European Commission

FVLLMONTI

Call: **H2020-FETPROACT-2020-01**

Grant Agreement no. **101016776**

Deliverable D4.3

Versatile and scalable 3D interconnect framework

Start date of the project: 1st February 2021
Duration: 60 months
Project Coordinator: Cristell MANEUX - University of Bordeaux
Contact: Cristell MANEUX - cristell.maneux@ims-bordeaux.fr

DOCUMENT CLASSIFICATION

Title	Versatile and scalable 3D interconnect framework
Deliverable	D4.3
Estimated Delivery	31/10/2023 (M30+4)
Date of Delivery Foreseen	31/10/2023 (M30+4)
Actual Date of Delivery	31/10/2023 (M30+4)
Authors	Giovanni Ansaloni – P5 - EPFL
	David Atienza – P5 – EPFL
	Oskar Baumgartner – P4 – GTS
	Alberto Bosio – P3 – ECL-INL
	Bastien Deveautour – P3 – ECL-INL
	Sara Mannaa – P3 – ECL-INL
	Ian O'Connor – P3 – ECL-INL
Approver	Cristell Maneux – P1 – UBx
Work package	WP4
Dissemination	PU (Public)
Version	V0.4
Doc ID Code	D4.3_FVLLMONTI_P3-ECL-INL-20231031
Keywords	Neural network, logic design, interconnect

DOCUMENT HISTORY

Version status	V0.4
Date	16/10/2023
Document revision	V0.1 – 04/10/2023 – Initial draft
	V0.2 – 14/10/2023 – Document restructuring
	V0.3 – 16/10/2023 – Added mechanisms and building blocks chapter
	V0.4 – 16/10/2023 – Finalized introduction, abstract and conclusion
	V0.5 – 16/10/2023 – Minor additions, document finalization

DOCUMENT ABSTRACT

This document describes the 3D interconnect framework enabling inter-N^2C^2 connectivity in a flexible and scalable way, as a necessary step towards the complete N^2C^2-based neural network computation accelerator. In order to enable a regular 3D matrix of configurable logic functions, the intent is to deliver a versatile and scalable inter-cube interconnect framework capable of housing multiple (10^6) non-volatile N^2C^2s structured in the x,y,z planes and routing all inter-cell data, control signals and power lines in an efficient, regular and organized way.

This deliverable D4.03 is intended to serve as a reference document, containing a description of requirements for inter-N^2C^2 communication in the context of a compute accelerator, and leading naturally to a detailed description of mechanisms to be devised for 3D interconnect as well as corresponding FVLLMONTI platform building blocks. This includes fixed interconnect strategies for 2D and for 3D; design of building blocks for reconfigurable 2D/3D interconnect; design of non-volatile switchbox elements for 2D/3D architectures.

This is then put into perspective with a description of the main application of the interconnect framework – the implementation of the communication infrastructure for the N^2C^2-based Systolic Array (N^2C^2-SA). We cover the interface architecture, i.e. the external black-box view as well as signal definitions for configuration and use of the N^2C^2. We then relate this to the level of single N^2C^2s, and combine the two to describe the internal structure of the N^2C^2-SA.

We also cover, in the final chapters of this deliverable, prospective work to explore future extensions to the N^2C^2-SA: non-volatile operation implementing ferroelectric technology variants JLFE1 and JLFE2; and 3D operation with multiple levels (z-dimension) of a single N^2C^2-SA folding matrix computation elements in 3D to create a true 3D compute cube.

This information will be used jointly in WP4 (to focus further logic cell design work in living document D4.1 Library of optimized VNWFET-based logic cells, and to refine logic synthesis libraries in view of D4.5.2 Virtual scalable N^2C^2design and Pareto-front data scheduled for M44) and WP5 (to enable architectural exploration in D5.2).

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101016776.

TABLE OF CONTENT

LIST OF FIGURES AND TABLES

LIST OF ACRONYMS / GLOSSARY

D: Deliverable

LUT: Look Up Table

M: Month of the project

MAC: Multiply Accumulate

N^2C^2: Neural Network Compute Cube

N^2C^2-SA: Neural Network Compute Cube Systolic Array

NN: Neural Network

P: Partner

PU: Public

V: Version

VNWFET: Vertical Nanowire Field Effect Transistor

WP: Work Package

1. Introduction

The N^2C^2-based computation accelerator, capable of implementing complex and configurable matrix-vector multiplications and other operations for transformer-based neural networks, requires not only the highly versatile and configurable N^2C^2 block, but also a 3D interconnect framework enabling inter-N^2C^2 connectivity in a flexible and scalable way. To enable a regular 3D matrix of configurable logic functions, the inter- N^2C^2 interconnect framework should be both versatile and scalable, as well as capable of housing multiple (10^6) non-volatile N^2C^2s structured in the x,y,z planes and routing all inter-cell data, control signals and power lines in an efficient, regular and organized way.

Indeed, projecting the technology scaling as proposed by the FVLLMONTI project, will enable large-scale integration of N^2C^2 processing elements within a single accelerator architecture. However, as the number of processing elements increases, the burden on the interconnect infrastructure becomes more pronounced.

Dedicated buses may be sufficient as the interconnect fabric for specific accelerator architectures at limited scale, but ultimately more sophisticated power-efficient on-chip interconnect infrastructures are required to meet delay, power and cost constraints. In particular, reconfigurable communication paradigms enable the exchange of data through a packet switching network, rather than through fixed directly connected wires. Data is routed from a source node to a destination by traveling through a series of routers and links. This work tackles both fixed and reconfigurable interconnect infrastructures, and further explores the opportunities of 3D architectures and of non-volatile functionality for communication.

The aim of this document is thus two-fold – to supply an architecture for a scalable and versatile interconnect infrastructure enabling the construction of complete N^2C^2-based accelerators in conjunction with work in WP5; and to materialize the building blocks necessary for the infrastructure, beyond those developed in logic cell library work (D4.1), including more prospective elements leveraging 3D and non-volatile technology characteristics.

This deliverable is organized as follows. We begin in Section 2 by reviewing N^2C^2 functionality, as a basis around which the interconnect architecture is designed in order to formulate high-level requirements for the target versatile interconnect infrastructure. Section 3 contains a description of more detailed requirements for inter- N^2C^2 communication in the context of a compute accelerator, and leads naturally to a discussion of mechanisms to be devised for 3D interconnect as well as corresponding FVLLMONTI platform building blocks. This includes fixed interconnect strategies for 2D and for 3D; design of building blocks for reconfigurable 2D/3D interconnect; design of non-volatile switchbox elements for 2D/3D architectures. This is then put into perspective in Section 4 with a description of the implementation of the communication infrastructure for the N^2C^2-based Systolic Array (N^2C^2-SA). We cover the interface architecture and relate this to the level of single N^2C^2s, and combine the two to describe the internal structure of the N^2C^2-SA. More prospective work to explore future extensions to the N^2C^2-SA is covered in Section 5 which focuses on non-volatile operation implementing ferroelectric technology variants JLFE1 and JLFE2, and in Section 6 which targets the implementation of a single N^2C^2-SA by folding matrix computation elements in 3D. Finally, Section 7 concludes this deliverable.

2. N²C² functionality and implications for N²C²-based accelerator

The following text describes the functionality of the N²C² block, as a basis around which the interconnect architecture is designed. This description first appeared in D4.5a (Virtual scalable N²C² design and Pareto-front data) and is reproduced here (with minor updates) for the sake of readability.

The Neural Network Compute Cube (N²C²) is a central concept to the FVLLMONTI project. It represents a flexible computing hardware block for transformer-based neural networks. As illustrated in Figure 1, it is to be implemented based on a dedicated library of 3D logic cells leveraging VNWFET devices developed in T4.1 (Logic cell design, optimization and validation) and using technological hardware and data developed in WP1, WP2 and WP3. It also connects through a reconfigurable 3D interconnect framework developed in T4.2 (Inter-cube interconnect framework) to implement a scalable and versatile 3D architectural model in connection with WP5. Its fundamental properties of physical regularity, functional versatility and in-memory vector processing make it suitable to explore hardware/software co-design techniques in the context of transformer-based neural networks for machine translation applications as well as quantization-based approximate computing to reduce resource usage and energy consumption as well as enable more complex network topologies.

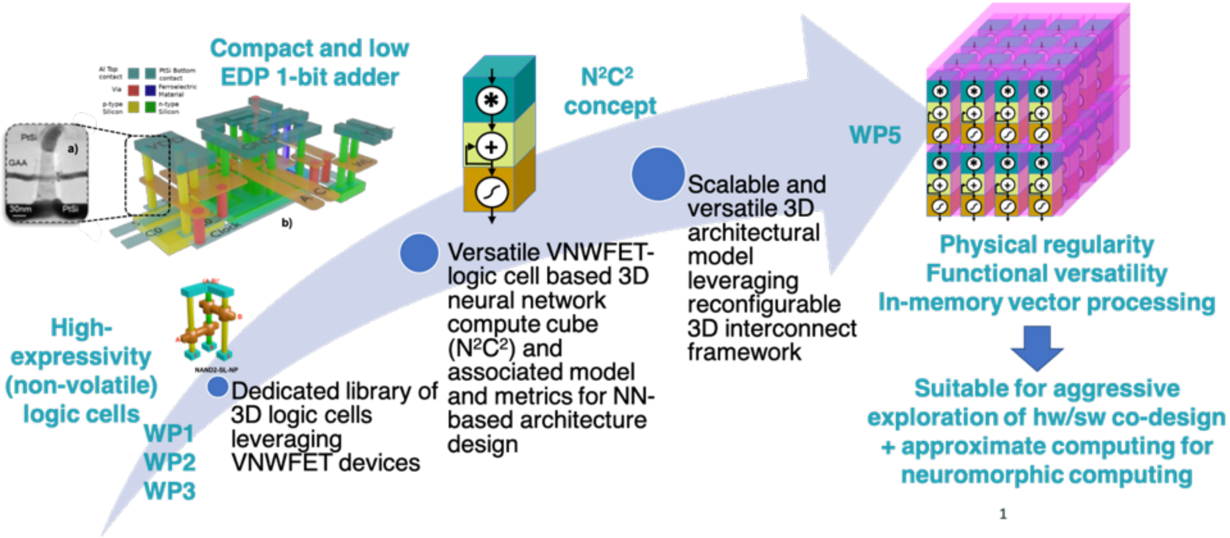
Compact and low EDP 1-bit adder
Al Top contact
PtSi Bottom contact
Via
Ferroelectric Material
p-type Silicon
n-type Silicon
PtSi
a)
GAA
30nm
PtSi
b)
C0
Clock
VDD
GND
A
C1
High-expressivity (non-volatile) logic cells
NAND2-SL-NP
WP1
WP2
WP3
Dedicated library of 3D logic cells leveraging VNWFET devices
N²C² concept
Versatile VNWFET-logic cell based 3D neural network compute cube (N²C²) and associated model and metrics for NN-based architecture design
Scalable and versatile 3D architectural model leveraging reconfigurable 3D interconnect framework
WP5
Physical regularity
Functional versatility
In-memory vector processing
Suitable for aggressive exploration of hw/sw co-design + approximate computing for neuromorphic computing
1

Figure 1 : Neural Network Compute Cube (N²C²) – the big picture

The principal function of the N²C² 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 resources 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 intended for the activation function to be implemented in memory elements in a coarse-grain logic-in-memory approach.
- 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 N²C² (T4.2 – Intercube interconnect framework) 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 it is also possible to handle inter-N^2C^2 scaleup to 64 bits, 128 bits, 256 bits, 512 bits.

Each N^2C^2 is intended for use in a versatile 3D architectural model. In other words, 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.

In order to consider multiple use-case scenarios, we define the array of N^2C^2 blocks as being of 2 dimensions (x,y) or 3 dimensions (x,y,z) where x,y,z represent the number of columns, rows and layers of cells, respectively.

A versatile interconnect infrastructure requires:

- physical mechanisms and building blocks to support the needs of the interconnect infrastructure architecture in 2D and 3D
- an architecture, defined at design time, to support the communication requirements between all N^2C^2 blocks in the accelerator
- a configuration and communication protocol to a) define the application-targeted functionality of the interconnect infrastructure at run-time and b) define means for actual communication streams to operate between N^2C^2 blocks

In the following sections, we will cover these points in the 2D and 3D use-case scenarios.

3. Mechanisms and building blocks for 3D interconnect

In this section, we cover the requirements for the various cases of interconnect to be considered.

In a generic 3D matrix, we define x,y,z as representing the coordinates of a given N2C2 block in the array :

- x represents the column coordinate, from 0 (left) to X-1 (right)
- y represents the row coordinate, from 0 (top) to Y-1 (bottom)
- z represents the layer coordinate, from 0 (lowest) to Z-1 (highest)

This notation enables us to define connections between N^2C^2 blocks (e.g. the propagated input $Y_{x,y,z}$ for one N^2C^2 block is the output $Y_{x,y-1,z}$ from the preceding N^2C^2 block in the y direction).

By expressing the interconnect structure in 3 dimensions, it is trivial to simplify to 2 dimensions (by removing z) or to even 1 dimension (by further removing either x or y).

In terms of hardware for the interconnect, three cases can also be identified:

- fixed interconnect
- reconfigurable interconnect
- reconfigurable non-volatile interconnect

In all cases, the datawidth must also be considered, as well as the potential need for reconfigurability to inhibit parts of the communication infrastructure when it is not needed (e.g. in the case of reduced precision data, requiring fewer bits than the actual physical infrastructure provides).

Similarly, when data inputs should be simply propagated over an N^2C^2 block (e.g. in the case of sparse matrices, requiring fewer operations than the physical infrastructure provides), means should be available to circumvent the N^2C^2 and route the data through the block without any data transformation.

I. FIXED INTERCONNECT FOR 2D

In a fixed interconnect approach, we consider that x and y routing channels surround the N^2C^2 block in a 2D tile, where:

- Weights (B_x) are shared across each column x
- Data inputs (A_y) are propagated at each cycle from column to column along each row y
- Data outputs ($Y_{x,y}$) are computed at each cycle from $A_y(t)$, B_x, $Y_{x,y-1}$ to generate $Y_{x,y}$

Figure 2 shows the overall organization for a single 2D tile.

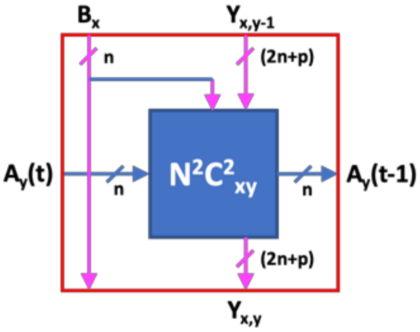

Figure 2 : x and y routing channels surrounding the N^2C^2 block in a tile-based approach

From the physical design point of view and as shown in the figure, we use (arbitrarily) metal layer **M1** for lines in the x direction, **M2** for lines in the y direction.

To incorporate timing constraints, buffering is implemented using inverter cells such as INV1_1_CStatic_JL1 (see D4.1).

Finally, in order to handle flexible datawidth requirements and since this cannot be programmed in the fixed, interconnect, it must be handled within the N^{2}C^2 block generating the results by setting unused bits to 0.

II. FIXED INTERCONNECT FOR 3D

In a 3D fixed interconnect approach, we consider that x, y and z routing channels surround the N^{2}C^2 block in a 2D tile, where:

- Weights (B_x) are shared across each column x and layer z
- Data inputs (A_y) are propagated at each cycle from column to column along each row y and layer z
- Data outputs ($Y_{x,y,z}$) are computed at each cycle from $A_y(t)$, B_x, Y_{in} to generate $Y_{x,y,z}$, where:
  - $(x, y, z) \in \mathbb{N}$
  - if z even and y>0, $Y_{in} = Y_{x,y-1,z}$; else if z>0 and y=0, $Y_{in} = Y_{x,y,z-1}$; else $Y_{in}=0$
  - if z odd and $y<y_{max}$, $Y_{in} = Y_{x,y+1,z}$; else $Y_{in} = Y_{x,y,z-1}$

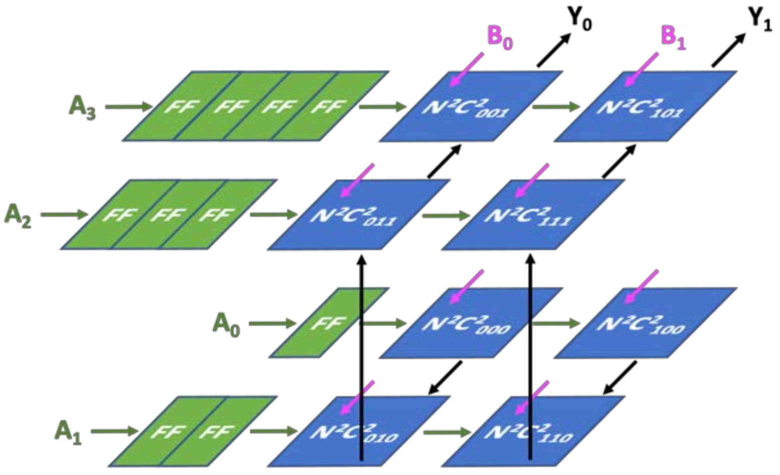

Figure 3 : x y z routing scheme for N^{2}C^2 blocks in a 3D tile-based approach implementing a 2x4 systolic array

III. RECONFIGURABLE SWITCHBOX FOR 2D

While the main use case (Systolic Array) focuses on fixed interconnect, it is of interest for the future to explore the potential for reconfigurable interconnect resources between N^{2}C^2 blocks in order to adapt the interconnect infrastructure to communication requirements and/or network topologies. Indeed, reconfigurable approaches enable hardware performance to be tuned to real application requirements at run-time, rather than fixing hardware at design time and facing possible severe mismatches between hardware performance and application requirements.

The approaches described below stem from conventional FPGA structures. The principle explored here is the design of a tile composed of one N^{2}C^2 surrounded by routing resources [1], as shown in Figure 4.

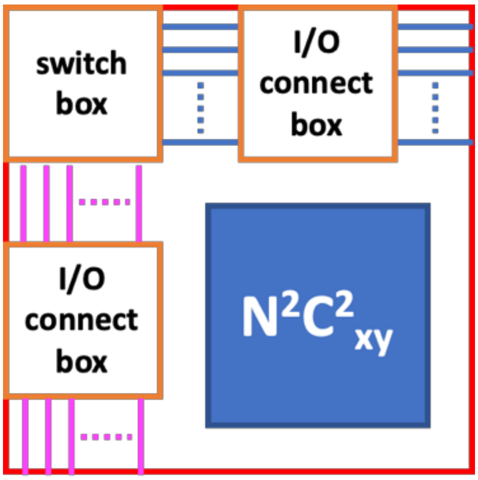

Figure 4 : Reconfigurable routing functions surrounding the N^2C^2 block in a tile-based approach

Routing resources are composed of x/y routing channels (as in the previously described fixed interconnect approaches) but here also include:

- input/output connection switches to/from the N^2C^2 block to enable access to channels for data transfer
- a 2D switchbox between 4 x,y points (North, East, South, West)

Input and output connection switch functions are shown in Figure 6. The main building blocks are 3-state buffers (e.g. based on INVT1_1_CStatic_JL1 or INVT1_1_CStatic_JL2 as shown in Figure 5), switches/multiplexers and memory (SRAM) bitcells.

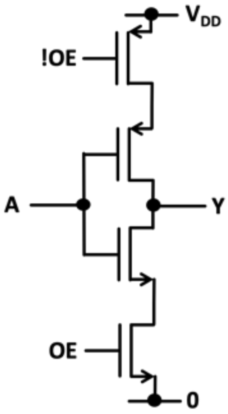

Figure 5 : 3-state inverter (INVT1) standard cell in complementary static logic, with implementation as INVT1_1_CStatic_JL1 or INVT1_1_CStatic_JL2

It should be noted that the connections can be implemented with single transistors (VNWFETs) as switches, or through multiplexers. The single device solution is more compact but can also lead to slower communication and logic degradation. The multiplexer-based solution is faster and suffers no logic degradation but costs more in terms of device count.

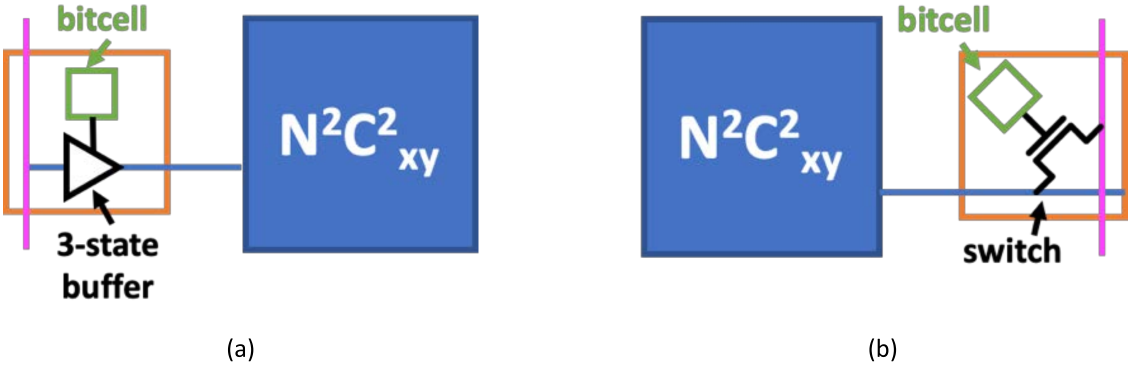

Figure 6 : Input / output connection switch structures. a) 1-bit input connect box. b) 1-bit output connect box

2D switchboxes implement programmable connections between 4 I/O points (North, East, South, West) at an intersection between x and y data channels. For a fully-connected switchbox, this means 6 potential connections, as shown in Figure 7(a). A solution based on switches controlled by memory (SRAM) bitcells is shown Figure 7(b) – as with the I/O connections, this approach is subject to logic degradation and low data transfer rate. Alternative solutions, using 3-state buffers and multiplexers, for both bidirectional and directional communication cases [2], are shown Figure 7(c) and Figure 7(d) respectively.

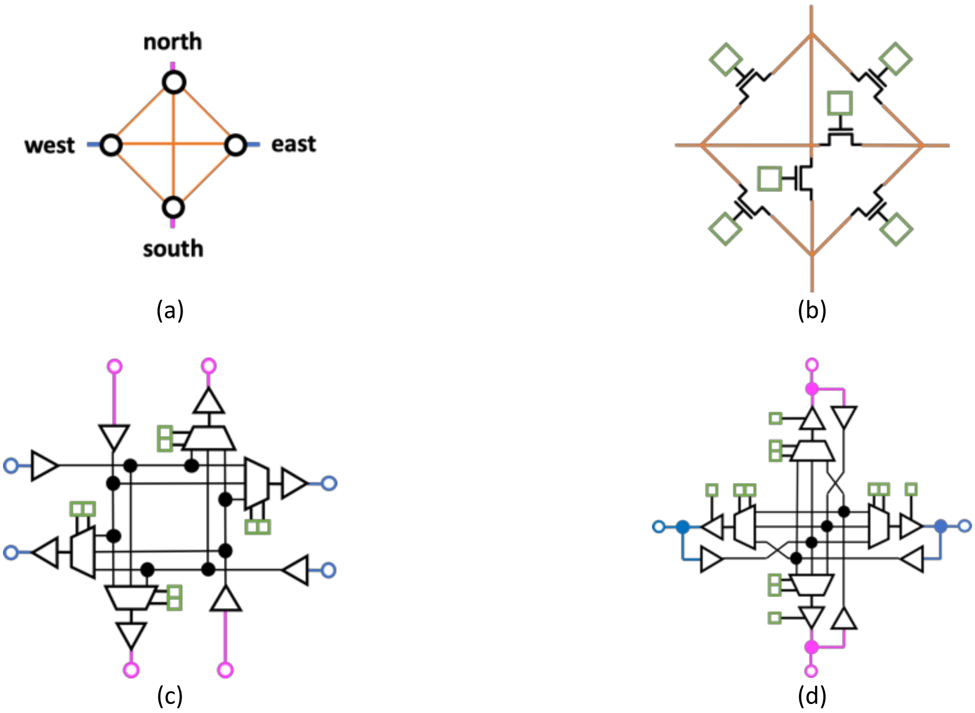

Figure 7 : Switchbox structures. a) Principle of 4-point (North, East, South, West) 6-connection 2D structure. b) Switch+SRAM structure (Switchbox4_1_PTL_JL1). c) Directional 3-state buffer and MUX-based structure. d) Bidirectional 3-state buffer and MUX-based structure.

In terms of datawidth, one of the reconfiguration requirements of the N^2C^2 is to enable flexible means to adjust datawidth to required computation accuracy. Since all connections are programmed through SRAM cells, it is a simple matter to inhibit connections for unused bits.

Finally, it is anticipated that sparse matrices will only require specific N^2C^2 blocks to carry out operations at specific points in the array. Unused N^2C^2 blocks at other points in the array should merely propagate the column input to the output i.e. $Y_{x,y}=Y_{x,y-1}$. In order to implement aggressive power-down strategies, it is of interest to enable full disconnection of unused N²C² blocks from the supply, necessitating the addition of propagation functionality to the y channels. This structure is shown in Figure 8.

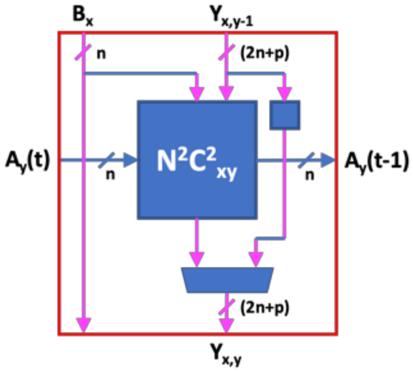

Figure 8 : Input / output connection switch structures

IV. RECONFIGURABLE SWITCHBOX FOR 3D

In the case of a 3D array of N²C² blocks, we extend the design of a tile composed of one N²C² surrounded by routing resources to the vertical (z) dimension, as shown in Figure 9.

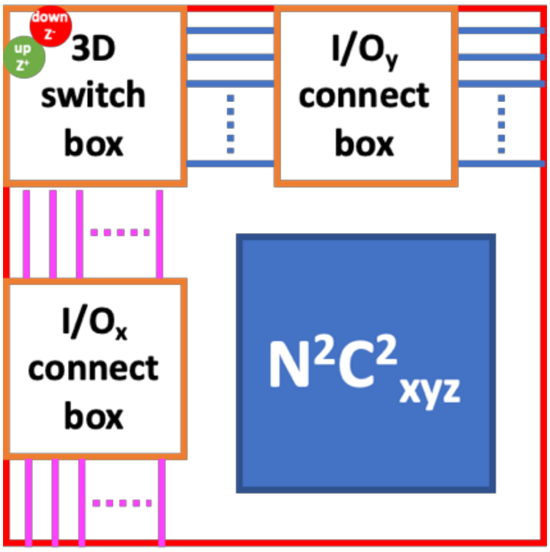

Figure 9 : Reconfigurable 3D routing functions surrounding the N²C² block in a tile-based approach

Routing resources are composed of x/y/z routing channels (as in the previously described fixed interconnect approaches) but here again also include:

- input/output connection switches to/from the N²C² block to enable access to channels for data transfer
- a 3D switchbox between 6 x,y,z points (North, East, South, West, Up, Down)

Input and output connection switch functions are as described previously.

3D switchboxes implement programmable connections between 6 I/O points (North, East, South, West, Up, Down) at an intersection between x, y and z data channels. For a fully-connected switchbox, this means 15 potential connections as shown in Figure 10. Solutions based either on switches controlled by memory

(SRAM) bitcells, or on 3-state buffers and multiplexers, for both bidirectional and directional communication cases, are extensions of the 2D solutions described previously.

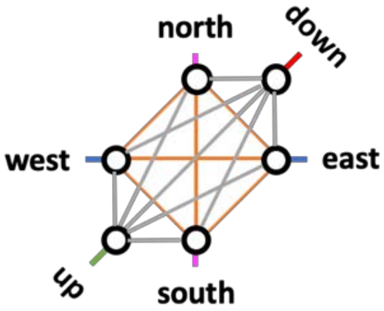

Figure 10 : Principle of 6-point (North, East, South, West, Up, Down) 15-connection 3D switchbox structure.

Mechanisms for reduced datawidth and sparse matrices are identical to those devised for the 2D reconfigurable interconnect.

V. NON-VOLATILE SWITCHBOX ELEMENTS (2D/3D)

While the reconfigurable circuit elements presented in the previous sections enable hardware flexibility, they require SRAM bitcells to store the configurations of the state of each switch. This is costly in terms of hardware resources (1 SRAM bitcell = 6 transistors) and is a volatile approach, implying that any power-down leads to loss of configuration and significant time and energy overhead for wakeup to reprogram the bitcells; and leading to long power-on times during which standby (leakage) energy consumption becomes high.

The availability of non-volatile components in the FVLLMONTI platform (JLFE1, JLFE2) was already explored with the objective of developing non-volatile logic cells in D4.1. In the context of designing the interconnect infrastructure, the same non-volatile properties can be leveraged to enable non-volatile storage of routing resource configurations, opening the way to aggressive power-down strategies incurring no overhead during wakeup.

The three elements that can be designed with JLFE1 or JLFE2 to introduce non-volatile operations into their functionality are the following: switchbox (switch-based, bidirectional and directional 3-state buffer and multiplexer-based), input connection structures, output connection structures.

FeFETs can directly replace switch+SRAM devices in switch-based 2D switchbox structures, as shown in Figure 11. Further to enabling non-volatile functionality, the device count is also drastically reduced. Considering the types of FeFET explored in D4.1, two types can be used:

- type / (non-volatile on/off) – n/FeFET: in this case, the shift of threshold voltage is such that the device can be programmed to implement a switch in either on- or off-state, without applying any further voltage to the gate terminal.
- type • (AND) – n•FeFET: here, the device can be considered either to be always off (independent of the gate voltage) or on if a '1' is applied to the gate voltage. This is used to maintain dynamic control over the switch state during run time.

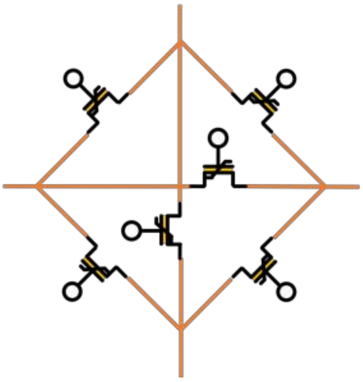

Figure 11 : Non-volatile FeFET-based 2D switchbox (Switchbox4_1_PTL_JLFE1)

Within the other types of switchbox and in the input/output connection structures, 3-state buffers and multiplexers are required.

A non-volatile version of the 3-state buffers can be implemented as shown in Figure 12(a). The figure shows the use of p/ and n/FeFET type devices, leading to always on / always off operation. As with the switchbox, p• and n•FeFET devices can also be used to maintain dynamic control. Finally, a non-volatile 4-1 multiplexer (NVMUX4) is shown in Figure 12(b).

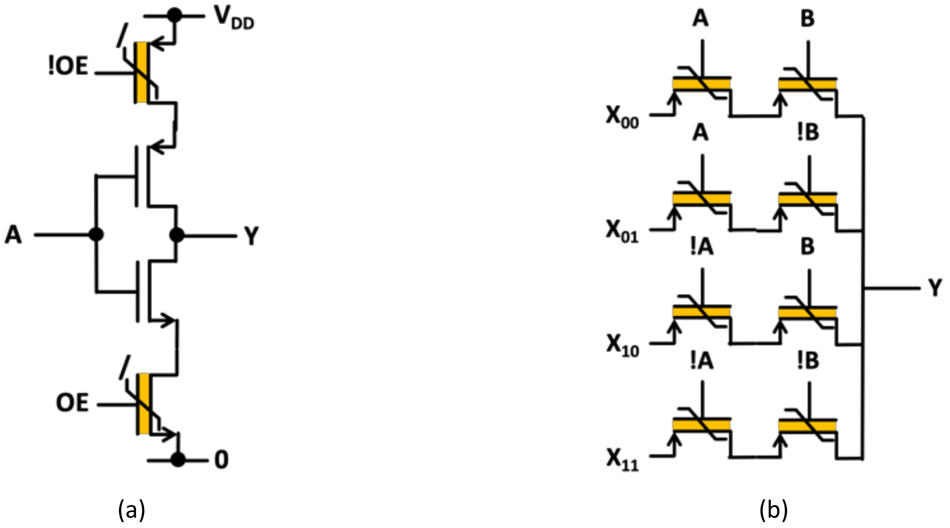

Figure 12 : Non-volatile FeFET-based circuits for switchboxes and input/output connection boxes. a) 3-state buffer (INVT1_1_CNVL_JLFE1). b) Non-volatile 4-input multiplexer cell (NVMUX4_1_PTL_JLFE1, NVMUX4_1_PTL_JLFE2)

4. N^2C^2 Systolic Array Interface

In this section, we examine the black-box view of the Neural Network Compute Cube Systolic Array (N^2C^2-SA), as well as the design of its interface. This is particularly important as handover point for work in WP5 considering N^2C^2-SA used as an accelerator in conjunction with a RISC-V based compute core or microcontroller. The interface must be able to handle both programming (configuration, weights loading) and execution (data in, data out) phases of operation. We then review the internal structure of the N^2C^2 in order to indicate how the individual blocks are then connected in an actual array structure (i.e. the internal structure of the N^2C^2-SA, to serve as use-case for the 2D and 3D interconnect infrastructure using the building blocks discussed in section 3).

I. BLACK-BOX VIEW

The implementation of the N^2C^2-SA and its external interface is depicted in Figure 13.

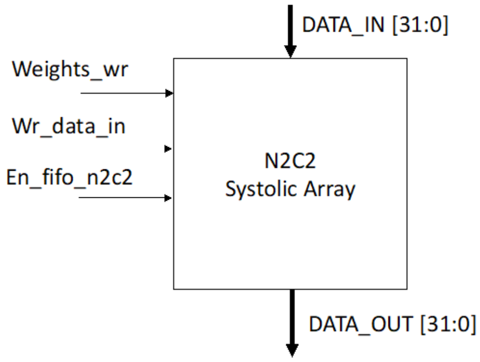

Figure 13 : N^2C^2 SA top-level entity and external interface

The designed interface allows the host controller (a RISC-V microcontroller) to access the N^2C^2-SA through two instructions:

- *SA_LW (value):* Systolic Array Load Weights. It writes the application weights into the array.
- *SA_ IOC (value)*: Systolic Array In-Out and Compute. It writes in the input value into the array and, at the same time, reads back 32 bits from the N^2C^2-SA output that corresponds to the computed results.

The above instructions allow the N^2C^2-SA to be used in two phases:

1. Weights programming: communication of individual weights from external memory to within each individual N^2C^2 block in the accelerator, over the 32-bit data bus. It corresponds to the *SA_LW (value).*
2. Execution (after weights programming): data in is passed to the accelerator over the 32-bit data bus and output is read back from data out. It corresponds to the *SA_ IOC (value).*

Apart from the input and output DATA BUS (32 bits), the external interface is also composed of the following three control signals:

- Weights_WR (active high): enables N^2C^2-SA configuration during the weights programming phase.
- Wr_data_in (active high): configures the N^2C^2-SA for data input-output.
- En_fifo_n2c2 (active high): configures N^2C^2-SA for computation during the execution phase.

II. N²C² INTERNAL STRUCTURE

Internally, each N²C² has the structure depicted in Figure 14, including the internal interconnect structure. This structure was fully described in D4.5a (Virtual scalable N²C² design and Pareto-front data) from the conceptual point of view, and was also considered as the reference structure for the physical design point of view in D3.3 (3D-place-and-route prototype).

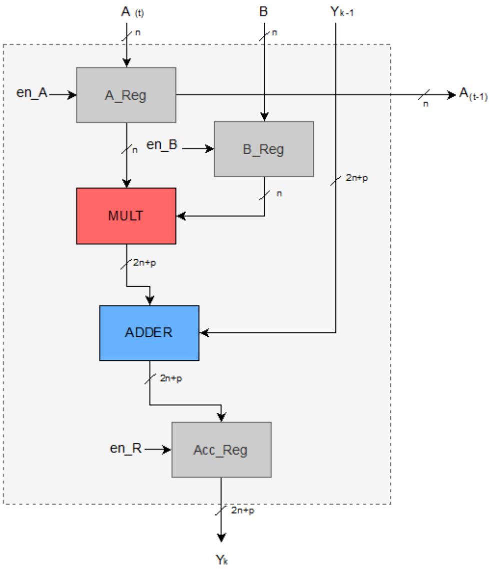

Figure 14 : N²C² structure and interconnections

The N²C² has three data inputs: A, B and Y_{k-1}. The n-bit inputs A and B 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 N²C² 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" N²C² and is in practice related to an actual block in a 2D or 3D matrix such as $Y_{x,y-1,z}$ as denoted as an illustrative example in section 3.

In terms of internal interconnect, there are:

- two internal n-bit buses connecting A_reg and B_reg to the multiplier
- the (2n+p)-bit bus connecting the output of the multiplier to the input of the adder
- the (2n+p)-bit bus connecting the output of the adder to the accumulator register

Finally, the N²C² has three control signals (enables):

- en_A, connected to Wr_data_in
- en_B, connected to Weights_WR
- en_R, connected to En_fifo_n2c2

III. N²C²-SA INTERNAL STRUCTURE

Figure 15 shows how N²C² elements exchange stream data in a 2x2 cropping of the most "bottom-left" elements that are part of a larger 2D array.

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 (B) and its incoming operand (A) 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.

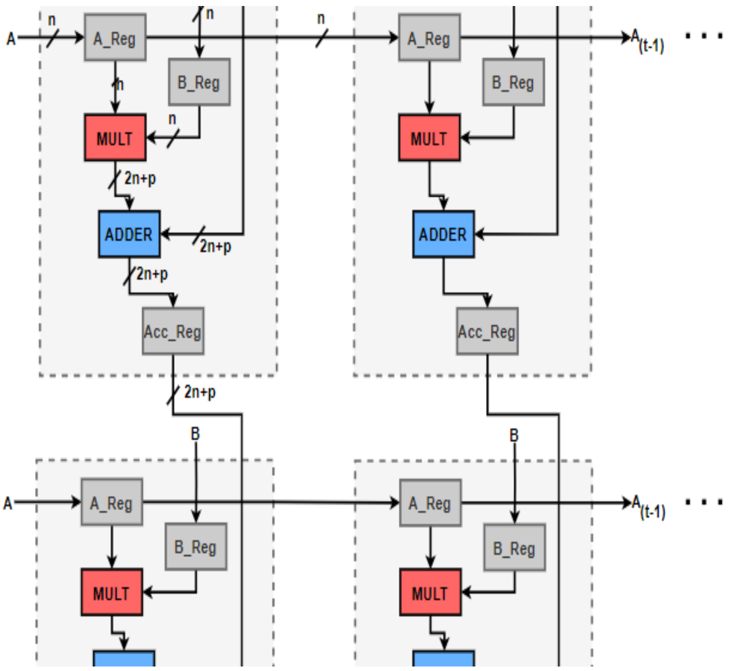

Figure 15: N^2C^2-N^2C^2 data flow within an N^2C^2-SA

Figure 16 depicts the complete interconnect infrastructure within an example 4x4 systolic array, and where Weights_WR and Wr_data_in are also shown.

As previously mentioned, the N^2C^2-SA is used in two phases:

1. Weights programming: when Weights_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).

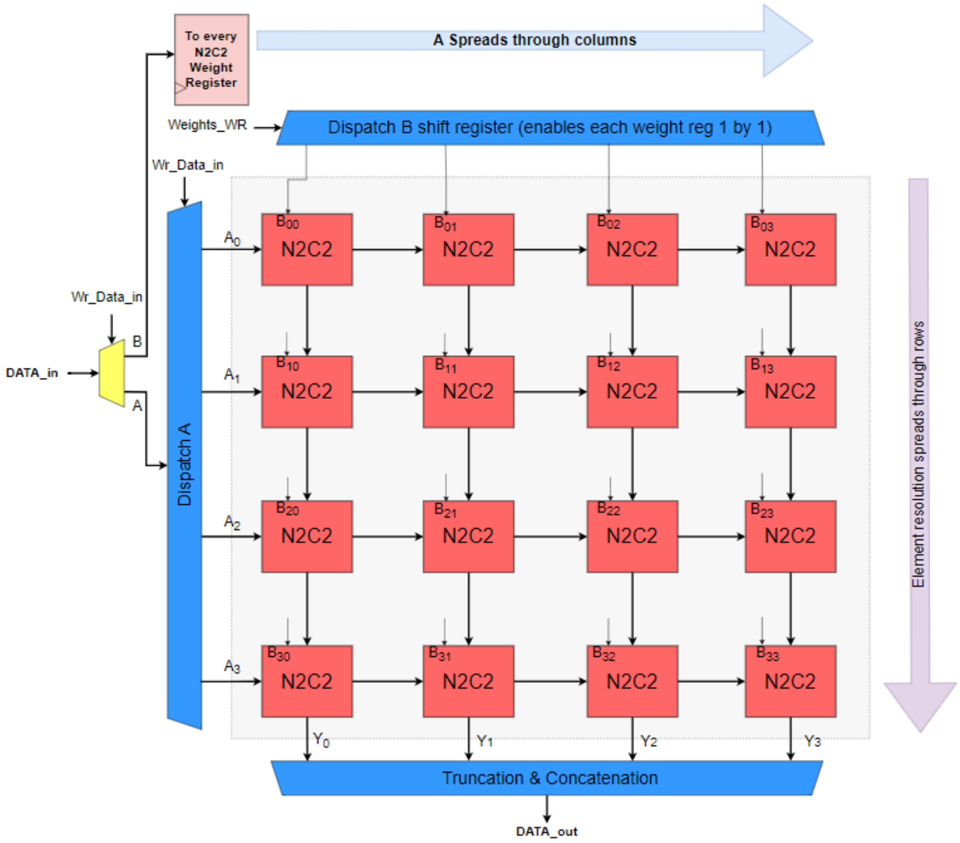
To every N2C2 Weight Register
A Spreads through columns
Weights_WR
Dispatch B shift register (enables each weight reg 1 by 1)
Wr_Data_in
Wr_Data_in
DATA_in
B
A
Dispatch A
A0
A1
A2
A3
B00
B01
B02
B03
B10
B11
B12
B13
B20
B21
B22
B23
B30
B31
B32
B33
N2C2
Element resolution spreads through rows
Y0
Y1
Y2
Y3
Truncation & Concatenation
DATA_out

Figure 16: Example of a 4x4 N²C² array and its interconnections

5. Non-volatile operation of N^2C^2-SA

As described in D4.1 (Library of optimized VNWFET-based logic cells), the use of ferroelectric VNWFETs enables the design of non-volatile logic gates [1][5], for which one input can be permanently stored [6][7][8]. In our application context, we will implement **non-volatile multipliers**, for which input B will be permanently stored (i.e., programmed).

The programming phase of ferroelectric devices requires necessarily higher voltages (in order to modify, i.e. write, the polarization state of the ferroelectric layer) than that used during normal operation (in order to probe without modifying, i.e. read, the polarization state) [9]. For this reason, the non-volatile version of N^2C^2-SA (NV-N^2C^2-SA) must be modified to adapt to this additional functionality, as depicted in Figure 17.

To enable FeFET programming, it is mandatory to introduce two different power domains:

- One carrying higher voltages to set or erase the input B (i.e. the weight), for use only during the weights programming phase
- One carrying CMOS-compatible voltages, for use during the execution phase

The "FeFET control logic" block is thus responsible for converting the weight value from the CMOS-voltage power domain to the higher-voltage power domain. A "Power Mux" [10] block selects between CMOS-compatible data (DATA_IN, to be used for input A) and high-voltage data (HV_DATA_IN). The control signal Weights_WR selects between the weights programming phase (Weights_WR = '1') and the execution phase (Weights_WR = '0'). In both blocks, logic devices must be capable of withstanding the higher electric fields and will thus require careful and distinct optimization with respect to the other VNWFET devices considered in the project.

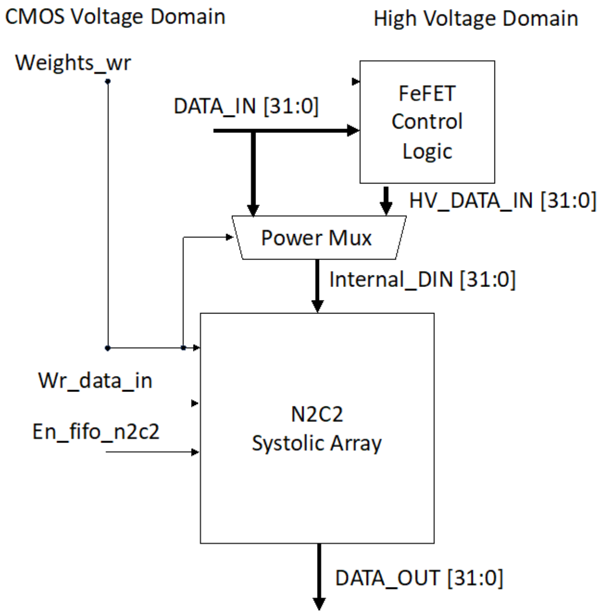

Figure 17 : Non-Volatile N^2C^2-SA (NV-N^2C^2-SA)

6. 3D operation of N²C²-SA

An additional anticipated evolution of N²C²-SA is to exploit VNWFETs to implement a 3D systolic array as depicted in Figure 18. As shown in the figure, this structure stores a number (two in the example) of different tiles of weights on different planes, with the same inputs for all planes. Output of tiles on layer z=2 (W2) will be "added" to those on layer z=1 (W1) thanks to dedicated vertical interconnect between tiles. Notice that only the adders performing the accumulation between outputs in different 'z' planes and the input/output FIFOs are planar structures, while the N2C2 array itself has a regular 3D layout. Furthermore, increasing the number of planes has no effect on the Input/output bandwidth, because inputs are broadcasted to multiple planes and outputs are reduced (accumulated) among planes.

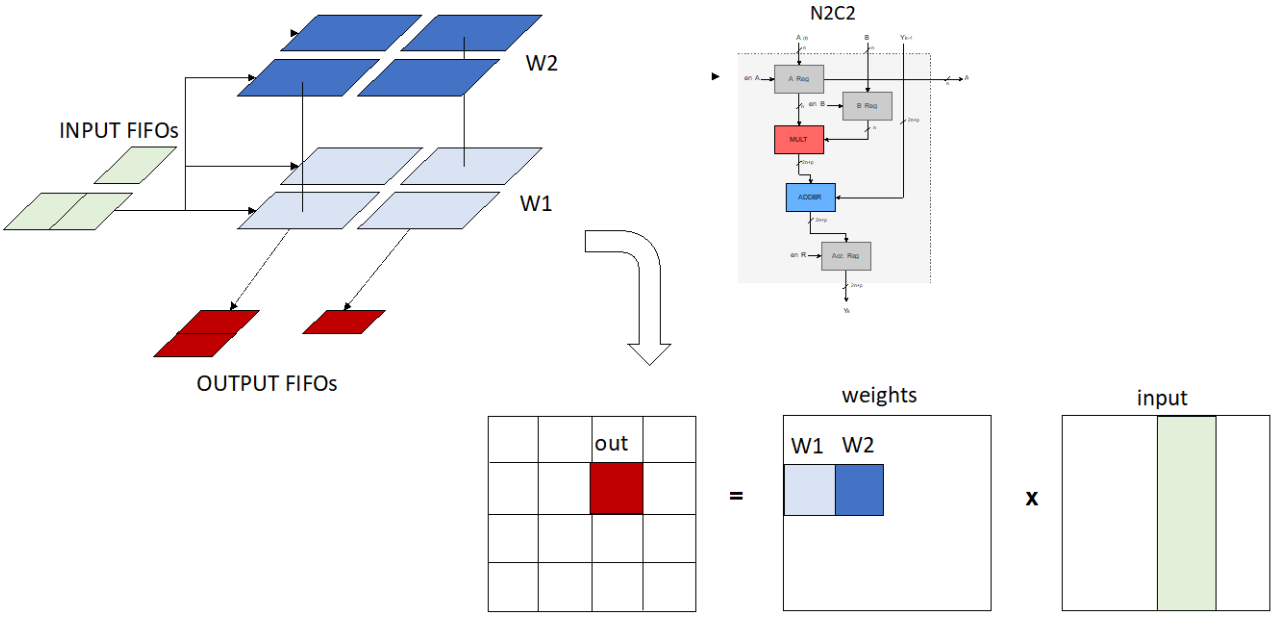

Figure 18 : 3D N²C²-SA

7. Conclusion

In this deliverable, we described a proposal for a versatile and scalable 3D interconnect framework to enable inter-N^2C^2 connectivity. This is a necessary step towards the complete N^2C^2-based neural network computation accelerator. In order to target support communication within a regular 3D matrix of configurable N^2C^2s, we propose an interconnect framework capable of housing multiple (10^6) non-volatile N^2C^2s structured in the x,y,z planes and routing all inter-cell data, control signals and power lines in an efficient, regular and organized way.

We first reviewed N^2C^2 functionality, as a basis around which the interconnect architecture is designed in order to formulate high-level requirements for the target versatile interconnect infrastructure.

We then described in more detail requirements for inter-N^2C^2 communication in the context of a compute accelerator, as well as mechanisms devised for 3D interconnect and corresponding FVLLMONTI platform building blocks. This includes:

- fixed interconnect strategies for 2D and for 3D
- design of building blocks for reconfigurable 2D/3D interconnect
- design of non-volatile switchbox elements for 2D/3D architectures

We then described the implementation of the communication infrastructure for a 2D N^2C^2-based Systolic Array (N^2C^2-SA), using fixed interconnect topologies. We cover the interface architecture and relate this to the level of single N^2C^2s, and combine the two to describe the internal structure of the N^2C^2-SA.

We also carried out more prospective work exploring future extensions to the N^2C^2-SA:

- non-volatile operation, implementing ferroelectric technology variants JLFE1 and JLFE2
- implementation of a single N^2C^2-SA by folding matrix computation elements in 3D

8. References

[1] A. Rahman, S. Das, A. Chandrakasan, R. Reif, "Wiring requirement and three-dimensional integration technology for field programmable gate arrays," in IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol. 11, no. 1, pp. 44-54, Feb. 2003 (DOI: 10.1109/TVLSI.2003.810003)

[2] U. Farooq, Z. Marrakchi, H. Mehrez, Tree-based Heterogeneous FPGA Architectures: Application Specific Exploration and Optimization, Springer, 2012 (DOI: 10.1007/978-1-4614-3594-5)

[3] C. Mineo, R. Jenkal, S. Melamed, W. Davis, "Inter-die signaling in three dimensional integrated circuits," IEEE Custom Integrated Circuits Conference (CICC), pp. 655-658, San Jose (CA), USA, 21-24 Septembre 2008 (DOI: 10.1109/CICC.2008.4672171)

[4] C. Marchand, A. Nicolas, P.A. Matrangolo, D. Navarro, A. Bosio, I. O'Connor, "FeFET based Logic-in-Memory design methodologies, tools and open challenges," 31st IFIP/IEEE Conference on Very Large Scale Integration (VLSI-SoC), Abu Dhabi, UAE, 16-18 October 2023

[5] A. Ram, K. Maity, C. Marchand, A. Mahmoudi, A.R. Kshirsagar, M. Soliman, T. Taniguchi, K. Watanabe, B. Doudin, A. Ouerghi, S. Reichardt, I. O'Connor, J.F. Dayen, "Reconfigurable Multifunctional van der Waals Ferroelectric Devices and Logic Circuits," ACS Nano, accepted for publication (2023)

[6] A. Bosio, M. Cantan, C. Marchand, I. O'Connor, P. Fiser, A. Poittevin, M. Traiola, "Emerging Technologies: Challenges and Opportunities for Logic Synthesis," International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), p. 93-98, Vienna, Austria, 7-9 April 2021 (DOI: 10.1109/DDECS52668.2021.9417062)

[7] C. Marchand, I. O'Connor, M. Cantan, E.T. Breyer, S. Slesazeck, T. Mikolajick, "FeFET based Logic-in-Memory: an overview," 16th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS), Apulia, Italy, 28-30 June 2021, (DOI: 10.1109/DTIS53253.2021.9505078)

[8] C. Marchand, I. O'Connor, M. Cantan, S. Slesazeck, T. Mikolajick, "A FeFET-based hybrid memory accessible by content and by address," IEEE Journal of Exploratory Solid-State Computational Devices and Circuits, vol. 8, no. 1, pp. 19-26, June 2022 (DOI: 10.1109/JXCDC.2022.3168057)

[9] E. T. Breyer, H. Mulaosmanovic, T. Mikolajick and S. Slesazeck, "Reconfigurable NAND/NOR logic gates in 28 nm HKMG and 22 nm FD-SOI FeFET technology," 2017 IEEE International Electron Devices Meeting (IEDM), pp. 28.5.1-28.5.4, San Francisco (CA), USA, 2017 (DOI: 10.1109/IEDM.2017.8268471)

[10] G. E. Biccario, O. Vitrenko, R. Nonis and S. D'Amico, "A 5-V Switch for Analog Multiplexers With 2.5-V Transistors in 28-nm CMOS Technology," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 31, no. 5, pp. 636-643, May 2023 (DOI: 10.1109/TVLSI.2023.3240002)