2024 Design, Automation & Test in Europe Conference (DATE 2024) – Multi-Partner Project

FVLLMONTI: The 3D Neural Network Compute Cube (N^2C^2) Concept for Efficient Transformer Architectures Towards Speech-to-Speech Translation

Ian O'Connor[*], Sara Mannaa[*], Alberto Bosio[*], Bastien Deveautour[*], Damien Deleruyelle[*], Tetiana Obukhova[*], Cédric Marchand[*], Jens Trommer[†], Cigdem Cakirlar[†], Bruno Neckel Wesling[†], Thomas Mikolajick[†], Oskar Baumgartner[‡], Mischa Thesberg[‡], David Pirker[‡], Christoph Lenz[‡], Zlatan Stanojevic[‡], Markus Karner[‡], Guilhem Larrieu[§], Sylvain Pelloquin[§], Konstantinous Moustakas[§], Jonas Muller[§], Giovanni Ansaloni[¶], Alireza Amirshahi[¶], David Atienza[¶], Jean-Luc Rouas[||], Leila Ben Letaifa[||], Georgeta Bordea[||], Charles Brazier[||], Chhandak Mukherjee[**], Marina Deng[**], Yifan Wang[**], Marc Francois[**], Houssem Rezgui[**], Reveil Lucas[**], Cristell Maneux[**]

[*]*Lyon Institute of Nanotechnology* – Lyon, France
[†]*NaMLab gGmbH* – Dresden, Germany
[‡]*Global TCAD Solutions GmbH* – Vienna, Austria
[§]*LAAS-CNRS* – Toulouse, France
[¶]*Embedded Systems Laboratory (ESL), Ecole Polytechnique Fédérale de Lausanne (EPFL)* – Lausanne, Switzerland
[||]*LaBRI CNRS UMR 5800, Univ. Bordeaux* – Talence, France
[**]*IMS, University of Bordeaux, Bordeaux INP, CNRS UMR 5218* – Talence, France

cristell.maneux@u-bordeaux.fr

***Abstract*—This multi-partner-project contribution introduces the midway results of the Horizon 2020 FVLLMONTI project. In this project we develop a new and ultra-efficient class of ANN accelerators, the neural network compute cube (N^2C^2), which is specifically designed to execute complex machine learning tasks in a 3D technology, in order to provide the high computing power and ultra-high efficiency needed for future edgeAI applications. We showcase its effectiveness by targeting the challenging class of Transformer ANNs, tailored for Automatic Speech Recognition and Machine Translation, the two fundamental components of speech-to-speech translation. To gain the full benefit of the accelerator design, we develop disruptive vertical transistor technologies and execute design-technology-co-optimization (DTCO) loops from single device, to cell and compute cube level. Further, a hardware-software-co-optimization is executed, e.g. by compressing the executed speech recognition and translation models for energy efficient executing without substantial loss in precision.**

***Index Terms*—ANN, translation, DTCO, emerging technologies**

I. Introduction

Ever since the invention of the integrated circuit [1], digitalization has massively changed our way of living. Today, smart electronic gadgets assist our daily life seamlessly by the continuous collection and processing of data in our environment. This step is the entry point for the next industrial revolution, where new technologies such as artificial intelligence (AI), robotics, the metaverse, or the internet-of-things (IoT) will immersively blur the lines between the physical, virtual, and even biological worlds. As part of this vision, the FVLLMONTI project (Ferroelectric Vertical Low energy Low latency low volume modules FoR Neural network Transformers In 3D) pursues the implementation of direct speech-to-speech translation embedded in a lightweight in-ear-device. Such modern AI applications are nowadays powered by machine learning techniques, in particular artifical neural networks (ANNs) [2]. However, as traditional computer architectures cannot handle AI tasks efficiently, it is clear that such EdgeAI devices need to operate under disruptively different computational paradigms, as they also have to fulfil the boundary conditions set by battery capacity and heat dissipation. Dedicated heterogenous computing architectures have to be developed, which increase throughput and lower power dissipation. The main source of power dissipation in AI applications is data movement. The transfer of data between the processor and the memory requires several orders of magnitude more energy and time compared to a single compute operation [3]. This problem is called 'von-Neumann-bottleneck' as it is closely related to the von-Neumann paradigm, the foundation of all modern computing systems. [4]

In this work, we present a new and ultra-efficient class of ANN accelerators, the neural network compute cube (N^2C^2), which is specifically designed to execute complex machine learning tasks in a 3D technology, in order to provide the high computing power and ultra-high efficiency needed for future edgeAI applications. We showcase its effectiveness by targeting the challenging class of Transformer ANNs [5], tailored for Automatic Speech Recognition (ASR) and Machine Translation (MT), the two fundamental components of speech-to-speech translation.

This work was funded by the European Union's Horizon 2020 research and innovation program under grant agreement No 101016776 (FVLLMONTI).

II. The Neural Network Compute Cube

A. N^2C^2 as a functional block

The principal function of the N^2C^2 is to carry out element-wise non-volatile matrix multiplication, accumulation and activation through a non-linear function. It features multiple means of configuration:

- Firstly, it is function-configurable. As a baseline operation, we define a 32-bit integer multiply-accumulate function (MAC) which can also be broken down into its individual operations (multiply, addition, accumulation and combinations of these). We also include mechanisms to efficiently program an activation function (e.g. sigmoid, tanh, rectified linear – ReLU, softplus . . .) that can be switched in and out of the datapath.
- It is connectivity-configurable, meaning that it is possible to input from 2-8 operands as number of inputs to each cell. Further, it is compatible with routing resources outside of the N^2C^2 in order to (for example) handle feedback in recursive networks, or to configure the vertical routing of data between layers in both directions.
- It is coefficient-configurable, meaning that it is possible to program synaptic coefficients in memory elements and connect them to the multiplier function blocks.
- It is datawidth-configurable, in that it is possible to implement both intra-N^2C^2 scaledown from 1*32 bits to 2*16 bits, 4*8 bits or 8*4 bits; and that, ultimately, it will also be possible to handle inter-N^2C^2 scaleup to 64 bits, 128 bits, 256 bits, 512 bits.

The conceptual view of the N^2C^2 is shown in figure 1(a). Each N^2C^2 is intended for use in a versatile 3D architectural model, as depicted in figure 1(b). The target accelerator architecture is composed of an array of interconnected N^2C^2 blocks to make an efficient hardware implementation of functions used heavily in transformer-based neural networks. Figure 1(c) depicts the internal schematic view of the N^2C^2 block.

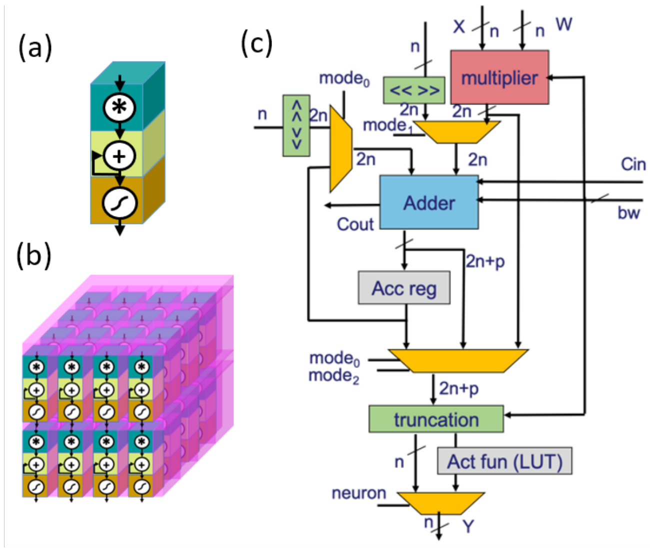


Fig. 1. Neural network compute cube (N^2C^2). (a) Conceptual view. (b) Use in a 3D accelerator. (c) Internal schematic structure.

B. N^2C^2 accelerator in a Systolic Array configuration

In this section, we describe the use of the N^2C^2 block in the context of an accelerator organized in a Systolic Array architecture. Each N^2C^2 has three data inputs: X, W and Y_{k-1}. The n-bit inputs X and W correspond to the input data and the weight value respectively. The datawidth n is set to 8 in the current implementation for both inputs. Y_{k-1} represents the data coming from the preceding N^2C^2 and the $(2n + p)$ datawidth is set in the current implementation to 20 bits, where p=4 represents the number of guard bits needed to manage overflow due to accumulation operations. Here, Y_{k-1} represents a generic "preceding" N^2C^2 and is in practice related to an actual block in a 2D or 3D matrix such as $Y_{x,y-1,z}$ considering a classical naming convention for cells in a 3D structure.

A systolic array architecture $N^2C^2 - SA$, as depicted in figure 2 as a 4-bit example, is used in two phases:

1) Weights programming: when weight W_{WR} is high, the weights programming phase begins. A shift register activates the enable signal of the N^2C^2 weight registers in a given order so each N^2C^2 is programmed with its correct weight.
2) Execution (after weights programming): When Wr_{data_in} is high, the operands are dispatched through the rows (as previously explained in terms of N^2C^2-N^2C^2 data flow).

During execution, the first data to be streamed from one N^2C^2 to another are the operands (A). This data passes between N^2C^2s from left to right (i.e. with incrementing column coordinate x), from the first column (x=0) until it reaches the last column (x=X-1) of the array. This operand remains the same for every N^2C^2 on the same row (i.e. with identical row coordinate y).

The other data to be streamed is the MAC computation that flows through columns. One N^2C^2 implements a local multiplication of its weight (W) and its incoming operand (X) before adding the result to the computation result coming from the preceding N^2C^2 ($Y_{k-1} = Y_{x,y-1}$ in a 2D column-connected array). The result ($Y_k = Y_{x,y}$) is then downstreamed to the N^2C^2 below and added to the local multiplication ($Y_{k+1} = Y_{x,y+1}$) until the last row is reached ($Y_{x,Y-1}$). When this happens, the data is truncated back to 8-bit precision and concatenated to 3 other 8-bit data coming from neighboring columns.

III. From Vertical Transistors to 3D Libraries

While the N^2C^2 concept itself can also be implemented in planar technologies, it was specifically designed to excel in a 3D technology. To evaluate the 3D N^2C^2 hardware, it is necessary to build a circuit-scale logic cell library based on multiple, interconnected vertical devices. Development objectives here are twofold - one leading to circuit-scale hardware demonstrations, in order to prove the viability of the technology; and another to develop flexible physical design strategies and tools to pursue realistic technology projection through parasitic extraction on actual physical designed cells. This enables more refined performance estimation of the N^2C^2 block for injection into architectural scale explorations. In particular, the N^2C^2 block is described in a synthesizable format to be used with a library file, and can be targeted to

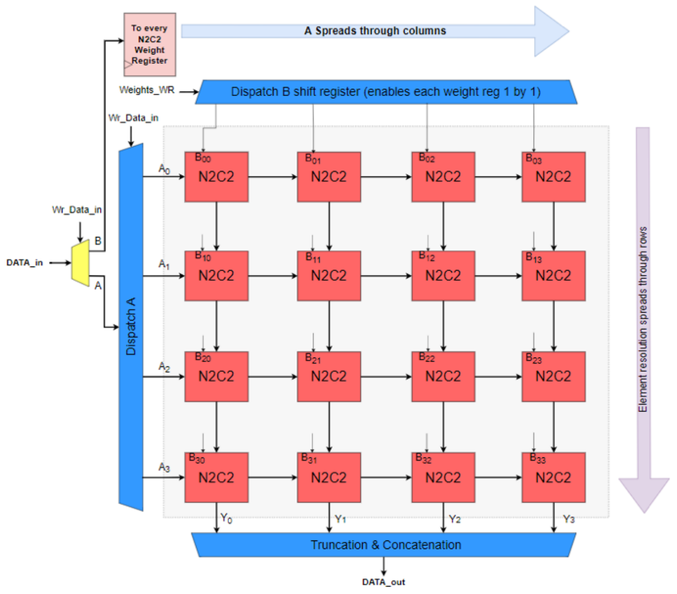


Fig. 2. Example 4x4 $N^2C^2 - SA$ and its interconnections

any data width. In a first approach, we target a 4-bit N^2C^2, which uses 230 logic cells. The logic cell library itself is based, again in a first approach, on single-gate stack vertical junctionless nanowire transistors (JLNTs), but can be adapted to more disruptive technologies, later in the development cycle.

A. Vertical Junctionless FETs

Vertical gate-all-around (GAA) technology represents a significant evolution from traditional planar and FinFET architectures. In vertical GAA designs, the gate material completely surrounds the nanowire channel, forming a cylindrical control region around the vertical silicon nanowires [6]. This architecture offers multiple advantages for device design and integration. First, the vertical orientation of the nanowires allows for denser device packing on a chip, maximizing the use of available surface area. This makes vertical GAA technology highly scalable, providing a pathway to increase transistor density without expanding the chip footprint. Second, the gate-all-around structure ensures superior electrostatic control of the channel, leading to reduced leakage currents and lower power consumption. This is particularly beneficial for mobile and high-performance computing applications where energy efficiency is critical. Additionally, a junctionless configuration in vertical process technology, such as in silicon nanowires, offers a simpler and more efficient design compared to traditional junction-based transistors. In this design, the absence of highly doped junctions eliminates the need for complex doping profiles and high-temperature diffusion processes, thereby streamlining the manufacturing workflow. The current flows through the core of the nanowire, which is less susceptible to surface defects typically occurring at the gate oxide interface. This core conduction path ensures more stable and reliable device performance, as it is shielded from surface scattering and traps that can affect carrier mobility. Vertical GAA transistors also exhibit better immunity to short channel effects due to improved gate control. This allows for further scaling down of

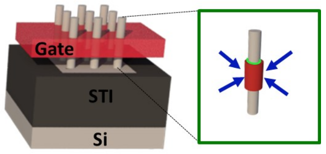


Fig. 3. Vertical gate-all-around junctionless nanowire transistor (JLNT).

the gate length, pushing the limits of transistor miniaturization while maintaining performance.

B. Design-Technology-Co-Optimization (DTCO) Loops

Advanced 3D technologies show an especially strong interaction between individual components, making accurate predictive simulations important for DTCO. The DTCO simulation flow applied here relies on an accurate extraction of the parasitic network (PEX) from a detailed 3D TCAD model of the logic cells built from layout and technology information [7]. Measurement data combined with insight from TCAD are key enablers of the compact model completing the description of the key components. The automatically extracted PEX netlists are combined with the compact models of the vertical GAA JLNTs to run transient simulations of 3D logic cells enabling power, performance, and area analysis. Fig. 4 details the information flow template for the three increasing DTCO loops going from device to cell to compute cube level.

The first DTCO loop aims to optimize individual devices (i.e., transistors). Device dimensions of modern technologies like vertical GAA JLNTs have a huge impact on the device characteristics (e.g., random discrete dopants, metal gate granularity, oxide defects, spacers, gate length or geometrical variations). Additionally, device variability linked to well known physical phenomena increases. Therefore, TCAD simulations are utilized to optimize the devices without conducting fast experimental studies of device characteristics [8].

In the second DTCO loop, circuits of several devices (i.e., cells) are investigated (e.g., INV1, XOR2, . . .). These cells introduce additional complications into the design process (e.g., coupling capacitance, metal contacts, layout issues). Again TCAD simulations are utilized to avoid pre-PDK evaluations of the technology [9].

The final DTCO loop utilizes the generated cells to construct a physical representation of an electric circuits (e.g. adders, multipliers, functional blocks of the N^2C^2, . . .). This physical representation is then further used to conduct timing and energy/power analysis. Furthermore, the wires that are generated during this process introduce an additional level of complexity into the process (e.g., interconnect parasitics). Depending on the resulting parasitics, affected wires have to be rerouted.

C. Compact Modelling, Thermal Modelling

To ensure design compatibility between the compact model and the DTCO flow as well as associated circuit simulation tools, the model specifically developed for the JLNT technology in the FVLLMONTI project, based on the understanding of carrier transport studied using numerical multiphysics and TCAD simulations, is written in Verilog-A [10]. Following model calibration against experimental data at room temperature, the compact model is modified to take into account

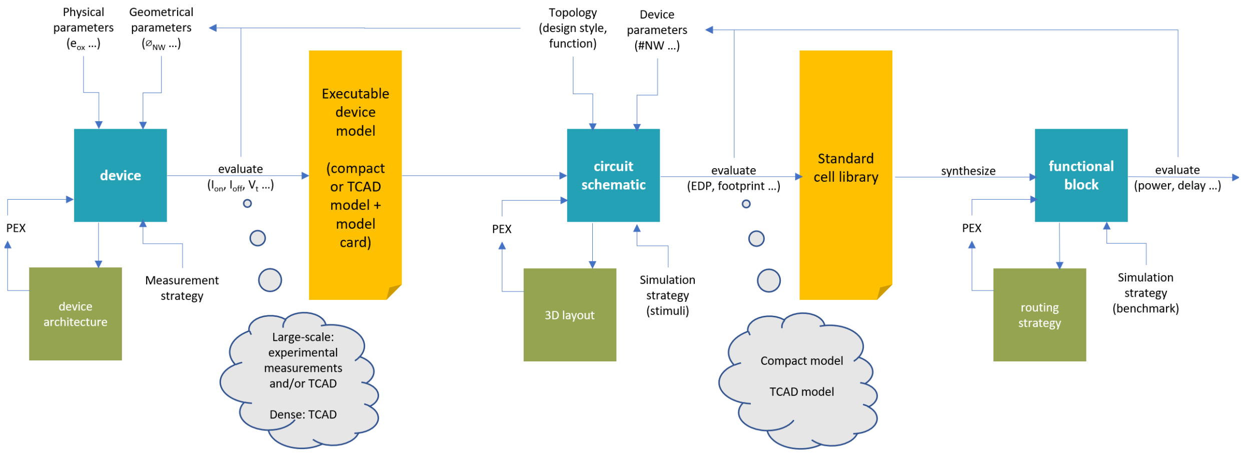

Physical parameters (e_{ox} ...)
Geometrical parameters ($\varnothing_{NW}$ ...)
device
evaluate (I_{on}, I_{off}, V_t ...)
Executable device model (compact or TCAD model + model card)
PEX
device architecture
Measurement strategy
Large-scale: experimental measurements and/or TCAD
Dense: TCAD
Topology (design style, function)
Device parameters (#NW ...)
circuit schematic
evaluate (EDP, footprint ...)
Standard cell library
PEX
3D layout
Simulation strategy (stimuli)
Compact model
TCAD model
synthesize
functional block
evaluate (power, delay ...)
PEX
routing strategy
Simulation strategy (benchmark)

Fig. 4. DTCO flow illustrating the three optimization loops. Device loop, circuit schematic loop and functional block loop. Each loop expands the scope and complexity of the examined configuration, starting with single devices over logic cells up to entire circuits (compute cubes).

electrothermal effects, one of the critical issues in nano-scale junctionless transistors that can potentially impact dynamic circuit performance. For this, on-wafer DC measurements over a wide temperature range are exploited that reveal a linear increase of drain current with temperature in junctionless transistors [11], [12]. This has been explained by a strong temperature dependence of the threshold voltage in junctionless devices, which is further pronounced by a weak temperature dependence of the mobility [11]. Multiphysics simulations [13] further revealed that the thermal conductivity in JLNTs reduces significantly with temperature, thus creating temperature hot-spots leading to possible device self-heating.

To translate these effects into the existing compact model, we first modified the equation of the threshold voltage, which was obtained following a linear temperature dependence extracted from the experimental data. In addition, temperature coefficients of other relevant model parameters such as the Schottky barrier height, series access resistances and intrinsic carrier concentration were also implemented in the model equations. Lastly, an equivalent electrical sub-circuit consisting of the thermal resistance and capacitance, extracted from low-frequency S-parameter measurements [12], was added to the compact model, in order to capture the dynamic self-heating effects.

D. *Vertical Logic Cell Library*

Using the compact model, conventional static logic cells are characterized in terms of worst case transition delay and worst case energy per transition for varying numbers of nanowires per transistor (and consequent logic cell drive strength). The measured Pareto fronts are shown in Fig. 5 for the four most important static cells. Finally, complete cell layouts are generated semi-automatically from cell templates and applying design rules adapted from ASAP 7nm FinFET PDK. An example is shown for a 2-input XOR cell using 8 nanowires per device (schematic, layout, 3D view) in Fig. 6.

IV. Future Technology Variants

The derived static library discussed here is still based on a single layer of JLNTs. More compact designs are expected for stacking of multiple vertical transistors within a single cell.

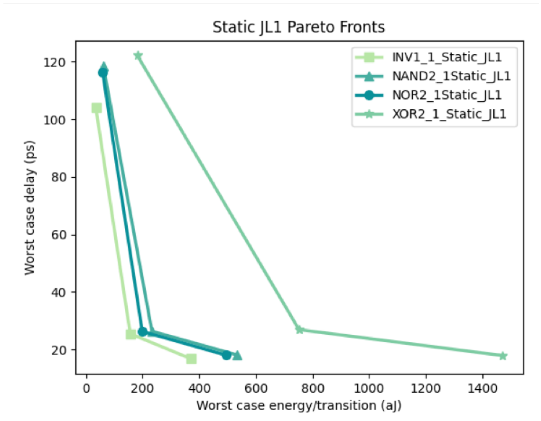


Fig. 5. Pareto fronts for conventional static logic cells based on single-gate stack vertical junctionless nanowire FET technology in terms of worst case transition delay vs energy per transition.

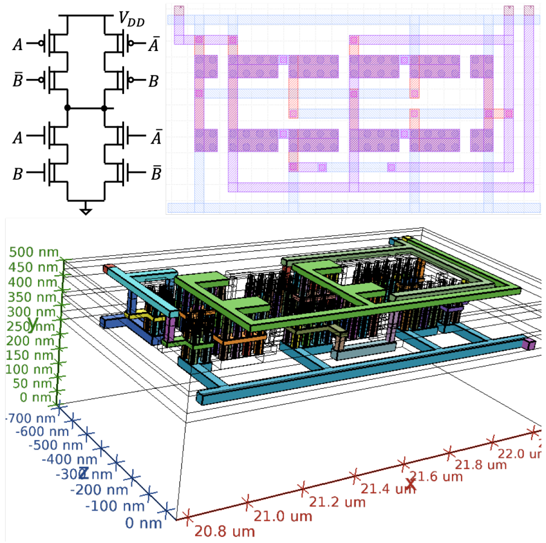


Fig. 6. Schematic, layout, 3D view of a 2-input XOR2 cell of 8 JLNTs.

Further, non-volatile logic or even purely analog logic styles have a high potential to reduce footprint and power consumption even further [14]. These disruptive variants would have to be enabled by an integration of emerging device concepts into our vertical platform. However, the added benefit also comes with new constraints added to the design of the N^2C^2 hardware. We will discuss this briefly for two emerging device options ferroelectric transistors, and reconfigurable transistors.

A. Ferroelectric Gate Integration

Ferroelectric hafnia is among the premium materials under consideration for seamless non-volatile memory integration into an exisiting transistor technology as HfO_2 is already widely used as gate dielectric material in scaled process nodes [14]. By simply adding doping or strain to these gate layers, an internal remanent polarization is added to the gate, which can be reprogrammed between two stable states [15]. The state can be changed dynamically by applying a program voltage pulse having an about 2-3 times higher voltage as the reference V_{DD}. Most interestingly, after programming the state, a volatile input can be superimposed at the same gate, thus compacting designs of basic logic units. Indeed, it is possible to program ferroelectric transistors to the following states:

- Always on / Regular switch: '1' $\leq V_{TFL}$ and '0' $\leq V_{TFH}$ '1' – type "+" (OR)
- Regular switch / ALWAYS OFF: '0' $\leq V_{TFL} \leq$ '1' and '0' V_{TFH} – type "•" (AND)

where V_{TFL} (resp. V_{TFH}) represents the low (resp. high) ferroelectric shifted threshold voltage. This is utilized in figure 7 for a non-volatile (NV) NAND-gate built from junctionless transistors. In NV logic, single gate devices represent two sequential inputs – one non-volatile destined to remain stored for a considerable length of time, one volatile destined to change at every compute cycle. Here, we denote the non-volatile input with an underscore.

In this way, ferroelectric transistors can be used to perform MAC operations in a very compact analog manner, potentially further improving N^2C^2 performance and power consumption. However, to yield the best circuit results, a variety of device metrics such as memory window, endurance, and retention have to be carefully co-optimized with respect to the target circuit applications. For example a high coercive field is needed to generate a large memory window, while a too high coercive field on the other hand widens the electric field distribution needed to induce polarization switching and thus degrades endurance [14]. Thereto, a specialized DTCO procedure is again needed out to ensure optimized results.

B. Vertical Reconfigurable Transistors

Reconfigurable field effect transistors (RFETs), are an emerging type of device combining the functionality of p- and n-type FETs in a single transistors [16]. RFETs are built on doping free-channels, exploiting electrostatics for operation. They can be utilized on the circuit level to perform a higher number of functions with a reduced number of elements. Especially interesting with respect to the N^2C^2 design is that multiple functionalities, like NAND, NOR, XOR and MUX, can be mapped onto the exact same basic circuit layout [17]. Being doping-free, this concept allows for a

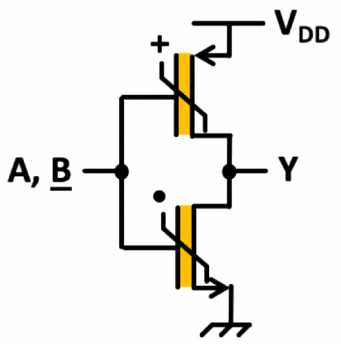


Fig. 7. Ferroelectric NV-NAND circuit

stacking of multiple functional blocks on top of each other if built in a true-vertical arrangement. This has the potential to combine the flexibility of logic design with the high layout regularity of a memory array, albeit coming on the cost of a lower individual device performance as compared to the JLNT counterpart. The other option for RFETs is to serve as an add-on functionality if seamlessly co-integrated within a another base technology. For example, the dynamic reconfiguration can be used for signal routing purposes, while computation itself is carried out on cells built from JLNTs. In [18] we have introduced the disruptively new concept of common-channel RFET, in which multiple individual source contacts can be routed directly to a single drain. The current output at the drain is the linear summation of the source currents. This feature is both interesting for flexible signal routing, as well as direct analog data processing in future design variants of the N^2C^2 concept.

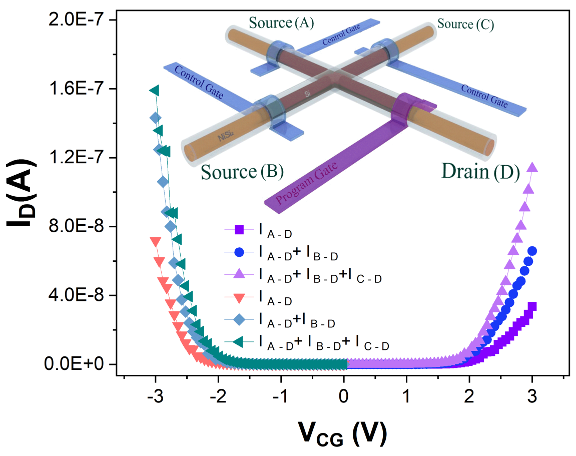


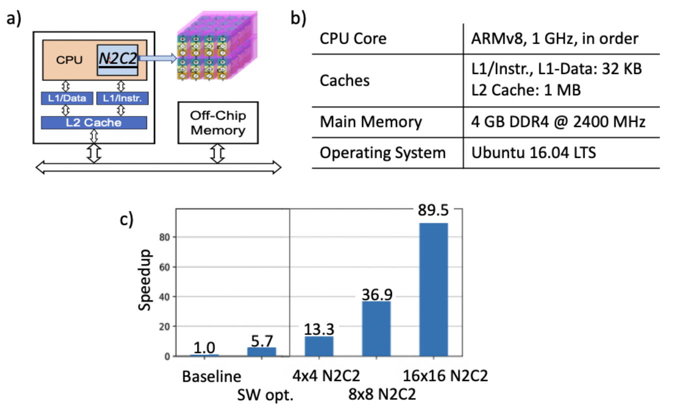
Fig. 8. Structure and transfer characteristics of a cross-shape reconfigurable field effect transistor [18] illustrating analog current summation within a single device.

V. Performance Benchmark and Application

A. Full system simulation

To fully assess the potential of N^2C^2 accelerators, we modelled them as parts of entire systems, measuring the ensuing benefits when executing Transformer inferences. To this end, we extended the gem5-X system simulation framework [19], [20] to include a novel architectural component characterizing the N^2C^2 behaviour [21]. This component was then interfaced as a dedicated functional unit, tightly coupled to processor pipelines (Fig. 9). N^2C^2 functional units are governed by custom extensions to the ARM instruction set architecture, which allow software applications to enqueue inputs, dequeue outputs and start N^2C^2 computations. The effectiveness of N^2C^2 in accelerating benchmark Transformer inferences has been tested on realistic system architectures

Obtained speedups for the BERT-Large Transformer [5] are reported in Figure 9c.) Therein, a baseline naïve implementation is assumed as a baseline, and compared with a fully optimized software implementation (*SW. opt.*) and N^2C^2-accelerated systems with arrays of increasing size ($N \times N$ N^2C^2). Results highlight that even a very small 4×4 N^2C^2 is able to achieve a speed up factor of 2X with respect to software optimization alone. The motivation for this outcome is twofold. First, N^2C^2s can process Matrix-Matrix Multiplications (MMMs) in constant time among tiles of data, greatly reducing the computational cost of this computational pattern. In turn, MMMs account for more than 99% of the execution time in non-accelerated transformers (and, still, 97% of execution time in the *16×16* N^2C^2 case). Second, N^2C^2s can store operands in the accelerator fabric, lowering the "Memory Wall" by easing the pressure on the cache hierarchy, hence reducing misses and the ensuing CPU stalls.



CPU Core	ARMv8, 1 GHz, in order
Caches	L1/Instr., L1-Data: 32 KB L2 Cache: 1 MB
Main Memory	4 GB DDR4 @ 2400 MHz
Operating System	Ubuntu 16.04 LTS

Fig. 9. a) N^2C^2 interfaced as a tightly-coupled accelerator. b) Test system configuration. c) Inference run-time for BERT-Large [5], normalized to the baseline non-accelerated configuration.

B. Speech-to-Speech Translation

In order to be able to achieve efficient speech-to-speech translation on an autonomous ear-sized device, we need to develop and optimize neural network software models. We first identified a common framework for building Transformer Deep Neural Networks for both Automatic Speech Recognition (ASR) and Machine Translation (MT), the ESPNet toolkit [22]. Using this toolkit, we build models that are able to replicate state-of-the-art performances for both ASR and MT. We then studied compression techniques that can be easily applied without retraining, namely quantization and pruning, and evaluated their impact on the accuracy of the models [23]. Furthermore, after studying the weights distribution in the Transformer model in [24], we developed a new pruning method called *Variable Scale pruning* which aims at setting different pruning rates according to the depth of the considered layer in the network [25]. This method, combined with 8-bit quantization, enables to reach a compression rate of 59.5% with a degradation in relative accuracy of less than 10%.

VI. Conclusion

The FVLLMONTI project being at intermediate stage, we have introduced the concept of the neural network compute cube (N^2C^2), which is a functional unit to carry out element-wise non-volatile matrix multiplication, accumulation and activation through a non-linear function. We described an N^2C^2 accelerator organized in a Systolic array and provided insights into our DTCO and STCO approaches in order to built this system on a vertical nanowire 3D technology platform. We tailored the system towards automatic speech recognition and machine translation, the two fundamental components of speech-to-speech translation. The concept is ready to be adapted to facilitate non-classical computing paradigms enabled by emerging device technologies in the future.

References

[1] J. S. Kilby, US Patent 3,115,581, Dec. 1963.
[2] Y. LeCun et al., *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[3] E. Strubell et al., *arXiv preprint arXiv:1906.02243*, 2019.
[4] J. Backus et al., *Communications of the ACM*, vol. 21, no. 8, pp. 613–641, 1978.
[5] J. Devlin et al., *arXiv preprint arXiv:1810.04805*, 2018.
[6] G. Larrieu et al., *Solid-State Electronics*, vol. 130, pp. 9–14, 2017.
[7] C. Maneux et al., in *2021 IEEE International Electron Devices Meeting (IEDM)*, 2021, pp. 15–6.
[8] G. Rzepa et al. in *2022* IEEE *International Electron Devices Meeting (IEDM)*, 2022, pp. 15.1.1–15.1.4.
[9] S. Mannaa et al., *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits,* pp. 1–1, 2023.
[10] C. Mukherjee et al, *Solid-State Electronics*, vol. 183, p. 108 125, 2021.
[11] C.-W. Lee et al., *IEEE transactions on electron devices,* vol. 57, no. 3, pp. 620–625, 2010.
[12] C. Mukherjee et al., *IEEE Transactions on Electron Devices,* 2023.
[13] H. Rezgui et al., *IEEE Transactions on Electron Devices,* 2023.
[14] T. Mikolajick et al., *Advanced Materials*, vol. 35, no. 37, p. 2 206, 042, 2023.
[15] E. T. Breyer et al., *IEEE Journal of the Electron Devices Society*, vol. 8, pp. 748–756, 2020.
[16] T. Mikolajick et al., *Solid-State Electronics,* vol. 194, p. 108 381, 2022.
[17] G. Galderisi et al., *IEEE Electron Device Letters*, accepted, 2024.
[18] C. Cakirlar et al., *Materials Today Electronics,* vol. 4, p. 100 040, 2023
[19] Y. M. Qureshi et al., *ACM Transactions on Architec-ture and Code Optimization (TACO),* vol. 18, no. 4, pp. 1–27, 2021.
[20] J. Klein et al., *IEEE Transactions on Computers,* 2022.
[21] A. Amirshahi et al., *in Proceedings of the 28th Asia and South Pacific Design Automation Conference*, 2023, pp. 657–663.
[22] S. Watanabe et al., *in Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH,* vol. 2018-September, 2018, pp. 2207–2211.
[23] L. Ben Letaifa et al., *in EUSIPCO 2022*, 2022.
[24] L. Ben Letaifa et al., *in 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA),* Dec. 2022, pp. 897–902.
[25] L. Ben Letaifa et al., *Al-gorithms,* vol. 16, no. 9, p. 398, Sep. 2023.