in wide use today that television and digital video device makers find it necessary and preferable to use programmable digital video processors such as the ARC Video subsystem within their designs. We will give an overview of several notable compression standards and techniques below.

H.120

In1984,the International Telecommunication Union (ITU) standardized recommendation H.120, the first digital video coding technology standard. H.120 used differential pulse code modulation (DPCM), scalar quantization and variable-length coding techniques to transmit NTSC or PAL video over dedicated point-to-point data communications lines. The application target then and for all ITU H.xxx standards since is video conferencing. H.120 is essentially deprecated today.

H.261

Practical digital video compression started with the ITU H.261 standard in 1990. The target was to transmit video over ISDN lines, with multiples of 64 Kbit/s data rates and CIF (352x288-pixel) or QCIF (176x144-pixel) resolution. The standard was a pioneering effort and used a hybrid video coding scheme that is still the basis for many video coding standards today.

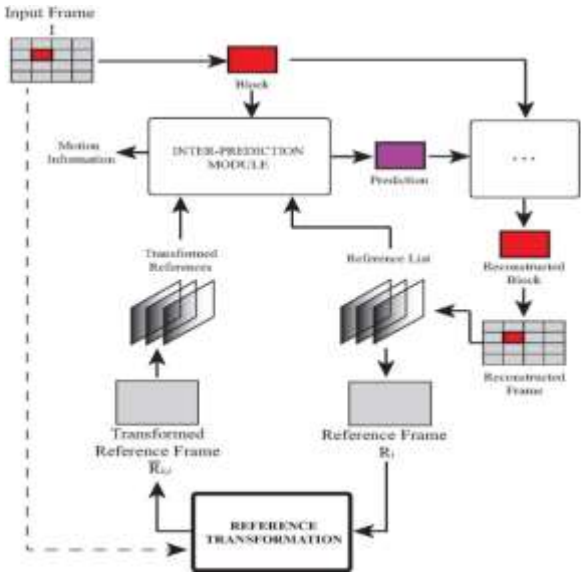


Fig : Basic Hybrid Video Encoder.

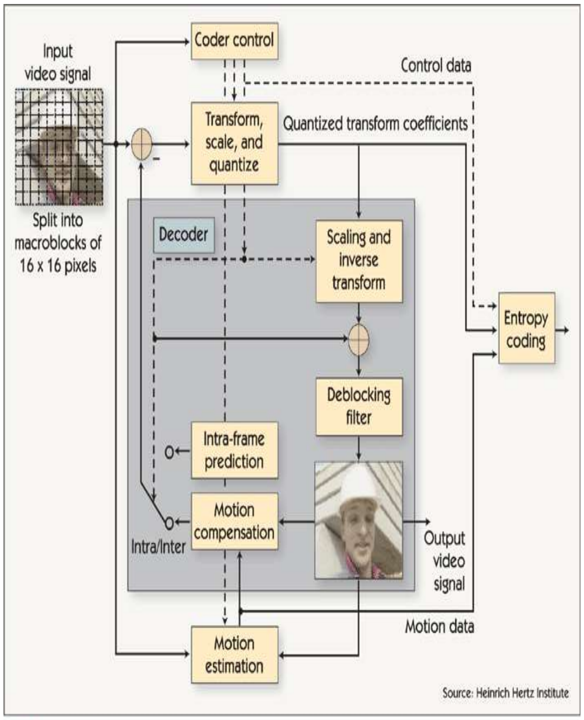
The standard was a pioneering effort and used a hybrid video coding scheme that is still the basis for many video coding standards today. Hybrid video coding combines two methods:

1. Motion from frame to frame is estimated and compensated for by predicting data from previously coded frames.

2. The residual difference after prediction is Encoded by decorrelating the data in the spatial domain through transformation to the 2D frequency domain. The transformed data are quantized, a stage in which information is lost, after which the data are encoded with a lossless compression scheme such as Huffman coding or an arithmetic coder. H.261 uses 4:2:0 data sampling, in which there are twice as many luminance samples as chrominance samples. The human eye is more sensitive to light intensities than to color. H.261 has 16x16-pixel macro blocks with motion compensation, an 8x8-pixel discrete cosine transform (DCT), scalar quantization, zigzag scanning and Huffman-based variable-length entropy coding. The standard operates at 64–2048 Kbit/s. H.261 is still in use but has largely been overtaken by H.263.The organizational structure and processes used by the H.261 international committee to develop the standard have remained the basic operating process for subsequent standardization efforts.

H.264 / MPEG-4 part 10 / AVC

Where the original MPEG-4 standardization Committees put a lot of effort into novel media coding techniques; H.264 took a "back to basics" approach. The goal was to compress video at twice the rate of previous video standards while retaining the same picture quality. The standard was originally known as H.26L or JVT, for the Joint Video Team, in which the ISO and ITU organizations worked together to complete the standardization.



Source: Heinrich Hertz Institute

Fig. 2 H.264 macroblock coding structure.

Figure: Typical Structure of an H.264/MPEG4-AVC Video Encoder.

H.264 is the ITU name for the standard; MPEG-4 part 10, Advanced Video Coding (AVC) is the ISO name. Due to its improved compression quality, H.264 is quickly becoming the leading standard; it has been adopted in many video coding applications such as the iPod and the Play station Portable, as well as in TV broadcasting standards such as DVB-H and DMB. Portable applications primarily use the Baseline Profile up to SD resolutions, while high-end video coding applications such as set-top boxes, Blue ray and HD-DVD use the Main or High Profile at HD resolutions. The Baseline Profile does not support interlaced content; the higher profiles do.

Current standardization efforts center around extending today's standards rather than developing completely new video coding methods. There's an effort to add coding of multiple views to H.264 for 3D video. Scalability, which allows streams to be encoded once but transmitted and decoded at Different resolutions and frame rates, is also an area of research and standardization. Both MPEG-2 and MPEG-4 have scalable profiles, but these have not been widely adopted by the industry. It remains to be seen whether an H.265 standard will follow the H.261 to H.264 series of increasingly complex advanced standards. The major goal would likely be a further 50% savings in bandwidth. One way to achieve this is to focus on a perceptual quality metric, replacing the quantitative peak-signal-to noise ratio metric that has been used in the past.

Advantages

- Better Reliability.
- Low power consumption.
- Less Execution time.

CONCLUSION

Proudly we conclude this paper we implemented a new Error compensation adder with high performance and less time delay by using Xilinx 13.1 Version. The new H.264/MPEG4-AVC video coding standard was developed and standardized collaboratively with only 4-bit compensation .but in previous method that is 9 bit compensation.H.264/MPEG represents the of advances in standardized video coding technology, in terms of both coding efficiency enhancement and flexibility for effective use over a broad variety of network types and application domains. Its video coding layer design is based on conventional block-based motion-compensated hybrid video coding concepts, but with some important innovations relative to prior standards. Also finally we conclude this paper in the view of execution time i.e also less when we are comparing with previous paper.

REFERENCES

[1] M. Budagavi, A. Fuldseth, G. Bjontegaard, V. Sze, M. Sadafale, "Core Transform Design for the High Efficiency.Video Coding (HEVC) Standard," Selected Topics in Signal Processing, IEEE Journal of, vol. 7, no. 6, pp.1029–1041, 2013.

[2] M. Potkonjak, M.B. Srivastava, and A. Chandrakasan,"Efficient Substitution of Multiple Constant Multiplications by Shifts and Additions using Iterative PairwiseMatching," in Design Automation, 1994. 31st Conference on, june 1994, pp. 189 – 194.

[3] M. Tikekar, C.-T. Huang, C. Juvekar, and Chandrakasan .A., "JCTVC-G265: Core transform property for practical throughput hardware design," Joint Collaborative Team on Video Coding (JCT-VC), 2011.

[4] D. F. Finchelstein, Low-power Techniques for Video Decoding.Thesis, Massachusetts Institute of Technology,2009.

[5] M. Budagavi and V. Sze, "IDCT pruning and scan dependent transform order," Joint Collaborative Team on Video Coding (JCT-VC), 2011.

[6] M. Tikekar, Chao-Tsung Huang, C. Juvekar, V. Sze, and A.P. Chandrakasan, "A 249-Mpixel/s HEVC Video-Decoder Chip for 4K Ultra-HD Applications," Solid-State Circuits, IEEE Journal of, vol. 49, no. 1, pp. 61–72,2014.

[7] T. Xanthopoulos, Low Power Data-Dependent Transform Video and Still Image Coding, Thesis, Massachusetts Institute of Technology, 1999.

[8].Gary J.Sullivan, Pankaj Topiwala, and Ajay Luthra, "The H.264/AVC Advanced Video Coding Standard:Overview and Introduction to the Fidelity Range Extensions" SPIE Conference on Applications of Digital Image Processing XXVII. Special Session on Advances in the New Emerging Standard: H.264/AVC, August, 2004.

[9].Gary J. Sullivan and Jens-Rainer Ohm "Recent developments in standardization of high efficiency video coding (HEVC)"SPIE.