Combinational Logic Optimization for a Hardware based HEVC Transform

Anish Tamse, Hyuk Jae Lee, *Chae Eun Rhee,
Seoul National University, *Inha University

Abstract

In a 2-dimensional (2D) Discrete Cosine Transform (DCT) hardware, a significant fraction of the total hardware area is contributed by the combinational logic used to perform 1-dimensional (2D) transform. The size of the non-combinational logic i.e. the transpose memory is dictated by the size of the largest transform supported. Hence, the optimization of hardware area is performed mainly for 1D-transform combinational logic. This paper demonstrates the use of Multiple Constant Multiplication (MCM) algorithm to reduce the combinational logic area. Partial optimizations are also described for the cases where the direct use of MCM algorithm doesn't meet the timing constraint. Experimental results show that 46% improvement in gate count is achieved for 32 point 1D DCT transform logic after using MCM optimization.

1. Introduction

High Efficiency Video Coding (HEVC) is the successor to the H.264 coding standard. The HEVC standard achieves a significantly higher compression performance compared to its predecessor. One of the major reasons for this is that HEVC supports a large size of 2-dimensional (2D) Discrete Cosine Transform (DCT) ranging from 4×4 to 32×32. However, this large size of DCT increases the hardware complexity significantly. Hence an efficient transform architecture design is an important issue to be tackled for a hardware-based HEVC encoder.

The DCT for residual matrices in HEVC is performed by multiplying them with DCT integer coefficient matrices of corresponding sizes. These coefficient matrices in HEVC are chosen such that the 2D transform can also be performed equivalently by a 1-dimensional (1D) row transform followed by a 1D column transform [1]. Hence, the transform architectures generally consist of three components, a combinational 1D row transform, a transpose memory and a combinational 1D column transform [2]. The size of the transpose memory required for a given transform architecture is dictated by the size of the largest transform it supports. Hence reduction in the hardware area of transpose memory is not feasible. Therefore, 1D transform logic needs to be optimized to improve the hardware cost. The 1D transform implements multiplication of coefficients to the input residuals and thus, is combinational in nature. 1D transform modules contribute a significant fraction of the total hardware cost of the complete architecture (approximately 35% for 32 point DCT architecture; and even higher for lower sized transforms as the area taken up by transpose memory reduces a lot). Therefore, optimizing its hardware area does contribute to major reduction in overall hardware area.

This paper presents the optimization techniques that can be applied to the 1D transform modules. The first optimization is the partial butterfly transform implementation which reduces the total number of multiplications required. The next optimization is to use algorithm for Multiple Constant Multiplication (MCM) [3] which reduces gate count significantly. The focus of this paper will be on using the MCM algorithm. The MCM algorithm allows for the trade-off between the levels of gates and the total number of gates. When implemented in hardware, this translates to the amount of latency versus the hardware area.

2. 1D - DCT TRANSFORM

The DCT integer coefficients are chosen in HEVC such a manner that the 2D transform calculation can be broken down into two equivalent 1D transforms [1]. This 1D transform when being implemented in hardware can be further optimized as follows.

A. Partial Butterfly Transform

1D transformation calculation for an n-point DCT involves multiplication of n coefficients with one residual value each to generate one output transformed coefficient. Hence, there are n^2 multiplications in total. However, the calculations can be optimized mathematically to yield the same result with much fewer multiplications by performing smaller transforms and using the result for the larger transform, called the butterfly structure. The DCT for HEVC allows a partial butterfly structure for generating the coefficients. This implementation reduces the number of calculations to approximately $n^2/3$. The partial butterfly implementation for DCT used in the HM reference software is used as the reference implementation in this paper. Since the number of multiplications is reduced, it causes an equivalent reduction in hardware cost as the multipliers are the major contributors to the hardware area.

B. Multiple Constant Multiplication

Once the number of multiplications required is reduced

through the partial butterfly structure, the next step is to implement the multiplications. Firstly, since the coefficients for multiplications are constants and are known beforehand, these multiplications can be implemented with just shifting and addition operations, which reduces the gate count. For the next optimization, in a discrete transform, the number of coefficients is fixed and the input residuals are shifted by one position every time and multiplied by the coefficient at that position to generate the transformed coefficients. Thus, for generating n point transform, each residual is shifted n times, and is multiplied with each corresponding coefficient. Therefore each residual is multiplied with same set of constants. Thus, a module needs to be designed which, given an input, gives the results of multiplication with these constants. Optimizing the gate count of such a module is a well-studied problem called Multiple Constant Multiplication. The problem of finding the most optimal gate structure is a non-deterministic polynomial (NP) hard problem and thus, various heuristic algorithms exist. The basic idea while optimizing such a module is that if a certain sub-expression exists for two different multiplications, it is calculated only once and the result is used in both places as shown in Fig. 1.

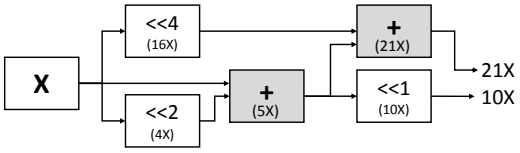

Fig. 1. Sub-expression reuse demonstration

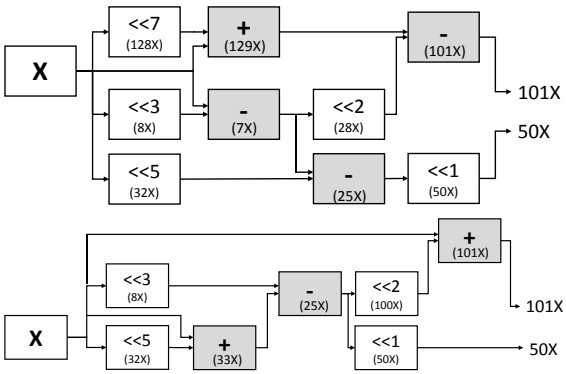

Fig. 2. Adder depth of two and three to generate result of multiplication with 101 and 50. Using depth of three requires one shifter and one adder less.

The most optimal MCM implementation will have many sub-expressions being reused. This will lead to multiple levels of adders. Considering the trade-off between gate count and the path latency, one option is to have the same implementation processed in multiple stages in a pipelined manner. This will add delay in the final result arrival and require the use of multiple registers to store the intermediate results. However, if the timing requirement is not missed by a large margin, the number of gate levels can be reduced while increasing the total gate count slightly. As shown in Fig. 2, the same logic can be implemented using different depths of adders, while trading off with the total gates required. One algorithm for MCM [3] allows for the limit on the maximum depth of adders in the design. With this constraint, the levels of gates can be reduced so that the latency in the combinational logic meets the timing requirement.

2. Experimental Results

The implementation is done in Verilog and synthesized with TSMC 65nm technology. This section shows the experiments performed using the optimization technique for the 1D DCT transform in HEVC with different adder depth constraints. The use of MCM algorithm is demonstrated for the 32 point DCT. The partial butterfly implementation of HM11 is used as the starting point for 1D row and column transforms. Different constraints on the number of gate level are enforced and the total gate count is calculated for each case. The result is summarized in Table 1.

Table 1. Experimental result summary

Depth of Adders	Total Gate Count	Maximum Clock frequency
2	139K	279 MHz
3	124K	273 MHz
4	120K	269 MHz

2. Conclusion

The combinational logic required by 1D DCT transform contributes a significant fraction to the overall hardware area. It can be effectively optimized using algorithm for MCM. The gate count is reduced from 223K when MCM isn't used to 120K for 32 point DCT transform after using the MCM optimization. The maximum clock frequency possible for operation for the corresponding two cases reduces from 304MHz to 269MHz. Since the algorithm also allows for tradeoff between area and gate levels, partial optimizations can also be performed for in-between clock frequencies as shown in the previous section. The MCM algorithm described is generic and can be also applied to other types of transforms as well as other sizes of DCT.

Acknowledgement

This research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)(NIPA-2013-H0301-13-1011).

REFERENCES

[1] M. Budagavi, A. Fuldseth, G. Bjontegaard, V. Sze and M. Sadafale, " Core Transform Design in the High Efficiency Video Coding (HEVC) Standard," IEEE Journal of Selected Topics in Signal Processing, 7(6):1029-1041, 2013

[2] J. D. Bruguera, R. R. Osorio, " A Unified Architecture for H.264 Multiple Block-Size DCT with Fast and Low Cost Quantization," 9th EUROMICRO Conference on Digital System Design: Architectures, Methods and Tools, 407-414, 2006

[3] A. G. Dempster and M. D. Macleod, " Constant integer multiplication using minimum adders," IEEE Proceedings - Circuits, Devices and Systems, 141(5):407-413, 1994