

On-Chip Implementation of High Resolution High Speed Floating Point Adder/Subtractor with Reducing Mean Latency for OFDM

¹Rozita Teymourzadeh, ¹Yazan Samir Algnabi, ²Nooshin Mahdavi and ¹Masuri Bin Othman

¹Department of Electrical and Electronic, Faculty of Engineering,
Institute of Microengineering and Nanoelectronics, Blok Inovasi 2,
University Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

²Department of Electrical and Electronic, Arak University, Iran

Abstract: Problem statement: Fast Fourier Transform (FFT) is widely applied in OFDM trance-receiver communications system. Hence efficient FFT algorithm is always considered. **Approach:** This study proposed FPGA realization of high resolution high speed low latency floating point adder/subtractor for FFT in OFDM trance-receiver. The design was implemented for 32 bit pipelined adder/subtractor which satisfied IEEE-754 standard for floating-point arithmetic. The design was focused on the trade-off between the latency and speed improvement as well as resolution and silicon area for the chip implementation. In order to reduce the critical path and decrease the latency, the novel structure was designed and investigated. **Results:** Consequently, synthesis report indicated the latency of 4 clock cycles due to each stage operated within just one clock cycle. The unique structure of designed adder well thought out resulted 6691 equivalent gate count and lead us to obtain low area on chip. **Conclusion:** The synthesis Xilinx ISE software provided results representing the estimated area and delay for design when it is pipelined to various depths. The report showed the minimum delay of 3.592 ns or maximum frequency of 278.42 MHz.

Key words: OFDM, FFT, FPGA, Synopsys design compiler, CMOS layout

INTRODUCTION

Recently Orthogonal Frequency Division Multiplexing (Bingham, 1990) (OFDM) techniques have received immense attention in high-speed data communication systems such as Wireless local Area Network (WAN), digital audio/video broadcasting and beyond 3G research (Tian, *et al.*, 2004).

OFDM is a multi-carrier transmission technique, which divides the available spectrum into many carriers, each one being modulated by a low rate data stream. Then Inverse Fast Fourier Transform (IFFT) or Fast Fourier Transform (FFT) transforms are used to get the signal in time domain or spectrum frequency. FFT reduces the complexity of the OFDM. Hence the performance of the FFT has highly effect on the OFDM system result.

High resolution, high speed and low latency, FFT would not be achieved unless with efficient elements in particular such the butterfly component. The performance of butterfly is greatly depending on its adders and subtractors. Hence, this research with emphasis of the simple structures, focused on design

and implementation of efficient high performance floating point adder/subtractor for FFT algorithm.

Narasimhan *et al.* (1993) in 1993 proposed 8 stage floating point adder for FPGA. The specifications for the adder stated that it should work at 62.5 MHz.

In 2004, (Thompson *et al.*, 2004) presented a decimal floating-point adder with 5 stages that compiled with the current draft revision of IEEE-754 Standard. The adder supported operations on 64 bit (16 digit) decimal floating-point operands. Initial synthesis testing and evaluation was done and performed using Synopsys Design Compiler and LSI Logic's Gflxp 0.11 micron CMOS standard cell library.

In 2005, (Huang *et al.* 2005) presented high speed double precision floating point adder using custom-designed macro modules and other advanced optimization technology. Based on SMIC six-layer-metal CMOS process, he achieved a 4 stage pipelined double precision floating point adder which could complete a floating point addition in 7.72 ns. The total gate count in this system is 37977 gates.

All the results show that the effort is taken to achieve low latency and area, high resolution and speed

Corresponding Author: Rozita Teymourzadeh, VLSI Design Center, Faculty of Engineering, Blok Inovasi 2, UKM, 43600 Bangi, Selangor, Malaysia

by introducing the new algorithms. In this study the advanced floating point adder with pipelined structure is presented.

The similar work also represented within year 2005 to 2009 (Shi Chen *et al.*, 2008; Xenoulis *et al.*, 2005; Yu *et al.*, 2008), to show engineers effort for increasing the capability of floating point calculation. However still there are not enough available resources regarding floating point arithmetic.

Design and implementation: Based on IEEE-754 standard for floating point (Hollasch, 2005) arithmetic 32 bit data register is considered to allocate mantissa, exponent and sign bit in a portion of 23, 8 and 1 bit respectively. The advantage of this adder is that it can be easily switched to 12 bit mantissa 8 bit power and 1 bit sign bit arithmetic calculations. Additionally the floating-point adder unit performs the addition and subtraction using substantially the same hardware as used for floating-point operations. This advantage causes saving the core area by minimizing the number of elements. Fig. 1 shows the new structure of the floating point adder when it is divided to the four separate blocks. The purpose is that to share total critical path delay into three equal blocks. These blocks calculate the arithmetic function within 1 clock cycle. However the propagation delay can be associated with continues assignment (Ciletti, 2002) so that to increase the overall critical path delay and the reason of slowing down the throughput. Hence in this study, the effort is taken to reduce the worst effect of delay for arithmetic calculations.

Based on combinational circuit design, the output of each stage is only depending on its input value at the time. As shown in the Fig. 1, each block creates the output within 1 clock cycles at the time. The unique structure of this adder enables us to feed the output result in pipeline registers after every clock cycles. Hence, the sequential structure is applied for overall pipelined add/subtractor algorithm to combine the stages.

The pipeline floating point adder structure consists of compare stage, aligned mantissa, add/subtractor stage and finally normalized stage.

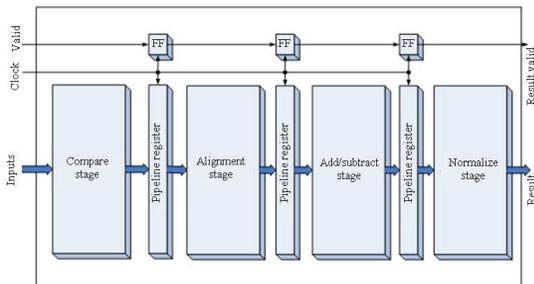


Fig. 1: Block diagram of the proposed adder

While the effective operation is determined the blocks, compare stage and aligned mantissa lead us to align the mantissa and exponent accordingly to make sure having the same exponent in two operands. The significant point is to design the function so that all calculation performs within one clock cycle. The basic operation of the aligned mantissa and normalized block is shifting. Every shifting need one clock cycle and it causes huge delay to align 32 bit operand. Hence, the advanced algorithm is applied to avoid having many delays on aligning stage. This is the third advantage of the mentioned adder.

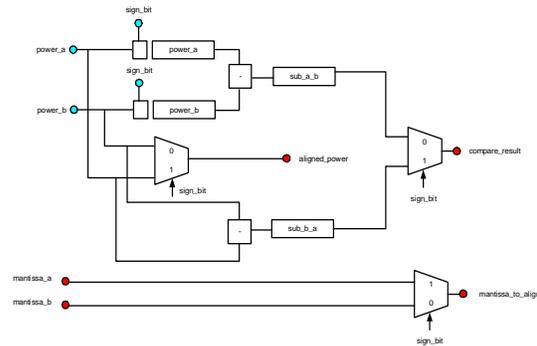


Fig. 2: Compare stage structure

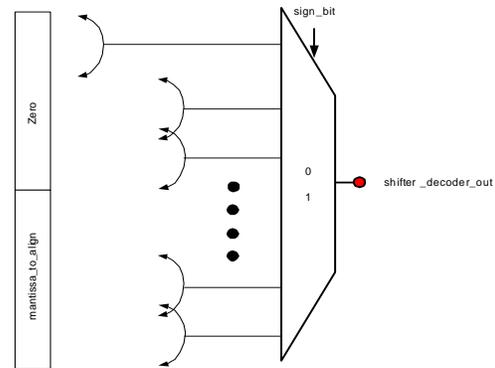


Fig. 3: Mantissa aligned stage structure

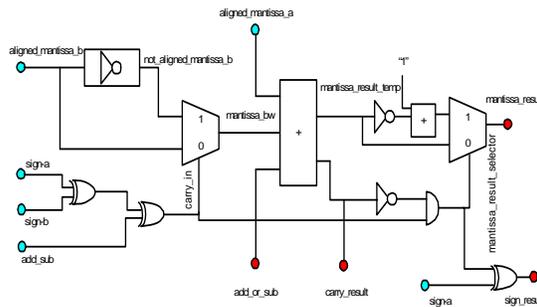


Fig. 4: Add/subtractor stage structure

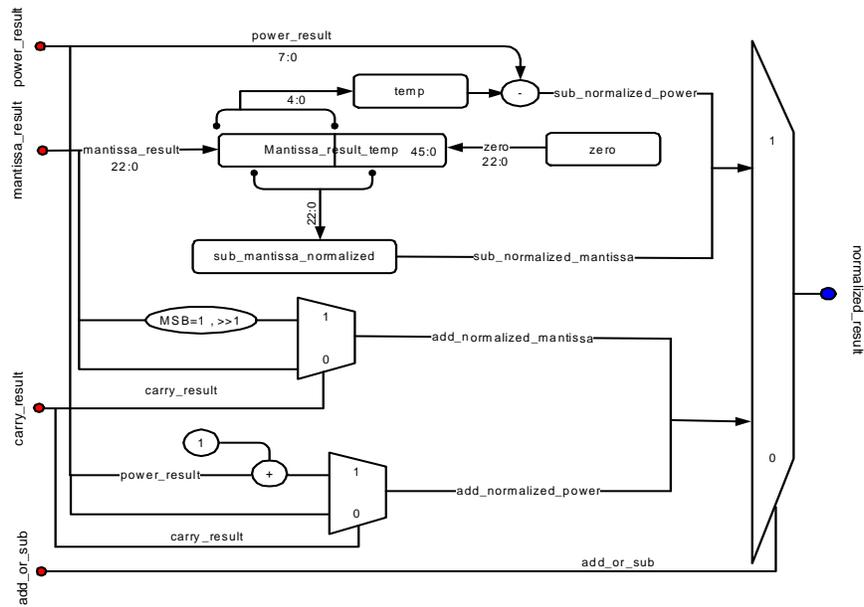


Fig. 5: Normalized stage structure

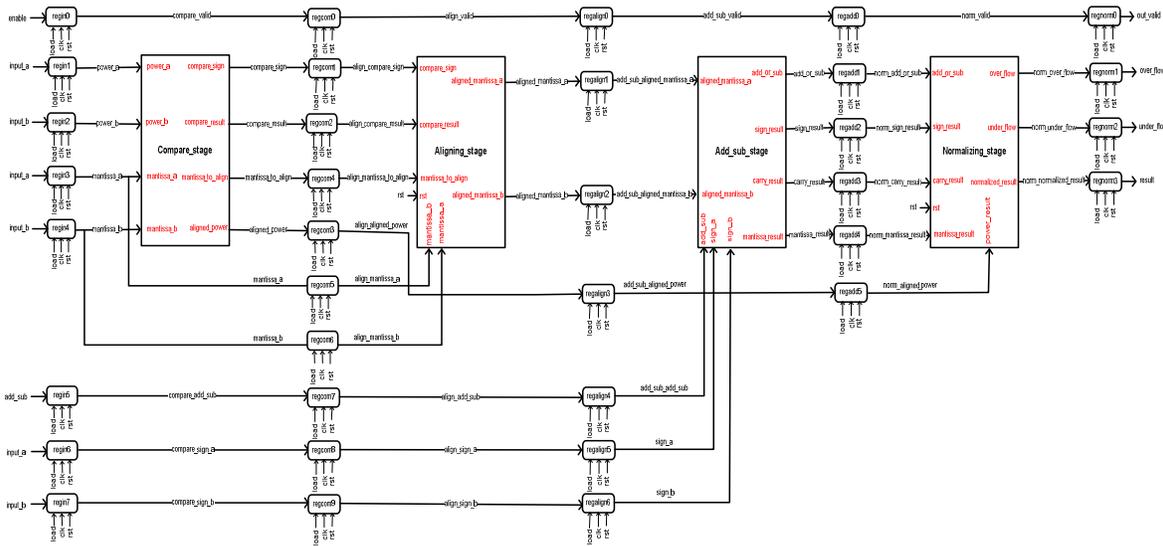


Fig. 6: Overall pipelined adder/subtractor structure

Figure 2-5 show the internal structure of proposed pipelined adder/subtractor algorithm whereas Fig. 1 and 6 present the overall structure of pipelined adder. As shown in the Fig. 1-6, the smart algorithm is designed for add/sub stage to apply two's complement for the operand if it is required and combine addition and subtraction in the same hardware block diagram.

In addition to save power consumption, the proposed architecture also offers power savings due to

the simplification of data paths. Ignoring the shifting blocks and combine the adder and subtractor has the significant role to reduce power-effective in the overall system.

The total number of gate count in the system is 6691 which is given by synthesis report of Xilinx ISE synthesis software. This total gate count proves the low area and low power consumption for proposed floating point adder/subtractor, whereas the almost similar

project implements the adder with the equivalent gate count of 37977 (Huang *et al.*, 2005).

MATERIALS AND METHODS

The Floating point adder designed according to the IEEE 754 standard. The design process was to create software model of efficient pipeline floating point adder algorithm using Verilog HDL programming language. The system was simulated by Modelsim and its function was fully covered accordingly. The design was emulated the following characteristics; Floating point arithmetic computation, Pipelined addition and subtractions and finally data and factor precision. The proposed adder was fabricated on FPGA Virtex-II chip and the layout core was submitted respectively.

RESULTS AND DISCUSSION

Implementation result: The implementation was modeled in Verilog HDL code and simulated by Modelsim software. The design was synthesized by Xilinx-ISE software and downloaded to FPGA Virtex II.

The synthesis result shows the system specification of every stage of the pipelined floating point adder/subtractor separately. These stages are introduced as Compare, Align, Add/Sub and Normalized stages (Table 1-4). Finally the synthesis result of overall system by combining of the mentioned stages is given (Table 5).

From Xilinx ISE synthesizer report; it was found minimum clock period is 3.592 ns (Maximum Frequency is 278.428 MHz) for floating point adder/subtractor. As shown in the result, total slice number in overall pipeline floating point adder is less than the sum of the each cell blocks. It is due to advance configurations which avoid repeating the similar slice and try to reuse the similar slice again.

Table 1: Compare stage specification

HDL synthesis report	
	QTV
Design statistics	
No. of IOs	103
No. of multiplexer	4
No. of Xor	0
Slice number	42 (2%)
Timing report	
Minimum period (ns)	1.031
Maximum frequency (MHz)	969.744
Minimum input arrival time before clock (ns)	1.988
Maximum output required time after clock (ns)	2.775
Total equivalent gate count for design	893
Total memory usage (MB)	59

Furthermore, the minimum clock periods increased sharply after add/sub stage was applied. This issue can be explained by utilizing FPGA adder to calculate fixed point arithmetic. However the speed result is high enough to cover the subject and for future study, enhancing the maximum frequency will be obtained by applying high speed prefix adder (Burgess, 2004) to calculate fix point arithmetic. The chip layout on Virtex II FPGA board has been shown in Fig. 7.

The estimated latency of the design for 32 bit resolution is 4 clock cycle due to its pipeline structures.

Table 2: Align stage specification

HDL synthesis report	
	QTV
Design statistics	
No. of IOs	126
No. of multiplexer	2
No. of Xor	0
Slice number	158 (10%)
Timing report	
Minimum period (ns)	0.847
Maximum frequency (MHz)	1181.056
Minimum input arrival time before clock (ns)	4.032
Maximum output required time after clock (ns)	2.775
Total equivalent gate count for design	2211
Total memory usage (MB)	65

Table 3: Add/sub stage specification

HDL synthesis report	
	QTV
Design statistics	
No. of IOs	77
No. of multiplexer	1
No. of Xor	1
Slice number	68 (4%)
Timing report	
Minimum period (ns)	2.377
Maximum frequency (MHz)	420.76
Minimum input arrival time before clock (ns)	2.299
Maximum output required time after clock (ns)	2.779
Total equivalent gate count for design	1615
Total memory usage (MB)	69

Table 4: Normalized stage specification

HDL synthesis report	
	QTV
Design statistics	
No. of IOs	70
No. of multiplexer	4
No. of Xor	0
Slice number	201 (13%)
Timing report	
Minimum period (ns)	2.062
Maximum frequency (MHz)	484.919
Minimum input arrival time before clock (ns)	5.522
Maximum output required time after clock (ns)	2.775
Total equivalent gate count for design	3687
Total memory usage (MB)	61

Table 5: Floating point adder/subtractor specification

HDL synthesis report	
	QTV
Design statistics	
No. of IOs	104
No. of multiplexer	11
No. of Xor	1
Slice number	349 (22%)
Timing report	
Minimum period (ns)	3.592
Maximum frequency (MHz)	278.428
Minimum input arrival time before clock (ns)	2.534
Maximum output required time after clock (ns)	2.779
Total equivalent gate count for design	6691
Total memory usage (MB)	72

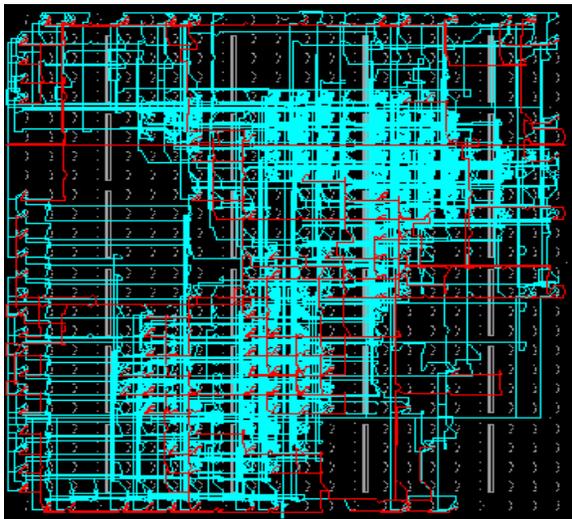


Fig. 7: The core layout on FPGA board

The advance design algorithm led us to have minimum equivalent of gate count (6691 gate) resulted saving area and power consumption satisfied low area low power on chip core. This value is 37977 gates in previous research (Huang *et al.*, 2005).

CONCLUSION

Efficient algorithm of high resolution high speed low area floating point adder/subtractor with reducing mean latency for OFDM applications was designed and investigated. The structure of the adder is consisting of 4 parts called compare stage, aligned stage, add/sub stage and finally normalized stage. Each block calculates the arithmetic operation within 1 clock cycle. The result will be inputted to the pipeline register and it is the reason to reduce the latency. The unique structure of this adder ignores all the shifter cells and replace it with multiplexer to reduce delay

propagation through each cells. The evaluation indicates that the proposed pipelined adder is attractive due to high resolution (32 bit floating point), low area (6691 gate count) and low latency of 4 clock cycle. The maximum frequency for this adder is 278.428 MHz. To increase the speed, prefix adder can be replaced with FPGA structural adder to calculate.

REFERENCES

- Bingham, J.A.C., 1990. Multicarrier modulation for data transmission: An idea whose time has come. *IEEE Commun. Mag.*, 28: 5-14. http://wsl.stanford.edu/~ee359/ofdm_bingham.pdf
- Burgess, N., 2004. Prenormalization rounding in IEEE floating-point operations using a flagged prefix adder. *IEEE J. Very Large Scale Integrat. Syst.*, 13: 266-277. DOI: 10.1109/TVLSI.2004.840764
- Ciletti, D.M., 2002. *Advanced Digital Design with the Verilog HDL*. 1st Edn., Springs, ISBN: 10: 0130891614.
- Hollasch, S., 2005. *IEEE Standard 754 Floating Point Numbers*. IEEE, Inc., New York. <http://steve.hollasch.net/cgindex/coding/ieeefloat>
- Huang, C., X. Wu, J. Lai, C. Sun and G. Li, 2005. A design of high speed double precision floating point adder using macro modules. *Proceedings of the Asia and South Pacific Design Automation Conference*, Jan. 18-21, ACM Press, New York, USA., pp: D11-D12. <http://portal.acm.org/citation.cfm?id=1120891>
- Narasimhan, D., D. Fernandes, V.K. Raj, J. Dorenbosch, M. Bowden and V.S. Kapoor, 1993. A 100 MHz FPGA based floating point Adder. *Proceeding of the IEEE Conference on Custom Integrated Circuits*, Sept. 6-6, IEEE Xplore Press, USA., pp: 3.1.1-3.1.4. DOI: 10.1109/CICC.1993.590348
- Shi Chen, R. Venkatatesan and P. Gillard, 2008. Implementation of vector floating-point processing unit on FPGAs for high performance computing. *Proceeding of the IEEE Canadian Conference on Electrical and Computer Engineering*, May 4-7, IEEE Xplore Press, USA., pp: 881-886. DOI: 10.1109/CCECE.2008.4564662
- Tian, J., Y. Xu, H. Jiang, H. Luo and W. Song, 2004. Efficient algorithms of FFT butterfly for OFDM systems. *Proceeding of the IEEE 6th CAS Symposium on Emerging Technologies: Mobile and Wireless Communication Shanghai*, May 31-June 2, IEEE Xplore Press, USA., pp: 461-464. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1321923

- Thompson J., N. Karra and M.J. Schulte, 2004. A 64-bit decimal floating-point adder. Proceedings of the IEEE Computer Society Annual Symposium on VLSI Systems Design, Feb. 19-20, IEEE Xplore Press, USA., pp: 297-298. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1339563
- Xenoulis, G., M. Psarakis, D. Gizopoulos and A. Paschalis, 2005. Test generation methodology for high-speed floating point adders. Proceeding of the IEEE International Conference on On-line Testing Symposium, July 6-8, IEEE Computer Society, Washington DC., USA., pp: 227-232. DOI: 10.1109/IOLTS.2005.67
- Yu, C.W., A.M. Smith, W. Luk, P.H.W. Leong and S.J.E. Wilton, 2008. Optimizing coarse-grained units in floating point hybrid FPGA. Proceeding of the IEEE International Conference on Field-Programmable Technology, Dec. 8-10, IEEE Xplore Press, USA., pp: 57-64. DOI: 10.1109/FPT.2008.4762366