# Approximate 32-bit Floating-point Unit Design with 53 % Power-area Product Reduction

Vincent Camus†, Jeremy Schlachter†, Christian Enz
Integrated Circuits Laboratory (ICLAB)
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Neuchatel, Switzerland

Michael Gautschi, Frank K. Gurkaynak
Integrated Systems Laboratory (IIS)
Swiss Federal Institute of Technology in Zurich (ETHZ)
Zurich, Switzerland

vincent.camus@epfl.ch, jeremy.schlachter@epfl.ch
†These authors contributed equally to this work

***Abstract*—The floating-point unit is one of the most common building block in any computing system and is used for a huge number of applications. By combining two state-of-the-art techniques of imprecise hardware, namely Gate-Level Pruning and Inexact Speculative Adder, and by introducing a novel Inexact Speculative Multiplier architecture, three different approximate FPUs and one reference IEEE-754 compliant FPU have been integrated in a 65 nm CMOS process within a low-power multi-core processor. Silicon measurements show up to 27 % power, 36 % area and 53 % power-area product savings compared to the IEEE-754 single-precision FPU. Accuracy loss has been evaluated with a high-dynamic-range image tone-mapping algorithm, resulting in small but non-visible errors with image PSNR value of 90 dB.**

## I. Introduction

With the forecasted end of Moore's law and the increasing complexity to design and fabricate integrated circuits, power and reliability have become the main challenges to technology scaling. Power has definitely emerged as a critical issue due to the poor scaling of $V_{DD}$ and $V_{th}$, while transistor miniaturization reaching the nanoscopic scale has led to extreme Process-Voltage-Temperature (PVT) variations. Unfortunately, achieving low power and robustness against PVT variations requires complicated and conflicting design constraints. As a consequence, designers are being pushed to seek for new energy-efficient circuit design and computing techniques to meet the exploding demand of data processing from mobile devices and cloud services.

*Approximate computing* [1, 2] has emerged as a promising solution to sustain computing advancement and overcome the limitations in technology scaling. This approach explores a new trade-off between energy or circuit costs versus application accuracy. A myriad of applications could tolerate trading off a little bit of accuracy without compromising their functionality or user experience. In multimedia applications for instance, a small proportion of errors remains imperceptible to humans.

To design approximate systems, several approaches have been investigated at different hardware levels, such as voltage-frequency over-scaling [3] at physical level or significance-based memory protection [4] at algorithmic level. At circuit level, an interesting approach is to perform computations using approximate arithmetic operators, such as adders and multipliers, allowing a controlled and limited amount of errors against significant power saving or performance increase. This paper focuses on two of these techniques: Gate-level Pruning [5] and Inexact Speculative Adder [6], which have both demonstrated significant savings simultaneously in energy, delay and area at the cost of reasonable errors.

This work was supported by the NanoTera IcySoC project and the Swiss National Science Foundation project IneSoC under grant N° 200021-144418.

Although approximate techniques have been thoroughly investigated on fixed-point arithmetics, approximate floating-point circuits have so far not received much attention. Floating-Point Units (FPU) are key building blocks of Digital Signal Processing (DSP), graphics and high-performance workloads. They feature a mathematically superior alternative to fixed-point computing with higher computational ability and flexibility, but with a much higher complexity, power consumption and circuit cost. High-Dynamic Range (HDR) imaging is a rapidly growing area in computer graphics. Extensively using floating-point computations, tone-mapping is an increasingly used process of HDR image contrast optimization and correction. To that extent, it is an ideal target application to demonstrate the interest and *error tolerance* of approximate FPUs.

This paper aims at investigating the benefits of approximate circuits in the mantissa datapath of a FPU and at evaluating its use in a HDR image tone-mapping algorithm. The contributions of this work are as follows. Section II sums up the techniques used to design imprecise hardware and introduces a novel Inexact Speculative Multiplier (ISM) architecture. Section III presents the architecture and measurement results of three approximate FPUs and of the multi-core platform in which they have been implemented. Finally, section IV details the case study of the HDR image tone-mapping application implemented to evaluate and validate the use of the imprecise FPUs.

## II. Approximate Arithmetic Techniques

### A. Gate-Level Pruning and Inexact Speculative Adder

Two approximate circuit design techniques, namely Gate-Level Pruning [5] and Inexact Speculative Adder [6], as well as their combination [7], have been adapted and fitted in the FPU mantissa.

Gate-Level Pruning [5], shown in Fig. 1, is a CAD technique to automatically generate inexact circuits from the original one. In order to reduce the design cost, nets are pruned to remove or simplify gates based on significance and switching activity.

The Inexact Speculative Adder [6] is a generalized and optimized architecture for speculative compensated addition. As depicted in Fig. 2, it splits the carry chain in multiple paths executed concurrently. Each path consists of a carry speculator block with a determined dynamic or static carry guess, a local

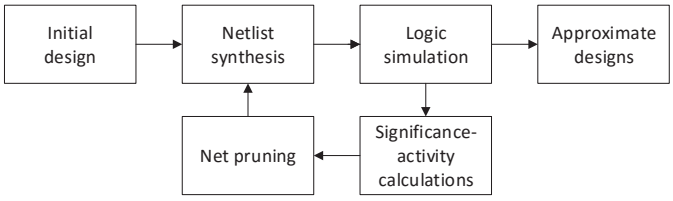


Fig. 1: Gate-Level Pruning CAD framework.

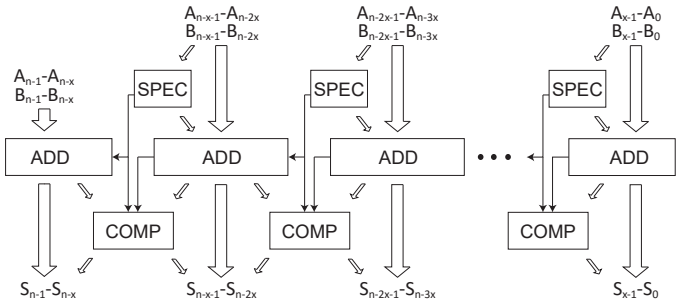


Fig. 2: Block diagram of the Inexact Speculative Adder.

adder and an error compensation block to correct the local sum or to balance the preceding sum.

Individual use of those techniques has shown great power saving abilities. While producing different types of errors, it has also been shown to be worth combining those two techniques in arithmetic adders [7].

### B. Inexact Speculative Multiplier

Multiplier circuits have much higher area, power consumption and delay than their adder counterparts. Yet, few works in literature have addressed the case of speculative multiplication. This section briefly introduces the Inexact Speculative Multiplier (ISM), a new approximate multiplier circuit derived from error-compensated speculative architectures.

Conventional parallel multiplier architectures are based on computing a set of partial products and summing them together. To be integrated in high-performance blocks such as a FPU, this process is generally pipelined with several stages. The ISM is based on a two-stage multiplier architecture. First, a Partial Product Multiplier generates and merges partial products with a compressor tree into two partial sums. Then, an Inexact Speculative Adder [6] adds them in a speculative way in the last stage. This approach strongly reduces the overall critical path, and with a retiming step, used for instance in the case of pipelining, it significantly relaxes the timing constraints, leading to smaller overall area and power consumption.

Sizing of the different speculative elements of the adder stage directly allows to trade worst-case and average errors in a delay-accuracy approach in the case of unsigned operation, as in [6]. In the case of two's-complement signed multiplication, a dynamic carry guess of the inverse of the expected sign is required on all speculative paths to avoid any sign error (i.e. a XNOR of the two operand's MSBs). Other parameters are selected in the same approach as for unsigned operation.

As the mantissa multiplier is in the critical path of the FPU circuit, even the slightest level of approximation can significantly relax the timing constraints. Moreover, the ISM error compensation and the FPU rounding unit both share the same philosophy that a few bits in one direction are equivalent to a single one at adjacent position. For instance, the FPU rounding would approximate the sequence '0.111' by '1.000', while the speculative error '0.000' instead of '1.000' would be compensated by '0.111'.

## III. Chip Implementation

Gate-level Pruning and Inexact Speculative Adders have demonstrated their efficiency on isolated arithmetic blocks by simulation [5, 6]. For the first time, these two techniques have been used to approximate the mantissa computations of FPUs implemented in a small multi-core processor.

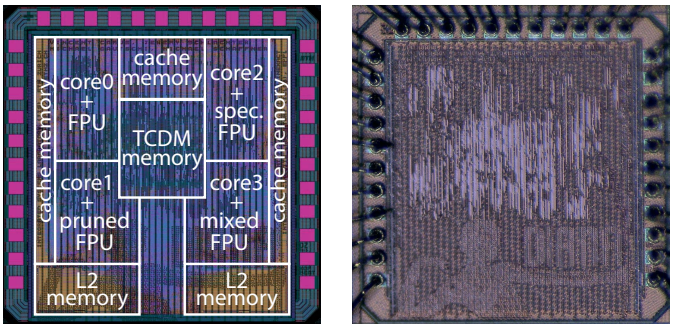


Fig. 3: Floorplan and die microphotograph of the chip. Die size is $1.56\,\mathrm{mm}^2$.

### A. Chip Architecture

As depicted in Fig. 3. a chip has been realized based on the PULP architecture [8] with 4 Or10n cores, 16 kB of L2 memory, 16 kB of tightly coupled data memory (TCDM) organized into 8 banks and 4 kB of instruction cache. Each core has a dedicated FPU capable of additions, subtractions and multiplications with 2 cycles of latency. One of them is compliant to the IEEE-754 single-precision standard while the three others are approximate variations of it. The chip has been fabricated with UMC 65 nm standard process technology and has been designed to run at a maximum frequency of 500 MHz with a power supply of 1.2 V.

### B. Approximate Floating-point Units

All the FPUs share the same architecture, the only difference is the replacement of the original mantissa adder and mantissa multiplier by approximate versions of them. In a *pruned* FPU, Gate-Level Pruning has been used to generate the approximate adder and multiplier. The Inexact Speculative Adder and the new Inexact Speculative Multiplier have been implemented in a *speculative* FPU. At last, in a *mixed* FPU, both speculation and pruning techniques have been combined to obtain even higher power and area savings.

To ensure a minimal guaranteed precision and in order to better compare the three techniques, all the approximate FPUs have been chosen to maintain exactly 10 bits of exact arithmetic computation.

### C. Error Characterization

Approximate circuits are commonly characterized and validated through the simulation of random sets of inputs since extensive simulation or measurements would be too time consuming. Hence, each of the approximate FPUs has been fed with a set of twenty million uniformly-distributed random inputs to get a statistical estimation of their approximate behavior. The hardware used for floating-point additions and subtractions is different from the one used for multiplications and is implemented with different speculative circuits and pruning levels. For this reason, floating-point additions/subtractions and multiplications have been characterized independently.

The metrics used to characterize approximate FPUs in this work are based on the *Relative Error* (*RE*), defined as:

$$RE = \left| \frac{S_{approx} - S_{correct}}{S_{correct}} \right| \tag{1}$$

where $S_{approx}$ and $S_{correct}$ are the approximate and correct results of an addition, subtraction or multiplication.

The first metric that has been considered is the *Maximum Relative Error* ($RE_{MAX}$), that represents the largest relative error

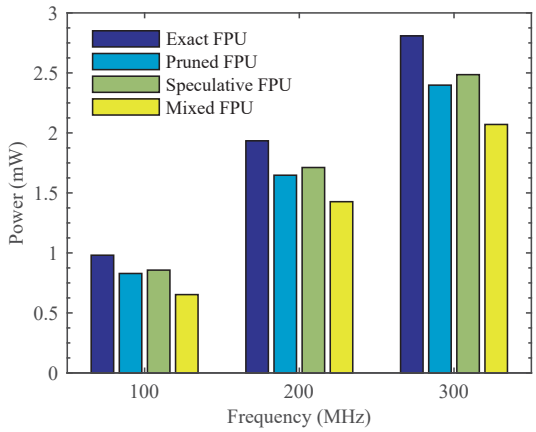


Fig. 4: Measured power consumption of the 4 FPUs for 3 frequencies.

of a floating-point operation and defines its worst-case accuracy. However, it cannot fully portray the error characteristics. For instance in the case of pruning, the $RE_{MAX}$ holds at 1 (100%). Indeed, in the pruning process, some LSBs are set to a fixed value. If the LSB of an adder is set to logic '0', the operation $1 + 0 = 0$ immediately gives a $RE_{MAX}$ of 1. To that extent, the *Relative Error RMS* ($RE_{RMS}$) has also been considered. Directly proportional to the SNR, it is a good accuracy estimator for many applications, particularly in multimedia processing.

Table I summarizes all the error characteristics. The errors characteristics of the speculative FPU are about two orders of magnitude lower than the other FPUs, but the $RE_{RMS}$ remains quite low for all operations and all approximate FPUs.

### D. Chip Power Measurements

The total power consumption of the chip has been measured by running a vector multiplication and addition benchmark, one core at a time. In order to be able to extract the consumption of a single FPU, i.e. without the overhead of the cores and memories, a second set of power measurement has been performed by running the same benchmark with all the assembly floating-point add and multiply instructions replaced by *No Operations* (NOPs). This test has been performed over 9 chips and with frequencies ranging from 100 MHz to 300 MHz[1].

Measurements shown in Table II and Fig. 4 show that the pruned FPU achieves 15 % power and 11 % area savings, whereas the speculative FPU enables 12 % power and 14 % area savings. Thanks to the switching activity criteria, pruning generally achieves better power reduction than speculation but with higher errors. Despite speculation requires extra hardware for carry generation and error compensation, it strongly relaxes the timing constraint, allowing to simplify the architecture and reduce the use of buffers or up-sized cells, leading to smaller silicon area. Combining pruning and speculation leads to 27 % power, 36 % area and 53% Power-Area Product (PAP) savings thanks to their radically different circuit approaches, and with similar error levels to pruning.

TABLE I: Error characteristics of the approximate FPUs.

| FPU | Addition/subtration | | Multiplication | |
|---|---|---|---|---|
| | $RE_{RMS}$ | $RE_{MAX}$ | $RE_{RMS}$ | $RE_{MAX}$ |
| Pruned | 1.15E-3 | 1 | 1.4E-3 | 1 |
| Speculative | 2.36E-6 | 5.69E-3 | 2.6E-5 | 1.17E-1 |
| Mixed | 2.27E-4 | 1 | 1.4E-3 | 1 |

TABLE II: Power, area and power-area product of the FPUs, measured at 1.2 V, 300 MHz, room temperature, and implemented in a UMC 65 nm technology.

| FPU | Power (mW) | Area ($\mu m^2$) | Power-area product ($W \cdot \mu m^2$) |
|---|---|---|---|
| Exact | 2.81 | 13 200 | 37.1 |
| Pruned | 2.40 | 11 850 | 28.4 |
| Speculative | 2.48 | 10 070 | 25.0 |
| Mixed | 2.07 | 8 550 | 17.7 |

[1]Measurements were not accurate above 300 MHz with the available tool.

## IV. Application to HDR Image Tone-Mapping

Carried by the high demand for consumer digital cameras integrated in phones, tablets or Internet-of-Things (IoT) devices, HDR tone-mapping is an excellent application to evaluate and validate the use of approximate hardware in the FPU by comparing the end-user impact of the image quality loss.

### A. Tone-mapping Algorithm

In this work, a tone-mapping application using non-linear masking algorithm [9] has been implemented in C and compiled to be executable on the realized chip. This method has been chosen as it is less computationally-intensive than other algorithms and particularly because it minimizes the use of floating-point divisions as those have not been implemented.

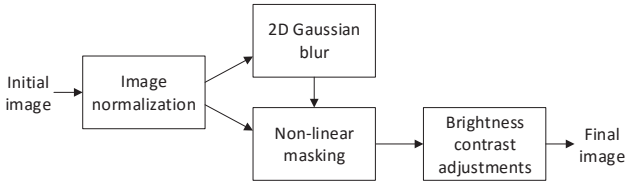


Fig. 5: Flowchart of the implemented tone-mapping algorithm.

As depicted in Fig. 5, the implemented tone-mapping algorithm consists in multiple operations. First, the initial image has to be normalized. Then, a low-pass version of the normalized image is generated by applying a 2D Gaussian-blur effect. Finally, the main tone-mapping operation is applied on the normalized image by performing a pixel-by-pixel gamma correction using the coefficients of the blurred image. Using floating-point exponent and logarithm operations built out of additions and multiplications, this step combines a high number of floating-point operations together, therefore it is a good indicator of the robustness of the approximate FPU since the propagation and accumulation of errors could have a significant impact on the final image. A brightness and contrast adjustment step is added to further improve image quality.

### B. Results

Fig. 6a shows an HDR image before tone-mapping, the landscape is not visible at all and the sun is too bright, hiding part of the clouds. Fig. 6b shows the same image after tone-mapping and brightness-contrast correction computed with the exact FPU, the entire scenery is now discernible. Fig. 6c-e show the tone-mapped images computed with the pruned, the speculative and the mixed FPU, respectively, with PSNR ranging from 76.4 dB using the pruned FPU to 127.3 dB using the speculative FPU, in line with the error characterizations. There is absolutely no difference discernible by the human eye between the picture processed by the exact FPU and the ones processed by the approximate FPUs.

To further investigate the quality loss, the approximate tone-mapped images have been compared to the exact one, pixel by

(a) Original image (b) Exact (c) Pruned (76.4 dB) (d) Speculative (127.3 dB) (e) Mixed (90.0 dB)

Fig. 6: Original image (a) and tone-mapped images obtained by each of the 4 cores (b-e). PSNR is indicated for images processed by the approximate FPUs. Image size is 512×512 pixels.

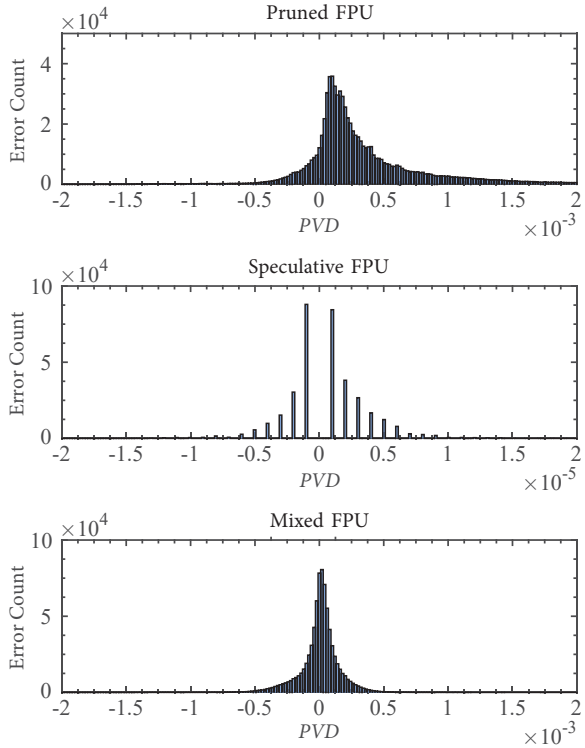


Fig. 7: Error distributions of the images tone-mapped by the approximate FPUs. X-axis of the speculative and Y-axis of the pruned FPU are scaled differently.

pixel and color by color. The *Pixel Value Difference* (*PVD*) has been used to show the error on each individual pixel and color component between the image processed with an approximate FPU and the one processed with the exact FPU. It is simply defined as the arithmetic difference between the approximate pixel value and the exact pixel value.

Fig. 7 plots the *PVD* distribution for each of the approximate tone-mapped images. The speculative FPU produces very small errors of specific magnitudes due to the specific positions of the cuts in the carry chains. On the other hand, errors produced by the pruned and mixed FPUs are spread by two orders of magnitude more than for the speculative FPU, but large errors remain rare. The error distribution of the tone-mapped image processed by the mixed FPU combines the continuous distribution as with the pruned FPU and high error-count around zero as with the speculative FPU.

## V. Conclusion

By combining Gate-Level Pruning and Inexact Speculative Adder together with a novel Inexact Speculative Multiplier, three approximate single-precision FPUs have been implemented by approximating the mantissa adder and multiplier. The FPUs have been integrated in a 65 nm CMOS process within a quad-core PULP processor in order to demonstrate their functionality in a computing system. Measurements have shown 15 % power and 11 % area savings for the pruned FPU and 12 % power and 14 % area savings for the speculative FPU. Producing different types of errors, pruning and speculation can be combined to achieve 27 % power, 36 % area and 53 % power-area product reductions. The use of those FPUs have been validated by running a floating-point-intensive tone-mapping algorithm on high-dynamic range images. Results have shown no visible quality loss, with image PSNR ranging from 76.4 dB using the pruned FPU to 127.3 dB using the speculative FPU. Additional error measurements have confirmed that each technique produces a specific error distribution with errors remaining small and centered around zero.

## Acknowledgment

The authors would like to thank the Integrated Systems Laboratory at ETHZ for supporting the fabrication costs and for providing support and equipments for the design and measurement of the chips.

## References

[1] C. M. Kirsch and H. Payer, "Incorrect systems: It's not the problem, it's the solution," in *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*, June 2012, pp. 913–917.

[2] K. Palem and A. Lingamneni, "Ten years of building broken chips: The physics and engineering of inexact computing," in *ACM Transactions on Embedded Computing Systems*, vol. 12, no. 2s, May 2013, pp. 87:1–87:23.

[3] S. Ghosh, S. Bhunia, and K. Roy, "CRISTA: A new paradigm for low-power, variation-tolerant, and adaptive circuit synthesis using critical path isolation," in *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, no. 11, Nov 2007, pp. 1947–1956.

[4] S. Basu, P. G. d. Valle, G. Karakonstantis, G. Ansaloni, and D. Atienza, "Heterogeneous error-resilient scheme for spectral analysis in ultra-low power wearable electrocardiogram devices," in *2015 IEEE Computer Society Annual Symposium on VLSI*, July 2015, pp. 268–273.

[5] J. Schlachter, V. Camus, C. Enz, and K. Palem, "Automatic generation of inexact digital circuits by gate-level pruning," in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, May 2015, pp. 173–176.

[6] V. Camus, J. Schlachter, and C. Enz, "Energy-efficient inexact speculative adder with high performance and accuracy control," in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, May 2015.

[7] J. Schlachter, V. Camus, and C. Enz, "Near/sub-threshold circuits and approximate computing: The perfect combination for ultra-low-power systems," in *VLSI (ISVLSI), 2015 IEEE Computer Society Annual Symposium on*, July 2015, pp. 476–480.

[8] F. Conti, D. Rossi, A. Pullini, I. Loi, and L. Benini, "Energy-efficient vision on the PULP platform for ultra-low power parallel computing," in *Signal Processing Systems (SiPS), 2014 IEEE Workshop on*, Oct 2014.

[9] N. Moroney, "Local color correction using non-linear masking," in *Color Imaging Conference (CIC), 8th IS&T/SID*, Nov 2000, pp. 108–111.