Speed and Area Efficient FXP Adders and Multipliers: A Comparative Analysis for LNS System

M S S M Basir¹,², R C Ismail²,³, S Z M Naziri², M N M Isa², S A Z Murad², A Harun²

¹Department of Electrical Engineering, Politeknik, 96400 Mukah, Sarawak, Malaysia
²School of Microelectronic Engineering, Universiti Malaysia Perlis, Malaysia
³School of Computing and Informatics, Albukhary International University, Malaysia

Abstract. In this paper, a variety of adder and multiplier are compared to be implemented in a new logarithmic number system (LNS). Both adder and multiplier are designed with a generic very high-speed integrated circuit hardware description language (Verilog) program. This makes it possible to achieve the optimum performance in latency and area of 0.18µm CMOS technologies LNS chip. Consequently, the optimal configurations vary with speed and area of the schemes and in some cases can be compact area, O(n), fast in latency O(log₂ n) or optimized. The program was scripted based on fixed-point (FXP) adders and multipliers that yet will be implemented in LNS system. The functionality of the scheme was tested before synthesized. Outcomes show that Ladner Fisher (LF) adder and modified Baugh Wooley multiplier contribute to fast in latency and consume minimal area.

1. Introduction

Designing a high-performance digital signal processing (DSP) architectural is crucial in advanced computation area of time-varying signal processing, image processing and machine vision (Yao et al, 2015; Khirade and Patil, 2015 and Wang et al, 2016). Operation of DSP microprocessor involves with executing the input data through arithmetic functions (i.e. addition, subtraction, multiplication, division, square, and square root) to perform the calculation. For a real-time application that usually involved with a wide dynamic range of numbers required DSP algorithm to be computed in high speed with lowest delay. Modification in the computation of arithmetic functions becomes an interesting topic (Rodríguez-Andina et al, 2015) in improvising the DSP architecture for wordlength (which governs range and precision), accuracy (i.e., error, both in quantisation and processing), speed and area (Coleman and Ismail, 2016).

In an early stage of DSP architecture, fixed-point (FXP) technique is employed where the operand involved with an integer using a fixed-length fraction. The FXP can perform well in complex DSP task with minimal bit-width, resulting minimize overall area, power, and delay for finite word-length (Fang et al, 2003). Drawback of FXP is the accuracy problem results from the use of fixed-length fraction. As an alternative, floating-point (FLP) is proposed with better precision and higher dynamic range of number compared to FXP. In addition, advantage in high-speed multiplication process follow IEEE 754 Standard gives benefit towards lowest delay for a dynamic range of the number required by real-time embedded system (Dido et al, 2002). However, FLP cannot tolerate complex operation i.e., division and square root make the executing time much slower. Thus, Coleman et al. (Coleman and
Ismail, 2016; Coleman et al, 2000; Ismail and Coleman, 2011 and Coleman et al, 2001) proposed microprocessor based on logarithmic operation known as a logarithmic number system (LNS), which gives simplification in multiplication, division, square, and square root operations.

Using similar architecture for addition and subtraction performs by FXP, LNS provide simple calculation for multiplication and division cause a great improvement in accuracy and speed. Unlike multiplication and division operations, involvement of non-linear function occurs at LNS addition and subtraction operations lead to time consuming. Despite simply ignoring the trade-off, research conduct by Coleman et al, 2000; Naziri et al, 2015 and Naziri et al. 2014 proposed an interpolation and co-transformation methods towards LNS addition and subtraction operations. These methods provide apparent reduction for the look-up table (LUT) used in terms of size and complexity makes LNS comparable with FLP (Coleman and Ismail, 2016). Since then, studies in improvising the LNS architecture have been increased and additional works are focused on non-linear function.

2. Addition and Multiplication

Based on the review conducted by Pudi and Sridharan, 2011, conventional ripple carry adder (RCA) structured by the direct composition of full adder is proved to be area efficient with a complexity of $O(n)$. To overcome the counterbalance of RCA by a high delay, carry select adder (CSLA) is proposed. Unlike RCA, CSLA is structured by connecting two RCA in parallel, one input is served by 0 and another one is 1, hence giving an early prediction of the outcomes. In the worst-case scenario, a complexity can reach up to $O(\sqrt{n})$, and possibility for faster in term of speed is guaranteed.

Another promising adder concept is by implementing carry look ahead adder (CLA), through the self-reliant circuit to bypass outputs of inferior rank. Increment in latency, results from the direct computation of carrying at the MSBs position, although sum has not yet been computed. The direct translation of this structure leads to a reduction in the wait time as compared to RCA, on the other hand downgrade the area performance. Based on this CLA paradigm, a couple of parallel prefix adders namely Brent Kung (BK), Kogge Stone (KS) and Ladner Fisher (LF) are mapped. These adders architecture are structured taking a number of stages, associative operations and majority gates as criteria (see Figure 1). The complexity of $O(\log_2 n)$, which is one of the fastest scheme as compared to conventional adder. The concept of parallel prefix adder, taking an example for LF for 32 bit (mapped with 5 stages) as graphically shown in Figure 2 is by converting blue-striped square ($a_i, b_i$) to propagate and generate ($p, g$) bits and red-striped square will revert ($p, g$) to output $s_i$. Dark circle represents ($p, g$) compressors and the process is by expending from ($p_i, g_i$) to a couple of ($p'_i, g'_i$). The architecture of parallel prefix adder is differentiated by its interconnections at compressors taking KS as a favourite because of its minimum fan-out (Koren, 2001), BK which possesses maximal depth which accounts for increased latency by simple tree structure as well as proposed LF (Weste and Harris, 2015) using improved Sklansky and BK topology resulting faster computation at speed path (Shilpa et al., 2018).
Unlike addition, multiplication seemsly a bit simpler with low control overhead (rounding logic, normalization, management of particular cases) to perform a computation. In FLP format, both inputs at mantissas can be simply multiplied by a conventional integer multiplier (similar to FXP), while adder is used at exponents. Based on the comparative study of FLP adder and multiplier conducted by Barrois (2017), 32 bits FLP multiplier is 48% smaller in area compared to FXP multiplication. In addition, due to the large area consume by FXP multiplication, the power consumption recorded is 4.7 times higher than FLP multiplier. Thus, computing using the FLP numbering format is less penalizing and this arrangement is expected to give an advantage in proposing a compact LNS system.

3. Measurement of Speed and Area

In a 32-bit LNS ALU architecture, measurement of speed and area for addition-subtraction, multiplication-division and subtraction co-transformation are itemized. Logic gates and flops of
flowthrough LNS units in a form of silicon lattice are routed in the Faraday CMOS 0.18 µm technology, rendered using Synopsys compiler. For a synthesizable design process as graphically shown in Figure 3, the Verilog coding scripted in CAD tool (Quartus II) must first attest through the functional verification stage. In this case, each set of modules, includes proposed adder and multiplier are verified using a test bench with every possibility in 32 bits numbers. The process will be repeated if there is an error during debugging considering minor error prescribe for incorrect coding and a necessity to redesign if major error that can incriminate system correctly, functionality and reliability.

After the functional verification shows a correctness, synthesized procedure is taken place, by selecting the target library for a pre-synthesis schematic. While executing, the unconstrained performance of speed and area are estimated, targeting for optimum design. To meet a specific goal that required either speed or area to be constrained, these parameters are varied by repeating the
synthesis process until target design is achieved. For the purpose of illustration, gate-level netlist consists of logic gates and flops are saved in Verilog HDL format. The design is deemed to be successful if all the tests are passed, hence further actions like prototyping that involved with FPGA board or fabricating which are not covered in this research can take place.

4. FXP Adders Performance Comparison
This subchapter substantiated by performance comparison of variety of well-known adder schemes namely RCA, carry skip adder (CSA), CSLA, carry increment adder (CIA), CLA as well as chosen parallel prefix adder such as BK and LF synthesized under VLSI 0.18\textmu m CMOS technologies with 50MHz clock frequency. Most cases of the adder schematic entry describe (see Table 1), for 32 bits wordlength is constructed by cascading 8 × 4 bits full adder block in series, followed by verification of design functionality using test bench before synthesized. Synthesis emphasis is focuses on optimum performance using automatic timing and area minimization. Further, digital circuit adder that exhibit large area will consume more power since both of them is proportional (Dutt et al, 2019).

| Logical Utilization       | Area (\textmu m^2) | Latency (ns) |
|---------------------------|--------------------|--------------|
| Ripple Carry              | 396.9670           | 11.45        |
| Carry Increment           | 473.5241           | 8.92         |
| Carry Look Ahead          | 397.6269           | 6.45         |
| Carry Select              | 759.3769           | 5.38         |
| Carry Skip                | 529.7861           | 12.97        |
| Brent Kung                | 485.5570           | 3.02         |
| Ladner Fisher             | 508.9519           | 2.65         |

In addition to the investigations described in literature (Pudi and Sridharan, 2012; Ismail et al, 2014 and Rooj et al, 2018), straightforward RCA results seem to be agreed with the findings in Figure 4 indicated by longest delay but small in silicon area. By considering both speed and area design criteria, attempt to have a high scalable processor with faster speed may arise problems of interface bottlenecks, and this case totally happened if the design area is larger than what FPGA board offering or a compact single chip fabrication (Savich et al, 2007). With negotiable area of 435.5755 \textmu m^2 in average illustrates by CIA and CLA, in addition with short delay indicates by CLA which is 78% faster than RCA has attracted a lot of attention for delay/area frontier of conventional adder.

Design philosophy for parallel prefix adder i.e. BK and LF for \( n \)-bit conventional adder in term of area is unattractive, since grey and black cells in speed path need to be considered. Although the structure of BK is considerably immune to the timing variations as well as able to perform in short delay with \( O(\log_2 n) \), this adder still suffer from large silicon area, \( 2n - (\log_2 n - 2) \) due to the tree structure occupied with CLA (Al-Haija et al., 2019). On the other hand, due to the carry propagation delay cause by RCA, a proposed LF which sharing similar procedural stages as BK but difference in the interconnection networks between group generation and propagation leads to short delay. The outcome shows that LF has the higher area, \( (n / 4) \log_2 n + (3n - 2) \) in the case of the parallel prefix adder, while for the conventional adder case, the highest area is for the CSLA.
5. FXP Multipliers Performance Comparison

Designing a simple, but yet fast computation for binary multiplier right now is one of the aspects in the design of LNS microprocessor, follows through the process steps by Ercegovac and Lang, 2004, is necessary to be fused with complex wiring and block elements. A steps includes 1) recording the digital multiplier based on n-bit system design; 2) computational of individual digit by the multiplicand, hence gives an outcomes in certain number of partial products; 3) multioperand addition technique is applied to sum up all the partial product array; and 4) final result is computes from carry-propagate adder of two operands. In case of proposed LNS system, two multiplier unit that work in parallel are required in the interpolation. To ensure the recording process of signed number have a minimal redundant digit set, a radix-\( r \) (\( r = 2^m \)) were employed by segregate the \( m \)-bits into group. On the other hand, the \( n \)-bits operand, with a total of \( \frac{n}{m} \) partial products are represented in two’s complement, and \( \frac{m-1}{m} \) for unsigned number (Antelo et al., 2016).

Hypothetically, high radix can reduce the number of partial products, but is less attractive in industrial microprocessor due to complex shift process, and additional hardware is needed for carry-propagate additions. Highly demand signed recording scheme is radix-4, which provide a benefit in fast latency, but still radix-8 (Muhammad et al., 2001 and Colon-Bonet and Winterrowd, 2008) and radix-16 (Riedlinger et al., 2012) are considerable due to lower power multipliers. These kind of concept, using add and shift, taking an example radix-4 \([-2, -1, 0, 1, 2]\), radix-8 \([-4, -3, \ldots, 0, \ldots, 3, 4]\) and radix-16 \([-8, -7, \ldots, 0, \ldots, 7, 8]\) can be summarize as an array multiplier. Besides, another alternative using the fusion between modified Booth encoding algorithm and the Wallace tree structure, called Booth-Wallace multiplier as reported by Qian et al., 2016 promised a fast latency as well as flexibility in area and power consumption when synthesized under 45nm CMOS technology. This is because the simplification of partial product generation and compression, by reducing the number of rows make the structure relative mature.

In finding the best multiplier, two multiplier, array multiplier and Booth-Wallace multiplier are compared with recent multiplier derived from Vedic mathematic (Sudeep et al., 2014). The Vedic multiplier algorithm discuss in this research is based on Urdhva–Tiryagbhhyam sutra, by means the multiplication is conducted vertically and crosswise. The hardware itself is not as an expensive as and Booth-Wallace multiplier, indicates by four blocks of multiplier adjoint with two blocks of carry-save adder as in Figure 5. Based on comparative studies conducted by Pohokar et al., 2015, the latency of
the system using Vedic multiplier as compared to array multiplier is proved to give a reduction of 53%.

![Figure 5. 16 × 16 bit Vedic multiplier.](image)

In order to determine the functionality and reliability of results, six multipliers include unsigned, array, Booth-Wallace, modified Booth Wallace, modified Baugh Wooley and Vedic designed are synthesis using EDA tools, whereby similar testbench are scripted. The two main parameters of multipliers were greatly optimized as reported in Table 2, indicates the benefit of Vedic multiplier over the conventional array multiplier in term of speed with 44% of penalty in area. Besides, outcome from Booth-Wallace multiplier in contrast with Vedic multiplier is strongly agreed with Pushpangadan et al., 2009, with the impact of additional hardware required by Booth-Wallace is quite troublesome, taking an advantage of Booth-Wallace multiplier for fast in speed.

![Table 2. Multiplier utilization summary.](image)

| Logical Utilization         | Area ($\mu$m$^2$) | Latency (ns) |
|-----------------------------|-------------------|--------------|
| Unsigned                    | 2001.0193         | 28.39        |
| Array                       | 2608.7074         | 22.40        |
| Booth Wallace               | 4648.0219         | 15.93        |
| Modified Booth Wallace      | 3182.8302         | 24.10        |
| **Modified Baugh Wooley**   | **2895.6683**     | **16.12**    |
| Vedic Multiplier            | 4636.2191         | 17.93        |

To obtain a better grasp of relative performances of the multiplier, graph in Figure 6 show the relation between the area and latency. Based on the above discussed outcomes, Booth Wallace is nominated as the fastest multiplier while unsigned performed in term of area. By considering the trade off between speed and area, finding shows modified Baugh Wooley is the best among others. Note that these results are in term of technology file are similar, by means 0.18 $\mu$m with 50 MHz are standardized.
6. Conclusion

A comparative study for various adder and multiplier has been conducted in this paper. The selected LF adder and modified Baugh Wooley preserved benefit in fast in latency, at the same time offering reduced silicon area. Under CMOS 0.18 µm implementation, using LF, which is one of parallel prefix adder increase system speed by three times compared to conventional adder and surprisingly the area is much lower that CSA and CSLA. In addition, the selected multiplier is a runner-up in latency, but manage to maintain the small area. Hence, adder and multiplier that can provide optimal performance can truly benefit for a high-speed data paths as well as a compact area for a proposed new LNS microprocessor.

7. References

[1] Yao, J., Tjuatja, S. and Huang, H., 2015. Real-time vibratory strain sensing using passive wireless antenna sensor. IEEE Sensors Journal, 15(8), pp.4338-4345.
[2] Khirade, S.D. and Patil, A.B., 2015, February. Plant disease detection using image processing. In Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on (pp. 768-771). IEEE.
[3] Wang, L., Zhao, Y., Zhou, Y. and Hao, J., 2016. Calculation of flexible printed circuit boards (FPC) global and local defect detection based on computer vision. Circuit World, 42(2), pp.49-54.
[4] Rodríguez-Andina, J.J., Valdes-Pena, M.D. and Moure, M.J., 2015. Advanced features and industrial applications of FPGAs—A review. IEEE Transactions on Industrial Informatics, 11(4), pp.853-864.
[5] Coleman, J.N. and Ismail, R.C., 2016. LNS with co-transformation competes with floating-point. IEEE Transactions on Computers, 65(1), pp.136-146.
[6] Fang, C.F., Rutenbar, R.A. and Chen, T., 2003, November. Fast, accurate static analysis for fixed-point finite-precision effects in DSP designs. In Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design (p. 275). IEEE Computer Society.
[7] Dido, J., Geraudie, N., Loiseau, L., Payeur, O., Savaria, Y. and Poirier, D., 2002, February. A flexible floating-point format for optimizing data-paths and operators in FPGA based DSPs. In Proceedings of the 2002 ACM/SIGDA tenth international symposium on Field-programmable gate arrays (pp. 50-55). ACM.
[8] Coleman, J.N., Chester, E.I., Softley, C.I. and Kadlec, J., 2000. Arithmetic on the European logarithmic microprocessor. IEEE Transactions on Computers, 49(7), pp.702-715.
[33] Shilpa, K.C., Shwetha, M., Geetha, B.C., Lohitha, D.M. and Pramod, N.V., 2018, April. Performance analysis of parallel prefix adder for datapath VLSI design. In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT) (pp. 1552-1555). IEEE.