A New Complex Floating-Point Representation Using Auto- Correlation Algorithm for DSP Processors

E.Thenmozhi Ramyah, S.Pavithra, V.Prabhakaran

Abstract— Complex number computation is a big deal in mathematical calculations and it is well practiced in all modern processors, as we all familiar with real and imaginary parts in complex numbers, computation against the real and imaginary parts becomes a difficult task over a specific inputs. Here a DSP processors is taken for signal manipulation and a new method is introduced which reduces the bits size for better output. In this paper, a new complex floating point representation for complex numbers is introduced and is compared with IEEE 754 standard and a common DSP fixed point. The resulting system will use fewer bits than IEEE 754, which keeps the dynamic range and precision. It is also proposed that it has good quantization noise analysis using auto-correlation algorithm. This new algorithm reduces the former IEEE 754 Single precision representation total bit by 1 bit and also retains variables with a better exponent and mantissa parts.

Index Terms— Complex Float ing Point Algorithmg, DSP Processors, Auto-Correlation, Multiplier

1 Introduction

The digital signal processor is a special type of microprocessor which is shortly called as DSP block and it is particularly designed for Signal Processing. The main aim of DSP is to measure, manipulate, approximate, filter and compress the continuously varying real-world analog signals. The analog signals are digitalized by sampling process then all mathematical manipulation is done. Many general microprocessors also execute digital signal processing algorithms but a separate DSP processor is designed to make the processing easier with better power efficiency which makes the more suitable for many portable devices. All mathematical computation is designed with the help of algorithms which works on variables. All the fixed numbers and floating numbers are stimulated for analytical calculation. Generally a DSP processor has a capability to handle a fixed point data and a variable point data with fixed point architecture and variable point architecture. There are several aspects are considered for selecting a fixed point or floating point ranging from precision, range and also characteristics of DSP Processor which is well suited for cost, speed, and power consumption. Floating point allows a wide dynamic range automatically whereas fixed point is limited and requires the user to track the magnitude of the numbers. The computational range of floating point DSP greater and that's the reason for the fixed point DSP processor is usually less cost with shorter cycle time and less power consumption. There are many ways to improve the floating point benefits without a heavy cost. A common method used is to prefix an exponent part on to a vector of real and complex fixed point variables which performs the arithmetic calculation with complexity close to fixed point complexity. In this approach the elements in the vector will be at a similar energy level, if not, grave quantization effects may occur.

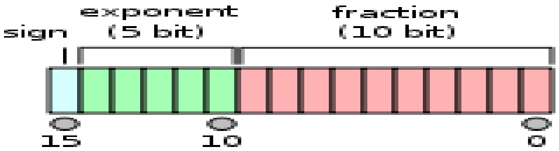
Few works are done to improve the dynamic range of fixed point by implementing "dual-fixed-point" representation, which will adds an exponent of 1 bit to fixed point representation, and some made an attempt to achieve this by "static floating-point" which indicates the values as fractional parts, similar to mantissa part of floating point representations which uses static techniques to normalize the values with implicit exponent tracking in software tools.

Many other methods are also introduced to improve the performance of floating point architecture to achieve cheaper and faster floating point solutions, by using a internal representation with redundancy which also simplifies the hardware modules of by improving the arithmetic module itself. Complex number computation is the heart of common signal processing algorithms, where a very complex task can be completed with an ease in time, all DSP processors inherently support complex number storage and processing. Complex number has a real and imaginary component, where each component can be expressed in fixed or floating point representation. In this paper a new system is designed for complex floating point representation and an improved version of quantization noise analysis using auto-correlation algorithm is introduced. This representation maintains the dynamic range and precision of 16-bits IEEE 754 standard floating point representation. First time Xilinx implementation of variable is done and its footprint is analyzed and compared with 16-bit IEEE 754.

- *E.Thenmozhi Ramyah is currently workring as Assistant Professor in Department of Electronics and Communication Engineering at Saveetha School of Engineering, Saveetha University, Thandalam, India, PH-9677071763. E-mail : ramyah.bloom@gmail.com*
- *S.Pavithra is currently working as Assistant Professor in Department of Electronics and Communication Engineering at Saveetha School of Engineering, Saveetha University, Thandalam, India, PH-9884246585. E-mail : pavithrra1286@gmail.com*
- *V.Prabhakaran is currently working as Assistant Professor in Department of Energy and Environmental Engineering at Saveetha School of Engineering, Saveetha University, Thandalam, India, PH-9884634252. E-mail : prabhakaranprof@gmail.com*

1.1 16-bit IEEE 754 Floating Point

It is a binary floating point computer number which uses 16 bits to hold a real part and another 16 bits to hold imaginary part of complex numbers in IEEE 754-2008 standard. This representation is similar in floating point Processors.



1.2 Basic format for IEEE 754 binary floating-point

This format has a floating point number which has a sign bit (1 bit wide), an exponent (5 bits wide) and an unsigned mantissa (10 bits wide), although mantissa is 10bits wide it has 11-bit precision. If the mantissa is large then the 1bit called 11th bit is fixed, which can be explicitly stored, if the mantissa is small then only 10 bits are fixed, thus the 1 bit is called as a hidden bit. These numbers are called as subnormal numbers.

1.3 Representing of floating point numbers

SIGN BIT	EXPONENT BIT	MANTISSA BIT
0 – Positive	0<E<255	Magnitude of numbers
1 – Negative	e=E-127	Hidden integer

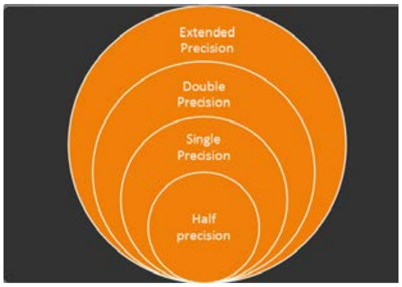
2. COMPLEX FLOATING POINT

- It uses a common exponent for real and imaginary components which reduces the size of the data word indirectly by expanding mantissa/exponent without enlarging the data word. This complex floating point is based on following criteria
- The real and imaginary component originate from the same source data which means there is a correlation between their dynamic range and the amount of correlation mainly depends on input.
- The fixed point DSP algorithm generally uses max levels for real and imaginary components and the scaling factor is same for both cases.

2.1 Performance factors for floating point processors

- The Dynamic range is increased (i.e) biggest numbers can be accommodated within the memory space
- The precision is increased with a correct value factor of exponent and mantissa which directly boost the degree of accuracy
- Both real and imaginary components can be processed at the same time.

2.2 Floating Point Precision Format



2.3 Merits of New Floating Point Representation

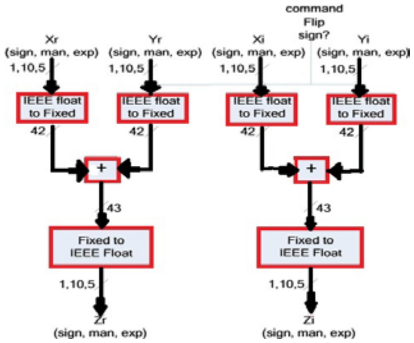
- Area of the register and memory used for the particular operation is reduced.
- It is very much cost effective.
- Quantization noise level is very much negligible.
- This new algorithm uses only 29 bits while the conventional method uses 32 bits for operation.
- The arithmetic module is comparatively bigger to handle complex calculations.

2.4 Comparison table of 32 bit IEEE 754 (vs) New Algorithm

S.NO	IEEE 754 PRECISION ALGORITHM		NEW ALGORITHM	
	Factors	Total No. of Bits used	Factors	Total No. of Bits used
1.	Sign Bit	2	Sign	2
2.	Exponent Bit	10	Exponent	5
3.	Mantissa Part	20	Mantissa Part	22
	Total	**32**	**Total**	**29**

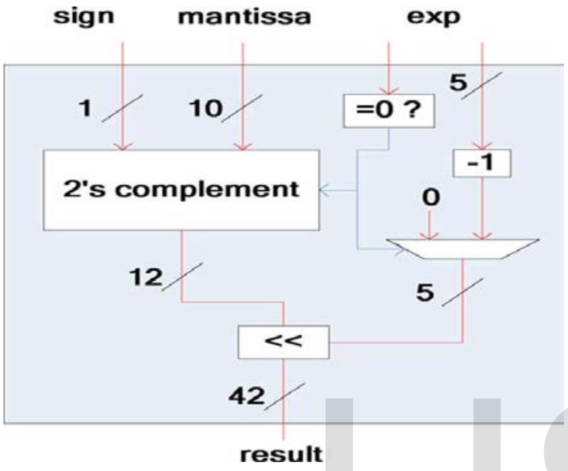
From the table its very well understood that the proposed new algorithm uses only 5 bits to hold their exponent and also it relatively increases the space for mantissa part which is used to hold 22 bits for complex computation, when the total bits consumed is reduced by 1 bit compared with former which is of 32 bit, the computation speed and the data transfer rate is increased also it can be easily accommodated in the register with better area efficient.

3. 1 - 16 bit IEEE 754 Implementation



The above shown is a basic structure of 16 bit IEEE 754 Representation with floating point algorithms. This representation is for two variable functions, initially the two data sets with sign, exponent and mantissa bits are fed into the IEEE float point algorithm unit which converts the floating point to fixed point with 2's complement conversion, and then the two data bits are added using an addition unit, finally the resultant is again converted into floating point variable which gives the desired output.

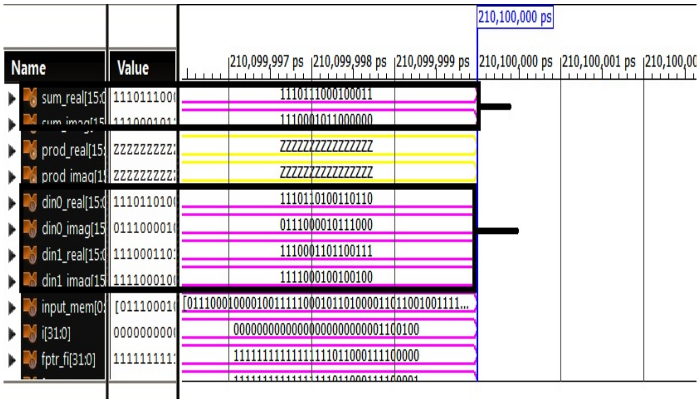
3.2 Floating to fixed point conversion point



Initially, the variable with sign and mantissa bit are fed to the 2's complement block where the 2's complement conversion is done, then the exponent part is fed to a multiplexer unit and the result is summed up with the 2'somplement output to derive the final output.

4. SIMULATION

4.1 IEEE 754 floating point implementation



```
210,100,000 ps
Name              Value        |210,099,997 ps |210,099,998 ps |210,099,999 ps |210,100,000 ps |210,100,001 ps |210,100,0C
sum_real[15:0     1110111000(        1110111000100011
sum_imag[15       111000101:         1110001011000000
prod_real[15:     ZZZZZZZZZ:         ZZZZZZZZZZZZZZZZ
prod imag[1!      ZZZZZZZZZ:         ZZZZZZZZZZZZZZZZ
din0_real[15:0    1110110100         1110110100110110
din0_imag[15      011100001(         0111000010111000
din1_real[15:0    111000110:         1110001101100111
din1 imag[15      111100010(         1111000100100100
input_mem[0:      [011100010   [0111000100001001111100010110100001101100100111...
i[31:0]           000000000(       00000000000000000000000001100100
fptr_fi[31:0]     111111111:       11111111111111111011000111100000
                                   11111111111111111011000111100001
```

4.2 Results for output sample in XILINX

4.3 Inference from Simulation Results

- It is shown that the data which are large enough holds less space with the reduced bit size.
- The above waveform shows a good data reliability in data with good dynamic range which holds large data for better integrity.
- The space in the registers is increased to hold more data bits with reduced bits computation.
- The precision is increased in exponent and mantissa parts which also boost the degree of accuracy

CONCLUSION

This paper shows the new method for "Complex floating point" representation which uses only 29bits to represents a complex value which is a 1 bit reduced format. The 29 bits has 5bits of exponent part which shares the real and imaginary parts. This new complex floating point is compared with standard IEEE 754 representation which uses 32 bits for its representation. The dynamic range and precision levels are also compared with the standard format and indicated with a good parameter which contributes the new parameter representation.

By implementing the new complex floating point algorithm in DSP Processor, user can get more intense on speed and reliability with a better area efficient which consumes smaller memory units at the expense of bigger logic and mirror quantization degradation.

REFERENCE

1. H. A. H. Fahmy, A. A. Liddicoat, andM. J. Flynn, *"Improving the effectivenessof floating point arithmetic,"* in *Proc. Asilomar Conf. Signals, Syst. Comput.*, 2001, vol. 1, pp. 875–879.
2. A. Beaumont-Smith, N. Burgess, S. Lefrere, and C. C. Lim, *"Reduced latency IEEE floating-point standard adder architectures,"* in *Proc.IEEE Symp. Comput. Arithmetic*, 1999, pp. 35–42.
3. P. S. Paolucci, *"Complex Domain Floating Point VLIW DSP With Data/Program Bus Multiplexer and*

Microprocessor Interface," U.S. Patent 7 437 540, Oct. 14, 2008.

4. S. Katayanagi, ***"Complex Vector Operation Processor With Pipeline Processing Function and System Using the Same,"*** U.S. Patent20030009502, Jan. 9, 2003.
5. R. G. Cox, M. W. Yeager, and L. L. Flake, ***"Single Chip Complex Floating Point Numeric Processor,"*** U.S. Patent 4996661, Feb. 26, 1991. *IEEE Standard for Floating-Point Arithmetic*, IEEE Std 754-2008, Aug. 29, 2008, PP. 1–58.
6. Nadav Cohen and Shlomo Weiss, ***"Complex Floating Point – A Novel Data Word Representation for DSP Processors"***, IEEE Transactions on circuits and systems-I : Regular papers, Vol.59, No.10, October 2012.
7. Swartziander,E.E, Saleh H.H, ***"Fused floating-point arithmetic for DSP"***, Signals, Systems and Computers, 2008 42 Asilomar Conference DOI: 10.1109/ACSSC.2008.5074512. Publication Year: 2008, Page(s): 767-771
8. Dong Da-ming, Fang Yong-hua, Xiong Wei, Lan Tiange, ***"Design of a high speed spectral signal processing system with a floating –point DSP for FTIR spectrometer"***, Electronic Measurment and Instruments, 2009. ICEMI'09, 9thInternatiional Conference, DOI:10.1109/ICEMI.2009.5274062. Publication Year : 2009, Page(s): 4-35 – 4-39.
9. Chun Te Ewe, ***"Dual fixed-point : an efficient alternative to floating-point computation for DSP applications"***, Field Programmable Logic and Appliucations, 2005, International Conference, DOI:10.1109/FPL.2005.1515822, Publication Year:2005, Page(s) : 715-716.
10. Palsodkar.P, Gurjar.A. ***"Improved fused floating point add-subtract and multiply-add unit for FFT implementation"***, Devices, Circuits and systems(ICDCS), 2014, 2nd International conference on DOI:10.1109/CDCSyst.2014.6926157. Publication Year : 2014, Page(s):1-5.