# Review and Benchmarking of Precision-Scalable Multiply-Accumulate Unit Architectures for Embedded Neural-Network Processing

Vincent Camus, *Member, IEEE*, Linyan Mei, *Student Member, IEEE*, Christian Enz, *Fellow, IEEE*, and Marian Verhelst, *Senior Member, IEEE*

***Abstract*— The current trend for deep learning has come with an enormous computational need for billions of Multiply-Accumulate (MAC) operations per inference. Fortunately, reduced precision has demonstrated large benefits with low impact on accuracy, paving the way towards processing in mobile devices and IoT nodes. To this end, various precision-scalable MAC architectures optimized for neural networks have recently been proposed. Yet, it has been hard to comprehend their differences and make a fair judgment of their relative benefits as they have been implemented with different technologies and performance targets. To overcome this, this work exhaustively reviews the state-of-the-art precision-scalable MAC architectures and unifies them in a new taxonomy. Subsequently, these different topologies are thoroughly benchmarked in a 28 nm commercial CMOS process, across a wide range of performance targets, and with precision ranging from 2 to 8 bits. Circuits are analyzed for each precision as well as jointly in practical use cases, highlighting the impact of architectures and scalability in terms of energy, throughput, area and bandwidth, aiming to understand the key trends to reduce computation costs in neural-network processing.**

***Index Terms*— ASIC, deep neural networks, precision-scalable circuits, configurable circuits, MAC, multiply-accumulate units.**

## I. Introduction

EMBEDDED deep learning has gained a lot of attention nowadays due to its broad application prospects and vast potential market. However, the main challenge to embrace this era of edge intelligence comes from the supply-and-demand gap between the limited energy budget of embedded devices, often battery powered, and the computationally-intensive deep-learning algorithms, requiring billions of Multiply-Accumulate (MAC) operations and data movements.

To alleviate this unbalanced relationship, many approaches have been investigated at different levels of abstraction. At algorithmic level, researchers have introduced hardware-friendly models that can keep up the original algorithm accuracy [1], [2]. At system-architecture level, dataflow schemes and memory hierarchies have been wildly studied to optimize data movement and storage [3]–[5]. Finally, new topologies have been proposed at circuit level to improve energy or performance beyond conventional design by exploiting data locality or error tolerance [6]–[8].

Among these techniques, reduced-precision computing has demonstrated large benefits with low or negligible impact on the network accuracy [9], [10]. It has been shown that the optimal precision not only varies from one neural network to another, it also can vary within a neural network itself [11], [12], from layer to layer or even from channel to channel. This has lead to a new trend of ultra-efficient run-time precision-scalable neural processors for embedded Deep Neural Network (DNN) processing in mobile devices and IoT nodes.

Accordingly, many run-time precision-scalable MAC architectures have been introduced in the recent years, built either with high parallelization capabilities [13]–[16] or serial approaches [17]–[21]. However, it has been difficult to assess their efficiency or to decide on a topology for several reasons. Firstly, they have been implemented in various process technologies, nonidentical bitwidths or scalability levels, or have been integrated into entirely different systems. Secondly, few works evaluate their performances against a baseline design, i.e. a MAC unit without scalability, to clearly show the configurability overheads. Finally, many designs are based on ad-hoc scalability techniques, lacking a systematic approach and a clear analysis of which scalability principles are common, respectively distinct, between different implementations.

This work ambitions to overcome these issues through the following contributions:

- A new taxonomy is proposed, which categorizes all the existing precision-scalable MAC architectures, uncovers their design patterns, and links MAC-level design decisions to array-level and algorithm-level choices (section II).
- An exhaustive and illustrated survey of state-of-the-art precision-configurable MAC architectures is presented, to help understand their circuit topologies and functionality differences. (section III).
- A detailed analysis of precision scaling and its overheads is conducted for each MAC architecture in terms of energy, area, bandwidth and throughput (sections IV-V).
- A comparative study of all scalable-precision MAC units is made across a wide range of performance targets, both at a nominal supply voltage and under Dynamic Voltage-Frequency Scaling (section VI).

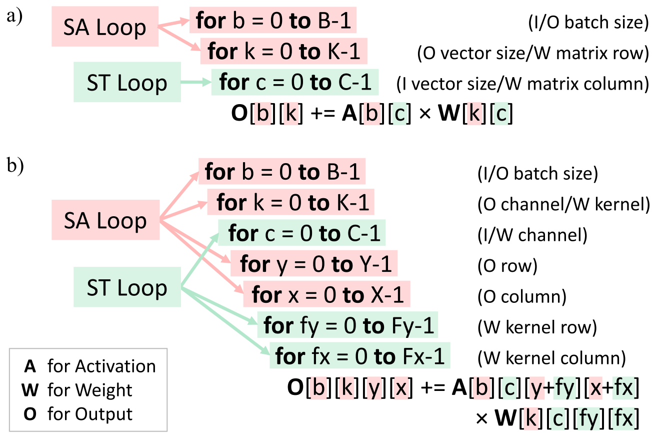
Manuscript received August 12, 2019; revised October 2, 2019; accepted October 6, 2019. Date of publication October 30, 2019; date of current version December 12, 2019. This work was supported in part by the European Research Council (ERC) under Grant ERC-2016-STG-715037 and in part by the Research Foundation - Flanders (FWO) under Grant G006718N. This article was recommended by Guest Editor M. Valle. *(Vincent Camus and Linyan Mei contributed equally to this work.) (Corresponding author: Vincent Camus.)*

V. Camus and C. Enz are with the ICLAB, EPFL, 1015 Lausanne, Switzerland (e-mail: vincent.camus@alumni.epfl.ch; christian.enz@epfl.ch).

L. Mei and M. Verhelst are with the ESAT-MICAS Laboratory, KU Leuven, 3001 Leuven, Belgium (e-mail: linyan.mei@esat.kuleuven.be; marian.verhelst@esat.kuleuven.be).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JETCAS.2019.2950386



```
a)
for b = 0 to B-1          (I/O batch size)
  for k = 0 to K-1        (O vector size/W matrix row)
    for c = 0 to C-1      (I vector size/W matrix column)
      O[b][k] += A[b][c] × W[k][c]

b)
for b = 0 to B-1          (I/O batch size)
  for k = 0 to K-1        (O channel/W kernel)
    for c = 0 to C-1      (I/W channel)
      for y = 0 to Y-1    (O row)
        for x = 0 to X-1  (O column)
          for fy = 0 to Fy-1   (W kernel row)
            for fx = 0 to Fx-1 (W kernel column)
              O[b][k][y][x] += A[b][c][y+fy][x+fx]
                               × W[k][c][fy][fx]
```

Fig. 1. Loop format for (a) fully-connected and (b) convolutional computations.

- An investigation of three practical cases of utilization is proposed, assuming different proportions of reduced-precision operations, exploring their impact on the optimal architecture selection (section VII).

Source code and supplementary materials are available online at: https://github.com/vincent-camus/benchmarking-precision-scalable-mac-units.

## II. Dataflow Implications of Precision Scalability

This section first introduces two dataflow scalability options, Sum Apart (SA) and Sum Together (ST), with implications at algorithmic, Processing Element (PE) array, and PE levels, showing that run-time precision scalability is tightly interwoven with neural-network dataflow considerations. It then maps the state-of-the-art precision-scalable MAC architectures into these categories and proposes a general MAC taxonomy.

### A. SA and ST at Algorithm Level

The concepts of Sum Apart (SA) and Sum Together (ST) were introduced at PE level by Mei *et. al* [16] to qualify two opposite ways of accumulating subword-parallel computations: SA keeps the parallel-generated products separately, while ST sums them together to form one single output result. These concepts can be applied to differentiate algorithm-level characteristics of neural-network workloads.

Fig. 1 illustrates fully-connected and convolutional DNN layer computations in a nested-loop format. Indexes appearing on the left-hand side of the MAC operation ($b$, $k$, $y$ and $x$) indicate that all their related partial results are accumulated and stored into *distinct* output variables, corresponding to SA-type accumulations. On the contrary, indexes which do not appear in the output ($c$, $fy$ and $fx$) indicate that those partial results are accumulated *together* along these dimensions, corresponding to ST-type accumulations. All loops are divided into either SA or ST types.

### B. SA and ST at PE-Array Level

Many state-of-the-art deep-learning processors rely on a multi-dimensional PE array to support convolutions or matrix multiplications. The architectures of such arrays are tightly connected to the targeted dataflow (spatial-unrolling scheme) [3], and can once again be categorized into SA and ST behaviors along each array dimension.

For example, a 2D PE array can be categorized into three types, SA-SA, SA-ST, or ST-ST, as shown in Fig. 2.

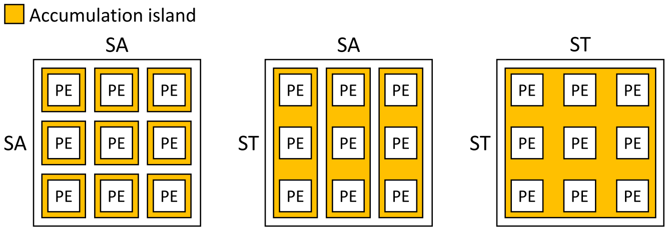


Fig. 2. Three types of two-dimensional PE array.

An SA-SA array hence supports spatial unrolling of two SA-type loops along its two geometric dimensions, while an SA-ST array unrolls one from each loop, and finally an ST-ST unrolls two ST-type loops. The choice of SA or ST array structure determines the accumulation islands in the array.

This allows categorizing neural-network accelerators into each of three different classes: SA-SA arrays contain only one single PE per accumulation island, also called Output-Stationary arrays, which typically accumulate results across clock cycles [22], [23]. SA-ST arrays contain a 1D vector of PEs per accumulation island whose results are added together, either in a systolic way [24], [25] or through a single-cycle accumulator [26], [27]. ST-ST arrays integrate all the PEs into a single accumulation island, e.g. with a 2D-accumulated adder tree [28], [29], a systolic array, or a 1D-accumulated adder tree plus systolic array.

### C. SA and ST at Precision-Scalable MAC-Unit Level

When scaling computational precision down to a fraction of the full-precision operation, spatial precision-scalable MACs typically subdivide the PE into several parallel smaller-resolution PEs. As such, each nominal PE then becomes a PE array on itself, which offers more spatial loop-unrolling opportunities along SA or ST dimensions. Many precision-scalable MAC architectures have been presented in literature, which can be categorized along this SA and ST concept.

In full-precision mode, SA and ST MACs behave identically: only one result is generated every clock cycle. In scaled-precision mode, several low-precision results are generated in parallel. The major difference between SA and ST MACs is that SA MACs keep these sub-products separately, thus generating several results in multiple output registers, while ST MACs sum them together to obtain a single accumulated result, requiring only one register.

In existing neural-network computing patterns, there is an implicit rule that if multiplications share an operand, their products *cannot* be added together. Alternatively speaking, only products that belong to different output values have the opportunity to share input values, either weight or activation. This algorithmic constraint contributes to the fundamental input-output bandwidth trade-off between SA and ST MACs. This will be covered in more details in Section III.

### D. Taxonomy

Building upon these concepts of ST and SA, Table I presents the complete taxonomy of precision-scalable MAC architectures used in the remainder of this paper.

Vertically, all precision-scalable MAC architectures are first organized into spatial and temporal precision-scaling techniques. In the category of spatial precision configurability, the MACs spatially split into several lower-precision arithmetic operators in reduced-precision modes. They are further

TABLE I

TAXONOMY OF PRECISION-SCALABLE MAC ARCHITECTURES

| Architecture types | | 1D scalable | 2D scalable | 2D symmetric scalable |
|---|---|---|---|---|
| Spatial | Sum Apart | **1D D&C SA**<br>*(DNPU [13])* | **2D D&C SA** | **Subword-Parallel SA**<br>*(DVAFS [15])* |
| | Sum Together | **1D D&C ST** | **2D D&C ST**<br>*(BitFusion [14])* | **Subword-Parallel ST**<br>*(ST [16])* |
| Temporal | Bit Serial | **1D bit serial**<br>*(UNPU [17])* | **2D bit serial**<br>*(LOOM [19])* | |
| | Multibit Serial | **1D multibit serial**<br>*(Multibit serial [18])* | **2D multibit serial**<br>*(LOOM [19])* | |

categorized using the ST and SA principles. The category of temporal precision-scalable MACs have a totally different working principle. They perform multiplication through temporal iteration of repeated add-shift operations. Reduced-precision operation is achieved by just reducing the number of temporal iterations. Temporal-based MACs are further grouped in Bit Serial and Multibit Serial, based on the bitwidth of the input data that are serially fed in at every iteration. In summary, reducing precision of input operands helps spatially-reconfigurable MACs to carry out more MAC operations in parallel, while it helps temporally-reconfigurable MACs to reduce the total number of clock cycles to finish one MAC operation.

Horizontally, architectures are categorized into three subtypes, depending on their input (weight and activation) scaling characteristics. "1D scalable" indicates that only one of the input operands is able to scale to lower bitwidth, while the other always stays at full-precision. "2D scalable" MACs break this constraint so that both input operands can scale independently. Finally, "2D symmetric scalable" MACs allow the two input operands to both scale, yet only to the same precision. Note that in 2D temporal-based MACs, the two operands are naturally independent, hence the last column is empty.

Table I maps the existing state-of-the-art precision-scalable MAC architectures onto this new taxonomy, and proposes new names for unification. As will become clear in the next section, all Divide-and-Conquer (D&C) architectures use a *bottom-up* design methodology: their multiplier logic is formed by combining several individual and identical sub-blocks, which are always active. Their configurability is in the interconnection and shift-add logic. Conversely, Subword-Parallel (SWP) architectures start from a *top-down* design philosophy: their single-block full-precision multiplier is selectively gated to reuse existing arithmetic cells in reduced-precision modes.

Note that a MAC unit is intrinsically a 2D architecture (with two input operands, activation and weight). By selecting a different precision scalability method listed in Table I for each dimension, a lot of hybrid precision-scalable MAC units can be created. For example, SA-ST MAC architectures or temporal-spatial mixed MAC architectures. This paper however focuses on discussing and comparing the pure architectural categories, as listed in Table I.

## III. SURVEY OF SCALABLE MAC ARCHITECTURES

This section is a tutorial of all precision-scalable MAC architectures listed in Table I. Their working principles, similarities and differences are fully discussed, and figures are shown to illustrate their computing states under different modes, assuming a 4-bit register headroom for output accumulation.

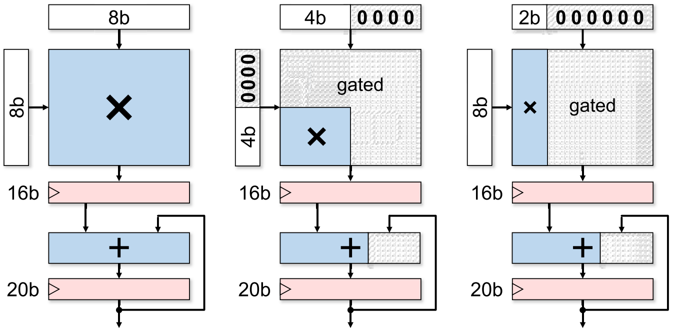


Fig. 3. Data-gated conventional MAC for either one full-precision 8b×8b, one symmetric 4b×4b, or one weight-only 2b×8b operation.

### A. Data-Gated Conventional MAC

The baseline of this study is a data-gated conventional MAC unit as in [11]. When scaled precision is applied, only the MSBs are used for computation while the LSBs are kept at zero. Hence, the switching activity is reduced. As the critical path going only through the MSBs is shorter, the frequency can dynamically be increased or the supply voltage can be lowered at equal throughput.

This is illustrated in Fig. 3, showing from left to right the use with 8b full precision, 4b symmetric scaling, and 2b weight-only scaling. When computing in reduced precision modes, some parts of the multiplier and accumulator logic are gated by zeroing input data, as shown by the shaded areas. However, as this MAC does not embed any configurability feature, such as selective clock gating, all registers remain clocked despite the fact that no data reaches their LSBs.

### B. 1D Divide-and-Conquer SA (DNPU)

Fig. 4 shows the MAC architecture of the Deep Neural Processing Unit (DNPU), introduced by Shin *et. al* [13] and characterized as a 1D D&C SA MAC unit in the new taxonomy.

Its 8b×8b multiplier is built out of four 2b×8b submultipliers followed by configurable shift-and-add logic blocks. Only one of its inputs is functionally scalable, hence its 1D scalability. More specifically, the top input is subword scalable, while the left 8b input is treated as one operand across all modes and shared by all sub-multiplier blocks. By gating the shift-and-add logic in reduced precision modes, 2 or 4 results are generated in parallel and kept separately, which explains its SA nature.

Note that like other SA-type MACs, each subword-output register requires its own accumulation headroom. For this reason, SA MACs will suffer more from a large headroom than temporal and ST MACs, which only require this headroom once.

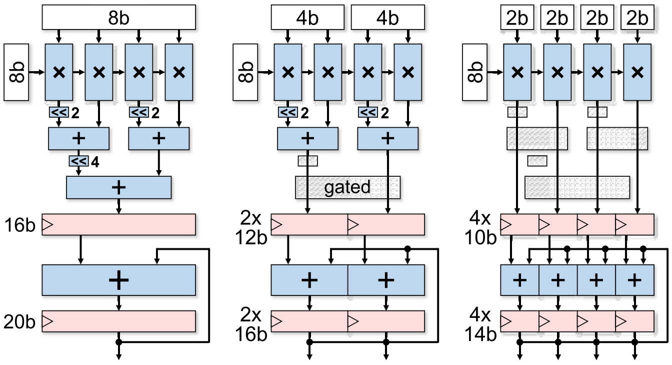


Fig. 4. Weight-only precision scaling in a 1D D&C SA MAC configured for either one 8b×8b, two 4b×8b, or four 2b×8b operations per cycle.

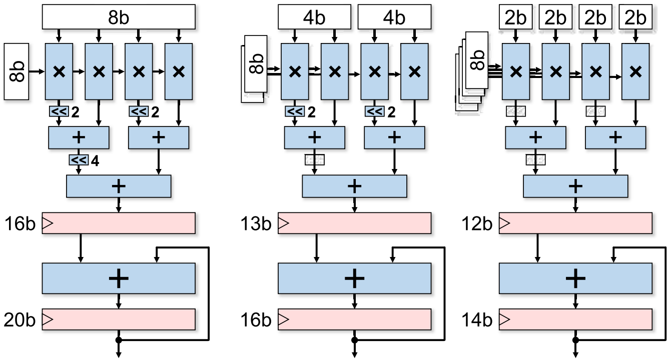


Fig. 5. Weight-only precision scaling in a 1D D&C ST MAC configured for either one 8b×8b, two 4b×8b, or four 2b×8b operations per cycle.

### C. 1D Divide-and-Conquer ST

A 1D D&C ST MAC is uncovered by the new taxonomy. This MAC is also only one-dimension scalable. However, at reduced precision, instead of storing parallel-generated results separately, the 1D D&C ST MAC accumulates them. As shown in Fig. 5, the intermediate adders in the configurable shift-add logic are not by-passed but reused for the ST accumulation.

Although the output bandwidth is reduced by only producing one result, the input bandwidth dramatically increases when precision scales down as it requires distinct 8-bit activations. This stems from the fact that neural-network computational kernels can share inputs across computations for different outputs, but are unable to share multiplicands across results accumulated into the same output.

### D. 2D Divide-and-Conquer SA

A 2D D&C SA architecture emerges from the newly introduced taxonomy. Its baseline 8b×8b multiplier can be implemented out of 16 2b×2b multipliers together with configurable shift-and-add logic blocks, which together enable it to perform either one 8b×8b, two 4b×8b, four 4b×4b, four 2b×8b, eight 2b×4b, or sixteen 2b×2b multiplications in one cycle. Fig. 6 demonstrates its four working states.

In reduced precision modes, the SA principle again results in an output bandwidth explosion, e.g. visible in the in 2D D&C SA 2b×2b mode of Fig. 6 (bottom-right). A second disadvantage of this architecture is the limited applicability. 2D D&C SA unit only supports convolutional workloads or batched workloads, as batch-1 fully-connected neural network layers do not allow data sharing across both operands.

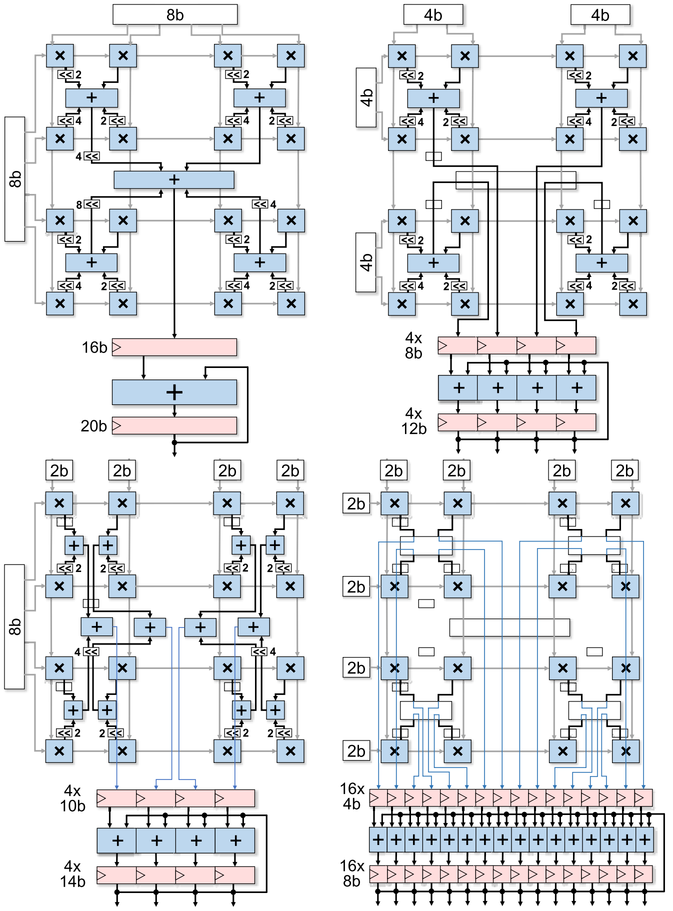


Fig. 6. Precision scaling in a 2D D&C SA MAC configured for either one 8b×8b, four 4b×4b, four 2b×8b, or sixteen 2b×2b operations per cycle.

### E. 2D Divide-and-Conquer ST (BitFusion)

BitFusion, proposed by Sharma *et. al* [14], is a 2D D&C ST MAC unit. Similar to the previous 2D D&C SA architecture, it is capable of executing one 8b×8b, two 4b×8b, four 4b×4b, four 2b×8b, eight 2b×4b, or sixteen 2b×2b multiplications in one clock cycle. Fig. 7 presents four of its computing modes. Instead of keeping parallelly-generated products apart, 2D D&C ST sum them together thus avoiding output bandwidth explosion. Yet, as a consequence, input sharing is not possible, boosting the required input bandwidth at reduced precision modes.

In BitBlade, a variant of BitFusion introduced by Ryu *et. al* [21], the shifting logic is pulled out of each MAC unit and shared across sixteen 2b×2b sub-multipliers of different MACs of the MAC array to reduce the configurability overhead. However, this work only focuses on MAC-level techniques, besides, this array-level optimization is applicable to many other MAC architectures. Therefore, BitBlade is not included in this benchmarking study.

### F. Subword-Parallel SA (DVAFS)

The so-called Dynamic Voltage-Accuracy-Frequency Scaling (DVAFS) MAC, developed by Moons *et. al* [23] can be categorized as a Subword-Parallel (SWP) SA MAC in the proposed taxonomy.

Unlike all previously discussed precision-scalable MAC types, SWP SA is a 2D symmetric scalable architecture, which follows a top-down design methodology. Its functional 8b

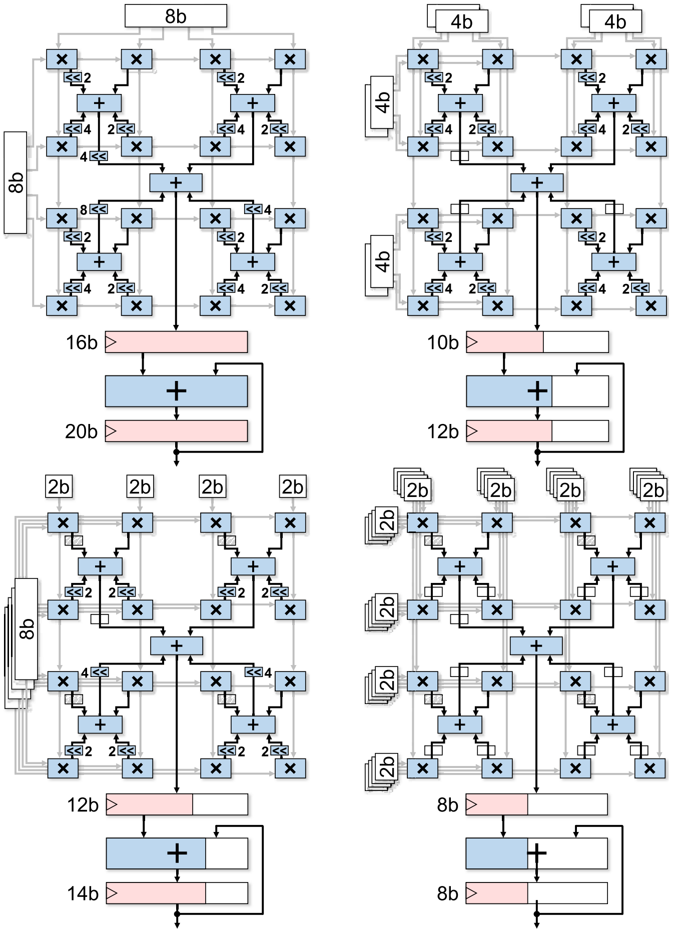


Fig. 7. Precision scaling in a 2D D&C ST (BitFusion) MAC configured for one 8b×8b, four 4b×4b, four 2b×8b, or sixteen 2b×2b operations per cycle.

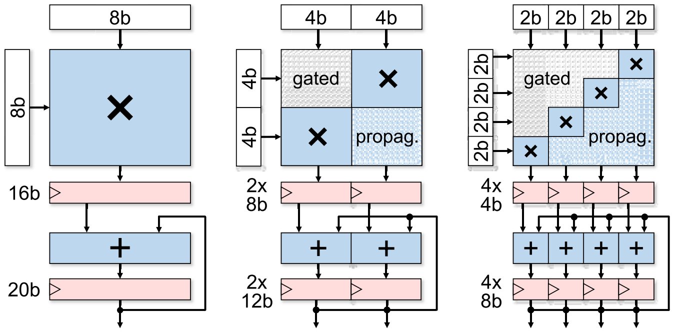


Fig. 8. Symmetric precision scaling in a SWP SA (DVAFS) MAC configured for either one 8b×8b, two 4b×4b, or four 2b×2b operations per cycle.

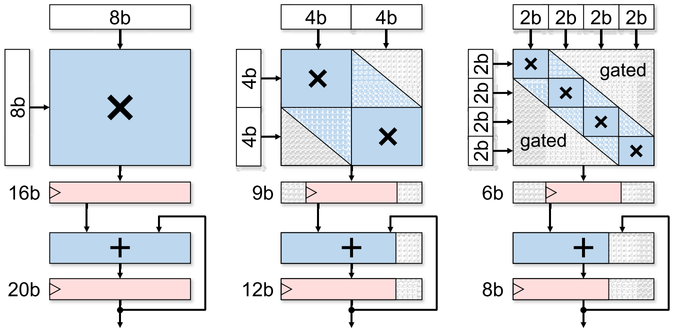


Fig. 9. Symmetric precision scaling in a SWP ST MAC configured for either one 8b×8b, two 4b×4b, or four 2b×2b operations per cycle.

multiplier is no longer realized by fusing several smaller multipliers with a configurable interconnect. Instead, it exploits the interconnection pattern of an array multiplier and selectively gates/activates parts of the arithmetic logic to perform one 8b×8b, two 4b×4b, and four 2b×2b MAC operation(s) within one clock cycle, as shown in Fig. 8. Large output registers are required due to the SA principle with its required duplicated register headroom.

Note that in reduced precision modes, three types of region appear in the array multiplier, depicted by differently shaded areas. The blue blocks marked with an "X" are active multiplication logic. The gray stripes represent fully-gated logic. And most importantly, the light-blue stripes localize cells which are not employed for arithmetic, but are active for propagating the intermediate results from the active multiplication logic to the bottom-right corner of the array multiplier, where the outputs are collected, hence preventing complete gating of these regions.

### *G. Subword-Parallel ST (ST)*

The Sum Together (ST) version of the SWP MAC unit, introduced by Mei *et. al* [16] is also a 2D symmetric scalable architecture, based on an array multiplier. But unlike SWP SA, SWP ST adds all subword results together by activating the array multiplier into an opposite diagonal pattern, as shown in Fig. 9. Such configuration saves addition logic, as it uses the multiplier array cells to implicitly perform the addition. The propagating region that can not be fully gated, shrinks along the precision down-scaling, while the critical path remains roughly the same, which are just opposite to SWP SA.

It is worthy notifying that, compared to other ST architectures, the input bandwidth of SWP ST remains the same across precision modes at the cost of partly gating its arithmetic blocks when precision scales down. So, there is a tradeoff for ST-type MACs between input bandwidth consistency and high hardware utilization.

### *H. 1D Bit-Serial Designs (UNPU)*

Bit-serial designs have recently gained attention with both the Unified Neural Processing Unit (UNPU) by Lee *et. al* [17] and the QUEST log-quantized 3D-stacked inference engine by Ueyoshi *et. al* [20]. Indeed, bit-serial operand feeding implicitly allows fully-variable bit precision.

Considered in this study, the UNPU bit-serial MAC receives weights through 1-bit iterations while activations are kept at full precision and sent in a parallel manner. Illustrated in Fig. 10, this design is thus categorized as a 1D bit-serial architecture. Scaling activations is possible by data gating.

The implementation chosen in this study assumes a right-shifting sequential multiplier as it requires a smaller firststage adder than a left-shifting design, preventing long carry propagation and sign-bit extension.

In [18], the original 1D bit-serial concept has been extended to *1D multi-bit serial* designs. Fig. 11 shows the example of a 1D 4-bit serial MAC, where weights are fed in 4 bits at a time. This scheme requires only 2 clock cycles for an 8-bit computation, hence reducing the energy consumed in the clock tree and registers. Lower precision can be obtained by gating the unnecessary bits, as in the right of Fig. 11. This survey includes both 1D 2-bit and 4-bit serial MACs.

### *I. 2D Bit-Serial Designs (LOOM)*

LOOM, designed by Sharify *et. al* [19], has extended the 1D bit-serial approach to a 2D scalable architecture. This

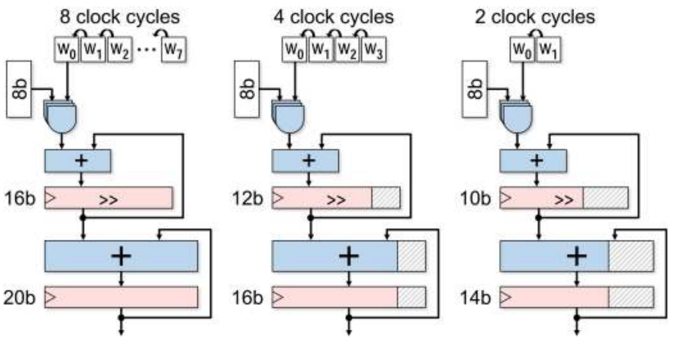


Fig. 10. Weight-only precision scaling in a bit-serial MAC configured for either 8b×8b, 4b×8b, or 2b×8b operations.

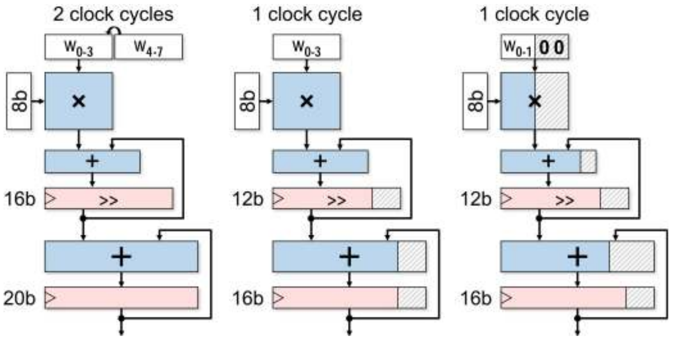


Fig. 11. Weight-only precision scaling in a 4-bit serial MAC configured for either 8b×8b, 4b×8b, or 2b×8b (by gating the 4b×8b) operations.

2D scalable design feeds both input operands bit-serially, as illustrated in Fig. 12. By doing so, the bit-wise AND logic from Fig. 10 is further simplified to a single AND gate, but the number of clock cycles to complete one MAC operation is increased to the product of activation and weight precisions.

Since both input operands are fetched bit by bit and all input bit combinations need to be traversed, a more complex input feed-in schedule and control logic are required. Fig. 13 presents the 4b×4b-operation scheduling used in this work.

Note that the input fetching pattern is not regular and contains repetitions. For example, in this 4b×4b case, the bit-by-bit weight fetching sequence is: $w_0, w_1, w_0, w_2, w_1, w_0, w_3, w_2, w_1, w_0, \ldots$. In addition, unlike 1D bit-serial units that perform a right shift at each clock cycle, 2D bit-serial units shift partial products irregularly, as indicated by the red arrows in Fig. 13.

2D bit-serial implementations can also be implemented in a multi-bit processing fashion. Fig. 14 exhibits a 2D 4-bit serial design under three different precision modes. By feeding in 4 bits at a time for both operands, the number of compute cycles is drastically reduced. Note that since the basic arithmetic block is a 4b multiplier, the precision scaling can only be pushed down to 4 bits. Scaling below 4 bits is again enabled through LSB data gating. This survey includes both 2D 2-bit and 4-bit serial MACs.

## IV. Design and Benchmark Methodology

The remainder of this paper puts together individual evaluations and comparative studies of all the aforementioned precision-scalable MAC architectures. To this end, the present section describes the methodology used for implementing and benchmarking these circuits for 5 precision modes: 8b×8b full precision, 2b×2b and 4b×4b symmetric scaling, and 2b×8b and 4b×8b weight-only scaling.

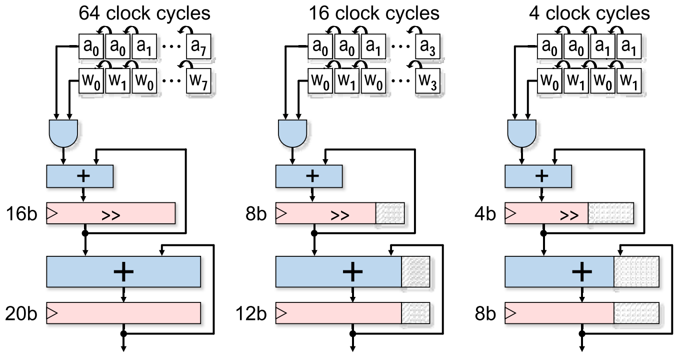


Fig. 12. Symmetric precision scaling in a 2D serial MAC configured for either 8b×8b, 4b×4b, or 2b×2b operations.

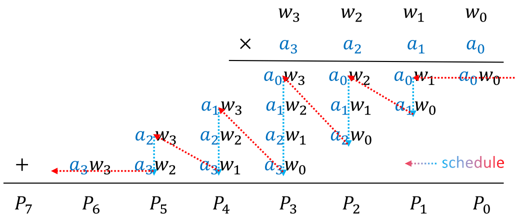


Fig. 13. A bit-wise feed-in schedule for 2D bit-serial circuit in 4b×4b mode.

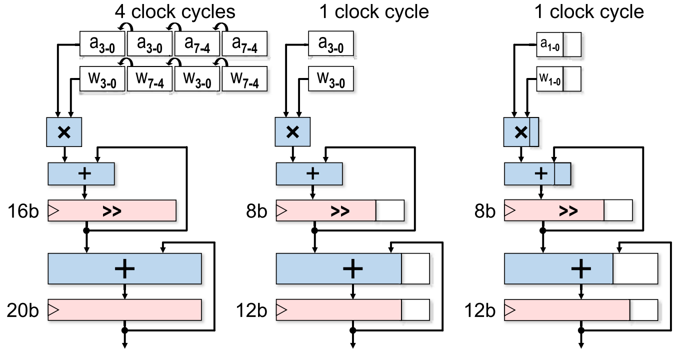


Fig. 14. Symmetric precision scaling in a 2D 4-bit serial MAC configured for either 8b×8b, 4b×4b, or 2b×2b (by gating the 4b×4b) operations.

### A. Design Considerations and Assumptions

To ensure the fairness of this study, all MAC architectures are designed with equivalent features and optimizations.

They are all built with a two-stage pipeline structure for the 8b full-precision baseline. To support the different precision modes, their accumulation registers are partitioned and gated as shown on Fig. 15. The accumulation headroom is 4b for any precision mode. Outputs are updated and stored separately for SA MACs (Fig. 15a), and into a single unpartitioned register for ST and temporal-based scalable MACs (Fig. 15b).

As the greater part of embedded neural networks uses ReLU activation in between layers, which only produces positive values, all designs assume a signed-weight and unsigned-activation representation. This choice, however, should have little impact on the relative architecture comparison if considering fully signed representation.

Input registers and overheads that can be shared among multiple PEs (e.g. control logic or finite state machines) are excluded from area and internal power reporting.

To keep this architectural study as general as possible, no individual optimization techniques nor features are considered for any design. For instance, LUT-acceleration [13],

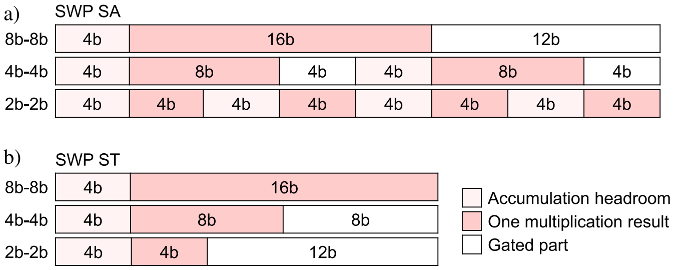


Fig. 15. Register settings for SWP SA (a) and SWP ST (b) MAC units.

or sparse-operation guarding [30] are not implemented. These orthogonal methods can generally benefit many architectures.

### *B. Design Space*

Multiple implementations of each architecture are made, varying the *level of scalability*, i.e. the scalability granularity. For instance, a 1-level scalable design allows to scale from 8b down to 4b by design, the 2b mode being carried out by data gating over the 4b mode. Data gating is applied from LSB to MSB to prevent unnecessary energy spent on carry propagation and sign-bit extension. A 2-level scalable design directly allows to scale down to 4b and 2b by design. Requiring different circuit and neural-network implementations, binary computations (1b precision) are not considered in this study.

Since the various MAC architectures should have very different optimal operating frequencies, this study explores a broad range of clock targets with frequencies from 600 MHz to 4 GHz. To allow each mode to find its optimum, constraint sweeps are applied individually to each mode.

### *C. Physical Implementation and Timing Analysis*

The performance of the MAC architectures is evaluated through synthesis and simulation in a commercially-available 28 nm low-leakage process in typical corner and with a nominal supply voltage of 1 V. Circuits are synthesized from abstract-level SystemVerilog descriptions using Cadence Genus with high-effort compilation options. Conservative power models are used for synthesis and power estimation.

The delay of mode-control signals is not optimized as it is assumed that the same precision is used for many cycles in a row (e.g. for processing a full DNN layer).

To enforce the scaling of the critical path at reduced precision, a multi-mode timing optimization is carried out to optimize the speed for each scaling scenario. Circuit delay is thus constrained and measured independently in each mode (even if the unit does not allow scaling by design, e.g. the conventional MAC still receives multiple constraints to optimize the shorter critical paths when data-gating inputs). For each precision mode, known static signals and unused registers are declared as such to prevent Static Timing Analysis (STA), which is workload-agnostic, to unnecessarily spend resources on them.

### *D. Power Estimation*

Power consumption is estimated through post-synthesis simulations at maximum operating frequency for each mode with ModelSim simulator and Cadence power extraction. This switching-activity-based estimation is very important as the gated or unused logic, depending of the precision mode, can lead to large differences in dynamic power.

The simulation for each mode consisted out of 10,000 MAC operations with an accumulator-register reset every 50 operations A set of Gaussian-distributed random numbers representative for quantized DNNs trained for CIFAR-10 is used for simulation.

Input registers and overhead shareable among several MACs within an array are not included in the reported power results.

### *E. DVFS*

Dynamic Voltage-Frequency Scaling (DVFS) provides an additional dimension for power-frequency optimization, allowing to dynamically move the operational point closer to a mode's optimal point and to further increase energy efficiency.

This study models the impact of DVFS on the circuits, assessing throughput and energy for each mode while sweeping the voltage from 1 V down to 0.8 V. To this end, the delay between the 0.8 V and 1 V characterized libraries is estimated using the Sakurai-Newton model [31], while the energy is quadratically interpolated from the previous gate-level simulations.

## V. Detailed Analysis

This section evaluates the circuit characteristics for one representative synthesized instance of each architecture for each scalability level. For this detailed analysis, only the circuit with the lowest energy-delay-area product is selected.

Figs. 16-19 show the breakdown of memory bandwidth, energy per operation, and silicon area of the circuits. The left subfigures display symmetric scaling scenarios while the right ones display weight-only scaling. Triplets of bars differentiate the precision scenarios (8, 4 or 2 bits) for each instance.

### *A. Bandwidth*

Figs. 16a-b present the required memory bandwidth per clock cycle of the different MAC units over each mode.

This shows the internal trade-off between input and output bandwidths. SA MACs produce multiple independent results in parallel, hence increasing the *output* bandwidth when precision scales down by design (i.e. down to 4b for 1-level scalable circuits and down to 2b for 2-level ones). However, since most neural networks allow data reuse over different outputs, inputs can be shared among the sub-computations, leading to a lower *input* bandwidth.

Inversely, ST MACs only store one result, which is the sum of multiple low-precision multiplications, leading to a relatively small *output* bandwidth. However, since those products are parts of one output element, they cannot share the same inputs. This results in a largely increased *input* bandwidth. For example, in order to keep its 16 arithmetic blocks busy in 2b×2b mode, the input bandwidth of 2D D&C ST explodes. Such highly-unsteady bandwidth may complicate its memory interface and its integration into a chip.

Finally, the much smaller bandwidth of temporal-based MAC units could be a huge advantage over spatial-based designs for narrow memory-band systems. However, as their computing capability is also lower, this result should be considered relatively to the computational throughput.

### *B. Bandwidth per Operation*

Figs. 17a-b therefore show the bandwidth per operation of each MAC, derived from above-mentioned bandwidth (bit/clock cycle) and computing capability (operation/clock cycle).

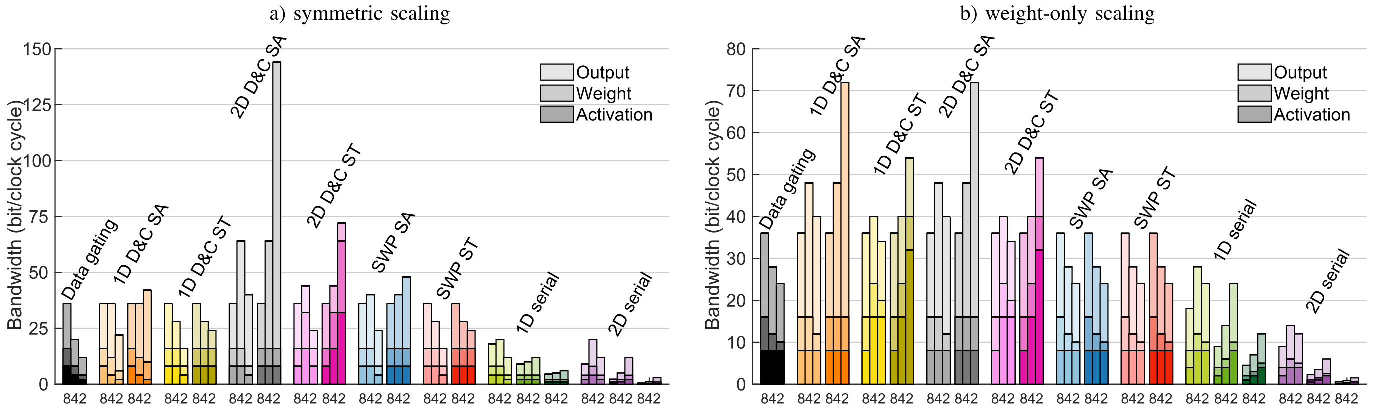


Fig. 16. Bar charts of the bandwidth per clock cycle of precision-scalable MAC units for symmetric (a) and weight-only (b) scaling scenarios.

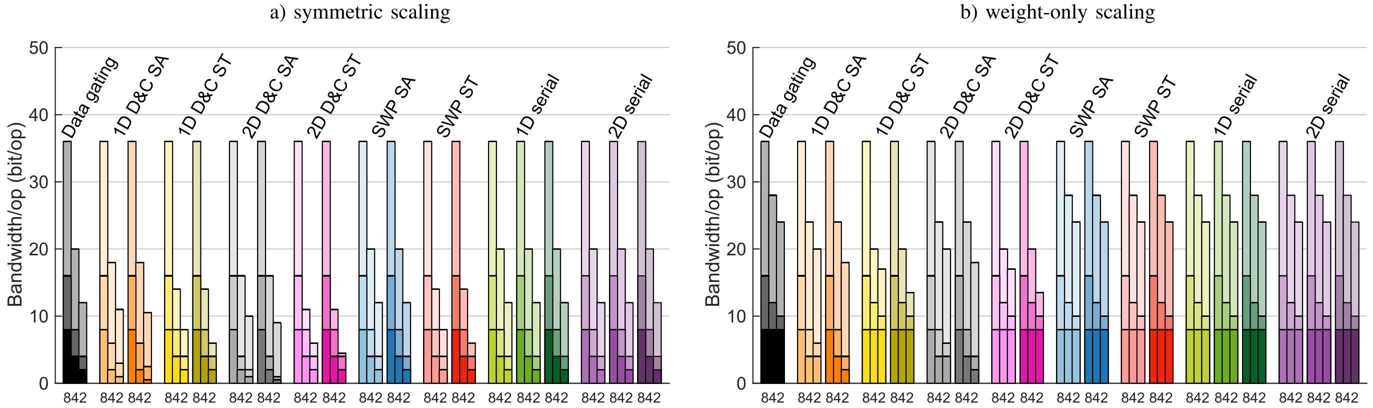


Fig. 17. Bar charts of the bandwidth per operation of precision-scalable MAC units for symmetric (a) and weight-only (b) scaling scenarios.

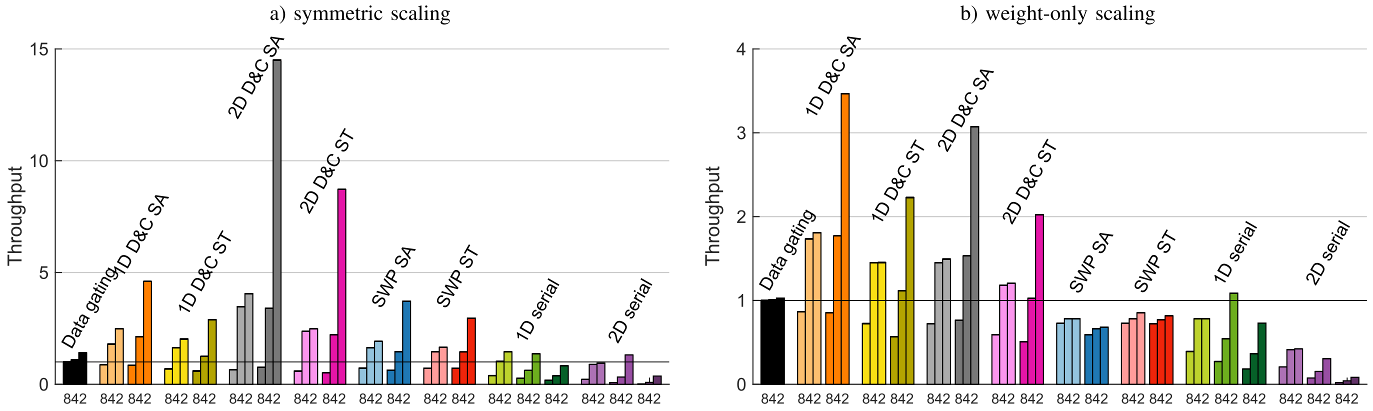
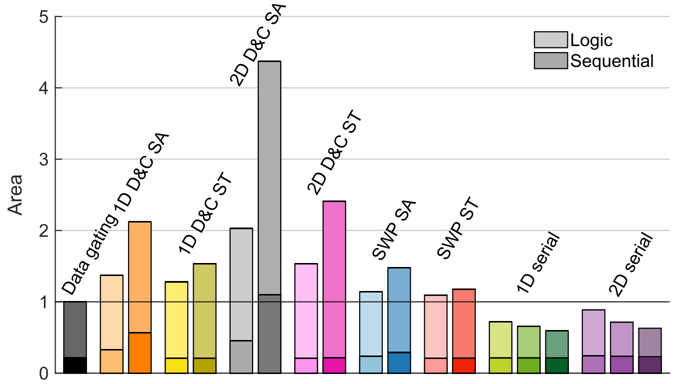


Fig. 18. Bar charts of the normalized circuit throughput of precision-scalable MAC units for symmetric (a) and weight-only (b) scaling scenarios.



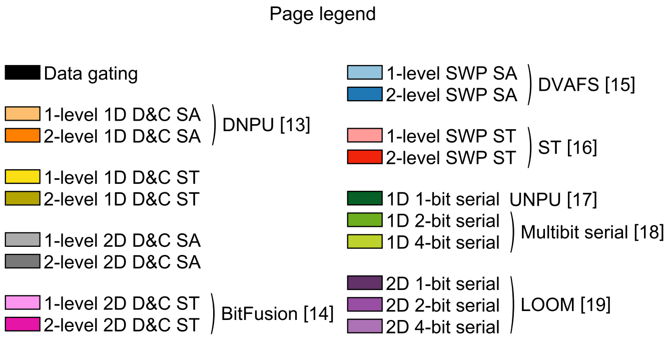


Fig. 19. Bar chart of the normalized area of precision-scalable MAC units.

As expected, at full precision, all MACs have the same bandwidth per operation. With precision scaling down, ST-type MACs show superiority over others, especially 1D and 2D D&C ST. Indeed, their arithmetic block remains fully utilized, which keeps their computing capability high, and their output bandwidth remains very low, which balances the increased input bandwidth.

For symmetric scaling, SWP MACs lose their advantage because of their unused arithmetic logic, leading to a relatively low computing capability. SWP ST is slightly better than SWP SA because with the same computing capability, it has lower bandwidth requirements. For asymmetric scaling, SWP MACs only benefit from data gating, behaving like the baseline circuit.

In temporally-scaled MAC units the low bandwidth and low compute capacity compensate for each other, resulting in similar bandwidth per operations as baseline circuit.

### C. Throughput Evaluation

Figs. 18a-b compare the measured throughput of the different architectures normalized to the data-gated conventional MAC in full-precision mode.

At 8b and 4b precisions, most designs display similar throughputs in their 1-level and 2-level variants. This result demonstrates that throughput is not affected by the level of scalability, contrarily to area and energy. Hence, systems targeting high performance can fearlessly consider embedding many precision modes.

The speed gain obtained by data gating the 1-level designs from 4b to 2b is negligible despite the multi-mode optimization. Conversely, scalability levels bring a near-quadratic boost to performance for all designs. 2-level scalable designs reach impressively high throughput factors in 2b precision. Above all, 2D D&C SA achieves 14.5× the base throughput for symmetric scaling, thanks to its 16 parallel sub-computations. Its compatibility with both symmetric and asymmetric scaling allows it to even reach a good second-ranking in weight-only scaling with 3.1× base throughput.

Interestingly, the 2-level 1D D&C SA appears versatile for both weight-only and symmetric scaling in spite of not being optimized for it. It reaches an excellent 4.6× and 3.5× base throughput for symmetric and asymmetric scenarios, respectively. This is due to its first-stage adder, which remains in the critical path in all modes, and thus, whose optimization jointly benefits to all modes. While for SWP designs, the multiplier hardware has to be co-optimized for different objectives as the critical path changes from one mode to the other. Another reason for its great speed is its SA accumulation stage which only contains short and therefore faster adders.

### D. Area Breakdown

Fig. 19 shows the area breakdown of the different MAC units synthesized in a 28 nm CMOS process normalized to the area of the data-gated conventional MAC circuit.

Not surprisingly, all scalable units based on sub-computation parallelism requires overhead for configurability. Among them, 1D and 2D D&C approaches are the most area-consuming due to customized internal structures, with 2D D&C designs requiring up to 4.4× the area of a conventional MAC (without including the input registers as mentioned in IV-A).

On the contrary, SWP MACs mitigate the overhead by reusing redundant arithmetic cells for subword-parallel computations. SWP ST circuits are particularly optimal, with only 10% to 18% overhead for 1 and 2-level scalability, respectively.

Serial designs are the only type requiring less area than the conventional multiplier, allowing area savings up to 40% on the MAC circuit for area-constrained systems.

Concerning the sequential area (dark bars), D&C SA designs require far more registers than others. This is due to their wide asymmetric sub-products (as one operand is kept full-precision), as well as the quadratic increase of sub-products for 2D D&C SA. Although SA-type, SWP SA keeps a low and slow-growing sequential area at the cost of a sacrificed throughput.

### E. Energy Overhead at Full Precision

Figs. 20-27 show the breakdown of energy per operation when scaling precision for each type of architecture. The left subfigures show symmetric scaling scenarios while the right ones show weight-only scaling. All energy values are normalized to the same full-precision data-gated conventional MAC drawn with a solid black line.

Processing at full precision with scalable designs *always* comes with some energy penalty. For 1D D&C and SWP MACs (Figs. 20-21, 24-25), energy overheads for 8b computations are in the order of 20% to 40% for 1 and 2 levels of scalability, respectively. For 2D D&C architectures (Figs. 22-23), these costs increase to 52% and 94%.

Serial designs (Figs. 26-27) require much more energy at full precision due to their need for several clock cycles per computation, diluting the power into the clock tree. At full precision, the 1D bit-serial MAC consumes 3.3× as much energy per operation as data gating, and the 2D bit-serial 14× as much.

Reassuringly, 1D multi-bit serial designs come at a lower energy penalty: the 2-bit serial MAC consumes 2.2× baseline energy, while it is also able to scale precision down to 2 bits by design, and the 4-bit serial MAC reduces the cost to 1.5× as much as data gating.

For nearly all circuits, the 2-level MAC consumes more at full precision than its 1-level version. Singularly, this is not the case for 1D D&C ST and SWP ST architectures, for which the second level of scalability costs barely more than the first. This is connected to their simpler ST-style accumulation coupled to their efficient reuse of multiplication logic.

### F. Energy Scaling

Figs. 20-27 allow to evaluate the scaling efficiency of each architecture.

In general, reducing precision of both weights and activations (left subfigures) with simple data gating leads to linear energy savings with respect to the bit precision. In comparison, all scalable MACs show a steeper slope between 8b and 4b precision, meaning that they save energy in a superlinear way with bit precision. Below 4 bits, 1-level scalable circuits (including 1D and 2D 4-bit serial MACs) can only scale precision through data gating, returning to the slope of the baseline. Despite less scalable, these designs exhibit decent energy savings compared to 2-level scalable MACs thanks to lower overheads.

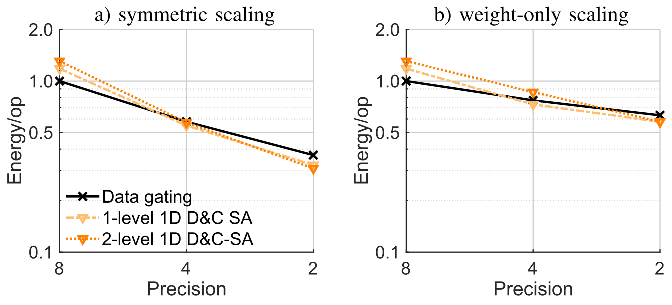


Fig. 20. Normalized energy/op for symmetric (a) and weight-only (b) scaling in a 1D D&C SA MAC (DNPU) [13].

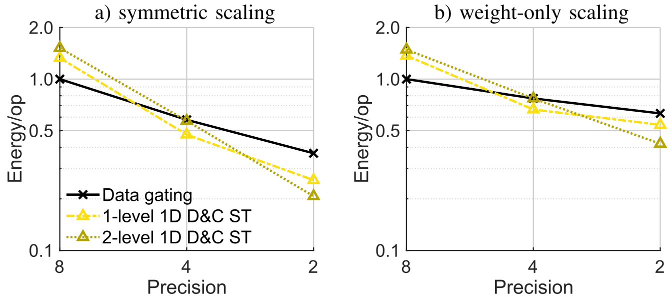


Fig. 21. Normalized energy/op for symmetric (a) and weight-only (b) scaling in a 1D D&C ST MAC.

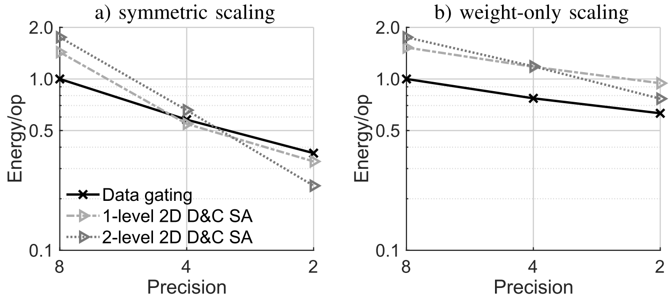


Fig. 22. Normalized energy/op for symmetric (a) and weight-only (b) scaling in a 2D D&C SA MAC.

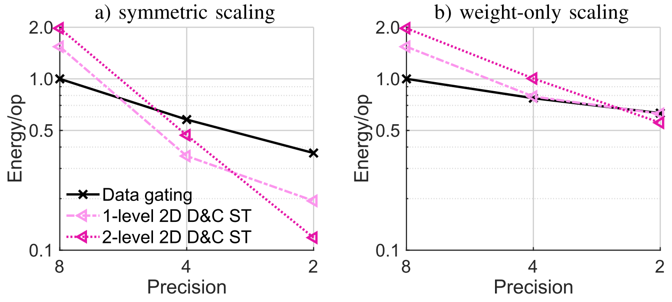


Fig. 23. Normalized energy/op for symmetric (a) and weight-only (b) scaling in a 2D D&C ST MAC (BitFusion) [14].

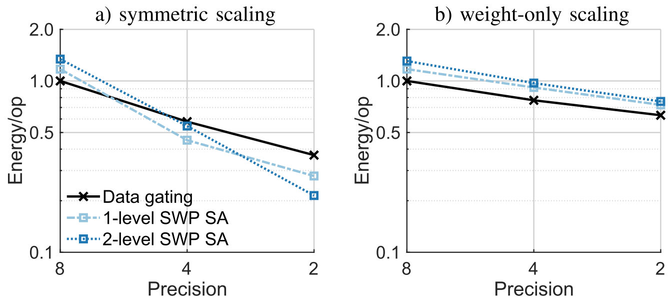


Fig. 24. Normalized energy/op for symmetric (a) and weight-only (b) scaling in a SWP SA MAC (DVAFS) [15].

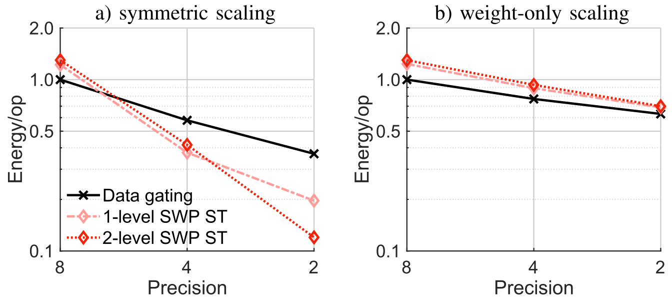


Fig. 25. Normalized energy/op for symmetric (a) and weight-only (b) scaling in a SWP ST MAC (ST) [16].

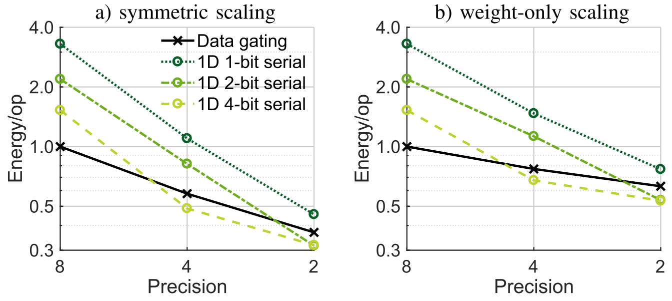


Fig. 26. Normalized energy/op for symmetric (a) and weight-only (b) scaling in 1D serial [17] and multibit-serial MACs [18]. Beware of the scale.

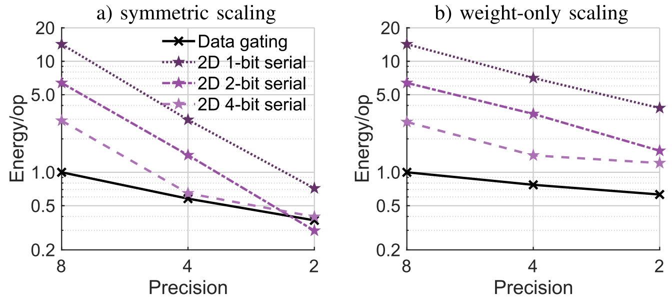


Fig. 27. Normalized energy/op for symmetric (a) and weight-only (b) scaling in 2D serial and multibit-serial MACs (LOOM) [19]. Beware of the scale.

When preserving full-precision activations (right subfigures), savings are without exception far lower. The two SWP architectures (Figs. 24-25) can only scale energy through data gating, hence they are unsuitable for weight-only scaling. Even though their architecture is built for both scenarios, 2D D&C (Figs. 22-23) and 2D serial (Fig. 27) MACs also seem ineffective for weight-only scaling. Only the 1D D&C ST MAC (Fig. 21) proves to be highly efficient for both scaling scenarios, distantly followed by the 1D 4-bit serial (Fig. 26) MAC circuit.

The slope of 1D D&C SA units (Fig. 20) is one of the lowest among spatial-unrolling architectures. In spite of being built for weight-only scaling, the 2-level 1D D&C SA circuit consumes more energy in 4b mode than data gating and saves merely as much in 2b mode (9%) as its 1-level variant. This demonstrates that the 1D D&C SA architecture poorly scales in terms of energy efficiency. On the contrary, the 1D D&C ST architecture (Fig. 21) is highly efficient and proves to sustain its efficiency with extra scalability levels.

Among spatially-unrolled architectures, the 2D D&C ST MAC (Fig. 23) and the SWP ST MAC (Fig. 25) have the sharpest energy drop when reducing precision symmetrically. They lead to comparable savings at 2b precision (48% for 1-level and 68% for 2-level scalable designs). SWP ST MAC is preferable when using 8b precision thanks to lower overheads, while 2D D&C ST MAC should be favored when requiring asymmetric scaling.

## VI. Comparative Study

This section presents an overall comparison of all MAC architectures in terms of energy per operation and throughput per area, for 1V nominal supply as well as under DVFS.

### A. Comparison of Scalable MACs at Nominal Voltage

Fig. 28 displays the feasible implementations and Pareto frontiers for each precision-scalable MAC design at each precision scenario, sweeping across a broad range of frequency constraints at nominal supply voltage. Circuits are compared in terms of energy per operation and throughput per area. The best circuits are towards the bottom-right corners of subfigures.

At 8b precision (Fig. 28c), data gating is undeniably the most efficient, capable of the highest throughput per area, followed first by 1-level (bright colored dashed lines) and then by 2-level (dark colored dotted lines) scalable designs which suffer from longer critical paths.

When scaling precision symmetrically (Figs. 28a-b), the 2D D&C ST and SWP ST architectures largely outperform other architectures in terms of energy per operation. 2D D&C SA circuits are by far the best for throughput-area performance, but not energy efficiency, given their large non-shared registers.

Note that 1-level scalable circuits stay the best compromise at 4b precision and above, but this trend reverses at 2b precision, which is visible by the inversion of bright and dark curves between left and center subfigures.

When only reducing weight precision down to 2b (Fig. 28d), 2-level 1D D&C ST architecture is optimal for energy while its 1D D&C SA companion outshines for throughput-area efficiency. At 4b precision (Fig. 28e), 1-level 1D D&C ST and 1D 2-bit serial are best for energy, however the gains over baseline are limited (20% at most). By-passing internal additions, 1D D&C SA is advantaged for throughput, while 1D 2bit serial benefits from a smaller circuit area.

### B. Comparison of Scalable MACs with DVFS

Fig. 29 displays the comparison and Pareto frontiers of precision-scalable designs with DVFS between 0.8 and 1 V.

The relative comparison between designs is fairly similar than at nominal voltage. However, the Pareto frontiers for all designs are perceptibly larger than at nominal voltage, especially regarding the range of achievable throughput-area solutions. This is manifest for the SWP ST architecture for example. DVFS offers one smooth degree of freedom to utilize the circuits at the optimal energy point within a target throughput or energy budget.

## VII. Comparative Study on Use-Case Ratios

### A. Introduction and Methodology

The best scalable-precision circuit is neither the one with the lowest consumption at reduced precision, nor the one with the lowest overhead, nor the one with the steepest energy scaling. The best circuit is the one that best optimizes its application needs. Hence, this section offers a comparative study for practical use cases of the MAC units, rather than for each precision mode individually. Such study is vital as it makes a direct link to the application level, and it simplifies the analysis by combining many results into a single trade-off.

Three case studies are assessed by defining the percentage of operations the MAC does under each precision mode. For simplicity, only the fraction of full-precision computations is set, while the ratios of 4b and 2b operations are assumed equal. Energy and throughput efficiencies are weighted considering these ratios to redraw a global Pareto frontier.

Fig. 30 shows the case studies with full-precision precision representing 33%, 20%, and 5% of the computations, respectively (2D 1-bit serial MACs have been omitted for clarity due to their sub-optimality). Only nominal voltage is considered here since the 1 V comparison results remain valid with DVFS.

### B. Equal Usage

First and foremost, when using all the precision modes uniformly (left subfigures), the conventional MAC with simple data gating is close to the most efficient architecture, but is also by far the best in terms of throughput and area. Indeed, no matter how efficient scaled operations are, being much slower and energy-consuming, 8b computations are largely dominant in the energy budget.

Note that even if 2b and 4b operations represent 66% of computations in this use case, since these are often 2× to 15× faster in scaled precisions (c.f. V-C), the circuits stay in these precision modes for a much shorter duration than to do the 8b operations.

For symmetric scaling scenarios, the conventional MAC unit is slightly outperformed by the 1-level SWP SA architecture and by both 1-level and 2-level SWP ST circuits. On the other hand, for weight-only scaling, the conventional units stay more energy efficient, before 1-level 1D D&C SA and ST MAC circuits.

### C. 20% Full-Precision Computations

Fortunately, DNNs have proven resilient to higher percentage of scaled-precision computations. Lowering the amount of full-precision operations down to 20% (central subfigures) makes scalable MACs worth the efforts to lower energy consumption for symmetric scaling scenarios. At this precision ratio, SWP SA and ST designs exceed in energy efficiency with up to 20% energy saving compared to data gating.

1-level SWP SA and ST circuits continue to surpass the conventional MAC for symmetric scaling. In general, 1-level scalable designs perform better than their 2-level equivalents, except for the 2-level SWP ST circuits that outshine.

However, for the weight-only scaling scenario, scalable MACs remain mostly inefficient in terms of energy. The 2-level 1D D&C ST architecture is hardly better than the conventional MAC at the cost of considerably worst area-throughput capability.

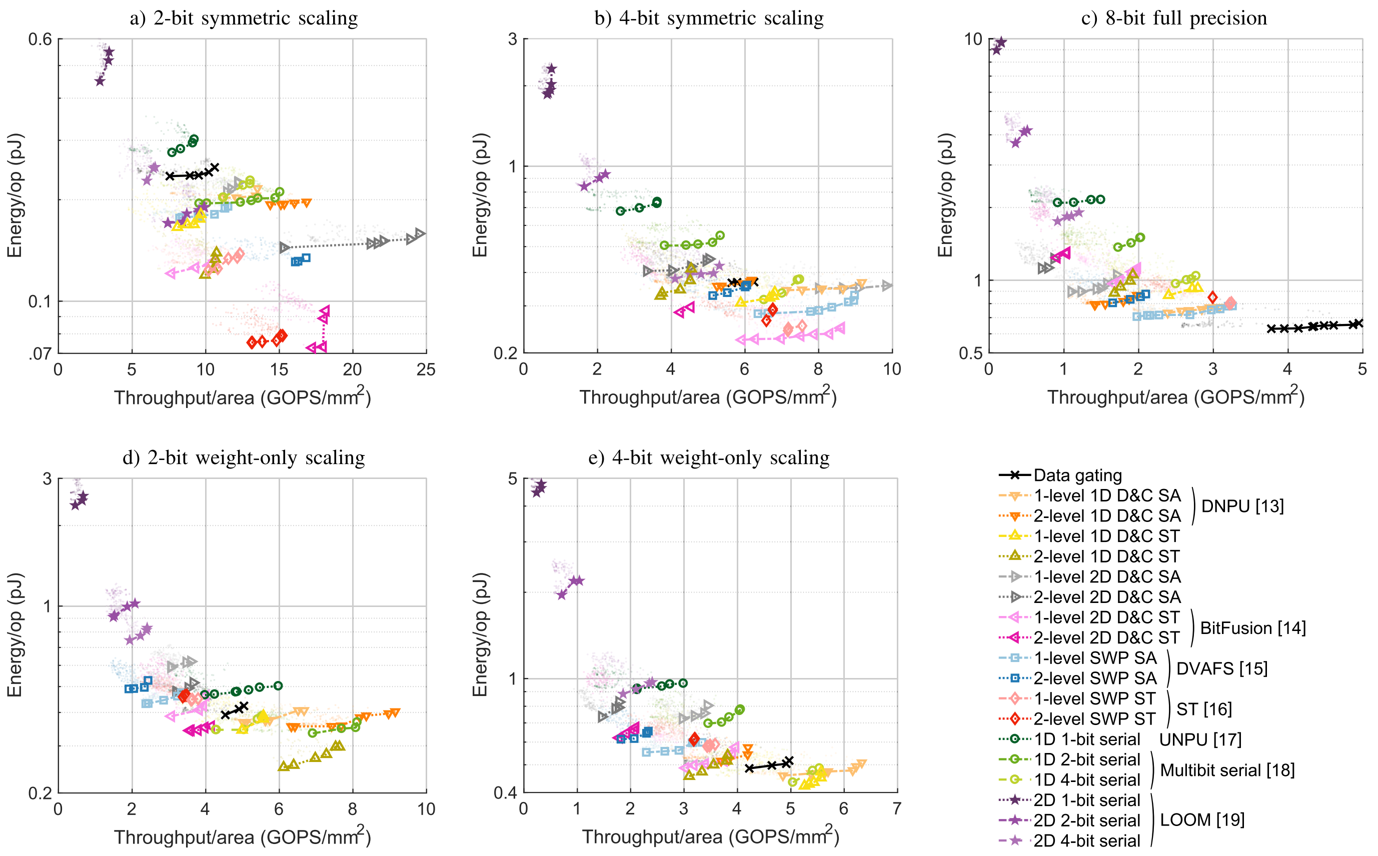


Fig. 28. Comparison of MAC architectures synthesized in a 28 nm CMOS process at 1 V supply voltage in terms of energy/op and throughput/area.

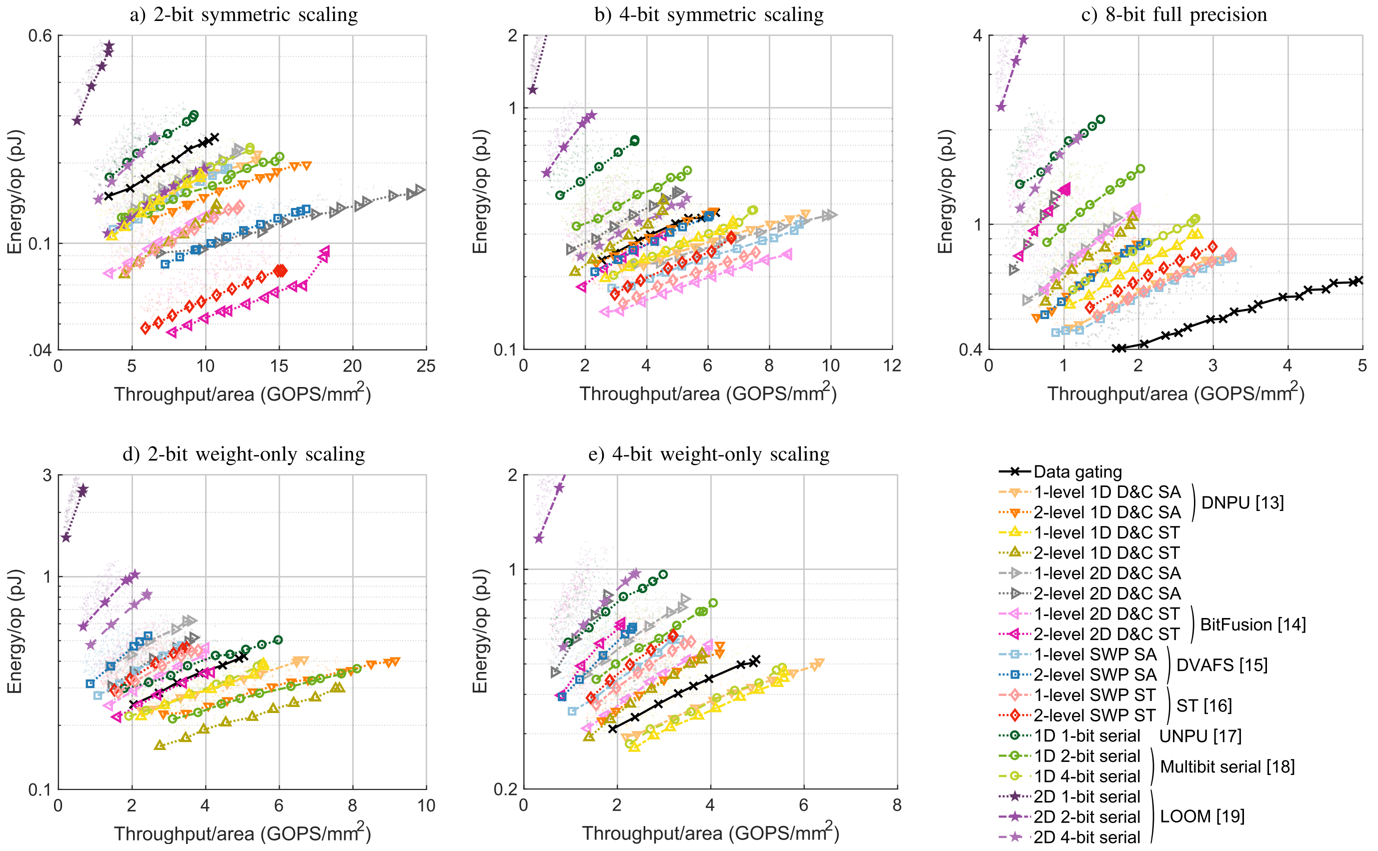


Fig. 29. Comparison of MAC architectures synthesized in a 28 nm CMOS with DVFS in terms of energy/op and throughput/area.

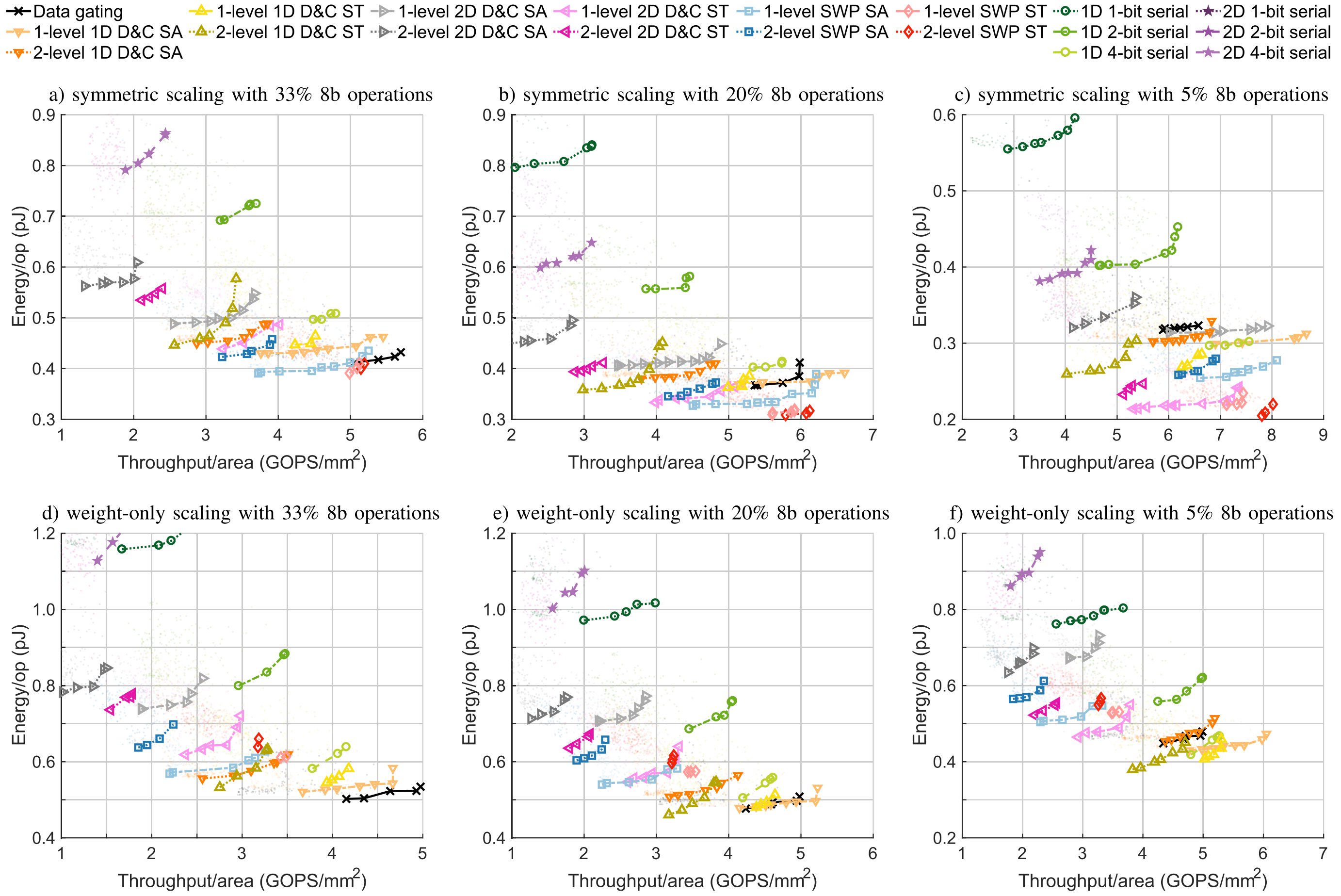


Fig. 30. Overall energy/op and throughput/area of MAC architectures utilized with 33%, 20% and 5% of 8b computations.

### D. 5% Full-Precision Computations

Across real-world DNNs, some low-complexity applications can use even less full-precision operations [14], e.g. only around 5%. This 8b-operation ratio is used for the third use case (right subfigures). At that proportion of 8b operations, SWP ST and 2D D&C ST largely outperform data gating for symmetric scenarios by 32% and 22%, respectively.

Eventually, some scalable MAC units are now beneficial for weight-only scaling scenarios, but energy gains are lower, at most 15% for 2-level 1D D&C ST, followed by 1-level 1D D&C ST and 1D 4-bit serial MACs (8%).

While 1D D&C SA is consistently equivalent to data gating in energy for both symmetric and weight-only scenarios, it is the best architecture with respect to throughput per area, bringing 10% to 30% higher performance than data gating.

Even at that level, leaving 95% operations shared by 2b and 4b precisions, no other MAC units can show benefit compared to a simple data-gated MAC. In particular, despite the recent trend for serial designs, all of them but the 1D 4-bit serial turn out inefficient in all scenarios.

These results confirm the importance for the designer to focus resources on what can bring the largest benefits rather than falling in the pitfall of implementing and optimizing to the lowest level, which can yield to limited or diminishing returns to the overall system. This strategy in the design trade-offs, recalling, at circuit level, Gene Amdahl's law [32], should be one of the most pervasive principles for designers.

## VIII. Conclusion

This work has introduced a new taxonomy of precision-scalable MAC architectures, categorizing them among multiple criteria: the type of unrolling (spatial or temporal), the dimensions they unroll (1D, 2D or 2D symmetric) and, for spatial-based designs, their type of accumulation (Sum Together or Sum Apart). This new classification has not only categorized all existing architectures, but it has also uncovered new design patterns that can give designers array-level and algorithmic-level insights to choose the right type of processing elements for their system. Along with this taxonomy, an exhaustive survey of state-of-the-art and further architectures has been carried out in order to clearly understand their ground principles, features, connections, and differences.

A benchmarking and comparison of these architectures has then been conducted. Therein, all circuits have been implemented in a 28 nm commercial CMOS process across a wide range of performance targets, with precision ranging from 2 to 8 bits. This study has thoroughly analyzed, theoretically and numerically, the different scalable MAC units in terms of energy, throughput, bandwidth and area, aiming to understand the key trends to reduce computation costs in neural-network processing via precision scaling.

The results of this comparative study have highlighted that 2D D&C ST (BitFusion) [14] and SWP ST (ST) [16] have the highest energy efficiency for symmetric scaling scenarios, while 1D D&C ST and 1D 4-bit serial [18] are best for weight-scaling scenarios. In addition to that, 1D D&C SA (DNPU) [13] and 2D D&C SA exceed with high throughput for all scaling scenarios, but suffer together with 2D D&C ST (BitFusion) from large varying bandwidth requirements. Despite the recent trend for 1D [17], [18] and 2D [19], [20] serial designs, these are strongly penalized for both throughput and energy efficiency.

This comparison has been concluded by an exploration of three practical case studies of usage under fixed proportions of operations at each precision, revealing the large impact of full-precision computations, even at a small proportion, in the overall energy budget. For symmetrical scaling scenarios, although being more efficient on low-precision computations individually, 2D D&C ST (BitFusion) is overtaken by SWP ST (ST) in any use case due to its higher overhead in full-precision mode. For weight-only scenarios however, precision-configurable designs are found to be mostly ineffective, where 2-level 1D D&C ST becomes beneficial only at a very low proportion of full-precision operations.

The rise of deep-learning applications, which have proven strongly resilient to the use of scaled precisions, has induced a monumental trend and a shift towards this new research direction. Among the blossom of countless techniques and designs related to deep-learning processing, this work has summarized and clarified the advances on precision-scalable MAC circuits in order to bridge the gap up to the software community and bring embedded neural-network processing to an adulthood phase. Future research will incorporate these MAC-level understandings at higher levels of abstraction in order to eventually build up to the application level.

## References

[1] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 2082–2090.

[2] J. Yu, A. Lukefahr, D. Palframan, G. Dasika, R. Das, and S. Mahlke, "Scalpel: Customizing dnn pruning to the underlying hardware parallelism," *ACM SIGARCH Comput. Archit. News*, vol. 45, no. 2, pp. 548–560, 2017.

[3] X. Yang *et al.*, "DNN dataflow choice is overrated," Sep. 2018, *arXiv:1809.04070*. [Online]. Available: https://arxiv.org/abs/1809.04070

[4] H. Kwon, P. Chatarasi, M. Pellauer, V. Sarkar, and T. Krishna, "A data-centric approach for modeling and estimating efficiency of dataflows for accelerator design," May 2018, *arXiv:1805.02566*. [Online]. Available: https://arxiv.org/abs/1805.02566

[5] A. Parashar *et al.*, "Timeloop: A systematic approach to DNN accelerator evaluation," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw. (ISPASS)*, Mar. 2019, pp. 304–315.

[6] V. Camus, M. Cacciotti, J. Schlachter, and C. Enz, "Design of approximate circuits by fabrication of false timing paths: The carry cut-back adder," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 8, no. 4, pp. 746–757, Dec. 2018.

[7] H. Jiang, C. Liu, L. Liu, F. Lombardi, and J. Han, "A review, classification, and comparative evaluation of approximate arithmetic circuits," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 13, no. 4, pp. 60:1–60:34, Aug. 2017.

[8] L. Yang and B. Murmann, "SRAM voltage scaling for energy-efficient convolutional neural networks," in *Proc. 18th Int. Symp. Qual. Electron. Design (ISQED)*, Mar. 2017, pp. 7–12.

[9] I. Hubara *et al.*, "Quantized neural networks: Training neural networks with low precision weights and activations," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6869–6898, Jan. 2017. [Online]. Available: https://dl.acm.org/citation.cfm?id=3122009.3242044

[10] L. Cavigelli and L. Benini, "Origami: A 803-GOp/s/W convolutional network accelerator," in *Proc. IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, Nov. 2017, pp. 2461–2475.

[11] B. Moons, B. De Brabandere, L. Van Gool, and M. Verhelst, "Energy-efficient ConvNets through approximate computing," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–8.

[12] C. Sakr and N. Shanbhag, "An analytical method to determine minimum per-layer precision of deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 1090–1094.

[13] D. Shin, J. Lee, J. Lee, and H.-J. Yoo, "DNPU: An 8.1TOPS/W reconfigurable CNN-RNN processor for general-purpose deep neural networks," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 240–241.

[14] H. Sharma *et al.*, "Bit fusion: Bit-level dynamically composable architecture for accelerating deep neural networks," in *Proc. 45th IEEE Int. Symp. Comput. Archit. (ISCA)*, Jun. 2018, pp. 764–775.

[15] B. Moons, R. Uytterhoeven, W. Dehaene, and M. Verhelst, "DVAFS: Trading computational accuracy for energy through dynamic-voltage-accuracy-frequency-scaling," in *Proc. Desing Automat. Test Eur. Conf. Exhib. (DATE)*, Mar. 2017, pp. 488–493.

[16] L. Mei *et al.*, "Sub-word parallel precision-scalable MAC engines for efficient embedded DNN inference," in *Proc. IEEE 1st Int. Conf. Artif. Intell. Circuits Syst. (AICAS)*, Mar. 2019, pp. 6–10.

[17] J. Lee, C. Kim, S. Kang, D. Shin, S. Kim, and H.-J. Yoo, "UNPU: A 50.6TOPS/W unified deep neural network accelerator with 1b-to-16b fully-variable weight bit-precision," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2018, pp. 218–220.

[18] V. Camus, C. Enz, and M. Verhelst, "Survey of precision-scalable multiply-accumulate units for neural-network processing," in *Proc. IEEE 1st Int. Conf. Artif. Intell. Circuits Syst. (AICAS)*, Mar. 2019, pp. 57–61.

[19] S. Sharify, A. D. Lascorz, K. Siu, P. Judd, and A. Moshovos, "Loom: Exploiting weight and activation precisions to accelerate convolutional neural networks," in *Proc. ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2018, pp. 20:1–20:6.

[20] K. Ueyoshi *et al.*, "QUEST: A 7.49 TOPS multi-purpose log-quantized DNN inference engine stacked on 96 MB 3D SRAM using inductive-coupling technology in 40 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2018, pp. 216–218.

[21] S. Ryu, H. Kim, W. Yi, and J.-J. Kim, "BitBlade: Area and energy-efficient precision-scalable neural network accelerator with bitwise summation," in *Proc. 56th Annu. Design Autom. Conf. (DAC)*, 2019, pp. 84:1–84:6.

[22] Z. Du *et al.*, "ShiDianNao: Shifting vision processing closer to the sensor," in *Proc. ACM/IEEE Int. Symp. Comput. Archit. (ISCA)*, Jun. 2015, pp. 92–104.

[23] B. Moons, R. Uytterhoeven, W. Dehaene, and M. Verhelst, "Envision: A 0.26-to-10 TOPS/W subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28 nm FDSOI," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 246–247.

[24] N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," *ACM SIGARCH Comput. Archit. News*, vol. 45, no. 2, pp. 1–12, 2017.

[25] Y. H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017.

[26] T. Chen *et al.*, "DianNao: A small-footprint high-throughput accelerator for ubiquitous machine-learning," *ACM SIGPLAN Notices*, vol. 49, no. 4, pp. 269–284, Feb. 2014.

[27] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing FPGA-based accelerator design for deep convolutional neural networks," in *Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays*, 2015, pp. 161–170.

[28] J. Qiu *et al.*, "Going deeper with embedded FPGA platform for convolutional neural network," in *Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays (FPGA)*, 2016, pp. 26–35.

[29] M. Song, J. Zhang, H. Chen, and T. Li, "Towards efficient microarchitectural design for accelerating unsupervised GAN-based deep learning," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2018, pp. 66–77.

[30] B. Moons and M. Verhelst, "A 0.3–2.6 TOPS/W precision-scalable processor for real-time large-scale ConvNets," in *Proc. IEEE Symp. VLSI Circuits (VLSI-Circuits)*, Jun. 2016, pp. 1–2.

[31] T. Sakurai and A. R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, Apr. 1990.

[32] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," in *Proc. AFIPS Spring Joint Comput. Conf.*, 1967, pp. 483–485.

**Vincent Camus** (S'15–M'19) received the Ph.D. degree from the Swiss Federal Institute of Technology in Lausanne (EPFL), Switzerland, in 2019.

He joined the Integrated Circuits Laboratory (ICLAB), EPFL, in 2013. In 2018, he was a Visiting Scholar with the ESAT-MICAS Laboratory, KU Leuven, Leuven, Belgium. He is currently a Research and Development Digital-Design and Software Engineer with The Swatch Group Research and Development Ltd., Marin, Switzerland. He has co-authored about 20 scientific articles or patents and has contributed to the organization of several conferences and public scientific events. His research interests include low-power digital circuits, electronic design automation, error-resilient and precision-scalable computing, and energy-efficient embedded deep neural networks.

**Linyan Mei** (S'18) received the B.Sc. degree in electronic science and technology from the Beijing Institute of Technology (BIT), China, in 2016, and the M.Sc. degree in electrical engineering from KU Leuven, Belgium, in 2018. She is currently pursuing the Ph.D. degree in the domain of digital design for embedded machine-learning processors with the ESAT-MICAS Laboratory, KU Leuven.

She was an Intern with imec, Leuven, Belgium, during her second year M.Sc. studies. Her research interests include energy-efficient deep neural network accelerators, algorithm-hardware co-design, and precision-scalable computing.

**Christian Enz** (S'83–M'84–SM'11–F'19) received the Ph.D. degree from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 1989.

Until 2013, he was the Vice President of the Swiss Center for Electronics and Microtechnology (CSEM) in Neuchâtel, Switzerland, where he was the Head of the Integrated and Wireless Systems Division. Prior to joining CSEM, he was a Principal Senior Engineer at Conexant (formerly Rockwell Semiconductor Systems), Newport Beach, CA, where he was responsible for the modeling and characterization of MOS transistors for RF applications. He is currently a Professor with EPFL, where he is also the Director of the Institute of Microengineering and the Head of the Integrated Circuits Laboratory. His technical expertise is in low-power analog and RF integrated circuits, wireless sensor networks, and semiconductor device modeling. He is the developer of the EKV MOS transistor model. He has co-authored more than 250 scientific articles and has contributed to numerous conference presentations. He is an individual member of the Swiss Academy of Engineering Sciences. He was an elected member of the IEEE Solid-State Circuits Society (SSCS) AdCom from 2012 to 2014. He is the Chair of the IEEE SSCS Chapter of Switzerland.

**Marian Verhelst** (SM'13) received the Ph.D. degree from KU Leuven, Belgium, in 2008.

She was a Visiting Scholar with the Berkeley Wireless Research Center of the University of California Berkeley in the summer of 2005, and was a Research Scientist with Intel Labs, Hillsboro, OR, USA, from 2008 until 2011. She is currently an Associate Professor with the MICAS Laboratories, Electrical Engineering Department, KU Leuven. Her research focuses on embedded machine learning, hardware accelerators, self-adaptive circuits and systems, sensor fusion, and low-power edge processing. She is a member of the DATE and ISSCC executive committees, is the TPC Co-Chair of AICAS 2020 and tinyML 2020, and a TPC member of DATE and ESSCIRC. She is an SSCS Distinguished Lecturer, was a member of the Young Academy of Belgium, a member of the STEM advisory committee to the Flemish Government, and an Associate Editor for TVLSI, TCAS-II, and JSSC. She currently holds a prestigious ERC Starting Grant from the European Union and was the laureate of the Royal Academy of Belgium in 2016.