

High Speed Radix-8 based MAC used for 2D-Image Compression

T.Kanagaraj
PG scholar
Kalaingar Karunanidhi
Institute of technology
Coimbatore, India.

M.Pradeepa
Asst.professor
Kalaingar Karunanidhi
Institute of technology
Coimbatore, India.

ABSTRACT

Field programmable gate arrays are ideally suited for the implementation of DCT based digital image compression. However, there are several issues that need to be solved. The Multiply-Accumulate Unit (MAC) is the main computational kernel in DSP and DIP architectures. The proposed MAC unit determines the power and the speed of the overall system; it always lies in the critical path. In this work, a fast and low power MAC Unit is proposed for 2D-DCT computation. The proposed architecture is based on modified booth radix-8 with merged MAC architectures to design a unit with a low critical path delay. The new architecture has reduces the hardware complexity of the summation network, it reduce the overall power. Increasing the speed of operation is achieved by feeding the bits of the accumulated operand into the summation tree before the final adder instead of going through the entire summation network. The FPGA implementation of the proposed booth radix-8 based MAC unit saves 64% of the area, to the regular MAC unit with conventional multiplier.

Keywords

Discrete Cosine Transform (DCT), Very Large Scale Integration (VLSI), Digital Signal Processing (DSP), Multiply-Accumulate Unit (MAC).

1. INTRODUCTION

The applications of the digital images are more and more extensive today than ever before. In recent years, the design for high speed with low power consumption has become one of the greatest challenges in high-performance very large scale integration (VLSI) design. As a consequence, many techniques have been introduced to reduce the power consumption with high performance of new VLSI systems. However, most of these methods focus on the power consumption and quality enhancement by replacing multipliers with distribute arithmetic approach [1]-[2]. But this increase the overall delay makes them unsuitable for high speed applications.

In the last decade many research have been carried out to reduce the computation complexity, and quality enhancement of DCT. Most of them are efficient with software implementation, but unfortunately due to their irregular structure and complex routing, these algorithms are not suitable for VLSI implementation.

Our goal is to design a high speed 8x8 DCT chip suitable for the application in HDTV system. To reduce the area of the circuit, the fixed-width multipliers will be only kept the most significant half part of the products. It will lead large error, many compensation methods are used to solve this problem [3]-[4]. Here we attempt to increase the speed of multiplication by reducing number of partial products

generated using high radix booth algorithm and combine it with partial addition of partial product addition using truncation technique. The main concerns are speed and error compensation in DCT [8] computation.

2. FPGA IMPLEMENTATION

Field programmable gate arrays were actually invented only for prototyping the digital design which is later to be used in IC's. But in recent days FPGA's are started to use as a product in many fields. So Field programmable gate arrays are ideally suited for the implementation of DCT based digital image compression. However, there are several issues that need to be solved. When performing software simulation of DCT, calculations are carried out with floating point precision. But in FPGA resources available to perform floating point arithmetic is unjustified, and measures to be taken to account for this. Another concern is the DCT coefficients value itself. Many techniques have been used to efficiently convert this floating point values into binary representation for digital implementation. Then only we can implement DCT in VLSI.

The two ways of floating point to binary conversion are

1. Both integral and fractional part is converted separately by repeatedly multiply 2, and considers each one bit as it appears left of the decimal.
2. Representing the floating number using IEEE 754 format (single or double precision).

Method 1 can be used efficiently if we use any shift and add based approach for DCT computation. For method 2 we need to have floating point arithmetic unit for computation.

3. BOOTH MULTIPLIER

In the modern world, we need system which will run at high speed. Multipliers play an important part in today's DSP and DIP applications. In our case for DCT computation multiplications are used larger in number. Therefore, speed improvement in multiplier is important. Advances in technology have permitted many researchers to design multipliers which offer both high-speed and unique hardware structure, thereby making them suitable for specific VLSI implementation.

In any multiplication algorithm, the multiplication operation is carried out by summation of decomposed partial products. For high-speed multiplication we need to apply a booth radix recoding multiplication algorithm. In recent days in all high radix booth algorithm recoding is changed from 2s-complement format to a signed-digit representation from the defined set. This is called modified booth algorithm.

3.1 Radix-8 Algorithm

Here we reduce the number of partial products using a higher radix in the multiplier recoding. Recoding of binary numbers was first invented by Booth [5]. The modified Booth's algorithm is done by appending a zero to the right of M. In radix-8 recoding is similar to radix-4 [6] but here we take four bits instead of three bits and then we represent that coded values in signed-digit representation shown in Fig. 1 using Table 1. The proposed radix-8 based MAC block diagram is shown in Fig. 2.

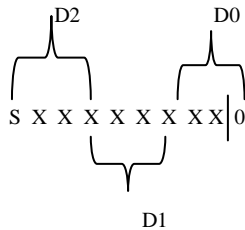


Fig. 1 Recoding Representation

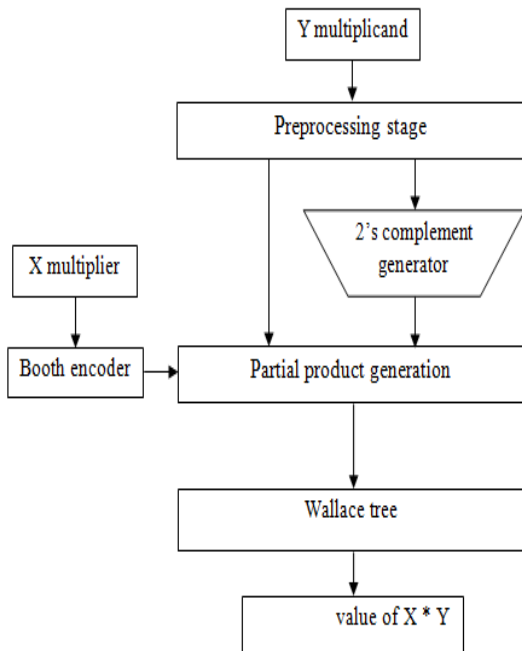


Fig. 2 Block Diagram of Proposed Booth Multiplier

Table 1. Radix-8 Sign Digit Values

Coded bits	signed-digit value
0000	0
0001	+1
0010	+1
0011	+2
0100	+2
0101	+3
0110	+3
0111	+4
1000	-4
1001	-3
1010	-3

1011	-2
1100	-2
1101	-1
1110	-1
1111	0

From the Table 1 we need to have 2N, 3N, 4N and its 2's complement respectively.

3.2. Preprocessing Stage

Both 2N and 4N is achieved by simple left shift of N. and 3N is calculated by adding 2N and N. If the bit width of N is high this will increase the delay in preprocessing stage. After this stage only we can generate partial products. In order to reduce the delay here we use Carry select adder. The resource used in CSA adder later will be used for partial product addition.

3.3. Wallace Tree

Here we use Wallace array to represent partial products array and its summation, which gives the multiplication result. First four partial products are processed using 4-2 compressor following Fig.3, which is made up of two full adders [7].

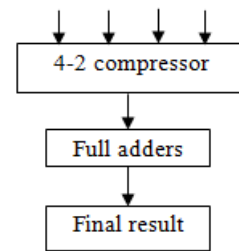


Fig. 3 Wallace Tree Structure

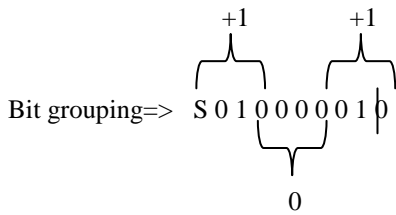
In later stages we used only Full adder for partial product addition. In radix-8 partial products are left shifted by three bits. So in partial products some of the LSB bits will become 0's. It is predefined one. So these bits are not considered here in Wallace tree structure in order to save the hardware resource.

In Wallace tree partial products are divided into main part and truncation part. Resource used in truncation part is reconfigured based on number of columns need to be selected for error compensation.

4. Example of Radix-8 Multiplier

In order to have a better understanding of the multiplier design we are going to show an example following the radix-8 recoding algorithm. Consider the multiplication of these 2s-complement binary numbers. The multiplier recoding has the result shown here (following Table 4.1),the bits are grouped as four bits. So here generate four bit groups shown in example.

Multiplier $x = -25 = 1110\ 0111$;
 Multiplicand $y = 65 = 0100\ 0001$;
 Product $z = -1625 = 1111\ 1001\ 10100\ 111$;
 Preprocessing unit $\Rightarrow x, 2x, \sim x, \sim 2x, 3x, \sim 3x, 4x, \sim 4x$



Bit grouping [0] = 0010 = +1 = x = -25;
 Bit grouping [1] = 0000 = 0 = 0;
 Bit grouping [2] = 0 010 = +1 = x = -25;
 The generation of three times the multiplicand gives:
 mul_prod [0] = 11100111;
 mul_prod [1] = 00000000;
 mul_prod [2] = 11100111;

To find final product
 The final product is calculated by each mul-prod are three bits left shift and add the product bits.

smul_product_result [0] => 11111111 11100111
 smul_product_result [1] => 00000000 00000000
 smul_product_result [2] => 11111001 11000000

Final product => 11111001 10100111

5. PERFORMANCE ANALYSIS

The conventional multiplier and proposed radix-8 MAC are simulate and synthesis using ModelSim and Quartus II. The parameters of speed, area and power dissipation are reported in table. The proposed radix-8 has high speed, and more area efficient than a conventional multiplier. The conventional and radix-8 multipliers area analysis result is shown in Fig. 4 and Fig. 5.

Flow Summary	
Flow Status	Successful - Tue Aug 27 13:43:50 2013
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	linebuffer
Top-level Entity Name	existing_mul
Family	Cyclone II
Met timing requirements	Yes
Total logic elements	349 / 4,608 (8 %)
Total combinational functions	349 / 4,608 (8 %)
Dedicated logic registers	256 / 4,608 (6 %)
Total registers	256
Total pins	34 / 89 (38 %)
Total virtual pins	0
Total memory bits	0 / 119,808 (0 %)
Embedded Multiplier 9-bit elements	0 / 26 (0 %)
Total PLLs	0 / 2 (0 %)
Device	EP2C5T144C6
Timing Models	Final

Fig.4 Conventional Multiplier Area Analysis Result

Flow Status	Successful - Tue Nov 05 09:56:16 2013
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	raddddwall
Top-level Entity Name	MBE_RADIX_8
Family	Cyclone II
Met timing requirements	Yes
Total logic elements	210 / 4,608 (5 %)
Total combinational functions	210 / 4,608 (5 %)
Dedicated logic registers	32 / 4,608 (< 1 %)
Total registers	32
Total pins	35 / 89 (39 %)
Total virtual pins	0
Total memory bits	0 / 119,808 (0 %)
Embedded Multiplier 9-bit elements	0 / 26 (0 %)
Total PLLs	0 / 2 (0 %)
Device	EP2C5T144C6
Timing Models	Final

Fig.5 Area Analysis Result of Radix-8 Multiplier

The radix-8 multiplier power analysis is result is shown in Fig. 6.

PowerPlay Power Analyzer Status	Successful - Tue Nov 05 09:58:55 2013
Quartus II Version	9.0 Build 132.02/25/2009 SJ Web Edition
Revision Name	raddddwall
Top-level Entity Name	MBE_RADIX_8
Family	Cyclone II
Device	EP2C5T144C6
Power Models	Final
Total Thermal Power Dissipation	33.02 mW
Core Dynamic Thermal Power Dissipation	0.00 mW
Core Static Thermal Power Dissipation	18.01 mW
I/O Thermal Power Dissipation	15.01 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

Fig.

6 Power Analysis Result of Radix-8 Multiplier

The conventional and radix-8 multiplier speed analysis results are shown in Fig.7 and Fig.8. Comparison result of different multipliers is tabulated in Table II.

Fmax Summary				
	Fmax	Restricted Fmax	Clock Name	Note
1	113.96 MHz	113.96 MHz	clk	

Fig. 7 Speed Analysis Result of Conventional Multiplier

Fmax Summary				
	Fmax	Restricted Fmax	Clock Name	Note
1	1488.1 MHz	420.17 MHz	clk	limit due to minimum period restriction (max I/O toggle rate)

Fig. 8 Speed Analysis Result of Radix-8 Multiplier

Table 2. Comparison Result of Different Multipliers

Sl.No	Conventional Multiplier	Radix-8 Multiplier	Radix-8 & Wallace tree Multiplier
1.	Total logic elements =349	Total logic elements = 226	Total logic elements = 210
2	Total register = 256	Total register = 48	Total register = 32
3	Thermal power dissipation = 38.2mw	Thermal power dissipation=33.10mw	Thermal power dissipation=33.02mw
4	Delay time = 7.99 ms	Delay time = 1.34 ms	Delay time = 0.67 ms

5. CONCLUSION

Proposed a unique high speed booth multiplier based MAC unit to improve throughput rate and to minimize the area complexity of 8x8 2D-DCT architecture. In the DCT architecture, DCT computation is performed with sufficiently high precision in DCT matrix multiplication yielding an acceptable quality. Our proposed MAC architecture achieves a maximum operating frequency of 1488.1MHz. In summary the proposed architecture is suitable for applications in HDTV system.

REFERENCES

- [1] S. Yu and E. E. S. „Jr., “DCT implementation with distributed arithmetic”*IEEE Trans. Comput.*, Vol. 50, No. 9, pp. 985–991, Sep. 2001.
- [2] P. K. Meher, “Unified systolic-like architecture for DCT and DST using distributed arithmetic,” *IEEE TransCircuits Syst. I, Reg. Papers*, Vol. 53, No. 12, pp. 2656–2663,Dec. 2006.
- [3] L. D. Van and C. C. Yang, “Generalized low-error area-efficient fixedwidth multipliers,” *IEEE Trans. Circuits Syst. I*, Vol. 52, No. 8, pp. 1608– 1619, Aug. 2005.
- [4] C. H. Chang and R. K. Satzoda, “A low error and high performance multiplexer-based truncated multiplier,” *IEEE Trans. VLSI Syst.*, Vol. 18, No. 12, pp. 1767–1771, Dec. 2010.
- [5] C. Lu, S. P. A.D. Booth, “A signed binary multiplication technique,” *Quarterly J. Mechan. Appl. Math.*, Vol. 4, Part 2, 1951.
- [6] Yuan-Ho Chen and Hsin-Chen Chiang “Modified Booth Encoding Radix-4 8-bit Multiplier” .2012.
- [7] Prasad, K, “Low-power 4-2 and 5-2 compressors” Vol. 1, pp. 129 – 133, Nov. 2001.
- [8] Jongsun Park · Kaushik Roy “A Low Complexity Reconfigurable DCT Architecture to Trade off Image Quality for Power Consumption”, Springer Science Business Media, April 2008.

- [9] Oscar T-C Chen, Sandy Wang, and Yi-Wen Wu (2003), 'Minimization of Switching Activities of Partial Products for Designing Low-Power Multipliers', IEEE Transactions on Very Large Scale Integration (VLSI) systems, Vol.11, No.3.
- [10] Prasad K (2001), 'A Low-Power 4-2 and 5-2 Compressors' Vol. 1, pp. 129 – 133.
- [11] Ron S, Waters and Earl E (2010), 'A Reduced Complexity Wallace Multiplier Reduction', IEEE Transactions on Computers, Vol. 59, NO. 8, August 2010.
- [12] Rizalafande Che Ismail , Hussin R (2006), 'High Performance Complex Number Multiplier Using Booth-Wallace Algorithm', IEEE International Conference on Semiconductor Electronics.
- [13] Soojin Swati Malik and Sangeeta Dhall(2012), 'Implementation of MAC Unit Using Booth Multiplier & Ripple Carry Adder', International Journal of Applied Engineering Research, ISSN 0973- 4562 Vol.7 No.11.
- [14] Sheen R, Wang S, Chen O T-C, and Ma R L (1999), 'Power Consumption of a 2's Complement Adder Minimized by Effective Dynamic Data Ranges', in Proc. IEEE Int. Symp. Circuits Syst., Vol. I, pp. 266–269.
- [15] Lakshmanan M, Othman M and Mohd.Ali M A (2002), 'High performance Parallel Multiplier Using Wallace-Booth Algorithm', proceedings ICSE 2002, IEEE International conference on Semiconductor Electronics.
- [16] Liao Y, Roberts D (2002), 'A High Performance And Low Power 32-Bits Multiply Accumulate Unit With Single Instruction Multiple Data (SIMD) Feature', IEEE Journal of solid state circuits, Vol.37, No.7, pp.926-931.
- [17] Mottaghi-Dastjerdi M, Afzali-Kusha A and M (2008), 'BZ-FAD - A Low-Power Low-Area Multiplier Based On Shift-and-Add Architecture', IEEE Trans. on VLSI Systems.
- [18] Meher P K(2006), 'Unified Systolic-Like Architecture For DCT and DST Using Distributed Arithmetic Technique', IEEE Trans. Circuits Syst. I, Reg. Papers, Vol. 53, No. 12, pp. 2656–2663.
- [19] Mahant-Shetti S, Balsara P, and Lemonds C (1999), 'High Performance Low Power Array Multiplier Using Temporal Tiling', IEEE Trans. VLSI Syst., Vol. 7, pp. 12.
- [20] Elguibaly F (2008), 'A Fast Parallel Multiplier – Accumulator using Modified Booth Algorithm', IEEE Transaction on circuits and systems –II: Analog and Digital Processing Vol.47, pp.902-908.
- [21] Farooqui A, Okolbdzija V (1998), 'General Data-path Organization of MAC Unit for VLSI Implementation', IEEE International symposium circuits and systems, pp.260-263.