Energy-Efficient Inexact Speculative Adder with High Performance and Accuracy Control

Vincent Camus, Jeremy Schlachter, Christian Enz
Integrated Circuits Laboratory (ICLAB)
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
vincent.camus@epfl.ch

***Abstract*—Inexact and approximate circuit design is a promising approach to improve performance and energy efficiency in technology-scaled and low-power digital systems. Such strategy is suitable for error-tolerant applications involving perceptive or statistical outputs. This paper presents a novel architecture of an Inexact Speculative Adder with optimized hardware efficiency and advanced compensation technique with either error correction or error reduction. This general topology of speculative adders improves performance and enables precise accuracy control. A brief design methodology and comparative study of this speculative adder are also presented herein, demonstrating power savings up to 26 % and energy-delay-area reductions up to 60 % at equivalent accuracy compared to the state-of-the-art.**

I. Introduction

As mobile devices become ubiquitous, the power efficiency of digital systems has become a primary concern. Unfortunately, achieving low-power and Process-Voltage-Temperature (PVT) robustness requires complex and conflicting design constraints and safety margins. Typically, integrated circuits are designed to always ensure accurate operation. In order to avoid faulty results, they finish most computations earlier than the worst-case permitted delay or with more accuracy than needed for normal operation. This results in an inefficient use of resources and leads to "over-engineered" circuits.

Inexact and approximate circuit design [1] is a radical approach to trade this counterproductive quest for perfection for substantial gains in power, speed, area and yield. The primary challenge, however, is to determine *where* and *how* to let an error or an approximation occur in the circuits without compromising the functionality or the user experience. With ever-increasing amount of data being processed, a wide variety of applications could tolerate inaccuracies. For example, in multimedia processing, a small proportion of errors is not perceptible to humans, and in highly computational algorithms such as data mining, search or recognition, the outcome is not required to be a single result but an adequate match. A promising approach to design inexact circuits is to use *speculation* to trade circuit accuracy for better power and speed. Taking advantage of such circuits would help to realize extremely energy-efficient and high-performance DSPs and hardware accelerators at lower integration cost and with higher speed, data rate or duty-cycling.

The main contribution of this paper is to introduce a novel speculative adder: the Inexact Speculative Adder (ISA). The ISA improves performance, energy efficiency and error management through an optimized speculative path and with a versatile dual-direction error compensation technique. A brief design methodology is presented along with results and a comparative analysis of adder architectures.

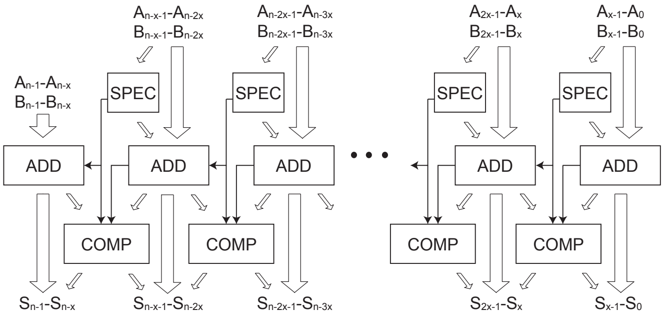

Fig. 1. General block diagram of the Inexact Speculative Adder.

II. Inexact Speculative Adder

A. Related Work

Speculative adders [1] exploit the fact that the typical carry propagation chain of an addition does not span the whole length of the adder, making it is possible to estimate an intermediate carry using a limited number of previous stages. Thus, the carry propagation chain, which is the critical path of the adder, can be split in two or more shorter paths, relaxing constraints over the entire design, reducing spurious glitching power, and improving the Energy-Delay-Area Product (EDAP) beyond the theoretical bounds of exact adders.

A number of speculative adders have been proposed in literature with different approaches in order to reduce the error frequency or magnitude. The ETAII adder [2] consists of regular sub-adder blocks with input carries speculated from Carry Look Ahead (CLA) blocks of the same length. In the ETAIIM version, several of the most significant CLA blocks are chained in order to increase accuracy. The ETBA adder [3], direct descendent of the ETAIIM, adds variable speculation signs and sub-adder sum balancing multiplexer blocks to mitigate relative errors. N. Zhu *et al.* [4] and Y. Kim *et al.* [5] have recently demonstrated adders with improved accuracy by considering two prior carry speculation blocks instead of one, coupled with a carry select (ETAIV) [4] or a carry skip [5] technique, with the latter also using sum balancing over several sub-adder blocks.

One room for improvement in the existing speculative adders is that the circuit hardware is not utilized efficiently enough. Indeed, the overhead due to carry speculation is huge and lies entirely in the critical path. On the other hand, the sum balancing blocks, containing parallel multiplexers, take also a part in the critical path, require large fan-outs, but are very weakly exploited due to the low probability of incorrect speculations. Thanks to its new speculative path and compensation scheme, the ISA architecture proposed herein greatly improves hardware efficiency upon the state-of-the-art and introduces a new way to control errors.

This work was supported by the NanoTera "IcySoC" project.

B. Proposed Architecture

The structural diagram of the ISA adder is depicted in Fig. 1. The ISA splits the carry propagation chain in multiple paths executed concurrently. Each path consists of a carry speculator block (SPEC), a sub-adder block (ADD) and an error compensation block (COMP). For each SPEC-ADD-COMP path, the SPEC block generates a partial carry signal that the sub-adder ADD uses as a carry-in to generate a local sum. The COMP block compensates faulty sums either by correcting the local sum or by reducing error magnitude as in [3]. The first speculative path, operating on the least significant bits (LSBs) of the adder, does not have SPEC nor COMP blocks since it uses directly the ISA carry-in.

The carry speculation of the SPEC block is generated from a few bits in a carry look-ahead approach sourced by either a static or a dynamic input. The latter, used in ETBA, can evenly distribute the errors at the cost of hardware and delay overhead. When a propagate chain covers the full SPEC block, the exact carry cannot be speculated from the partial product and the output carry is guessed at the input carry value. Long propagate sequences are uncommon in the case of uniform input distribution [2], thus the probability of speculation fault decreases by increasing the size of the SPEC block.

The sub-adder ADD calculates block sums locally from the speculated carry of the SPEC block. Without compensation, an internal overflow caused by an inconsistent carry could lead to a massive error. Therefore, the COMP block detects those speculation faults by comparing the carry generated from the SPEC with the carry-out coming from the prior ADD block. The COMP block is implemented in between two local ADD blocks. In case of faulty speculation, it attempts to fully correct a group of LSBs of the local sum. The full correction consists in incrementing or decrementing the group of LSBs, and is only possible when it does not lead to another internal overflow. If correction is not possible, the COMP block can flip a group of the most significant bits (MSBs) of the previous sub-adder sum to minimize the error magnitude.

The achieved addition arithmetic, illustrated in Fig. 2, is a 5-step process:

1. A carry-in is speculated from a very short carry propagation chain for each sub-adder block.
2. The sub-adder calculates the local sum based on this speculated carry-in.
3. Comparison of the speculated carry-in and the prior sub-adder carry-out allows detection of faulty speculation.
4. In case of wrong speculation, correction of the local sum is attempted.
5. If correction is not possible, error magnitude is reduced by balancing the preceding sum bits.

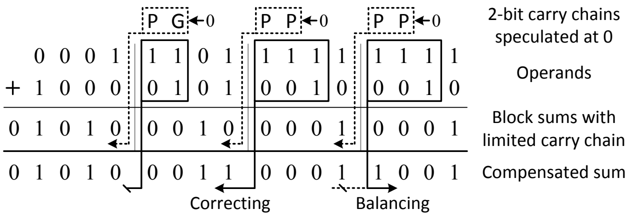

Fig. 2. Example of ISA addition arithmetic with 2-bit speculation, 1-bit correction and 1-bit error reduction. Faults only occur in the two right-hand paths. The 1st LSB of the central path can be corrected. The 1st LSB of the right path cannot be corrected, so the 1st MSB of the preceding sum is flipped.

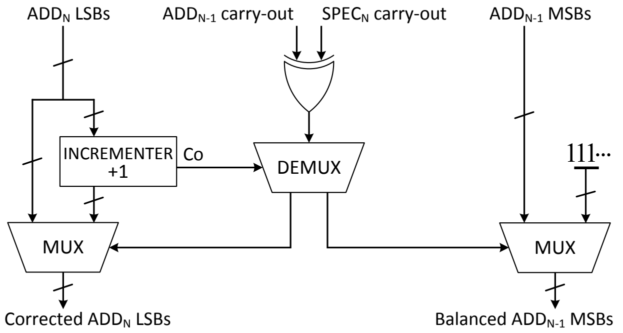

Fig. 3. COMP block implementation in the example of a fixed direction of errors leading to positive correction and balancing.

C. COMP Block Implementation

Fig. 3 presents a possible implementation of the COMP block. The COMP block detects inconsistencies between speculated carry and expected carry from the previous sub-adder with an XOR gate. This creates an error flag that triggers the activation of one of the two compensation techniques, namely error correction and error reduction.

The potential error always remains of the same nature as the input carry of the SPEC block. For example, speculations at 0 instead of 1 can only induce *too low* sums compared to expected ones, while speculations at 1 instead of 0 provoke *too high* sums. Therefore, the sign of the corrective compensation is always recognized. The error correction part of the COMP intercepts a group of LSBs of the local sum, at the position of the error, and performs an unsigned increment or decrement in the direction of this potential error (i.e. *too high* error is solved by a -1 and *too low* by a $+1$). The intercepted bus has a fixed number of bits, thus this operation is only possible if it does not cause an overflow. For instance, if the COMP operates error correction on 3 bits, incrementing 111_2 and decrementing 000_2 is irrelevant and leads to an overflow. The detection of this overflow allows to demultiplex the choice of compensation technique to apply.

When a speculation fault cannot be fixed by the error correction, the COMP balances in the opposite direction of the error a group of MSBs of the preceding sub-adder. This technique allows to attenuate an error at a given bit position by manipulating bits that are less significant with respect to this bit.

Assuming a small correction bus, the corrective increment or decrement operation allows to detect potential overflow and switch to the right compensation technique before the ADD block finish computing. Thus, a significant feature of this adder is that neither the pre-computing of error correction nor the compensation choice lie in the critical path of the ISA adder. The multiplexers are the only components of the COMP block lying in the critical path.

D. Analysis of Error Compensation

The objective of this section is to detail the conditions in which an error can be corrected or reduced by the COMP block.

Let a carry error C_{err} occur at the i^{th} bit of an adder. S_i, C_i and P_i denote respectively the sum, carry-in and propagate signals of the i^{th} stage addition. Hence, the sum is defined by:

$$S_i \triangleq P_i \oplus C_i = P_i \oplus C_{err}. \quad (1)$$

At the i^{th} stage, the condition under which correction is not possible is that S_i is already opposed to the error and a bit-flip would not compensate the error, meaning that:

$$S_i = \overline{C_{err}} \iff P_i = 1. \tag{2}$$

This results in propagating C_{err} to the $(i{+}1)^{\text{th}}$ stage where the same formulae apply again. As a result, the correction is infeasible if and only if all the bits of the COMP's correction bus are in propagation mode.

Even though similar to [3] and [5], the effect of the balancing error reduction is more complex in the present work due to the independent sizing between SPEC and COMP blocks. For instance, let an adder leave a non-correctable carry error at the i^{th} bit of the sum, speculated from a s-bit SPEC and followed by a r-bit error reduction scheme. Occurrence of the error presumes that the carry propagation chain prior to the error is longer than s bits, i.e. stages $i{-}1$ to $i{-}s$ are all in propagation mode. Those stages are calculated in the previous sub-adder carry chain, which is non-erroneous (or whose sum is compensated in a prior block). Following (1), this results in:

$$P_{i-k} = 1 \iff S_{i-k} = \overline{C_{i-k}} = C_{err} \text{ for } k \leq s. \tag{3}$$

This means that the sum bits $i{-}1$ to $i{-}s$ tend to follow the inverse of the real carry, that being the faulty carry C_{err}. Thus, they can always be flipped to compensate the error. However, (3) is only valid for stages known in propagation states. If the COMP's error reduction is larger than the SPEC size ($r{>}s$), the extra sum bits ($i{-}s{-}1$ to $i{-}r$) do not satisfy this condition, thus, do not follow (3). Those bits may not be able to flip the error. This supplemental balancing does not further reduce the error magnitude in worst case operations, although it does impact on the overall ISA accuracy in typical situations with shorter carry propagate chains.

E. The ISA Strategy

The ISA is a general topology of speculative compensated addition wherein the state-of-the-art adders are particular cases. It allows notable improvements concurrently in the circuit performance and accuracy control. This section describes the architectural advantages of the ISA.

1) Optimized block sizing: The ISA architecture can reduce speculative hardware overhead and improve speed. The state-of-the-art speculative adders always use blocks of the same size in the speculative paths. In the ISA architecture, with the flexibility provided by the advanced compensation scheme, the SPEC block lengths can be traded for longer ADD blocks to fit the same delay requirement. It is then possible to use fewer speculative paths and limit the in-critical path speculation-compensation overhead to a few stages of each path.

2) Speculation and correction tradeoff: The COMP's correction technique resolves carry errors if its intercepted bits are not all propagate signals. In other words, the combined SPEC and COMP's correction techniques prevent errors on carry propagation chains of their cumulated bit lengths. Thus, an easy tradeoff can be realized in order to fit and optimize both delay and error rate.

3) Error reduction and failed correction: All the speculative adders in the literature so far employ only LSBs balancing technique as introduced in [3] to reduce speculative errors. To be efficient in worst-case operation, such technique requires a SPEC block of the same size lying in the critical path. A significant feature of the novel COMP block is that even when the correction technique cannot compensate for the error, the uncorrected bits appear in the same state as they would be for error reduction. Thus, as illustrated in Fig. 4, the COMP's correction has the same effect as a shifted balancing scheme and is efficient even in worst-case operation.

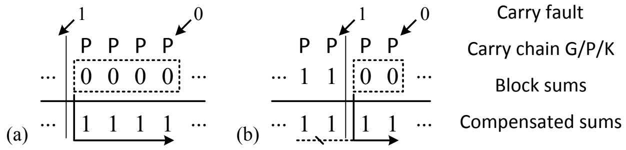

Fig. 4. Relative error equivalence between compensation with balancing only (a) and combined with correction (b) in the case of non-correctable error.

III. Design Methodology and Comparative Study

A. Methodology

The metrics used to characterize approximate adders in this work are based on the relative error (*RE*), defined as:

$$RE = \left| \frac{S - S_{cor}}{S_{cor}} \right| \tag{4}$$

where S and S_{cor} are respectively the approximate and correct sums of an addition. Interesting for many applications, particularly in media processing, the main metric considered is the RMS of the relative error (RE_{RMS}) that should be minimized. The worst-case accuracy (RE_{MAX}), i.e. the largest relative error of an adder, is also taken into account.

Most of the inexact adders are validated and characterized through the simulation of random sets of inputs since exhaustive simulation of large bit-width arithmetic blocks are too time-consuming. As a matter of fact, the metrics used are statistical estimators, dependent on the random input distribution and the chosen sample itself (occurrence of specific patterns initiates errors in specific adders). In this work, 32-bit adders are compared using two samples of five million unsigned random inputs. First, a formal uniform distribution is used to estimate RE_{RMS}. Then, a logarithmically uniform distribution exhibiting a very large dynamic range of scattered values is used to detect the worst-case accuracy RE_{MAX}. Delay, area and power of compared circuits are estimated using Synopsys Design Compiler with a 65 nm technology library.

While there are numerous approaches to implement inexactness and speculation within adder circuits, the choice of the right topology and design parameters remains cumbersome. A simple methodology is presented in Fig. 5 to find the optimal ISA strategies fitting timing and accuracy constraints with optimized performances. The adequate delay tradeoff is obtained by sizing SPEC and ADD blocks, principal slack elements of the ISA. The sizing of the COMP block can then be used to tune and fit the accuracy requirements.

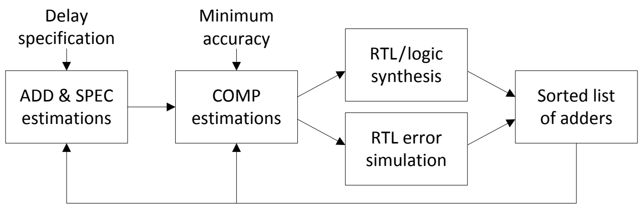

Fig. 5. CAD framework for ISA design.

B. Results and Comparison

Although it would be advantageous to optimize each speculative path according to the input patterns, the following comparison considers solely regular structures with identical speculative paths as in Fig. 1. Over 500 ISA adders with such implementations (2x16, 4x8, 8x4 and 16x2 concurrent paths) have been synthesized with diverse error characteristics.

Fig. 6 and 7 show the power and EDAP requirements and the error characteristics of selected implementations synthesized under 3.3 and 5 GHz constraints. Exact adder, ETAII and ETBA equivalents have been represented for the comparison. It is of interest to highlight the diversity of error engineering possibilities permitted by the ISA topology. It is also particularly clear that some ISA configurations achieve significant reduction of power consumption and EDAP at equivalent accuracy than the ETBA and ETAII. At identical RE_{RMS} and nearby RE_{MAX}, ISA circuits achieve between 11 % and 26 % of power reduction and between 24 % and 60 % of EDAP reduction upon ETBA.

The ISA architecture allows to tune and match any error specification. For instance, increasing SPEC and error reduction lengths minimizes progressively RE_{RMS} as depicted by the 8x4 adders (ISA and ETBA) in the middle of Fig. 7. However, the increasing in-critical path overhead leads to a breakdown in circuit efficiency to fit in the delay constraint. Shifting the speculative hardware towards error correction can limit the resource cost while continuing to increase accuracy (as with the 8x4 [s0 c4 r0] ISA on the left of Fig. 7). The ETAII, one of the most EDAP and energy efficient existing adders [5], is generally outperformed by ISA adders with short SPEC lengths.

On the other hand, reducing the in-critical path speculative hardware relaxes the design and the adder structure can collapse into less speculative paths, offering better accuracy with significant gains in power and EDAP. In Fig. 7, 4x8 ISA are sometimes twice as EDAP efficient for higher accuracy as the above-mentioned 8x4 ISA. Such structural contractions lead to large reductions of RE_{RMS} at relatively low costs as observable in Fig. 6.

Controlling RE_{MAX} requires an expensive combination of SPEC and COMP blocks. Relaxing a bit this requirement leads to significant savings in all aspects as demonstrated on the left-hand sides of Fig. 6 and 7. However, weakly compensated implementations such as ETAII and 4x8 [s0 c1 r4] ISA may be limited in practical use due to the occurrence of high relative errors. The variety of results and large variations of the chosen error characteristics point to the importance of also developing robust error metrics to match applications and circuits.

IV. Conclusion

This paper has proposed a general architecture of Inexact Speculative Adder (ISA) with high performance and adaptable accuracy. It features a novel error correction-reduction scheme that can improve overall and worst-case accuracy and shift speculative hardware overhead out of the critical path of the circuit. Thanks to a flexible sizing of speculative path elements and its new compensation scheme, the ISA architecture allows precise tuning of multiple error characteristics and greatly improves performance and efficiency upon the state-of-the-art. A simple methodology has been presented to design regular ISA adders with a delay-accuracy approach and has demonstrated power savings up to 26 % and EDAP reductions up to 60 % at equivalent accuracy compared to ETBA.

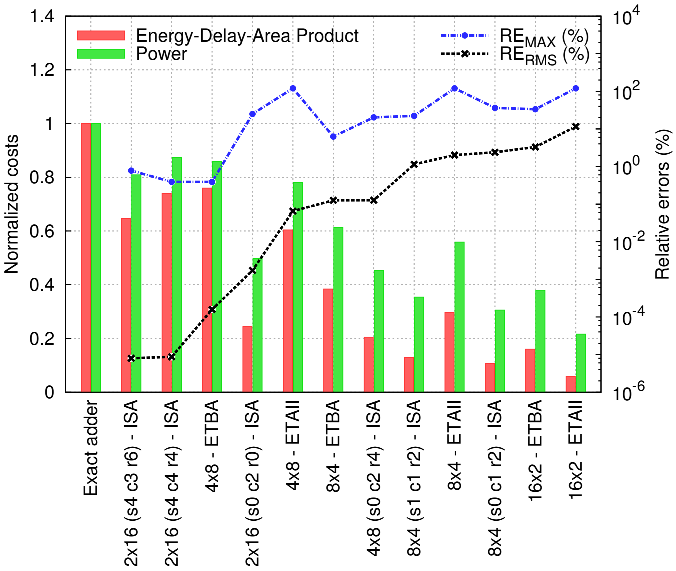

Fig. 6. Design costs and relative errors of selected ISA synthesized at 3.3 GHz. Specific ISA implementations are denoted by number and size of sub-adders, SPEC size s and COMP's error correction-reduction lengths c and r.

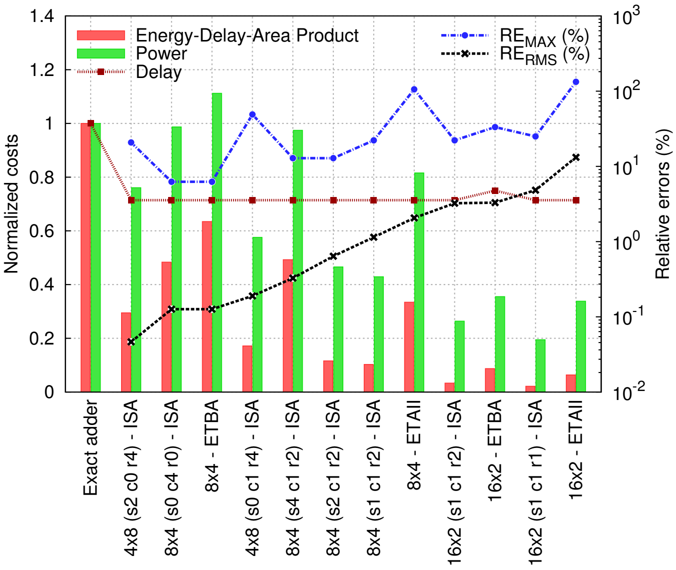

Fig. 7. Design costs and relative errors of selected ISA synthesized at 5 GHz. Delays are shown as the fastest exact adder cannot fit the 5 GHz constraint. 16x2 ETBA suffers from a drop of efficiency exactly at synthesized speed (q.v. section III-B), it is replaced with a slightly slower one for fair comparison.

References

[1] T. Liu and S.-L. Lu, "Performance Improvement with Circuit-level Speculation," in *Microarchitecture, 2000. MICRO-33. Proceedings. 33rd Annual IEEE/ACM International Symposium on*, 2000, pp. 348–355.

[2] N. Zhu, W.-L. Goh, and K.-S. Yeo, "An Enhanced Low-power High-speed Adder For Error-tolerant Application," in *Integrated Circuits (ISIC), Proc. of the 2009 12th International Symposium on*, Dec 2009, pp. 69–72.

[3] M. Weber, M. Putic, H. Zhang, J. Lach, and J. Huang, "Balancing Adder for Error Tolerant Applications," in *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, May 2013, pp. 3038–3041.

[4] N. Zhu, W.-L. Goh, G. Wang, and K.-S. Yeo, "Enhanced Low-power High-speed Adder for Error-tolerant Application," in *SoC Design Conference (ISOCC), 2010 International*, Nov 2010, pp. 323–327.

[5] Y. Kim, Y. Zhang, and P. Li, "An Energy Efficient Approximate Adder with Carry Skip for Error Resilient Neuromorphic VLSI Systems," in *Computer-Aided Design (ICCAD), 2013 IEEE/ACM International Conference on*, Nov 2013, pp. 130–137.