

Formal Error Analysis and Verification of a Frequency Domain Equalizer

Anis Souari and Sofiène Tahar

Department of Electrical and Computer Engineering
Concordia University
Montreal, Quebec, Canada
Email: {asouari, tahar}@ece.concordia.ca

Amjad Gawanmeh

Department of Electrical and Computer Engineering
Khalifa University of Science, Technology and Research
Sharjah, UAE
Email: amjad@ece.concordia.ca

Abstract—In this paper we provide an approach for the formal verification of a frequency domain equalizer using higher-order logic based theorem proving. We perform a multi-level formal error analysis to verify an implementation of the equalizer based on the Fast LMS (Least Mean Square) algorithm. The formal error analysis is performed at the floating-point, fixed-point, and real numbers domains. The expressiveness of higher-order logic allows us to model the equalizer in all the three number domains and value the errors generated by approximating the floating- and fixed-point designs to the real domain of the frequency domain equalizer. This application shows the efficiency of formal methods in analyzing and verifying complex systems such as the frequency domain equalizer.

I. INTRODUCTION AND MOTIVATION

Equalization is one of the applications of adaptive filtering. Its role consists of eliminating the inter-symbol interference caused by the noise in the transmission environment. To get an output matching as much as possible to the desired response, uncountable adaptive algorithms are used to regulate the filter or the equalizer coefficients. To decrease the filtering complexity, the equalizer can be implemented in the frequency domain using the Fast Fourier Transform (FFT) and IFFT (Inverse FFT), where time convolution is replaced by frequency multiplication. This method offers low complexity growth in comparison with the time domain method.

Data processing and filtering in the frequency domain requires dealing with data at different domains: real numbers, floating-point numbers, and fixed-point numbers domains. This conversion generates and accumulates errors due to the different level of accuracy provided by each number's domain. Therefore, a frequency domain multiplication based system must be tested thoroughly, and error analysis must be conducted to be sure about the correctness of its operation.

Verifying the correctness of the equalizer is very challenging because, first, the equalizer implementation is based on an iterative algorithm, second, it contains multiple FFT and IFFT blocks, finally, it contains multiple mathematical operators in different number domains. As a result, errors are naturally generated during data conversion between these domains, and can accumulate while performing various algorithmic iterations, FFT, and IFFT operations. Therefore, an implementation of such system must be verified in order to be sure that error accumulation is within acceptable limits. Traditionally,

this is achieved using simulation. For instance, the Simulink framework [1] can be used in order to develop and simulate a model for the system under test, where the error is estimated in every step of the simulation for a number of iterations. To achieve a certain level of assurance, specifically, in safety critical applications, estimation based simulation becomes inadequate and requires tremendous time. To overcome these problems in simulation, we will use a theorem proving based verification in order to provide formal error analysis for the equalizer.

Higher-order-logic (HOL) theorem proving [2] is a formal method that is used to conduct precise analysis of various systems. It is based on a system of deduction with a precise semantics and is expressive enough to be used for the precise specification of systems such as the frequency domain equalizer. In this work we will use the HOL theorem prover [2] in order to provide error analysis for an implementation of the frequency domain equalizer based on the Fast LMS algorithm [3]. This analysis is required to show that the error generated in the implementation of the Fast LMS algorithm conforms with the required accuracy of conversion in the equalizer design to operate properly. The formal analysis of the design intends to show that, when converting from one number domain to another, the algorithm produces the same values with an accepted error margin caused by the round-off error accumulation. To this end, we will adopt a similar approach to the DSP (Digital Signal Processing) verification methodology developed by Akbarpour [4].

John Harisson [5] used the HOL-Light theorem prover to approximate floating-point algorithms to their mathematical counterparts. He mainly proved that the floating-point exponential function has a correct overflow behavior and when this overflow is absent, the result is linked to a precise error value. Akbarpour [4] continued the work of [5] and proposed an error analysis framework based on theorem proving and dedicated specially to DSP algorithms. The methodology is based on the idea of representing the system in the three domains; the real, the floating-point and the fixed-point. Akbarpour applied his technique on digital filters [6] as well as on a 16 point radix 2 FFT [7]. Abdullah [8] adopted the methodology of Akbarpour to study the error analysis of an FFT-IFFT system which is a combination of a 64 point radix 4 FFT and IFFT

blocks. This proved that the work in [4] is reusable, scalable, and applicable on big case studies.

This paper is also considered as an application of the formal verification framework developed in [4] since it is dealing with the error analysis of a frequency domain equalizer. However, the application we verify in this work is considered more complex and error prone than the design in [9], where there is only a single combination FFT and IFFT blocks, whereas our system is composed of three FFTs and two IFFTs. In addition, the equalizer is based on more arithmetic operations that use complex numbers of type real, floating-point and fixed-point numbers. Hence, the formalization of error expressions and error analysis we intend to perform on the design requires more complex theorems for the different number domains and design blocks. Finally, error analysis for the equalizer requires formalizing input vectors to be able to store various symbols in each iteration of the Fast LMS algorithm.

II. FREQUENCY DOMAIN EQUALIZER

The implementation of the frequency domain equalizer [10] is based on the Fast LMS adaptive frequency domain algorithm [11]. As illustrated in Figure 1 below, it uses the overlap-save convolution algorithm [11]. First, the input signal is transferred from time domain into frequency domain using FFT. Then its conjugate is multiplied by the current error in the frequency domain, and gradient constraints is applied on the resulting signal. The result is multiplied by the step size parameter and added to the previous coefficients values. The output is then used to calculate the error for the next iteration. This operation is repeated again.

The implementation of the equalizer was tested using simulation in the Simulink environment [3]. In this work, we will use the theorem proving based verification in order to provide an accurate formal error analysis of the Fast LMS algorithm.

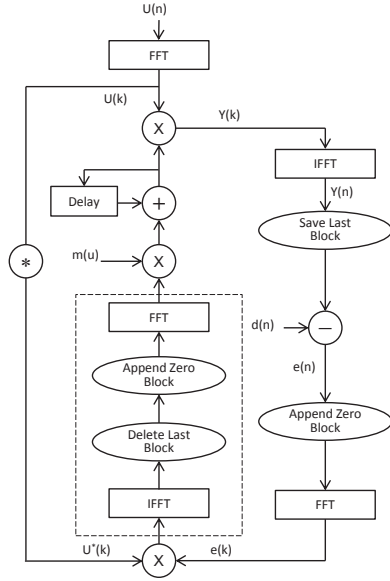


Fig. 1. Frequency Domain Equalizer using the Fast LMS Algorithm

III. FORMAL ERROR ANALYSIS IN HOL

To formalize the error due to the floating-point rounding in the HOL theorem prover, we use two fundamental theorems: the first one deals with floating-point rounding errors, and states that *if x is a real number within the floating-point range, then $x_R = x(1 + \delta)$, $|\delta| \leq 2^{-p}$* ; while the second one deals with fixed-point rounding errors, and states that *if x is a real number within the fixed-point range, then $x'_R = (x + \epsilon)$, $|\epsilon| \leq 2^{-fracbits(X)}$* , where p is the precision of the floating-point format, x is the real number, x_R is the floating-point value and x'_R is the fixed-point value. The rounding error in the floating-point domain is multiplicative, while it is additive in the fixed-point domain [12], [13]. Evaluating arithmetic operations, denoted as \star , is defined as $fl(x \star y) = (x \star y)(1 + \delta)$, where $|\delta| \leq 2^{-p}$ in the floating-point domain, and $|\delta| \leq 2^{-fracbits(X)}$ in the fixed-point domain.

The verification methodology, as depicted in Figure 2, is based on a formal model for numbers in three different domains: fixed point, floating-point, and real domains and a valuation procedure for numbers conversion. Based on this conversion, error analysis is performed between the actual real values obtained and the converted ones from both floating-point and fixed-point domains. Finally further analysis is performed to show the error analysis between fixed-point and floating-point.

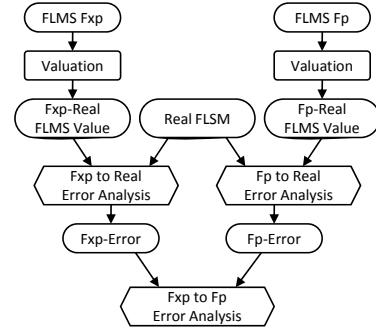


Fig. 2. Formal Error Analysis Methodology in HOL

In Figure 2, circular shapes refer to the Fast LMS algorithm in a specific number domain; real, floating-point or fixed-point, while the hexagon ones represent the error analysis that we conducted on the design in these domains.

To perform the error analysis of the Fast LMS algorithm, we used existing theories (libraries) in HOL and built on top of them the necessary theories to reason about error generation and accumulation in the equalizer. First, the construction of complex numbers was necessary for the implementation of the specification. Regarding the floating-point and the fixed-point modeling of the design, we used, respectively, the formalization of the IEEE 754 standard based floating-point arithmetic [6] and the fixed-point arithmetic HOL theorems developed by Akbarpour *et. al.* [14].

We formalize errors in the floating-point and fixed-point domains for the designs using the following two lemmas; *float_complex_val* and *fxp_complex_value*, denoted as Val_{fp} and Val_{fx} , respectively, and defined as follows:

```

float_complex_val =
⊢lemma ∀ z. float_complex_val z =
  complex(Val(float_Re z), Val(float_Im z))
fxp_complex_value =
⊢lemma ∀ z. fxp_complex_value z =
  complex(value(fxp_Re z), value(fxp_Im z))

```

where the function *complex* is used to compose a complex variable from two floating-point or fixed-point values. The effect of these functions on the arithmetic operations is inherited from the effect of the function *Val* and *value*. The function *Val* is used to define the rounding error due to the valuation of the floating-point in real number. The effect of the *Val* function on arithmetic operations is defined as:

$$\begin{aligned}
\forall a b. \exists e. Val(a + b) &= (Val a + Val b) * (1 + e) \\
\forall a b. \exists e. Val(a - b) &= (Val a - Val b) * (1 + e) \\
\forall a b. \exists e. Val(a * b) &= (Val a * Val b) * (1 + e)
\end{aligned}$$

where *a* and *b* be two floating-point numbers and *e* a real number. *e* in the above lemmas defines the error caused by the valuation of the fixed-point in real number. The effect of the *value* function on arithmetic operations is given as:

$$\begin{aligned}
\forall a b X. \exists e. value(FxpAdd X a b) &= value a + value b + e \\
\forall a b X. \exists e. value(FxpSub X a b) &= value a - value b + e \\
\forall a b X. \exists e. value(FxpMul X a b) &= value a * value b + e
\end{aligned}$$

These lemmas are dedicated to complete the error analysis for rounding numbers between different domains, and will be used to formalize FFT in the next Section.

IV. FORMALIZING THE FREQUENCY DOMAIN EQUALIZER IN HOL

In this section, we will discuss the major theorems we defined for the equalizer design. The structure of these theorems is shown in Figure 3. In order to perform complete error analysis for the whole design, each block of the Fast LMS algorithm described above in Figure 1 must be formalized in HOL. As shown in the figure below, the theorems for every block are intended to show the validity of rounding and error accumulation.

The first theorem deals with error analysis between the real and the floating-point representations of the FFT block. It consists of a simple subtraction between the floating point valuation and the real valuation of FFT. Using *C* for composing a complex number, *P_{f_p}* for float principal root function, and *P* for real principal root, this theorem, denoted as *EF_{RF_p}*, is formalized as follows:

$$\begin{aligned}
\vdash_{thm} \forall x, y, k. \exists e1, e2, e3. EF_{RF_p}(x, y, k) = \\
Val_{f_p}(P_{f_p}(0, k) * EL(0, y) * C(1 + e3, 0)) + \\
Val_{f_p}(P_{f_p}(1, k) * EL(1, y) * C(1 + e2, 0)) + \\
Val_{f_p}(P_{f_p}(2, k) * EL(2, y) * C(1 + e1, 0)) - \\
\sum_{n=0}^3 (P(n, k) * EL(n, x))
\end{aligned}$$

The real to fixed-point error analysis of the FFT is established by proving that the produced error is equivalent to the subtraction between the valuated fixed-point FFT expression and the real one. Using *MUL* for complex multiplication, and

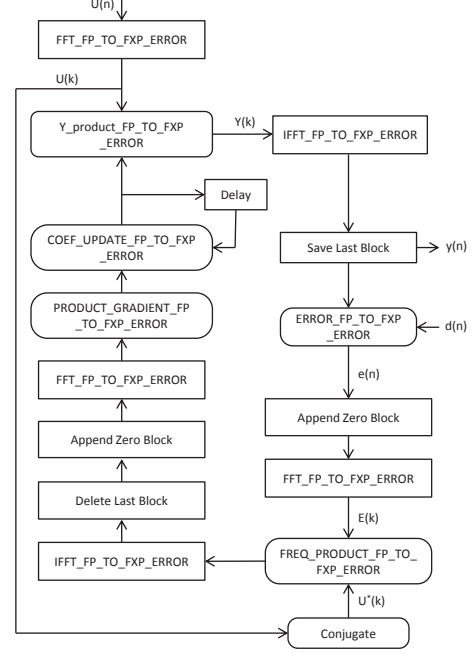


Fig. 3. Structure of HOL Theorems for the Equalizer

P_{f_x} for fixed-point principal root function, this error, denoted as *EF_{RF_x}*, is formalized as follows:

$$\begin{aligned}
\vdash_{thm} \forall x, y, X, k. \exists e1, e2, e3. EF_{RF_x}(x, y, X, k) = \\
Val_{f_x}(MUL(X, P_{f_x}(X, 0, k), EL(0, y))) + C(e3, e3) + \\
Val_{f_x}(MUL(X, P_{f_x}(X, 1, k), EL(1, y))) + C(e2, e2) + \\
Val_{f_x}(MUL(X, P_{f_x}(X, 2, k), EL(2, y))) + C(e1, e1) - \\
\sum_{n=0}^3 (P(n, k) * EL(n, x))
\end{aligned}$$

Once the real to fixed-point error analysis is achieved, the fixed-point to floating-point error analysis of the FFT block is obtained by deducting the results of the real to floating-point and the real to fixed-point error expressions as given in Figure 2 above. The following theorem, denoted as *EF_{F_pF_x}*, is defined in order to formalize the fixed-point to floating-point error analysis of the FFT. This error, denoted as *EF_{RF_x}*, is formalized as follows:

$$\begin{aligned}
\vdash_{thm} \forall x, x_p, x_f, X, k. \exists e1, e2, e3, e4, e5, e6. \\
EF_{F_p F_x}(x, x_p, x_f, X, k) = \\
Val_{f_x}(MUL(X, P_{f_x}(X, 0, k), EL(0, x_f))) + C(e3, e3) + \\
Val_{f_x}(MUL(X, P_{f_x}(X, 1, k), EL(1, x_f))) + C(e2, e2) + \\
Val_{f_x}(MUL(X, P_{f_x}(X, 2, k), EL(2, x_f))) + C(e1, e1) - \\
(Val_{f_p}(P_{f_p}(0, k) * EL(0, x_p)) * C(1 + e6, 0) + \\
Val_{f_p}(P_{f_p}(1, k) * EL(1, x_p)) * C(1 + e5, 0) + \\
Val_{f_p}(P_{f_p}(2, k) * EL(1, x_p)) * C(1 + e4, 0))
\end{aligned}$$

Similar theorems were defined and proven for the IFFT block. For illustration purposes, the theorem for real to floating point error, denoted as *EI_{RF_p}*, is formalized follows:

$$\begin{aligned}
\vdash_{thm} \forall L, L_x, k. \exists e, e1, e2, e3. EI_{RF_p}(L, L_x, k) = \\
Val_{f_p}(C_{4_f} * C(1 + e, 0) * \\
((Val_{f_p}(P_{i_{f_x}}(0, k), EL(0, L_x)) * C(1 + e3, 0) + \\
Val_{f_p}(P_{i_{f_x}}(1, k), EL(1, L_x))) * C(1 + e2, 0) + \\
Val_{f_p}(P_{i_{f_x}}(2, k), EL(2, L_x))) * C(1 + e1, 0)) - \\
C_4 * \sum_{n=0}^3 (P_{f_x}(n, k) * EL(n, x))
\end{aligned}$$

Having a HOL theorem defined and proved for every building block of the design, we then need to define one comprehensive theorem for the whole design. This algorithm combines all algorithms that define the blocks of the design together. This theorem is used to obtain the rounding error from each block and accumulate it together with the error produced by the successor block in the design as depicted in Figure 3. Eventually, the rounding error is obtained for the whole design and validated for the algorithm. This theorem is proved in HOL, which verifies that the rounding and accumulated error produced by steps of the algorithm are within the accepted range given in the specification of the design. Theorems for error analysis of the blocks of the frequency domain equalizer that were formalized and proved in HOL are presented in [15].

V. DISCUSSION

Many existing theories in HOL, e.g., *arithmeticTheory*, *realTheory*, *listTheory*, *pairTheory*, *realLib*, *numLib*, *floatTheory*, *fxpTheory*, *ieeeTheory*, ..., were used to derive the rounding error analysis of the frequency domain equalizer. The definitions of the Fast LMS algorithm were formalized and proved based on these theories. All definitions were formalized first in the real domain, and then all arithmetic operators were overloaded to build the design in the floating-point and fixed-point domains using *floatTheory* and *fxpTheory*, respectively.

The equalizer application shows that formal error analysis is applicable on larger scale systems such as the one we have analyzed, which is traditionally analyzed with paper and pencil or simulation techniques based on estimating the error. Formal analysis proves that the implementation meets its specification with 100% coverage, something that is not feasible in simulation. In addition compared to the classical analytical technique, this method is computerized and the theorems can be efficiently reused to verify other designs that make use of the same algorithm.

Some specific error analysis theorems that were used in this work were defined and proven by Akbarpour [4] and Abdullah [8]. However, we had to build our own theorems on top of these in order to formalize and verify every block of the Fast LMS algorithm, and consequently define one single theorem for the whole design. This shows that relevant theorems that are proven in HOL can be reused efficiently in order to verify complex systems of similar properties. In fact, scalability of HOL theorems is one of its best features, since proven theorems can be reused efficiently to verify other designs, which reduces time and effort, in particular while using the interactive HOL framework environment. The HOL formalization and error analysis of the frequency domain equalizer was approximately 2600 lines of code and took more than 800 man-hours of labor.

VI. CONCLUSION

In this work we apply theorem proving to provide the formal error analysis for a frequency domain equalizer design. The equalizer implementation using the Fast LMS algorithm is based on a number of mathematical operations in three

different domains: floating-point, fixed-point and real numbers. This required data to be converted between these domains, which in turn produces errors that can accumulate during several iterations of the algorithm.

Formal error analysis is used to show that errors in the equalizer algorithm occurring while converting from one number domain to the another are within the accepted range based on the design specification of the equalizer. However, this required building on top of existing HOL theorems for error analysis and the derivation of new expressions for the accumulation of round-off error in the algorithm. We developed a HOL theorem for every building block of the design, then, we defined one comprehensive theorem for the whole design to show the validity of rounding and error accumulation. Our framework scales well with the size of the systems and many of the theorems that we have proved can be reused in the error analysis of other signal processing applications.

As future work, we plan to extend the current work and perform error analysis using the GAPPA framework [16], where error analysis returns a numerical error range. Another interesting approach is to use first order theorem proving to verify properties about the functional behavior of the equalizer.

REFERENCES

- [1] Matlab / Simulink. <http://www.mathworks.com>, February 2012.
- [2] M. Gordon and T. Melham. Introduction to HOL: A Theorem Proving Environment for Higher Order Logic. Cambridge University Press, 1993.
- [3] A. Souari, FPGA Implementation of a Frequency Domain Equalizer. MASC Thesis, École Nationale d'Ingénieurs de Sousse, Tunisia, 2010.
- [4] B. Akbarpour. Formal Verification Methodology of DSP Designs. PhD Thesis, ECE Department, Concordia University, Montreal, Quebec, Canada, 2005.
- [5] J. Harrison. Floating Point Verification in HOL Light: the Exponential Function. Technical Report 428, University of Cambridge, Computer Laboratory, Cambridge, UK, 1997.
- [6] B. Akbarpour and S. Tahar. Error Analysis of Digital Filters using Theorem Proving. In Theorem Proving in Higher Order Logics, volume 3223 of LNCS, pages 1-16. Springer, 2004.
- [7] B. Akbarpour and S. Tahar. A Methodology for the Formal Verification of FFT Algorithms in HOL. In Formal Methods in Computer-Aided Design, volume 3312 of LNCS, pages 37-51. Springer, 2004.
- [8] N. Abdullah, Formal Analysis and Verification of an OFDM Modem Design. MASC Thesis, ECE Department, Concordia University, Montreal, Quebec, Canada, 2006.
- [9] N. Abdullah, B. Akbarpour and S. Tahar. Error Analysis and Verification of an IEEE 802.11 OFDM Modem using Theorem Proving. Electronic Notes in Theoretical Computer Science, 242(2):3-30. Elsevier, 2009.
- [10] P. A. Dmochowski. Frequency Domain Equalization for High Data Rate Multipath Channels. MASC Thesis, Queen's University, Kingston, Ontario, Canada, 2001.
- [11] M. O. Fril. Frequency Domain Adaptive Filtering. MASC Thesis, National University of Ireland, UK, 2005.
- [12] G. Forsythe and C. B. Moler. Computer Solution of Linear Algebraic Systems. Prentice-Hall, 1967.
- [13] J. H. Wilkinson. Rounding Errors in Algebraic Processes. Prentice-Hall, 1963.
- [14] B. Akbarpour, A. Dekdouk, and S. Tahar. Formalization of Fixed-point Arithmetic in HOL. Formal Methods in System Design, 27(1-2):173-200. Springer, 2005.
- [15] A. Souari, S. Tahar, and A. Gawamneh. Formal Error Analysis and Verification of the Frequency Domain Equalizer of a Frequency Domain Equalizer. Technical Report, Concordia University, Montreal, Quebec, Canada, 2012.
- [16] G. Melquiond, De l'Arithmétique d'Intervalles la Certification de Programmes. PhD thesis, École Normale Supérieure de Lyon, France, 2006.