

Energy Efficient Floating-Point Based Block LU Decomposition on Large Signal Systems

Anu Philip¹, Dr. M. Devaraju²

¹Student (M.E- VLSI), Dept. of ECE, KVCET, Anna University, Chennai, India

²Professor and Head of Dept. of ECE, KVCET Anna University, Chennai, India

Abstract: In this paper, we propose architecture for floating-point based Lower Upper decomposition of large signal in FPGAs which are dealing with large-sized matrices. Our proposed architecture is based on the well known concept of pipelined floating-point units and the double-precision based design helps to obtain effective throughput. According to the post layout report, the hardware efficiency is as high as 1.6× compared with the existing LUD methods, and the energy efficiency is also higher than the state-of-the-art LUD when the matrix dimension is 8×8 and larger.

Keywords: Energy efficiency, hardware compatibility, lower-upper decomposition (LUD), large signal system based FPGAs

1. Introduction

An essential algorithm in the linear solution problem. Especially, in large signal systems, matrix decomposition is the main burden for the implementation of hardware and the energy-efficient FPGAs. The existing matrix decomposition optimization methods aim at large-size matrices such as dimension with 16 kB. The practical system, the scale of antennas is limited by the area of antenna array. For example, in the long-term evolution (LTE) standards, the systems employ 4×4 dimension antenna array. Even in the large-scale signal system the required inversion matrix dimension is no more than 100. The large signal systems are interested in more hardware-efficient and energy-efficient VLSI implementation of matrix decomposition algorithm.

2. The Basic LU Architecture

LU Decomposition is essentially factoring a square matrix into an upper triangular matrix and a lower triangular matrix. A(ax;y), a n_n matrix, is decomposed into Lower triangular matrix L (lx;y) and Upper triangular matrix U(ux;y). Both are of size n_n, where x denotes the row index and y denotes the column index. In this paper, we assume that matrix A is a non-singular matrix thus we do not consider pivoting.

This architecture has been slightly modified to obtain the LU architecture function. Which is based on a circular linear array with n PEs where n is the problem size. The input and output ports to the architecture are connected to the first PE, PE1. PE1 is different from the other PEs in that it has a divider and doesn't have the multiplier/subtractor as in the other PE2 to PEn which are identical. The elements of the input matrix are fed in column-major order.

The output matrix is the combined L and U matrices. Each PE2_j_n has two input and two output ports. The last PE requires only a single output port and this output port is connected back to the second input port of PE1. This essentially facilitates scheduling for the division required for the L matrix, to happen in PE1. Thus the data is input into PE1 and passes through PE2 to PEn and comes back

to PE1 and is fed out as output. After division in PE1, the elements of the L matrix are both fed out as output and fed into PE2 through PEn for further iterations. We use the pipelined floating-point units (FPUs) to achieve high throughput. We proposed the scheme of stacked matrices to resolve the data dependencies incurred due to the large latencies of the deeply pipelined FPU's. The stacked matrices scheme is for computing the LU series of matrices.

3. Algorithm for Block LU Decomposition

Input: A_n_n matrix with elements aij

Output: LU Decomposition of matrix A

Step 1: Perform a sequence of Gaussian eliminations on the n_b matrix formed by A₁₁ and A₂₁ in order to Calculate the entries of L'₁₁, L'₂₁, and U'₁₁.

Step 2: Calculate U'₁₂ as the product of (L'₁₁)⁻¹ and A₁₂.

Step 3: Evaluate A'₂₂ = A₂₂ - L'₂₁ U'₁₂.

Step 4: Apply Step 1 to 3 recursively to matrix A'₂₂. During the kth iteration, the resulting submatrices L(k)₁₁, U(k)₁₁, L(k)₂₁, U(k)₁₂ and A(k)₂₂ are obtained.

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} L_{11} & \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} \\ & U_{22} \end{pmatrix}$$

The block LU decomposition architecture uses the floating-point matrix multiplication architecture and a single PE for performing the subtraction of matrix. Thus, two sets of PEs are invented: one set of b PEs for b_b LU decomposition and another set of b PEs for b_b matrix multiplication and a single PE for matrix subtraction

Hardware efficiency of stochastic process:

The method based on stochastic process models for machine RUL prediction. First, a new stochastic process model is constructed with the multiple variability sources of machine stochastic degradation processes simultaneously. The Kalman particle filtering algorithm is used to estimate the system.

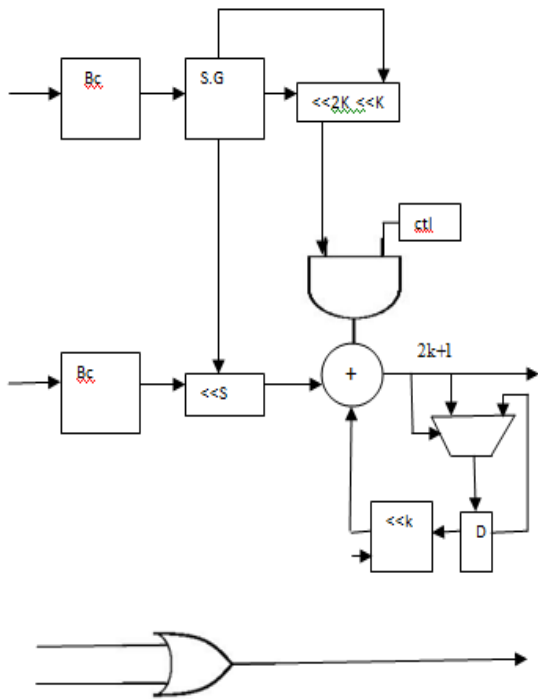


Figure 1: High accuracy SD

States and to predict the RUL. The effectiveness of this method is to demonstrate simulated degradation processes and to accelerate the degradation tests of rolling element bearings. Through comparisons with different methods, the proposed method presents its superiority to describe the stochastic degradation processes and to predicting the machine RUL.

A design and synthesize the LUD with 4×4 , 8×8 , and 16×16 dimension matrices.. The stochastic stream length is $2k = 256$. Hence, the stochastic LUD has almost the same performance as a 15-bit TCS. The implemented stochastic LUD with the synthesis results of the QR-CORDIC, direct mapped, and CSHM-based LUD. For fair comparison, we have scaled the throughput, area etc.. Since the area and throughput are quite dissimilar for every method. the hardware efficiency and energy efficiency to compare the performance as:

Hardware Efficiency = Scaled Throughput/ Gate Counts [(MS/s)/kGE]. Energy Efficiency = Scaled Power / Scaled Throughput [nJ/Symbol].

4. High Accuracy Hardware Design in LU Decomposition Unit

The high-performance multiplier and divider for DPC provide high computation accuracy with relative short stream and low hardware expense. The proposed stochastic multiplier and SD can achieve SNR of 70, 65 dB with 128-length bit stream, respectively. Also the stochastic multiplier and SD can be applied to other signal processing systems. Stochastic LUD can be applied in the practical signal detector, which is verified by SNR and MMSE-based packet-error-rate (PER) performance. The LUD has been implemented in a fully parallel form. Thus, the proposed LUD has a simple control and a computation

structure. According to the implementation report, after placing and routing, the hardware efficiency is $1.5 \times$ than existing LUD architectures. Energy efficiency also surpass the CSHM-based LUD when the dimension is equal or higher than 8×8 . The stochastic logic has been proposed decades ago and widely used in the neural network system. Most recently, researchers employed the stochastic computation in the wireless communication and signal processing systems to achieve inspired results.

a) Brent Kung Adder:

The Brent Kung adder computes the prefixes for 2 bit groups. These prefixes are used to detect the prefixes for the 4 bit groups, which in turn are used to calculate the prefixes for 8 bit groups and so on. These prefixes are then used to compute the carry out of the particular bit stages. These carries will be used along with the Group Propagate of the next stage to compute the Sum bit of that stage.

Brent Kung Tree uses $2 \log_2 N - 1$ stage. So for a 32-bit adder oin our design requires 9 number of stages and fanout for each bit stage is restricted to 2. The diagram below shows how the fanout being minimized and the loading on further stages are being reduced.

Brent-Kung Carry Network (8-Bit Adder)

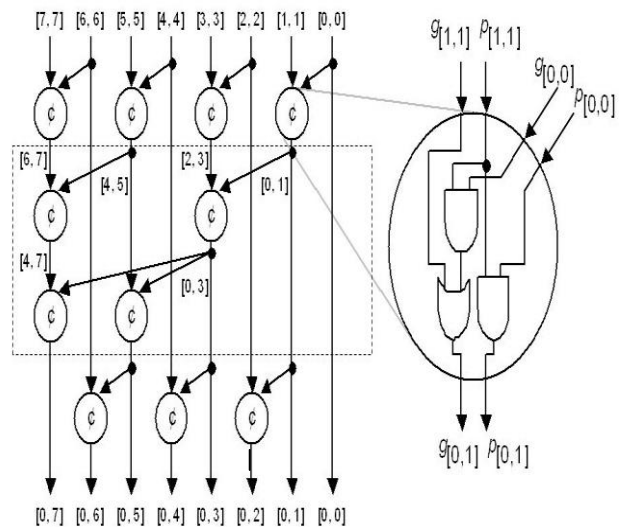


Figure 2: Brent Kung Adder

The logarithmic concept is utilized to combine its operands in a tree-like orientation. The logarithmic delay is acquired by restructuring the look-ahead adder. The restructuring merely dependence on the associative property, and delay is obtained equivalent to $(\log_2 N) t$, where 'N' denotes the number of input bits to the adder and t denotes the propagation delay time. Hence, for a 16-bit structure, the logarithmic adder has a delay equal to $4t$, while for a simple ripple carry adder the delay is given by $(N-1)t$ and is equal to $15t$ for 'N' and 't' being the number of input bits and the delay time, respectively. Hence it is seen that this structure greatly reduces delay, and makes it beneficial for a structures with large number of inputs. This advantage is, however, obtained at the expense of large area and a complex structure.

b)Vedic Multiplier

For the effective multiplication Conventional Vedic multiplication Hardware has been used which provide ultra precision in the carry propagation path. the proposed Vedic multiplier in at eight bit level has upshot optimized parameter characteristics compared to some other popular multiplier structures based on different multiplication algorithms at the eight bit level. For true and reliable comparison, proposed multiplier has been implemented on the same platform of target FPGA, which has been used by the reference papers.

Comparison with various multipliers in the target FPGA is shown in Table1.

In the following given table the target FPGA used belongs to Virtex 2P (family), XC2VP2 (device), FG256 (Package), 7 (speed grade).

Table 1: Comparative Table 1 for Different Multipliers at 8-Bit Level

Maximum Combination Path Delay for Different Multipliers at Eight Level in Nano Seconds						
Karatsuba [10]	Vedic Karatsuba [10]	Modified Booth Wallace [4]	Vedic with Partitioning [4]	Conventional Vedic [10]	Vedic with CSA [4]	Proposed [18]
31.039	18.695	15.815	15.685	15.418	13.07	11.886

5. Results

The proposed method has been simulated by using Modelsim 6.2c as well as synthesized using Xilinx ISE 14.5. The results produced shows that the proposed reduction of time delay method is optimized in terms of parameters such as power, area, delay, speed and ease of implementation.

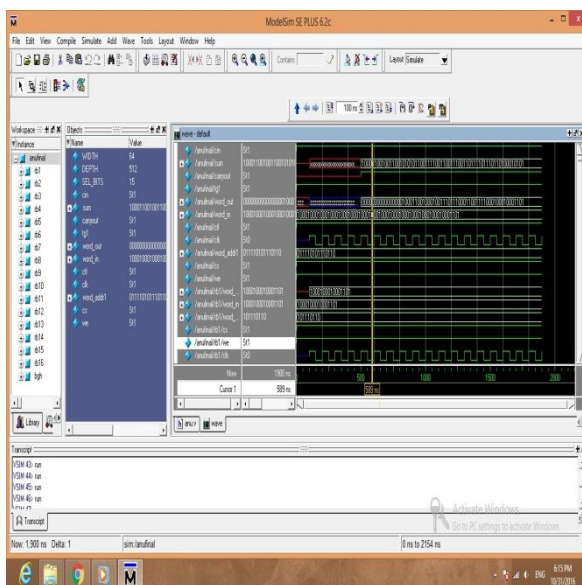


Figure 3: Simulation Result

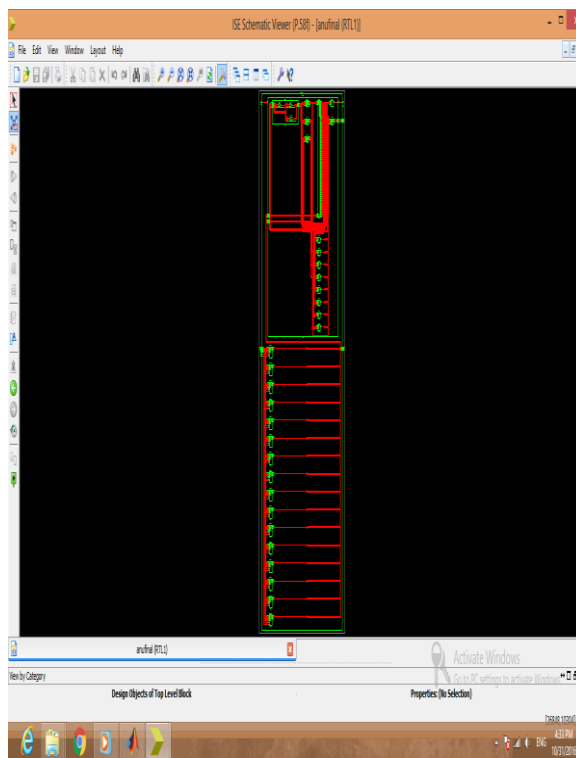


Figure 4: RTL Schematic View

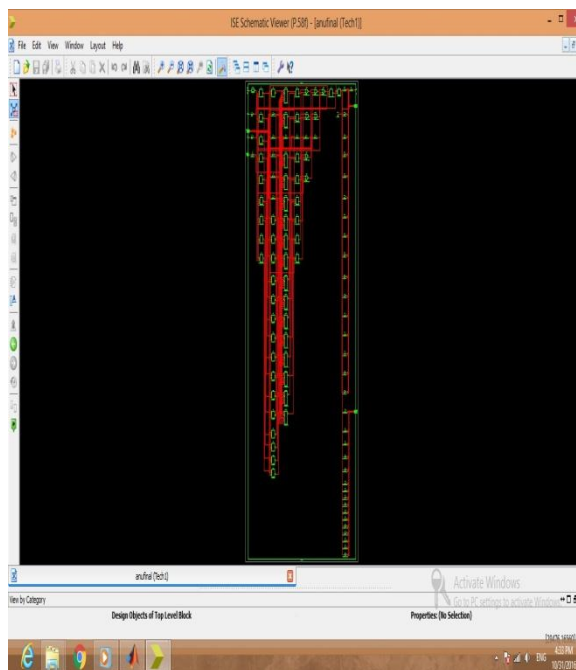


Figure 5: Technology Schematic

6. Conclusion

In this paper, energy efficient LU Decomposition Scheme for large signal systems is analyzed well. Extensive performed simulation results prove that Brunt Kung adder as well as Vedic multiplier along with the floating point pipelining in the architecture of FPGAs for large signal decomposing systems devised with LUD computation schematics is the efficient method to make energy efficiency in hardware operation.

The simulations show significant improvement over the state-of-the-art techniques on performance and energy

consumption, and effectively demonstrate the feasibility and efficiency of the approach.

References

- [1] A. Alaghi and J. P. Hayes, Eds. (2012), "Survey of stochastic computing," in ACM Trans. Embedded Comput. Syst. vol. 12, no. 2s, 2012, Articlenu. 92
- [2] M. Arzel, C. Lahuec, C. Jego, W. J. Gross, and Y. Bruned, (2012) "Stochastic multiple stream decoding of cortex codes, " IEEE Trans. Signal Process., vol. 59, no. 7, pp. 3486–3491, Jul. 2011
- [3] Chandrachoodan, M. K. Jaiswal and N.(2012)"FPGA-based high performance and scalable block LU decomposition architecture, " IEEE Trans. Comput., vol. 61, no. 1, pp. 60–72, Jan. 2012.
- [4] S. Choi, G. Govindu V. Prasanna, V. Daga, S. Gang3adharpalli, and V. Sridhar, (2004) "A high-performance and energy-efficient architecture for floating-point based LU decomposition on FPGAs," in Proc. 18th Int. Parallel Distrib. Process. Symp., Apr. 2004, pp. 149–156.
- [5] Y. Dou, G. Wu, J. Sun, and G. D. Peterson, (2012) "A high performance and memory efficient LU decomposer on FPGAs, " IEEE Trans. Comput., vol. 61, no. 3, pp. 366–378, Mar. 2012

Author Profile



Anu Philip, received B.E degree in Electronics and Communication Engineering in the year 2015 and Pursuing M.E in VLSI Design at KVCET affiliated to Anna University, Chennai-India



Dr. M. Devaraju, Professor and Head of Department of Electronics and Communication Engineering, KVCET, Chennai- India