The implementation of an efficient Zigzag scan

Article · April 2017

CITATIONS
8

READS
976

4 authors, including:

Sarifuddin Madenda
Universitas Gunadarma
177 PUBLICATIONS **563** CITATIONS

SEE PROFILE

Muhammad Subali
Universitas Gunadarma
6 PUBLICATIONS **8** CITATIONS

SEE PROFILE

The Implementation of an Efficient Zigzag Scan

Robby Candra[1], Sarifuddin Madenda[1], Sunny Arief Sudiro[2] and Muhammad Subali[3]
[1]Gunadarma University, Jakarta Indonesia.
[2]STMIK Jakarta STI&K, Jakarta Indonesia.
[3]Sekolah Tinggi Teknik Cendikia (STTC), Jakarta Indonesia.
robby.c@staff.gunadarma.ac.id

***Abstract*—The continuous growth of multimedia data capacity and the large capacity of communication lines had always become the obstacle in communication of multimedia data. To overcome this, data compression process in a shorter time is required. One part of JPEG algorithm that is able to determine compression ratio and compression process speed is Zigzag Coding or Zigzag Scan algorithm. This research is focused on developing the zigzag scan algorithm with mapping method. Zigzag Scan with mapping method is a sorting process of DCT-quantized data result according to the position sequence determined in a zigzag. Implementation of the Zigzag Scan mapping method into FPGA using ROM that serves as zigzag address generator and RAM that serves to write (put) and read (issued) data according zigzag address generated by ROM. The efficiency of Zigzag Scan with mapping method has been successfully developed. It is able to accelerate the sorting process of DCT-quantized coefficients period because input data can be immediately located in sequence position which has been determined without any value comparison and repetition process. Zigzag Scan with mapping method process time is 250 MHz or approximately 4ns per byte data (12 ns per pixel) with the delay time (latency) by 64 *clock*. It means that the generated IC Zigzag Scan prototype can be operated in real-time JPEG/MPEG compression with a maximum of 3 mega pixel per frame for video with 25 fps speed. The generated IC Zigzag Scan component (IP Core) needs 10 slices of Flip-flop (30 times less than Arafa method and 2 times less than Ketul method) and LUTs as many as 39 slices (20 times less than Arafa method and 2 times less that Ketul method).**

***Index Terms*—Compression; FPGA; Mapping; Zigzag Scan.**

I. Introduction

Today telecommunications facilities are not only used for sending message in voice or text form but also for multimedia data. The ever growing size of multimedia data, especially picture and video data, sometimes becomes the main obstacle in sending process due to communication lines capacity limitation and media storage availability. To overcome this obstacle, shorter time data compression process is required. One part of JPEG algorithm that is able to determine compression ratio and compression process speed is Zigzag Coding or Zigzag Scan algorithm. For video compression process speed in frame per second (fps), PAL (Phases Alternating Line) uses 30 fps, while NTCS (National Television System Commitee) uses 25 fps.

Zigzag Scan holds an important role in sequencing coefficient result of Discrete Cosine Transform (DCT)-quantized starting from the value related to low frequency information to value related to high frequency information. Matrix 8 x 8 of DTC-quantized result will be sequenced in 1 x 8^2 size vector, using zigzag reading. This sequencing greatly contributes to the compression ratio [3],[6].

The existing Zigzag scan algorithm has flaws, as mentioned in research by Arafa [2] and Ketul [3]. Arafa developed a zigzag scan algorithm using 40 memory (stack) elements to gather eight columns of input data with reading pattern based on zigzag pattern using insertion and coefficient shift method. The implementation of this zigzag scan method in IC-FPGA needs a lot of resources (303 Flip-flops and 811 LUTs), low speed process (86 MHz), and 320 clock cycles latency. In the algorithm developed by Ketul, there are three looping process and arithmetic counting (addition and division). Further, this algorithm has a condition to compare the result value of the arithmetic counting. The implementation of this zigzag scan method in IC-FPGA needs many resources (22 Flip-flops and 87 LUTs), slow speed process (234 MHz). In addition, the latency of this method is 960 clock cycles, indicating that the zigzag scan process needs more time.

Based on the researches developed by Arafa and Ketul, there is a need to speed up the compression process time and minimize the amount of resources for the implementation of zigzag scan into IC-FPGA. Thus, this research proposed to develop a zigzag scan algorithm with mapping method. Zigzag scan with mapping method is a process of placing DCT-quantized data result directly in accordance to the determined position order.

II. Methodology

Zigzag scan is a part of JPEG compression process (see Figure 1) that holds important role in grouping DCT process result and quantization values into DC low frequency and AC high frequency components. Zigzag scan contributes to the increase in picture and video compression ratio. This research discuss the efficient method of Zigzag scan algorithm implementation into electronic circuit that especially uses IC-FPGA equipment. This implementation result is directed to support a more efficient real-time JPEG compression process. In this case, efficient means faster process and using less circuit resources.

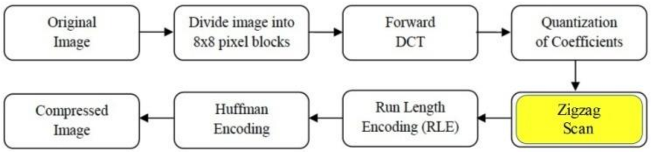
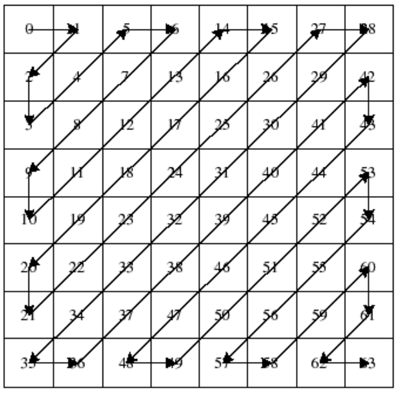

Original Image
Divide image into 8x8 pixel blocks
Forward DCT
Quantization of Coefficients
Zigzag Scan
Run Length Encoding (RLE)
Huffman Encoding
Compressed Image

Figure 1: Design of JPEG compression with efficient zigzag scan part [4]



0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Figure 2: Zigzag scan process on 8 x 8 picture [1]

Zigzag scan was positioned in the transformation procees from 8 x 8 matrix of quantification process result into 1 x 8^2 vector, through zigzag scan reading, as shown in Figure 2. Sorting index started from the top left coefficient and moved in the same direction with arrow in Figure 2, until it ended at the bottom right DTC matrix coefficient.

The efficiency of zigzag scan with mapping method is that it is able to accelerate the time used for data sorting to group the components from quantified coefficients. An overview of mapping process sequence is shown by scheme in Figure 3.

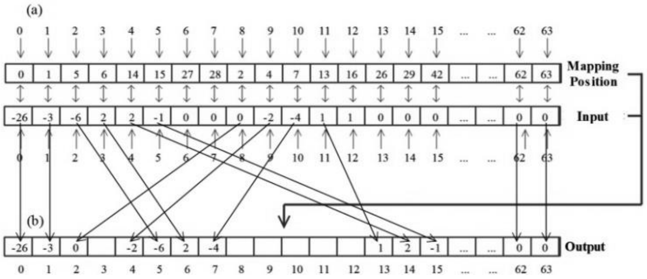


Figure 3: Diagram of zigzag scan with mapping method.

Mapping process is run between input data sequence and zigzag position sequence. The mapping happens according to the same element position between those two vectors (see mapping arrow in Figure 3). The end result of zigzag scan with mapping method is equal to the end result of previously developed zigzag scan method, as shown in Figure 2.

Based on a research done by Vijaya [5], for every 64 DCT coefficients from every 8 x 8 pixel block, those 64 DCT coefficients must be available before the scanning process. The reading or saving DCT coefficient input could slow down the scanning process. The architecture of zigzan scan in the research by Vijaya is as shown in Figure 4.

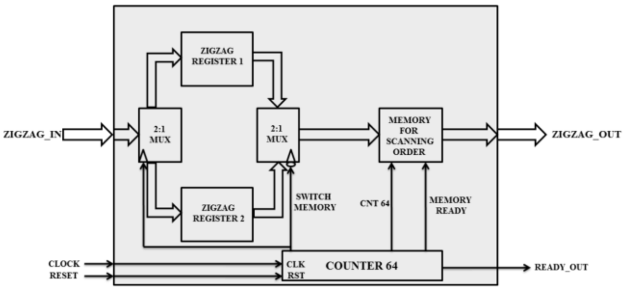


Figure 4: Architecture of zigzan scan [5]

To overcome problems that occured in research by Vijaya and to make faster scanning, new architecture design for zigzag scan with mapping method is proposed as shown in Figure 5. Implementation design of mapping method zigzag scan into FPGA using ROM which serves as zigzag address generator and RAM which serves to write (put) and read (issued) data according zigzag address generated by ROM.

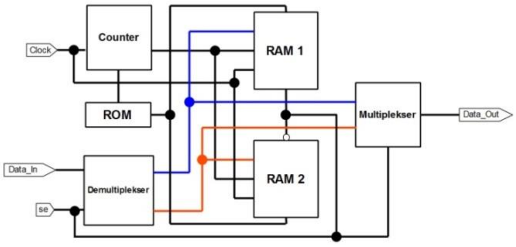


Figure 4: Implementation design of zigzag scan with mapping method into IC-FPGA

Data writing and reading process in RAM were carried out in turn between RAM 1 and RAM 2. When data writing process is conducted in RAM 1, data reading process is performed in RAM 2. On the other hand, when writing process is carried out in RAM 2, data reading process is carried out in RAM 1. The part that plays the role of determining RAM 1 and RAM 2 tasks is demultiplexer. This demultiplexer determines whether data is written in RAM 1 or RAM 2; therefore, the demultiplexer functions as data *distributors*. For the reading process from RAM 1 and RAM 2, it is determined by multiplexer part, which functions as data *selector*. Thus, the reading process is carried out in turn between RAM 1 and RAM 2.

III. Discussion

Implementation of the zigzag scan used mapping method to the FPGA through VHDL programming. Data input was a matrix (2D) with a size of 8x8. This matrix was changed to an array (1D) so that the size changed to 1x64. Changes in the form of a matrix input data into the array was intended to adjust the characteristics of the input data at the FPGA that uses the concept of the series. Data coming into the FPGA were individually alternately and sequentially.

Once the input data is defined, it then proceeds to define the sequence of positions that will be used to put the input data in accordance with the order of the position that has been determined. The result of this input data is then placed into zigzag scan using the mapping method. At the design stage using VHDL, sequence position data is placed on addressing the block mapping position.

VHDL programming for implementing the zigzag scan with mapping into the FPGA is as shown in the following program. Based on the program, the programming process of mapping starts, when every clock condition is at 1 (clk high condition). Signal se (enable signal) functions as selector to determine the memory is either a write or read process. During the first 64 clock cycles, every condition clock is at 1, the signal se is at 0 (se = '0'). Then, the first input data is written to the 1st memory (mem1) that is a mapping input data to the address position (addr_tls). There is no output that displays a reading of data from the 2nd memory because there is no process of writing data to the 2nd memory.

```
process(clk) begin
if (rising_edge(clk)) then
    if (i<65) then
        if (se='0') then
            mem1(conv_integer(addr_tls(i))) <= data_in;
            data_out<=mem2(i);
            i<=i+1;
        else
            mem2(conv_integer(addr_tls(i))) <= data_in;
            data_out<=mem1(i);
            i<=i+1;
            end if;
    else
        i<=1;
        if (se='0') then
            se<='1';
        else
            se<='0';
        end if;
    end if;
end if;
end process;
```

During the second of the 64 clock cycles, if the signal se is equal to 1 (se = '1') and the condition of the clock is 1 (clk conditions high), the second of input data are written to the 2nd memory (mem2), that is a mapping input data to the address position (addr_tls) and the 1st memory (mem1) reads the data that have been written at the time the signal se is 0. The reading of the first data from the 1st memory is the result of mapping the zigzag scan for the first input data that are displayed with data_out parameter. On the third of 64 clock cycles, if the signal se is worth 0 (se = '0') and the condition of the clock is worth 1 (clk conditions high), the third of data input are written to the 1st memory (mem1) and the 2nd memory (mem2) reads the second data that have already been written at the time of signal se is at 1. Any sequence of the writing and reading of data are generated by counter i in which the sorting began from the 1st sequence through 64th sequence (if i <65).

Data that become the input for zigzag scan process consist of 64 DCT quantified data result that are in series. Input data is represeted by four data blocks with the size of 8 x 8 for each block. Input data are mapped into position address, and the mapping result is saved in 2 RAMs that work alternately for writing and reading process based on the multiplexer which functions as a *selector*. To check the function of the zigzag scan with mapping, two types of simulation have been implemented: the behavioral simulation and post-route simulation. The result from the behavioral simulation of zigzag scan with mapping process is shown in Figure 6. While, the result from the post-route simulation of zigzag scan with mapping process is shown in Figure 7.

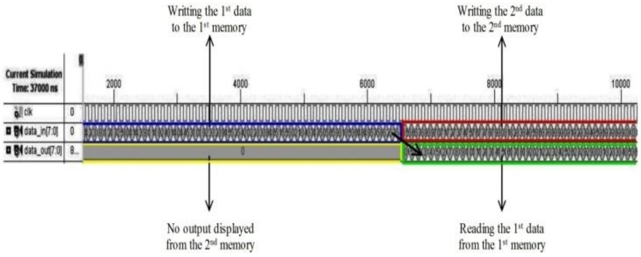


Figure 6: Result from behavioral simulation of zigzag scan with mapping process

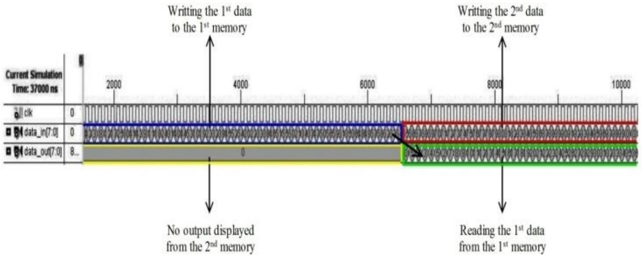


Figure 7: Result from post-route simulation of zigzag scan with mapping process

From the behavioral simulation, the obtained outcome agreed with the design which was proven by the resulting signal. This means that the design of zigzag scan with mapping method is able to sort DCT-quantified coefficient. This behavioral simulation outcome is still at the software level. Post-route simulation was implemented to adjust the zigzag scan with mapping design at the hardware level. The post-route simulation was the most accurate simulation that suits with the real hardware characteristic. In this post-route simulation, Xilinx conditioned the design by hardware so that the signal timing is suitable with the condition of hardware electronic components and space delay between components that was calculated. The post-route simulation resulted signal that was compatible with the design.

The RTL (Register Transfer Logic) circuit was obtained from the synthesis results, as shown in Figure 8. In Figure 8, the circuit consists of a counter, ROM, RAM. The Counter function functions to generate a sequence of writing and reading data to the RAM. The counter also functions to generate a sequence reading of address position stored in ROM.

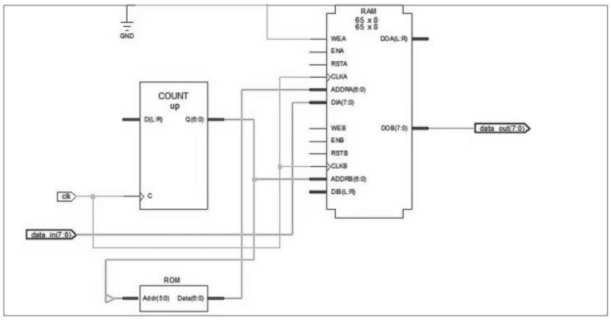


Figure 8: Zigzag scan schematic using the mapping method

IV. RESULTS

Algorithm developed by Arafa used 40 memory (stack) elements that consists of register 0 to 39 to save 8 input column. Storage was done based on the available cycle by inserting every input column element to different locations. Implementation of algorithm by Arafa into IC-FPGA needed resources in the form of 303 slices of Flip-flop, 811 slices of LUTs and low process speed (86 MHz). Latency in algorithm by Arafa was 320 clock cycles.

Zigzag scan from Ketul algorithm that was implemented into IC-FPGA needed 22 slices Flip-flops resource or about 0% from available resource which is 9.312 slices, 87 slices of LUTs or about 0% from available resource which is 9.312 slices, 9 bonded IOBs or about 4% from 190 available resources, and 1 GCLKs or about 4% from 24 available GCLKs resources. Performance comparison between the methods by Arafa, Ketul and this research is as shown in Table 1.

Table 1
Performance Comparison

No	Performance	Presented [2]	Presented [3]	This Paper
1.	Flip-flop	303	22	18
2.	LUT	811	87	33
3.	Max. Freq. (MHz)	86	234	250
4.	Latency	40	960	64

Based on Table 1, zigzag scan with mapping method has 64 *clock* latency, thus processing time to group DCT-quantified coefficients is faster than Arafa method (latency as many as 320 *clock*) and Ketul method (latency as many as 960 *clock*), due to every input data are immediately put in a determined position without many value comparison and repetition that can prolong processing time, thus zigzag scan with mapping method period is more efficient.

The process of zigzag scan with mapping method can be operated in real-time on frequency 250 MHz or for each data byte needs 4 ns process time and for 8 x 8 data matrix needs 256 ns process time. It means that the generated IC Zigzag scan prototype can be operated on real-time JPEG/MPEG compression with a maximum 3 mega pixel per frame/picture with 25 frame per second. This zigzag scan with mapping design needs 10 slices of Flip-flop (30 times less than Arafa method and 2 times less than Ketul method), LUTs as many as 39 slices (20 times less than Arafa method and 2 times less that Ketul method).

V. Conclusion

Based on the simulation result, this mapping process can be operated in real-time on frequency 250 MHz or for each data byte needs 4 ns process time and for 8 x 8 data matrix needs 256 ns process time. It means that IC Zigzag scan prototype can be operated on real-time JPEG/MPEG compression with maximum 3 mega pixel per frame/picture with 25 frame per second. The IC Zigzag Scan with mapping component (IP Core) generated need 10 slices of Flip-flop (30 times less than Arafa method and 2 times less than Ketul method), LUTs as many as 39 slices (20 times less than Arafa method and 2 times less that Ketul method), 9 bonded IOBs, 1 block of RAM and 1 GCLKs, with latency as many as 64 clock (5 times less than Arafa method and 15 times less than Ketul method); therefore. processing time and IC-FPGA resources requirement on zigzag scan with mapping could be more efficient.

Acknowledgment

Thanks to Yayasan Pendidikan Gunadarma for supporting this research as one part of research dissertation in Doctoral Program of Gunadarma University, Jakarta-Indonesia.

References

[1] Ann Malsha De Silva, Donald G. Bailey, Amal Punchihewa (2012), “Exploring the Implementation of JPEG Compression on FPGA”, *The 6th International Conference on Signal Processing and Communication Systems (ICSPCS).*

[2] Arafa, A. A., et al., “FPGA Implementation of Discrete Cosine Transform Based Image Compression Encoder”, *The 3rd Minia International Conference For Advanced Trends In Engineering (MICATE 2005),* 3-5 April 2005.

[3] Ketul Shah, Sagar Shah, “Zigzag Scanning of a Matrix”, Nirma University, March 2014.

[4] Mangesh Jadhav, Sneha Ghanekar, Jigar Jain (2012), “Image Compression Using Hybrid Quantization Method In JPEG”, *International Journal of Engineering Science and Technology – IJEST* (ISSN : 0975 – 5462), Vol. 4, No. 09, pp. 3975 – 3987.

[5] Vijaya Prakash. A.M, K.S. Gurumurthy, “VLSI Architecture For Low Power Variable Length Encoding And Decoding For Image Processing Applications”, *International Journal of Advances in Engineering & Technology,* ISSN: 2231-1963, Vol. 2, Issue 1, pp. 105-120, January 2012.

[6] Xinsheng Wang,et al., “Power Research of JPEG Circuits in FPGA”, *Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2011.