



Implementation Evaluation of Fixed-Point Multipliers for Complex Numbers

Downloaded from: <https://research.chalmers.se>, 2025-03-12 11:34 UTC

Citation for the original published paper (version of record):

Larsson-Edefors, P., Börjeson, E. (2025). Implementation Evaluation of Fixed-Point Multipliers for Complex Numbers. IEEE International Symposium on Computer Arithmetic

N.B. When citing this work, cite the original published paper.

© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

(article starts on next page)

Implementation Evaluation of Fixed-Point Multipliers for Complex Numbers

Per Larsson-Edefors
Chalmers University of Technology
Gothenburg, Sweden
 perla@chalmers.se

Erik Börjeson
Chalmers University of Technology
Gothenburg, Sweden
 erikbor@chalmers.se

Abstract—Complex multipliers are common in signal processing and scientific applications. A direct implementation of complex multiplication involves four real-valued multiplications and two real-valued additions. But it is well known that there are alternatives to the direct implementation in which some computations are shared. The rationale for these alternatives would be that fewer multiplications are used, potentially reducing hardware resource usage. We consider three distinctly different complex multiplication schemes, implement them as fixed-point complex multipliers using a 7-nm predictive technology and quantitatively evaluate their area usage and energy efficiency.

I. INTRODUCTION

Multiplication of complex numbers is at the heart of many applications in science and engineering. Fixed-point representation is favored over floating point in applications which face strict requirements on data throughput and latency, for example, in signal processing and communication systems. Fixed-point complex multipliers can have a significant impact on overall system properties. Take for example a high-throughput fiber-optic coherent receiver: The power dissipation of digital signal processing (DSP) circuits is dominated by equalization [1]. Moreover, in advanced MIMO equalizers complex multipliers occupy 70% of the total area [2].

To provide guidelines on efficient implementation of fixed-point complex multipliers, we review three complex multiplication schemes. Using the predictive ASAP7 7-nm cell library, we quantitatively evaluate the complex multipliers' area usage and energy efficiency in different design scenarios.

II. COMPLEX MULTIPLICATION SCHEMES

A common consideration in DSP algorithm design is how many real-valued multiplications and additions are required for a complex multiplication. This atomic, canonical view of hardware resources serves algorithm analysis well as long as algorithms are implemented in software. But as stricter hardware requirements are imposed, circuit customization is necessary to deliver performance in a power-efficient manner. For custom implementations, it is in general not meaningful to consider atomic operations, since operations and their logic are merged during logic synthesis to reduce area.

There exist a number of ways to construct one complex multiplication from real multiplications and additions, but as shown in [3], it is not possible to accomplish this with less than three real multiplications. In the following, we will review three distinctly different schemes for complex multiplication:

a) *Direct form*: Taking Z_r and Z_i to be the real- and imaginary-valued product, respectively, of inputs A and B , the direct form requires four multiplications and two additions. (We assume 2's complement signed numbers, in which sign change is trivial, allowing us to use addition for subtraction.)

$$\begin{aligned} Z_r &= A_r B_r - A_i B_i \\ Z_i &= A_r B_i + A_i B_r \end{aligned}$$

b) *Wenzler*: An exhaustive list of alternate computing structures for complex multiplication using only three real-valued multiplications is given in [4]. The three real multiplications can be arranged so that both Z_r and Z_i are of similar complexity. Our early exploration shows that the structure of this type that performs best is Structure XII in [4]:

$$\begin{aligned} Z_r &= A_r(B_r + B_i) - B_i(A_r + A_i) \\ Z_i &= A_r(B_r + B_i) + B_r(A_i - A_r) \end{aligned}$$

Here, we can isolate t as a shared computation:

$$\begin{aligned} t &= A_r(B_r + B_i) \\ Z_r &= t - B_i(A_r + A_i) \\ Z_i &= t + B_r(A_i - A_r) \end{aligned}$$

c) *Golub*: The three real multiplications can instead be arranged in a manner that better suits recursive computation. Our early exploration on alternate structures indicates that the recursive structure that performs best is Structure IV [4], which is also known as Golub's scheme [5]:

$$\begin{aligned} Z_r &= (A_r + A_i)(B_r - B_i) + A_r B_i - A_i B_r \\ Z_i &= A_r B_i + A_i B_r \end{aligned}$$

Here we can isolate two shared computations, t_1 and t_2 :

$$\begin{aligned} t_1 &= A_r B_i \\ t_2 &= A_i B_r \\ Z_r &= (A_r + A_i)(B_r - B_i) + t_1 - t_2 \\ Z_i &= t_1 + t_2 \end{aligned}$$

Both the Wenzler and Golub schemes require five real additions, assuming additions with two inputs. The order in which the operations are arranged in the three schemes will impact the noise caused by the fixed-point signal representation. The work in [4] includes an analysis of noise properties, however, it does not consider the circuit implementation aspects (timing, area, and energy dissipation) that we address here.

III. METHOD

The three complex multiplication schemes we selected are implemented and evaluated for different wordlengths: We assume that the inputs A_r , A_i , B_r , and B_i all have the same n -bit data. Additionally, we assume the outputs Z_r and Z_i are full-precision $(2n + 1)$ -bit words.

We use Cadence Genus [6] to synthesize each complex multiplication scheme. The logic synthesis tool starts from its own predefined implementation templates for datapath operators and as the tool employs different logic transformations, the actual gate-level netlist implementation becomes the result of proprietary tool algorithms, many of which are heuristics. Typically, synthesis minimizes circuit area under a timing constraint, which defines the maximal delay a circuit can have. To capture the impact of performance requirements on implementation, the timing constraint is an important design parameter. Generally, as we make the timing constraint stricter, the total circuit area grows, since logic cells with higher drive strength are required. But higher drive strength implies higher cell capacitance, which means the circuit area begins to grow exponentially as we approach the lowest possible timing constraint, which we call the *critical timing*. Our results only consider netlists whose delay satisfies the timing constraint.

To ensure the functional integrity of all results, every gate netlist is subjected to logic simulation in Cadence Xcelium [7]. By generating different input vector sets as well as reference outputs for Z_r and Z_i , we not only drive the logic verification, but we also set up various data switching scenarios for the ensuing netlist power simulations. We perform power analysis in Genus with backannotated data from Xcelium, using a clock rate of 500 MHz and a supply voltage of 0.7 V. In the default case, the testbenches are driven by a vector set for each input A and B which constitutes 10,000 randomly generated vectors. These sets are generated for each wordlength. Since vectors are random, our energy-per-operation values tend to overestimate those obtained in practical scenarios where the switching activity is significantly lower. However, our choice of vectors allows us to make comparisons between different implementations and others to replicate our setup.

We use regular-threshold-voltage cells of the open-source ASAP7 library [8], which was developed to represent a predictive 7-nm FinFET technology. We have also run evaluations with commercial cell libraries to confirm the trends.

IV. RESULTS

Complex multiplication is used for a wide array of applications, which means the requirement on accuracy and dynamic range will vary. For fixed-point DSP, these aspects are strongly associated with the resolution of analog-digital and digital-analog interfaces. On the one extreme, we find high-throughput communication systems with simple modulation formats, for which 10 bits or slightly less may suffice. On the other extreme, we find resolution-sensitive audio applications for which 24 bits are required.

A. Area

We plot in Fig. 1 the area ratio of 24-bit complex multipliers as function of timing constraint. The direct complex multiplier (CM) is faster than the other schemes as its critical timing, i.e. shortest delay possible, is the lowest. This is to be expected, since in this scheme there are no internal computations which are shared. In the other two schemes, shared computations tend to increase logic paths and fanouts, which have a detrimental effect on timing.

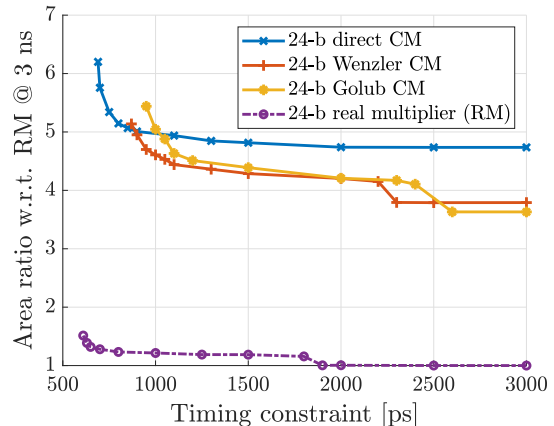


Fig. 1: Multiplier area ratio as function of timing constraint. This ratio is obtained by normalizing all area values to the area of a 24-bit real multiplier (RM) synthesized at 3 ns (158 um^2).

Considering their critical timing, the Wenzler and the Golub CM is 26% and 38% slower, respectively, than the direct CM implementation at 24 bits. For relaxed timing constraints to the right in the graph, the synthesis tool at certain timing points can opportunistically extend logic paths to further reduce area. The direct CM implementation has the largest area when timing is relaxed, which can be expected from a scheme which uses four real-valued multipliers.

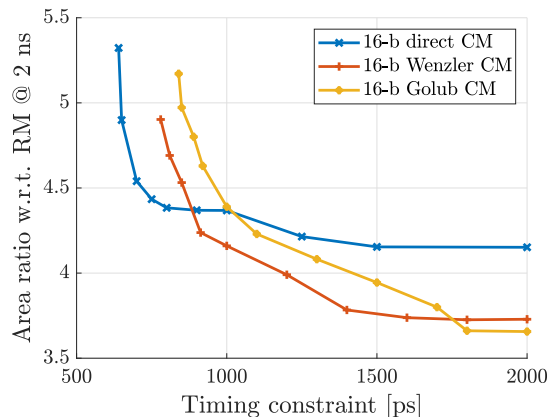


Fig. 2: Multiplier area ratio as function of timing constraint. A 16-bit real multiplier (RM) synthesized at 2 ns is used for normalization.

The area ratios for the 16- and 10-bit CMs are shown in Fig. 2 and Fig. 3. To enhance readability, we remove the data for the real multiplier from the graphs, however, all area values are still normalized to the real multiplier at the most relaxed timing constraint in the respective graph.

As we can see, the direct CM maintains its speed advantage also for shorter wordlengths. However, as we reduce n down to 10 bits, Wenzler and in particular Golub CMs lose their area advantage at relaxed timing constraints. This is to be expected as multiplier implementation area grows quadratically with input wordlength, while the area of adders grows linearly.

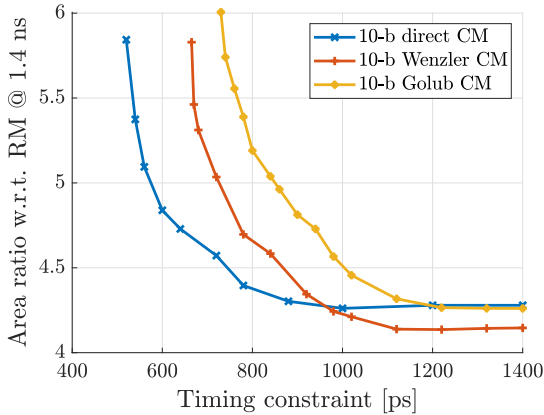


Fig. 3: Multiplier area ratio as function of timing constraint. A 10-bit real multiplier (RM) synthesized at 1.4 ns is used for normalization.

B. Energy per Operation

There is often a strong correlation between area and power dissipation, but arithmetic circuits represent an exception in that they are prone to glitching. In the following, we evaluate the energy per operation: We first synthesize the three CMs at a timing constraint which is 10% higher than the critical timing shown in Figs 1-3. This represents a design scenario of high throughput at a reasonable area cost. We show in Fig. 4 the energy per complex multiplication for the different CMs.

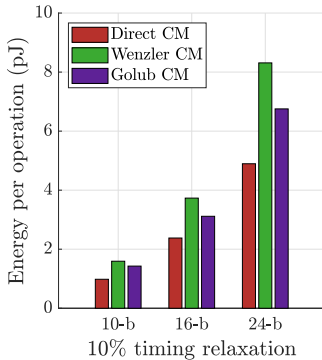


Fig. 4: Energy per operation at 10% timing relaxation.

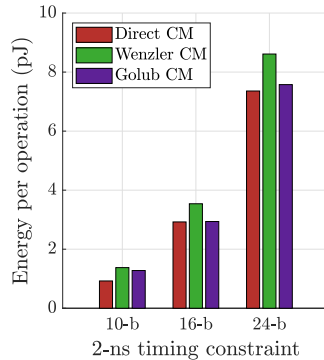


Fig. 5: Energy per operation at a fixed timing constraint of 2 ns.

At the strict timing constraints used, for larger wordlengths, the direct CM requires larger area than the other two schemes. But it is clear that this does not translate to higher energy per operation. In fact, as shown in Fig. 4, the direct CM is the most energy-efficient implementation for all wordlengths.

If we relax the timing constraint to 2 ns and resynthesize all CMs, we obtain Fig. 5. Since the area usage decreases as the timing constraint is relaxed, we would expect also the energy per operation to decrease. This does not happen for all the CM configurations that we evaluate here. For example, the 16-bit direct CM shows an increase in energy per operation.

As shown in Fig. 6, the explanation lies in the increasing glitching power dissipation that results from increasing delay differences between logic paths. Here, E_{zd} represents the energy per operation when we use zero-delay gate timing models, which effectively suppress glitching during simulation. Thus, E_{zd} is E_{tot} minus the energy used by glitches. As shown in Fig. 6, when strict timing is enforced for the direct CM, the symmetrical arrangement of operations is leveraged during synthesis into a delay-balanced netlist, leading to fewer glitches [9]. As the timing requirement is relaxed, the synthesis tool optimizes area at the cost of higher glitching.

In terms of area, the Wenzler CM is a better choice than the Golub CM for throughput-oriented applications. But the energy per operation is high for Wenzler CM and as we can see here, this is caused by glitching power.

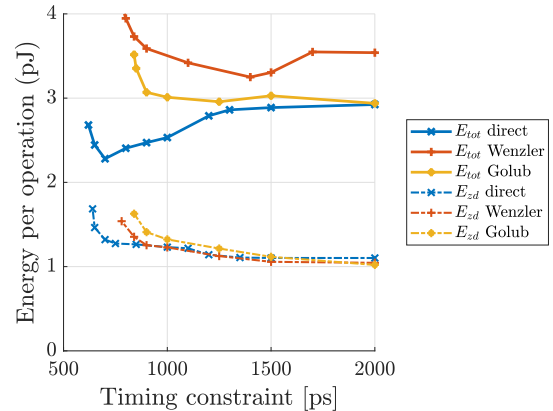


Fig. 6: 16-bit CM energy per operation as function of timing constraint. Both total energy E_{tot} and energy retrieved using zero-delay timing models E_{zd} are shown.

C. Pin Assignments

Multiplication and addition are commutative operations, but the order of inputs can have a significant impact on power dissipation. Above a certain wordlength, synthesis tools use Booth techniques to realize integer multipliers. This creates an asymmetric multiplier circuit, where inputs A and B do not face the same downstream logic: One of the inputs goes to the Booth encoder while the other is almost directly connected to the carry-save circuit, in which the two input operands reconverge. As shown for a direct CM, the order in which signals are assigned to input pins impacts power dissipation [10]. Interestingly, in recent EDA tools, support for low-power pin assignments is making inroads [11].

Since a complex multiplier is made up of several integer multipliers and adders, the arrangement of operations inside the CM scheme impacts the efficacy of pin assignments. To find out how the three CM schemes perform in terms of pin assignments, we run power analyses on the netlist implementations used for Fig. 4 and Fig. 5. In addition to the default *random* vectors described in Sec. III, we now generate 10,000 *reduced-activity* input vectors in which we decrease switching activity by 50% and dynamic data range by 1 bit. These vectors represent a common case of DSP filters in which some weights/coefficients have lower updating frequency and

dynamic range. We first assign random vectors to A and reduced-activity vectors to B and perform power analysis. Next we swap the vectors and perform another analysis. One assignment leads to lower power dissipation and this is marked optimal; the other assignment we call non optimal.

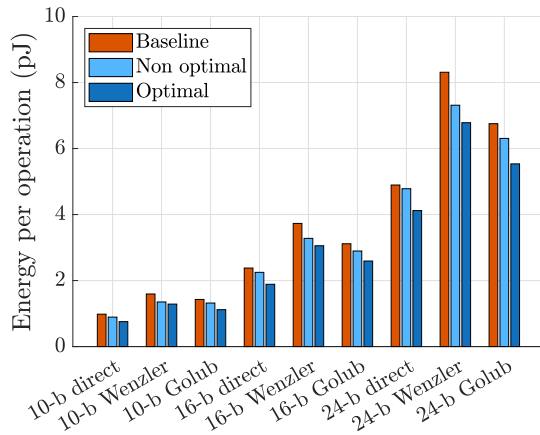


Fig. 7: Energy per operation for different pin assignments at 10% timing relaxation.

We show in Fig. 7 the impact of pin assignment on energy per operation, when a 10% timing relaxation is used. The baseline bar represents the case when both A and B receive random vectors. Because 1) there is a significant difference between the optimal and non-optimal assignments and 2) the non-optimal values are close to the baseline, the results suggest that the direct CM benefits substantially from pin assignment.

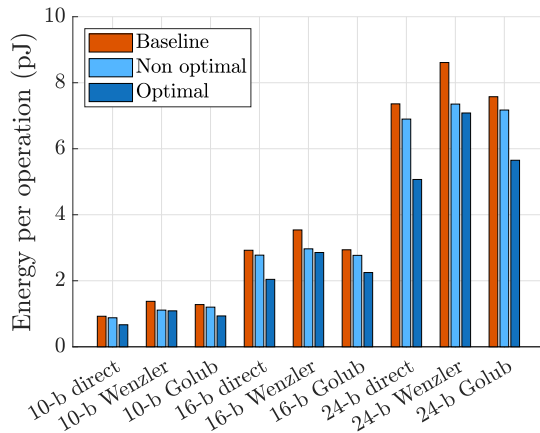


Fig. 8: Energy per operation for different pin assignments at a fixed timing constraint of 2 ns.

The data in Fig. 8 are based on a case where we during synthesis extend the timing constraint to 2 ns for all CMs. As shown in previous graphs, glitching power becomes significant as we relax the timing constraint. Fig. 8 demonstrates that pin assignment becomes a very powerful method to reduce CM power dissipation when glitches are present. Especially the direct CM benefits from an optimal assignment for all wordlengths considered. When we relax the timing even further in additional analysis runs not shown here, the impact of pin assignment increases for the two other CMs which are intrinsically slower than the direct CM.

D. Truncated Outputs

To avoid wordlength growth in multiplier-intensive designs, truncated complex products are often used in custom DSP circuits. If we retrieve the n most significant output bits, but discard the rest, the critical logic path will be similar to that of the full-precision multiplier. Because of this, area and energy trends of the three CM schemes should be similar for truncated and full-precision multipliers. We have performed additional synthesis runs to validate that this is the case. Since the least significant bits of the complex products are discarded, many logic gates can be removed during synthesis, leading to lower absolute area and energy values for truncated multipliers.

ACKNOWLEDGEMENT

We thank the Swedish Foundation for Strategic Research (SSF) for funding through the classIC and HotOptics projects.

V. CONCLUSION

There exist three fundamentally different schemes to perform complex multiplication based on real-valued multipliers and adders. To the best of our knowledge, no study in the open literature has been devoted to a thorough comparison of these three schemes and their resulting fixed-point implementations.

Using synthesis on a cell library of a 7-nm predictive technology, we have evaluated the schemes. Anecdotal evidence suggests that the direct complex multiplier (CM) is used more often than the other two CMs. We found that the direct CM has an area disadvantage, with the very important exception of design situations with tight timing constraints. The direct CM is intrinsically faster than the other two CMs, which means it is competitive in most practical situations. The direct CM is also more energy efficient than the other CMs, but as we have shown, glitching power starts to dominate total power for larger wordlengths and relaxed timing constraints. We showed that pin assignment can reduce power considerably, especially when applied in direct CMs implemented with relaxed timing.

REFERENCES

- [1] C. Fougstedt *et al.*, "ASIC design exploration for DSP and FEC of 400-Gbit/s coherent data-center interconnect receivers," in *Optical Fiber Communication Conf.*, Mar. 2020, p. Th2A.38.
- [2] E. Börjesson *et al.*, "Circuit implementation of pilot-based dynamic MIMO equalization for coupled-core fibers," in *Optical Fiber Communication Conf.*, Mar. 2024, p. W1E.4.
- [3] S. Winograd, "On multiplication of 2×2 matrices," *Linear Algebra and its Applications*, vol. 4, no. 4, pp. 381–388, 1971.
- [4] A. Wenzler and E. Luder, "New structures for complex multipliers and their noise analysis," in *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, vol. 2, 1995, pp. 1432–1435.
- [5] E. E. Swartzlander and H. H. Saleh, "Floating-point implementation of complex multiplication," in *Asilomar Conf. on Signals, Systems and Computers*, 2009, pp. 926–929.
- [6] Cadence® Genus®, v. 18.14, Cadence Design Systems, Inc., 2019.
- [7] Cadence® Xcelium®, v. 22.09, Cadence Design Systems, Inc., 2023.
- [8] V. Vashishtha, M. Vangala, and L. T. Clark, "ASAP7 predictive design kit development and cell design technology co-optimization," in *IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 2017, pp. 992–998.
- [9] R. Zimmermann, "Datapath synthesis for standard-cell design," in *IEEE Int. Symp. on Computer Arithmetic*, 2009, pp. 207–211.
- [10] P. Larsson-Edefors and E. Börjesson, "Low-power complex multiplier pin assignment based on spatial and temporal signal properties," in *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, 2025.
- [11] Genus® Datapath Synthesis Guide, Cadence Design Systems, Inc., 2024.