DATA MAPPING SCHEME AND IMPLEMENTATION FOR HIGH-THROUGHPUT DCT/IDCT TRANSPOSE MEMORY

Zheng Xie, Yanheng Lu, Yibo Fan , Xiaoyang, Zeng*
School of Microelectronics, Fudan University, Shanghai 200433, CHINA
*Corresponding Author's Email: fanyibo@fudan.edu.cn

ABSTRACT

In this paper, we proposed a generalized architecture for hardware implementation of the single port SRAM-based transpose memory for large size DCT/IDCT. Instead of shift-register array or multiport SRAM, only single-port SRAM is used in the proposed design. A novel data mapping scheme based on the theory of transpose of partitioned matrix is proposed to implement the transpose memory with less SRAM banks. Row access and column access can be perfectly supported under single port SRAM. This design can support DCT/IDCT of different transform sizes with different data throughput rates. Compared with the existed design [4], the proposed design can achieve 44.3% area saving. It is suitable for real-time processing of the video with the resolution up to 7680x4320 UHD.

INTRODUCTION

2-D DCT/IDCT have been widely used in many video coding standards such as MPEG-1/2/4, H.264, VC-1, AVS and the latest HEVC. To further improve the coding efficiency, 16 × 16 and 32 × 32 2-D integer DCT/IDCT are also used in HEVC.

Large size 2-D transform can be decomposed into two separate steps. First, the row-direction 1-D DCT/IDCT is performed. Then the column-direction 1-D DCT/IDCT is performed for the intermediate results of the row direction transform. Thus, a transpose memory is needed to get the transpose of the intermediate results in the first step.

Register-array based transpose memory is not area efficient for large size transform. Assumed each intermediate result stored in the transpose memory is 16-bit, a 4x4 2D DCT requires a 256-bit register array which is narrowly affordable. However, a 32x32 2D DCT will require a 16384-bit register array. SRAM is more area efficient than registers when massive amount of data are to be stored. So SRAM based transpose memory is a good choice for implementation of large size transform memory.

Single-port SRAM is used in [1]. The transpose memory is divided into four banks and data throughput is only two samples per cycle. In [2], 32 banks are used to get a read throughput of 32-samples per cycle, but the write throughput is only 4-samples per cycle. In [3], a diagonal data mapping scheme is proposed to implement single-port SRAM based transpose memory for Large Size 2-D DCT/IDCT. The diagonal data mapping scheme can reduce the number of SRAM banks used to implement the transpose memory. But the mapping scheme can only get a throughput which is not higher than N (the size of transform).

To support Ultra-High-Definition (UHD) 7680x4320 video, high throughput is desired for the regular DCT/IDCT used in video coding process. The design of 1D_DCT/IDCT in [4][5] can get a throughput of 32-samples per cycle irrespective of the size of transform unit(TU). This requires the transpose memory have a write and read throughput of 32-samples per cycle irrespective of transform size. The diagonal data mapping scheme is no longer applicable. To solve the problem, a novel data mapping scheme based on the theory of transpose of partitioned matrix is proposed. Also a generalized architecture for single port SRAM based transpose memory is presented.

GENERALIZED ARCHITECTURE FOR SINGLE PORT SRAM BASED TRANSPOSE MEMORY

The transpose memory is accessed row by row when the results of row DCT/IDCT are written into it and accessed column by column when the column DCT/IDCT module read data from it. It is very easy to access either a row or a column of registers in a register array. While the data can only be accessed in row direction in single port SRAM, the data of row/column must be stored in a mapping rule so that the data of column/row can be fetched out later.

The essence of different mapping scheme is making each data of the matrix in specified word of the specified bank. It can be achieved by two address: ADD and BADD. ADD is the address of word the data will be read or write; BADD is the number of bank the data will be stored, it can be implemented by rearrange the sequence of input or output.

Fig.1 shows the proposed generalized architecture for implementing the single-port SRAM based transpose memory. The 2:1MUXes is used for read and write, "0" denotes the process of writing data into SRAM and "1" denotes the process of reading data from SRAM. The architecture consists of three parts: address generation module (AGM), MUX_N array module (MAM), single-port SRAM.

The read and write throughput of every bank in the single port SRAM is 1-sample per cycle. Based on the write and read throughput of transpose memory, the SRAM should be physically divided into several banks. The number of banks used in the transpose memory equals the maximum value between write throughput and read throughput.

AGM is used to produce ADD and BADD based on the mapping rule. Every data need a pair of (ADD, BADD) to specify the bank and word to be write in or read out. So $2*N^2$

pairs of (ADD, BADD) should be produced for a NxN matrix. So a good mapping scheme is important for decreasing the implementation complexity of AGM. Using the diagonal data mapping scheme for a NxN transpose memory with a throughput of N-sample per cycle. When the row data is written, the add0, add1,⋯, add(N-1) can be 0,1,⋯, N-1 in every cycle and the badd0,badd1, ⋯, badd(N-1) can be produced by right or left shift the sequence of 0,1,2, ⋯ (N-1).when the column data is read, the add0,add1, ⋯ add(N-1) is equal to the number of column and the badd0,badd1,⋯, badd(N-1) can be produced by shift the sequence of 0,1,2,⋯(N-1) in a opposite direction with that in writing.

MAM is an array of MUXs used to specify bank for the data to be written or sort the data read from SRAM to get a right sequence in row/column based on BADD. The MUX is N:1 and N is the number of banks.

The proposed generalized architecture is flexible to support DCT/IDCT of different TU sizes and data throughputs by changing number and depth of banks. Modifying the AGM module, different mapping scheme can be implemented.

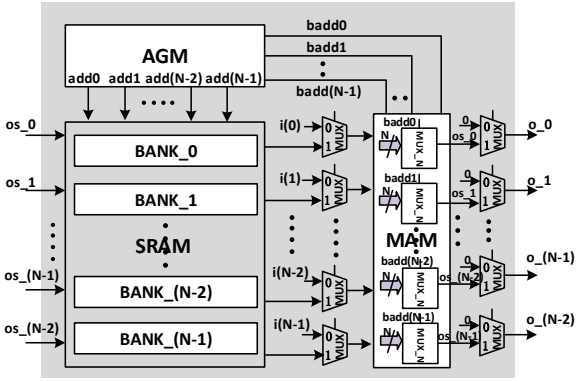

Fig.1. The generalized architecture for single-port SRAM based transpose memory;

A DATA MAPPING SCHEME BASED ON TRANSPOSE OF PARTITIONED MATRIX

In this section, the theory for transpose of partitioned matrix is introduced first. Then, a novel data mapping scheme based on the theory is proposed. Using the generalized architecture with the proposed data mapping scheme, an architecture which can support the DCT/IDCT design of [4] [5] is presented.

The theory for transpose of partitioned matrix.

It may be complicated to get the transpose of a large size matrix directly**,** the complexity can be decreased by partitioning the large size matrix into small size block matrices. Then the transpose of the large size matrix can be got by transposing the partitioned matrices and the sub-matrices respectively [6]. It can be described by equation-(1) with an example of size 4x4 matrix.

$$A_4 = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix}$$

$$A_4{}^T = \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix}^T = \begin{bmatrix} A_{00}{}^T & A_{10}{}^T \\ A_{01}{}^T & A_{11}{}^T \end{bmatrix} \quad (1)$$

The transpose of a MxM matrix can be got as follows:

- Partition the MxM matrix into small size NxN matrix and a (M/N)x(M/N) partitioned matrix is produced.
- Take a size NxN matrix as a basic unit and get the transpose of the (M/N)x(M/N) partitioned matrix first and then do transpose for every small NxN matrix.

The data mapping scheme based on transpose of partitioned matrix.

For the size NxN input matrix, if throughput (TP) is higher than N, the input data is (TP/N)-rows/columns per cycle. The transpose can be got in a way same with getting transpose of the partitioned matrix.

According to the generalized architecture, SRAM should be divided into TP banks with a depth of N^2/TP. The TP banks of the transpose memory are divided into N^2/TP groups, every group consist of $(TP/N)^2$ continuous banks, so the data of one sub-matrix can be stored in one group per cycle.

(TP/N) blocks of size $(TP/N) * (TP/N)$ are input in rows/ columns as the input data. The blocks are stored in the groups as the diagonal data mapping scheme. After the NxN matrix is stored in SRAM. The blocks are also read as the diagonal data mapping scheme. The transpose of blocks are get. The transpose of $(TP/N) * (TP/N)$ sub-matrices can be implemented by MAM, so the transpose of size NxN matrix can be achieved.

The transpose of size NxN matrix can be implemented as follows:

- Sorting the TP-point input as continuous blocks by MAM.
- Storing the row/column blocks in groups as the diagonal data mapping scheme.
- Reading column/row blocks in groups as the diagonal data mapping scheme, too.
- The transpose of the blocks are implemented by

MAM and the sequence of data is also rearranged.

A high throughput transpose memory for parallel architecture of DCT/IDCT

To meet the requirement of the DCT/IDCT design in [4][5] , the transpose memory should support to write or read eight 4x4 rows/columns, four 8x8 rows/columns, two 16x16 rows/columns and one 32x32 rows/columns per cycle.

To support size 32x32 matrix with a write and read throughput of 32 samples per cycle, the SRAM should be physically divided into 32 banks with depth of 32. The MAU consists of 32 MUXs, which is 32 to 1.

The transpose of two 4x4 blocks can be got by rearranging the sequence of input data through MAU without storing in SRAM. The 8x8 input matrix is partitioned into 2x2 block matrix with size 4x4 and the banks are divided into two groups: bank_0-bank_15 is group_0 and bank_16-bank_31 is group_1. The 16x16 input matrix is partitioned into 8x8 block matrix with size 2x2 and the banks are divided into eight groups consist of two continuous banks. The 32x32 matrix can be directly mapped as the diagonal data mapping scheme. The block matrix is mapped as diagonal data mapping scheme. In every group, the data have same ADD and successive BADD, this can simplify the hardware implementation of AGM.

IMPLEMENTATION RESULTS

The proposed single port SRAM based transpose memory to support DCT/IDCT design [4][5] is implemented in Verilog HDL and synthesized with the SMIC 0.13 μm standard cell library. The single port register file generators provided by VeriSilicon are used to generate the required SRAM banks for SMIC 0.13 μm process. The data stored in transpose memory is 16-bit. We have compared the proposed architecture with the existing architecture [2] [4].

TABLE I. area and performance comparing with exist architecture

design	[2]	[4]	proposed
Tech.	0.13um	90nm	0.13um
Gate count.	60.4k	105k	58.5k
Freq.	187	187	250
WTP/RTP.	4/32	32/32	32/32
Register	×	√	×
1-port SRAM	√	×	√

TABLE_I shows the comparing results of area and performance between the proposed architecture and the existed architecture. Comparing with the design [2], WTP of the proposed design is 32-sample per cycle, which is higher than design [2] with a WTP of 4-sample per cycle. The design [4] has the same throughput with proposed design, but the gate count of [4] is 44.3% more than the gate count of the proposed design.

SUMMARY AND CONCLUSIONS

In this paper, we have proposed a generalized architecture for hardware implementation of single-port SRAM based transpose memory. Also a novel data mapping scheme has been proposed to simplify the data mapping implementation and get a high throughput with less banks. The architecture can be used in parallel architecture of DCT/IDCT. The experiment results show that the single port SRAM based transpose memory can achieve 44.3% area saving comparing to register-array based transpose memory with the same throughput.

ACKNOWLEDGEMENT

This paper is supported by National Natural Science Foundation of China (61306023), Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP, 20120071120021), STCSM (13511503400), National High Technology Research and Development Program (863, 2012AA012001).

REFERENCES

[1] K. B. Lee, H. C. Hsu, and C. W. Jen, “A cost-effective MPEG-4 shapeadaptive DCT with auto-aligned transpos e memory organization,” in Proc. Int. Symp. Circuits Sy st., vol. 2. May 2004, pp. 777–780.

[2] S. Shen, W. Shen, Y. Fan, and X. Zeng, “A unified 4/8/1 6/32-point integer IDCT architecture for multiple video coding standards,” in *Proc.IEEE ICME*, Jul. 2012, pp. 7 88–793.

[3]Shang,Q, Fan, Y, Shen,W, Shen, S, Zeng, X, 2014, "Very Large Scale Integration (VLSI) Systems," IEEE Transactions on, Issue: 99, Volume: PP .

[4] Pramod Kumar Meher, SangYoon Park, 2014," Efficient Integer DCT Architectures for HEVC," IEEE transaction s on circuits and systems for video technology, Issue: 1. Page(s): 168 – 178.

[5] SangYoonPark ,Meher,P.K, "Flexible integer DCT archi tectures for HEVC ," in *Proc.IEEE* ISCAS, 2013, pp. 1376 - 1379.

[6] Eves,Howard (1980). *ElementaryMatrixTheory* (reprint ed.). New York: Dover. p. 37. ISBN 0-486-63946-0.