Christopher Bengel*, Leon Dixius*, Rainer Waser, Dirk J. Wouters and Stephan Menzel

Bit Slicing Approaches for Variability Aware ReRAM CIM Macros

Abstract: Computation-in-Memory accelerators based on resistive switching devices represent a promising approach to realize future information processing systems. These architectures promise orders of magnitudes lower energy consumption for certain tasks, while also achieving higher throughputs than other special purpose hardware such as GPUs, due to their analog computation nature. Due to device variability issues, however, a single resistive switching cell usually does not achieve the resolution required for the considered applications. To overcome this challenge, many of the proposed architectures use an approach called bit slicing, where generally multiple low-resolution components are combined to realize higher resolution blocks. In this paper, we will present an analog accelerator architecture on the circuit level, which can be used to perform Vector-Matrix-Multiplications or Matrix-Matrix-Multiplications. The architecture consists of the 1T1R crossbar array, the optimized select circuitry and an ADC. The components are designed to handle the variability of the resistive switching cells, which is verified through our verified and physical compact model. We then use this architecture to compare different bit slicing approaches and discuss their trade-offs.

ACM CCS: Hardware Emerging Technologies
Analysis and design of emerging devices and systems
Emerging Architectures

Keywords: Memristive Devices, Bit Slicing, Computation-in-Memory, Vector-Matrix-Multiplications

Further Metadata

Special Issue yes
If yes, please insert the title of the special issue or the name of the guest editor: Dietmar Fey
Column “Distinguished Dissertations” no
Column “Current Research Centers” no
Column “Regular Papers” no

* These authors contributed equally.

1 Introduction

The dominance and commercial success of machine learning algorithms for the processing of images, speech and video signals [1] in the last ten years, has largely been enabled by the utilization of Graphics Processing Units (GPUs) during training [1, 2]. These successes have further lead to the development of Application Specific Integrated Circuits (ASICs), specifically targeted for machine learning workloads. Examples of such chips are the Tensor Processing Units from Google [3] or Hanguang from Alibaba [4] further improving the efficiency of the hardware for machine learning algorithms. The great performance benefits have come at the cost of exponentially increasing energy cost for training and inference [5]. During the training phase of a machine-learning algorithm the parameters of a computational model, such as a multilayer neural network, are adapted to produce distinguishable mappings of different training inputs to output categories. The goal of the training phase is to enable the network to independently match unseen inputs to a range of trained output categories during the inference phase with as high accuracies as possible. Accelerators for machine learning algorithms can also be realized based on memristive devices such as filamentary Valence Change Mechanism (VCM) cells [6-8].

VCM cells are a type of two terminal and non-volatile redox based resistive switching random access memory, in which the resistive switching is based on the movement of oxygen vacancies. Filamentary switching VCM cells consist of an electronically active electrode (AE), a mixed ionic-electronic conducting layer and an ohmic counter electrode. The AE is characterized by a high work function from the metal to the oxide, while the ohmic electrode forms a low work function interface. The oxygen vacancies are then moved, via an electrical field, near the AE. An accumulation of them at the AE interface decreases the resistance, while a reduction of their number increases the resistance of the cell [6, 9]. These devices are also called memristive devices [10]. They can be switched between at least one high resistive state (HRS) and one low resistive state (LRS), although in the context of machine learning often the equivalent conductances, low conductive state (LCS) for the HRS and high conductive state (HCS) for the LRS are used. Using the conductances has the advantage, that they are directly proportional to the current values. The transition from a LCS towards a HCS is called a SET operation. The opposite direction is termed RESET operation. VCM cells can also switch in a more gradual fashion in the RESET/ SET direction, by controlling the maximum voltage/ current. For machine learning accelerator applications, VCM devices are often integrated together with N-type metal-oxide-semiconductor (NMOS) transistors in 1T1R crossbar arrays, to prevent sneak path currents [11] and to allow for a precise programming of the devices without program disturb of half-selected devices [12]. Recently, major semiconductor companies have displayed the feasibility of realizing large-scale reliable memories based on filamentary VCM systems for memory applications, displaying the maturity and reliability of this technology [13-15]. The main selling point of using memristive devices is their ability to unite computation and memory in a single physical location, alleviating the limitations of von-Neumann architectures [16]. Because of this advantage, many architectures have been proposed, that enable Computation in Memory (CIM) for the acceleration of machine learning algorithms. These architectures often focus on the central Vector-Matrix-Multiplication (VMM) or Matrix-Matrix-Multiplication

(MMM) operations to improve the energy efficiency of the inference phase for deep neural networks [7-8, 17-18]. It should be noted, that VMM and MMM operations are not only required for machine learning, but also for example in scientific computing [19, 20]. System level analyses have already shown the benefits of memristive matrix operation accelerators, and found especially benefits regarding the energy consumption, ranging from 10x to 1000x [17, 21-22]. In this work, we are proposing and investigating such an VCM-based accelerator on the circuit level. The required theoretical background as well as the different bit slicing techniques are introduced in Section 2. In Section 3, the designed circuits will be discussed in more detail, the bit slicing approaches will be compared and their variability tolerance will be discussed. Section 4 summarizes the paper and gives an outlook for future work.

2 Background and Simulation Architecture

Figure 1 shows the block level architecture, organized around the 1T1R crossbar array. The input vectors are applied serially to a shift register and then in parallel to the 1T1R array via the Wordline (WL) drivers. The Sourcelines (SL) are connected to the read drivers or RESET drivers via the SL select circuits. Similarly, the SET drivers and the ADC stage are connected to the Bitlines (BL), via the BL select circuit.

For the simulation of the VCM devices we used the physically motivated compact model JART VCM v1b Readvar [23-26], which can describe the programming variability between different devices (device-to-device variability), between different cycles in a single device (cycle-to-cycle) and between different read operations. Additionally, it can describe different types of filamentary VCM devices such as ZrO_x [24, 26] HfO_x [27][28] TaO_x [29] and $SrTiO_3$ [30]. The parameter set we are using was fitted to ZrO_x devices to describe the reliability properties relating to read disturb, read noise and programming variability for binary VMM [24]. We use the parameter set shown in Table I of [24] for the deterministic and variability parameters. Programming variability describes the variation of conductance states during switching. It can be reduced via program verify algorithms [31]. Read disturb and read noise affect the programmed conductance levels over time. Read noise is an undirected and random process [26], while read disturb depends on the voltage polarity and amplitude. For the ZrO_x devices we have shown that read disturb does not represent a problem, when reading in the RESET direction and keeping the read voltage below 500 mV [24]. This recommendation is followed here. The transistors in the arrays and all other circuits are modeled using a commercially available 180 nm CMOS technology.

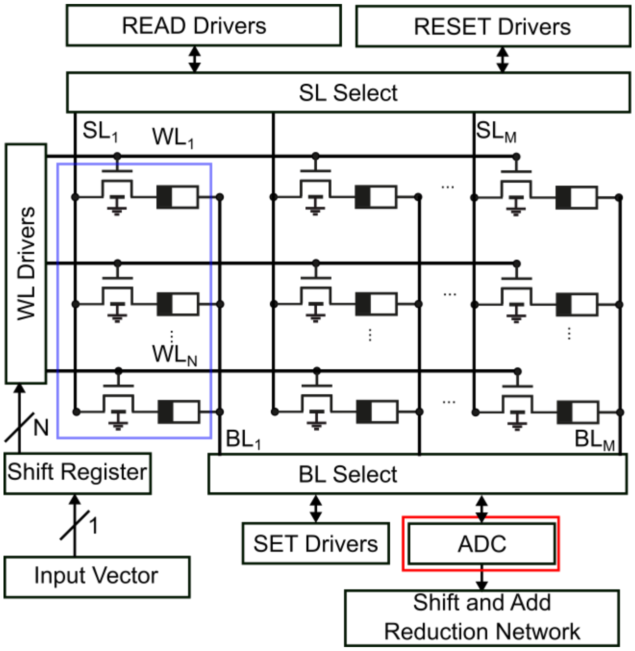

Figure 1: Block level analog VCM architecture. Bit slicing is investigated in the *N*x*M* 1T1R crossbar array (blue rectangle) or in the ADC stage (red rectangle).

To perform a VMM operation, a row of the matrix is mapped to a column of the crossbar. In the case of only binary weights, this means that '**0**' is represented by the LCS and '**1**' is represented by the HCS. In the case of multilevel weights, proportionally mapping the matrix values, by using equidistant conductance spacing, is the easiest mapping strategy. The crossbar is enabled to perform certain computational operations via the periphery. To perform VMMs, the periphery consists of SET and RESET drivers for the programming, digital to analog converters (DAC) to apply the input vectors and analog to digital converters (ADC) to convert the analog result of the VMM into a digital representation. While the DACs are often realized as buffers that can only apply binary voltages to the crossbar (0 V or V_{DD} = 5 V), the ADCs are more complex and therefore occupy a significant part of the total accelerator area and consume a lot of energy. For example, in [8] it consumed 58 % of the computing tiles power and occupied 31 % of the total area and in [32] it required more than 75 % of both energy and power. Consequently, the optimization of ADC designs is a large focus with a wide range of concepts being explored [8, 33-34]. Due to the high precision requirements of VMM for the training of deep neural networks (DNN) or for scientific computing applications, often a technique called bit slicing is used in memristive hardware accelerators. Bit slicing combines multiple lower resolution components together to realize higher resolution blocks. During each evaluation cycle, a dot product operation is computed in our proposed architecture. For a complete VMM or MMM these dot product results have to be shifted and added via digital CMOS circuits to the right bit positions [20, 35]. For the ADC to be able to convert all possible dot product results of the VMM to digital signals without any loss of information it needs a minimum resolution of [8]:

$$B_{\mathrm{ADC}} = \begin{cases} B_{\mathrm{w}} + B_{\mathrm{in}} + \lceil \log(N) \rceil, \text{if } B_{\mathrm{w}} > 1, B_{\mathrm{in}} > 1, \\ B_{\mathrm{w}} + B_{\mathrm{in}} \cdot + \lceil \log(N-1) \rceil, \text{ otherwise.} \end{cases} \quad (1)$$

In (1) B_{W} is the number of bits per weight, B_{in} is the number of bits per input and N is the number of rows in the crossbar being read out. Bit slicing can be applied to the weights (weight slicing), the input voltages (input slicing) and the number of rows that are evaluated by the ADC (ADC slicing). Input slicing can be performed, by breaking down a high-resolution input voltage into multiple equal pulses after one another at 1-Bit resolution (Pulse Frequency Modulation). Another approach for input slicing would be Pulse Width Modulation, where the proportion of 'On' time to the total pulse time period is varied. In our architecture, we only consider the 1-Bit input case. Weight slicing on the weight values can be realized in two different ways, either directly in the array or in the peripheral circuitry. In the direct approach, a multibit matrix value is mapped to different numbers of columns, depending on the resolution of the 1T1R cells. This approach is shown in Equation (2)

$$\begin{bmatrix} 3212 \\ 2103 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 2 \end{bmatrix}_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}_1 + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}_1 + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}_1, (2)$$

where the matrix on the left side represents the numerical matrix, from which the first row is mapped to the columns of the crossbar. In the case of 2-Bit resolution devices (possible values from '0' to '3') one column is required as indicated by the single matrix. If the devices only have a resolution of 1-Bit (right side) three columns of the crossbar are required. In the middle and on the right side, each row represents an individual 1T1R cell. Another approach of performing the weight slicing has been shown in an analog fashion in [33], where the individual columns are weighted differently through the ADC. Alternatively, weight slicing can be done after the ADC stage, in the digital periphery, via shift and add operations [20]. From equation (2) it can already be seen, that having higher resolution devices, saves considerable space or increases the information density of the array. Performing the weight slicing by increasing the resolution of the 1T1R elements has the additional advantage that the periphery does not have to be modified. However, it must then be designed, such that it can handle the increased resolution according to equation (1). Lastly, the bit slicing can also be applied to the ADC by adapting the number of rows that are evaluated in a given cycle. While bit slicing is conventionally used to realize higher precision components from lower precision components, it can also be used to reduce the requirements for the ADC.

3 Results

3.1 Peripheral Circuits

Figure 2 shows the selection circuits (a) as well as the ADC (b). The select circuitry for the BL and SL are structurally the same, but connected to different control signals and peripheral blocks for BL (marked in orange) and SL (marked in blue). The left branch uses a transmission gate to connect the RESET/SET drivers to the BL/SL respectively. Those drivers are required to program the VCM cells. At the opposite side of the active driver, the BL/SL is pulled to ground via an low ohmic NMOS transistor. The right branch of the select circuit is connected to the read driver /sensing stage for the BL/SL side. To increase the precision in the read path we have removed the transmission gates to reduce the IR drop. Instead, power gating is used to set this path high ohmic. Power gating is achieved via a PMOS transistor between the supply voltage and the supply voltage pin of the sensing stage. The ADC circuit is inspired from [33, 36], as the sensing stage from [33] is combined with the voltage controlled oscillator from [36].

A VMM is performed, by first turning on the read drivers. This is done by setting the 'VMM Control' signal and the 'READ' signal to '1'. The read drivers are implemented as digital buffers and apply a voltage of 2.2 V via the SL select circuit to the 1T1R crossbar. The ADC keeps the voltage at the output of the crossbar at 2 V, resulting in a voltage drop of 0.2 V across the 1T1R elements.

Then, the logical input vector is read in and applied via the Wordlines (WL_1 … $\mathrm{WL1}_{\mathrm{N}}$). The Wordline drivers (or input DACs) are realized as digital buffers, either applying 0 V or V_{DD} (5 V) to the transistor gates. This reduces the required driving power for the WL buffers, as they only have to drive a certain number of transistor gates, depending on the number of selected columns. The input vector is applied serially and fed through a shift register, to be applied in parallel to the WL drivers (see Figure 1). The correct column is chosen via the 'SELECT' signal, which is generated via a demultiplexer, filled by another shift register. In this way, single columns, multiple columns or all columns can be selected for an evaluation cycle. The maximum number of columns, which can be activated at the same

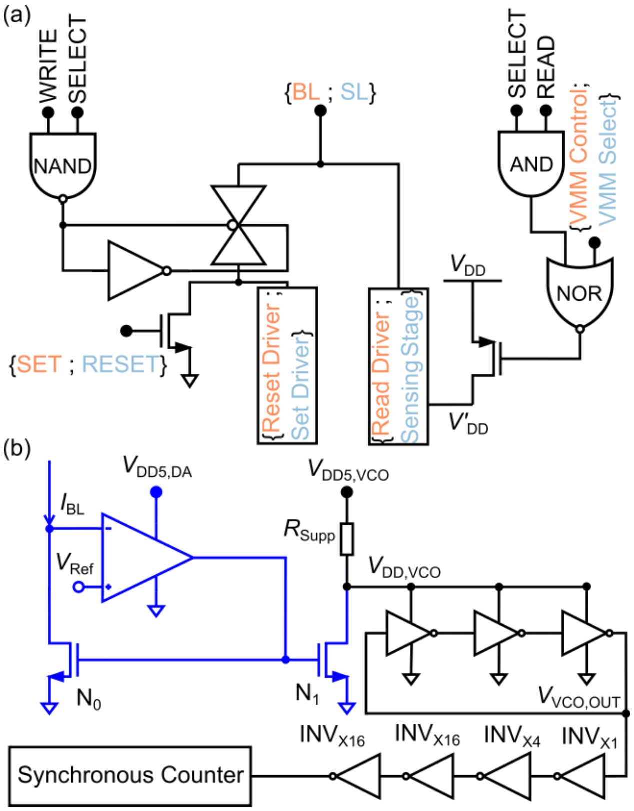

Figure 2: Circuit Blocks of the Bitline and Sourceline select circuits (a) and the ADC stage (b). In (a) signal names without curly braces are used in the BL and SL select circuits. For the signals in curly brackets the orange/ blue colored signals are applied in the BL/ SL select circuit.

time, depends on several factors, e.g. the layout size of the ADC, or the requirements of the application. For the highest throughput, all columns should be active at the same time, requiring one ADC per column. Our chosen approach allows for complete flexibility as to the number of activated columns. If multiple columns share a single ADC, e.g. every four columns have one ADC, the ‘VMM Select’ signal can be used to switch between these four columns. This approach is useful, if the columns should be weighed differently in the digital periphery as in the first cycle all first bit positions (2^0) can be evaluated, in the second cycle all second bit positions (2^1) can be evaluated and so on. Via the ‘VMM Select’ signal one can also switch between different numbers of columns to be evaluated. After all components have been activated, they are kept active for the evaluation time, which can also be chosen in a flexible fashion depending on the required accuracy. Programming of cells is performed individually, via the left side path of Figure 2 (a). To perform a write operation, first the ‘WRITE’ signal is set to one and the column is chosen via the ‘SELECT’ signal. As for the VMM, the correct row can be chosen via the logical input vector by only applying a ‘1’ to the programmed cell and applying ‘0’ to all other rows. Depending on the switching direction, the SET/RESET driver is connected at the BL/SL side while the SL/BL is set to ground via an NMOS transistor. The configuration of the periphery for the different operation modes is summarized in Table I. The signals in bold print are global signals, that are only required once for all columns.

The full ADC, shown in Figure 2 (b), can be split into three parts, namely the voltage stabilization stage (marked in blue in Figure 2), the ring oscillator stage and the counter stage. The sensing stage clamps the BL voltage to the reference voltage V_{Ref} = 2 V during the VMM operation to achieve a constant voltage drop across the 1T1R crossbar, which increases the resolution of the ADC [33]. With the read driver applying 2.2 V and V_{Ref} set to 2 V, 0.2 V drop over the 1T1R cells. This is the same voltage configuration for which the conductance levels are verified, to prevent the *IV* nonlinearity of the VCM cells from affecting the results. Via the differential amplifier it is ensured, that the resulting current levels are spaced linearly. N_1 mirrors the current from N_0 into the voltage controlled oscillator (VCO) stage, where it is converted into the supply voltage of the VCO, $V_{\mathrm{DD,VCO}}$ by R_{supp}. Thus, the VCO oscillates at different frequencies, depending on its supply voltage. To improve the counting of the pulses, the output signal of the VCO is buffered through an inverter chain. As the ADC converts a amplitude encoded signal (crossbar output current) into a time encoded signal (pulse train to be counted), higher accuracies can be achieved by increasing the conversion time [36]. The time for which the pulses are counted depends on the required precision as will be shown in the next section.

Table I: Configuration of BL/SL select circuits for the different operations in the selected column.

Signal Name	Write	Read	VMM
WRITE	1	0	0
READ	0	1	0
VMM CONTROL	0	0	1
SET/RESET	1/0 or 0/1	0/0	0/0
VMM SELECT	0	0	1
SELECT	1	1	1

3.2 Basic Performance Characteristics

In order to study the effects of the different approaches to bit slicing, first a baseline case needs to be defined. Here, we choose a simulation in which 16 rows with binary devices are read out with a single 4-Bit ADC. The 16 rows are chosen to not put to high requirements on the devices and the ADC. For the chosen array transistors (W=10 µm, L=500 nm) and the device states $N_{\mathrm{disc,\ LRS/HRS}} = 0.904\cdot10^{26}$ $1/\mathrm{m}^3$/ $0.148\cdot10^{23}$ $1/\mathrm{m}^3$ the resulting resistances of the 1T1R structure are

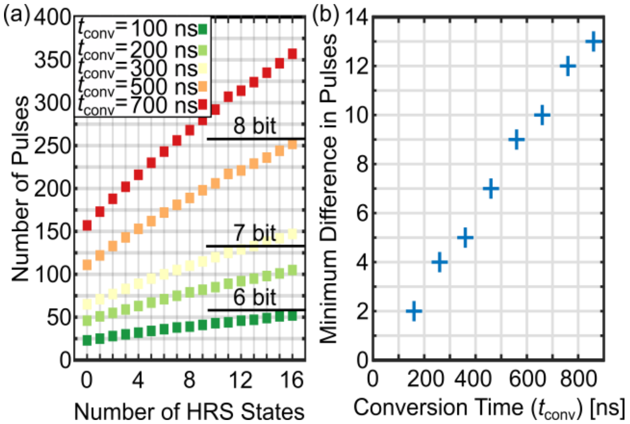

Figure 3: Generated number of pulses as a function of the number of devices in the HRS for different conversion times (a). The required counter length is indicated on the right side of (a). (b) shows the minimum difference in generated pulses for different conversion times.

4.65 kΩ/ 95.97 kΩ or 214.95 µS/ 10.42 µS. In the 1-Bit case HRS represents the logical '0' and the LRS represents the logical '1'. For 2-Bit the HRS still represents level '0',while the LRS represents level'3'. Both values were evaluated at a read voltage of 0.2 V in the RESET direction and a gate voltage of 5 V. For the investigation of 2-Bit devices we add two additional levels using proportional conductance mapping [37] at 6.78 kΩ (= 147.4 µS) for a '2' and 12.74 kΩ (= 78.47 µS) for a '1'. This leads to a HRS/LRS ratio of around 20. For this case, Figure 3 (a) shows the number of generated pulses as a function of the number of devices in the HRS and for different conversion times. In this simulation, all WL inputs are at V_{DD}. A higher number of HRS represents smaller numerical values stored in the matrix and results in a larger number of generated pulses, as the smaller crossbar current (I_{BL}) is mirrored from N_0 to N_1 and results in a bigger voltage drop over R_{Supp}. This in turn reduces the supply voltage of the VCO ($V_{\text{DD,VCO}}$) which reduces its oscillation frequency. If the conversion time is increased, the number of generated pulses is increased linearly. This makes distinguishing the different levels easier, as the differences between adjacent levels are increased linearly as well. The black solid lines at the right side of Figure 3 (a) indicate the required counter length which increases with increasing conversion times. Therefore, there exists a tradeoff between accuracy and variability tolerance on the one side and energy consumption and area on the other side, as longer counters consume more energy and require more space, but enable longer conversion times. Figure 3 (b) shows the minimum difference between the number of generated pulses over the conversion time for the same inputs as in (a). As expected from (a), the difference increases very linearly from two at 160 ns conversion time to 12 for a conversion time of 760 ns. The conversion time consists of a setup time until the VCO reaches its correct frequency and the actual counting time. Based on the results from Figure 3 we can see how the ADC can trade off between low precision, low energy consumption and small area towards high precision, high energy consumption and large area. In this way, the operational parameters and design choices are made to follow different optimization routes depending, for example, on the device technology, the timing, the energy or the accuracy requirements of the application. From these initial simulations we define our baseline case as a conversion time of 260 ns for a minimum guaranteed pulse difference of four. Apart from being able to tolerate more device variability, the guaranteed minimum pulse difference also allows the ADC to distinguish between more dot products. Those dot products can be realized by using multilevel devices with additional conductance states between the minimum and maximum values. It should be noted, that the worst case for 1-Bit devices is observed for the dot products (i) $(V_{\text{DD}}\ 0\ \ldots\ 0) \cdot (\text{LRS HRS} \ldots \text{HRS})^T = 1$ and (ii) $(V_{\text{DD}}\ V_{\text{DD}} \ldots V_{\text{DD}}) \cdot (\text{HRS HRS} \ldots \text{HRS})^T = 0$. In this case the ADC has to differentiate between 16 selected HRS states (= 95.92 kΩ/16 ≈ 6 kΩ) and one selected LRS (4.65 kΩ). For this case, the ADC produces a difference in the number of generated pulses of one for the baseline case (260 ns, 16 binary devices). To guarantee a perfect precision, this case needs to be distinguished. However, this worst case will occur with a very low probability. To mitigate this difficulty, higher HRS/LRS are required or less devices can be read out at the same time.

3.3 Comparing Different Bit Slicing Cases

In our analysis we compared two different types of bit slicing, namely weight slicing between 1-Bit and 2-Bit devices and ADC slicing, where either 16, 8 or 4 rows are read out at the same time. The 1-Bit and 16 rows case was previously discussed as the baseline case for a conversion time of 260 ns, giving a guaranteed pulse difference of four in Figure 3. The three other bit slicing cases are then simulated for conversion times long

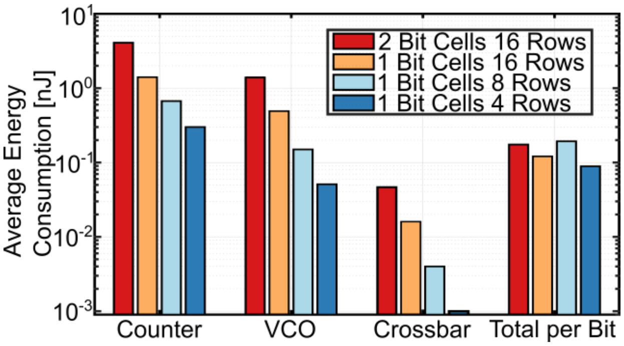

Figure 4: Average energy consumption of the different circuit components for the different bit slicing approaches.

enough to guarantee the same pulse difference, i.e. the same accuracy or noise resilience. In the case of 2-Bit devices this requires the counter to run for a longer time than in the case of 1-Bit devices, as the additional levels are added in between the 1-Bit levels. The longer run time of the counter can also require a bigger counter circuit, as indicated in Figure 3 (a). Both factors increase the energy consumption, but a higher number of performed computations compensates them. In the case of using only 8 rows or 4 rows, the counter can be implemented smaller, as the maximum number of pulses to be counted is smaller. However, to achieve the same number of bit operations, the computation of 8 rows has to be repeated twice. For 4 rows it must be repeated four times. Under the requirement of the same precision, the conversion time is increased to 760 ns for the 2-Bit 16 rows case and it is decreased to 130 ns for 1-Bit/8 rows and decreased to 65 ns for 1-Bit/4 rows. The counter width required is 8 Bit in the baseline case, 9-Bit in the 2-Bit/16 rows case, 7-Bit in the 1-Bit/8 rows case and 5-Bit in the 1-Bit weight/4 rows case. The average energy consumption of the different components and the average total energy per bit are shown in Figure 4. For the total energy per bit the energies of the three components are added together and then divided by the number of bit operations. This number is 32 for 2-Bit 16 rows, 16 for 1-Bit 16 rows, 8 for 1-Bit 8 rows and 4 for 1-Bit 4 rows. From the results it can be seen, that the counter is the largest energy consumer, followed by the VCO and then the crossbar. Additionally, the results suggest that it is more energy efficient in the given architecture to perform many small dot product operations with 1-Bit devices, as those can be finished in a much faster time and with smaller counters.

3.4 Variability Tolerance

VCM devices show different types of variability, such as device-to-device (d2d) variability, cycle-to-cycle (c2c) variability and read-to-read variability [38]. For the considered use case of VMM for machine learning accelerators, the most important variability effects are read disturb, read noise and programming variability [24]. Regarding the impact of read disturb, we showed in [24] that reading in the RESET direction is preferable to the SET direction and that read voltages up to 0.5 V over single devices are possible in the case of binary devices. As the read voltage here is 0.2 V in the RESET direction, read disturb will likely not be an issue. Regarding the read noise, we calculated it at the different conductance levels. Table II summarizes the VCM device conductances of the four considered levels and the minimum, median and maximum values for the amount of read noise and for the number of oxygen vacancies at these levels. With the chosen transistor dimensioning (W=10 µm, L=500 nm) the conductance of the 1T1R structure is almost completely determined by the resistances of the VCM cells. The JART VCM v1b Readvar model from [26] is then used to simulate the minimum, median and maximum amount of $\Delta G/G$ at the four different conductance levels, under the same read conditions as described above. The read variability model first discretizes the number of oxygen vacancies in the disc and then changes this number as described in [26]. The filamentary oxide in the JART VCM v1b model is simplified via a well conducting oxygen reservoir, the plug, and the variable resistance disc region. For the discretization, the disc volume is multiplied with the oxygen vacancy concentration at the different levels. The worst case noise at a given vacancy concentration is then obtained for the smallest filament volume, as this gives the smallest number of oxygen vacancies. Adding or removing one vacancy has then a larger influence on the conductance, compared to larger disc volumes. The rightmost column in Table II shows the floored number of vacancies at the different conductance levels for the different filament volumes. The minimum number of vacancies leads to the maximum $\Delta G/G$. As can be seen from the results, the worst-case noise can be very significant at the lower conductance levels (up to 1). The number of vacancies in this case is only two. However, most devices will have filament volumes close to the median case, where the read noise is between 5 % at the smallest conductance and 0.8 % at the highest conductance level. This read noise levels are then the best case accuracies for the different levels for the median devices. To program multilevel devices usually write verify algorithms are used [31, 39]. Even if it was possible to program levels with a 100 % accuracy, the intrinsic read noise would still lead to some uncertainty at the different conductance levels.

Table II: Minimum, Median and Maximum Read Noise Impact for the Considered Conductance Levels.

Level	G_{VCM} [µS]	Min/Med/Max $\Delta G/G$	⌊Max/Med/Min # of vacancies⌋
0	10.44	0.01/0.05/1	139/33/2
1	79.54	0.004/0.015/0.3	459/110/6
2	151.2	0.0024/0.01/0.19	672/161/10
3	223.1	0.0018/0.008/0.14	848/203/12

With the read noise results we can reconsider the 1-Bit worst case from Section 3.1, between 16 selected HRS or level 0 states (= 16*10.44 µS = 167.04 µS) and one selected LRS or level 3 (223.1 µS). In the JART VCM v1b Readvar model the read noise can change the number of vacancies by two at most. In the worst case, all the devices at level zero gain two additional vacancies, increasing their conductance, while the single level 3 device has two vacancies removed, decreasing its conductance. In this case, the equivalent conductance will be increased by 10 % for the level 0 devices and it will be decreased by 1.6 % for the level 3 device. The two conductance levels are then 183.74 µS and 219.53 µS. With these worst case conductances considering read noise we can calculate the programming accuracy at which the two levels will overlap, making a distinction impossible. This is calculated by 219.53 µS/183.74 µS = 1.19, meaning that the program verify algorithms have to be able to achieve errors smaller than 9.5 %. With the given ADC two neighboring conductance states can in principle always be resolved, if the conversion time is increased. In practice, again it would be a better solution to use higher conductance ratios, or to relax the accuracy requirements to allow for some errors in the VMM. With regard to the bit slicing approaches higher device resolutions make the differentiation much more difficult, as the levels are here inserted between the 1-Bit levels. Then the conductances have to be spread out over a larger range or fewer rows have to be read out at the same time. Reading out fewer rows at a time does not directly affect the accuracy, due to the linear conductance spacing. However, it improves the energy consumption as shown in section 3.3.

4 Conclusion

In this work, we have presented a CIM analog core architecture based on filamentary VCM devices. We have identified areas where different bit slicing techniques can be applied and shown the associated energy and latency tradeoffs between them. Understanding these tradeoffs is important, for future design space exploration. With our proposed ADC concept, it is possible to trade off accuracy for latency and energy consumption, especially interesting for machine learning or neuromorphic computing algorithms in which the accuracy of individual VMM might not directly affect the accuracy of the algorithm. Furthermore, we have discussed the effect of VCM intrinsic reliability issues and their impact on the architecture.

Funding

This work was in part funded by the German Research Foundation (DFG) under Grant No. SFB 917, in part by the Federal Ministry of Education and Research (BMBF, Germany) in the Project NEUROTEC (Project Nos. 16ME0398K and 16ME0399) and it is based on the Jülich Aachen Research Alliance (JARA-FIT).

Literature

[1] Y. LeCun, Y. Bengio and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436-444, 2015.

[2] V. Sze, Y. Chen, J. Emer, A. Suleiman and Z. Zhang, “Hardware for machine learning: Challenges and opportunities,” 2017 IEEE Custom Integrated Circuits Conference (CICC), pp. 1-8, 2018.

[3] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, R. Boyle, P. Cantin, C. Chao, C. Clark, J. Coriell, M. Daley, M. Dau, J. Dean, B. Gelb, T. V. Ghaemmaghami, R. Gottipati, W. Gulland, R. Hagmann, C. R. Ho, D. Hogberg, J. Hu, R. Hundt, D. Hurt, J. Ibarz, A. Jaffey, A. Jaworski, A. Kaplan, H. Khaitan, D. Killebrew, A. Koch, N. Kumar, S. Lacy, J. Laudon, J. Law, D. Le, C. Leary, Z. Liu, K. Lucke, A. Lundin, G. MacKean, A. Maggiore, M. Mahony, K. Miller, R. Nagarajan, R. Narayanaswami, R. Ni, K. Nix, T. Norrie, M. Omernick, N. Penukonda, A. Phelps, J. Ross, M. Ross, A. Salek, E. Samadiani, C. Severn, G. Sizikov, M. Snelham, J. Souter, D. Steinberg, A. Swing, M. Tan, G. Thorson, B. Tian, H. Toma, E. Tuttle, V. Vasudevan, R. Walter, W. Wang, E. Wilcox and D. H. Yoon, “In-Datacenter Performance Analysis of a Tensor Processing Unit,” *SIGARCH Comput. Archit. News*, vol. 45, pp. 1–12, 2017.

[4] Y. Jiao, L. Han and X. Long, “Hanguang 800 NPU – The Ultimate AI Inference Solution for Data Centers,” 2020 IEEE Hot Chips 32 Symposium (HCS) , pp. 1-29, 2020.

[5] N. C. Thompson, K. H. Greenewald, K. Lee and G. F. Manso, “The Computational Limits of Deep Learning,” *arXiv:2007.05558v2*, 2022.

[6] R. Dittmann, S. Menzel and R. Waser, “Nanoionic memristive phenomena in metal oxides: the Valence Change Mechanism,” *Adv. Phys.*, vol. 70, pp. 155-349 , 2021.

[7] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang and Y. Xie, “PRIME: A Novel Processing-in-Memory Architecture for Neural Network Computation in ReRAM-Based Main Memory,”, pp. 27-39, 2016.

[8] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams and V. Srikumar, “ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars,”, pp. 14-26, 2016.

[9] R. Waser, R. Dittmann, G. Staikov and K. Szot, “Redox-Based Resistive Switching Memories - Nanoionic Mechanisms, Prospects, and Challenges,” *Adv. Mater.*, vol. 21, pp. 2632-2663, 2009.

[10] D. B. Strukov, G. S. Snider, D. R. Stewart and R. S. Williams, “The missing memristor found,” *Nature*, vol. 453, pp. 80-83, 2008.

[11] M. A. Zidan, H. A. H. Fahmy, M. M. Hussain and K. N. Salama, “Memristor-based memory: The sneak paths problem and solutions,” *Microelectronics Journal*, vol. 44, pp. 176-183, 2013.

[12] H. Kim, M. R. Mahmoodi, H. Nili and D. B. Strukov, “4K-memristor analog-grade passive crossbar circuit,” *Nat. Commun.*, vol. 12, pp. 5198/1-11, 2021.

[13] C. Chou, Z. Lin, C. Lai, C. Su, P. Tseng, W. Chen, W. Tsai, W. Chu, T. Ong, H. Chuang, Y. Chih and T. J. Chang, “A 22nm 96KX144 RRAM Macro with a Self-Tracking Reference and a Low Ripple Charge Pump to Achieve a Configurable Read Window and a Wide Operating Voltage Range,” *2020 IEEE Symposium on VLSI Circuits*, pp. 1-2, 2020.

[14] P. Jain, U. Arslan, M. Sekhar, B. C. Lin, L. Wei, T. Sahu, J. Alzate-vinasco, A. Vangapaty, M. Meterelliyoz, N. Strutt, A. B. Chen, P. Hentges, P. A. Quintero, C. Connor, O. Golonzka, K. Fischer and F. Hamzaoglu, “13.2 A 3.6Mb 10.1Mb/mm2 Embedded Non-Volatile ReRAM Macro in 22nm FinFET Technology with Adaptive Forming/Set/Reset Schemes Yielding Down to 0.5V with Sensing Time of 5ns at 0.7V,” *IEEE International Solid-State Circuits Conference*, pp. 212-214, 2019.

[15] N. Kopperberg, S. Wiefels, K. Hofmann, J. Otterstedt, D. J. Wouters, R. Waser and S. Menzel, “Endurance of 2 Mbit based BEOL integrated ReRAM,” *IEEE Access*, vol. 10, pp. 122696 - 122705, 2022.

[16] M. Horowitz, “Computing's energy problem (and what we can do about it),”, pp. 10-14, 2014.

[17] A. Ankit, I. E. Hajj, S. R. Chalamalasetti, S. Agarwal, M. Marinella, M. Foltin, J. P. Strachan, D. Milojicic, W. Hwu and K. Roy, “PANTHER: A Programmable Architecture for Neural Network Training Harnessing Energy-Efficient ReRAM,” *IEEE Trans. Comput.*, vol. 69, pp. 1128–1142, 2020.

[18] C. Li, J. Ignowski, X. Sheng, R. Wessel, B. Jaffe, J. Ingemi, C. Graves and J. P. Strachan, “CMOS-integrated nanoscale memristive crossbars for CNN and optimization acceleration,” *2020 IEEE International Memory Workshop (IMW)*, pp. 1-4, 2020.

[19] J. J. Dongarra, J. Du Croz, S. Hammarling and I. S. Duff, “A Set of Level 3 Basic Linear Algebra Subprograms,” *ACM Trans. Math. Softw.*, vol. 16, pp. 1–17, 1990.

[20] B. Feinberg, U. K. R. Vengalam, N. Whitehair, S. Wang and E. Ipek, “Enabling Scientific Computing on Memristive Accelerators,”, pp. 367-382, 2018.

[21] A. Ankit, I. E. Hajj, S. R. Chalamalasetti, G. Ndu, M. Foltin, R. S. Williams, P. Faraboschi, W. W. Hwu, J. P. Strachan, K. Roy and D. S. Milojicic, “PUMA: A Programmable Ultra-Efficient Memristor-Based Accelerator for Machine Learning Inference,”, pp. 715–731, 2019.

[22] W. Wan, R. Kubendran, C. Schaefer, S. B. Eryilmaz, W. Zhang, D. Wu, S. Deiss, P. Raina, H. Qian, B. Gao, S. Joshi, H. Wu, H. P. Wong and G. Cauwenberghs, “A compute-in-memory chip based on resistive random-access memory,” *Nature*, vol. 608, pp. 504-512, 2022.

[23] JART, “Juelich Aachen Resistive Switching Tools (JART),”, vol., pp., 2019.

[24] C. Bengel, J. Mohr, S. Wiefels, A. Singh, A. Gebregiorgis, R. Bishnoi, S. Hamdioui, R. Waser, D. Wouters and S. Menzel, “Reliability aspects of binary vector-matrix-multiplications using ReRAM devices,” *Neuromorphic Computing and Engineering*, vol. 2, pp. 034001, 2022.

[25] C. Bengel, A. Siemon, F. Cüppers, S. Hoffmann-Eifert, A. Hardtdegen, M. von Witzleben, L. Helllmich, R. Waser and S. Menzel, “Variability-Aware Modeling of Filamentary Oxide based Bipolar Resistive Switching Cells Using SPICE Level Compact Models,” *IEEE Trans. Circuits Syst. I Reg. Papers,* vol. 67, pp. 4618-4630, 2020.

[26] S. Wiefels, C. Bengel, N. Kopperberg, K. Zhang, R. Waser and S. Menzel, “HRS Instability in Oxide based Bipolar Resistive Switching Cells,” *IEEE Trans. Electron Devices*, vol. 67, pp. 4208-4215, 2020.

[27] F. Cueppers, S. Menzel, C. Bengel, A. Hardtdegen, M. von Witzleben, U. Boettger, R. Waser and S. Hoffmann-Eifert, “Exploiting the switching dynamics of HfO_2-based ReRAM devices for reliable analog memristive behavior,” *APL Mater.*, vol. 7, pp. 91105/1-9, 2019.

[28] C. Bengel, F.Cüppers, M. Payvand, R. Dittmann, R. Waser, S. Hoffmann-Eifert and S. Menzel, “Utilizing the Switching Stochasticity of HfO_2/TiO_x-Based ReRAM Devices and the Concept of Multiple Devices for the Classification of Overlapping and Noisy Patterns,” *Frontiers in Neuroscience*, vol. 15, pp. 621, 2021.

[29] C. Bengel, A. Siemon, V. Rana and S. Menzel, “Implementation of Multinary Lukasiewicz Logic using Memristive Devices,” *2021 IEEE International Symposium on Circuits and Systems (ISCAS),* Daegu, Korea, 22-28 May, 2021.

[30] C. La Torre, “Physics-Based Compact Modeling of Valence-Change-Based Resistive Switching Devices,”, PhD thesis, RWTH Aachen, 2019.

[31] E. Perez, M. K. Mahadevaiah, E. P. Quesada and C. Wenger, “Variability and Energy Consumption Tradeoffs in Multilevel Programming of RRAM Arrays,” *IEEE Trans. Electron Devices*, vol. 68, pp. 2693-2698, 2021.

[32] P. Yao, H. Wu, B. Gao, J. Tang, Q. Zhang, W. Zhang, J. J. Yang and H. Qian, “Fully hardware-implemented memristor convolutional neural network,” *Nature*, vol. 577, pp. 641-646, 2020.

[33] A. Singh, M. Abu Lebdeh, A. Gebregiorgis, R. Bishnoi, R. V. Joshi and S. Hamdioui, “SRIF: Scalable and Reliable Integrate and Fire Circuit ADC for Memristor-Based CIM Architectures,” *IEEE Transactions on Circuits and Systems I: Regul*, vol., pp. 1-14, 2021.

[34] J. M. Hung, C. X. Xue, H. Y. Kao, Y. H. Huang, F. C. Chang, S. P. Huang, T. W. Liu, C. J. Jhang, C. Su, W. S. Khwa, C. C. Lo, R. S. Liu, C. C. Hsieh, K. T. Tang, M. S. Ho, C. C. Chou, Y. D. Chih, T. Y. J. Chang and M. F. Chang, “A four-megabit compute-in-memory macro with eight-bit precision based on CMOS and resistive random-access memory for AI edge devices,” *Nat. Electron.*, vol. 4, pp. 921+, 2021.

[35] M. Zahedi, M. Mayahinia, M. Abu Lebdeh, S. Wong and S. Hamdioui, “Efficient Organization of Digital Periphery to Support Integer Datatype for Memristor-Based CIM,”, pp. 216-221, 2020.

[36] M. Mayahinia, A. Singh, C. Bengel, S. Wiefels, S. Menzel, D. J. Wouters, A. Gebregiorgis, R. Bishnoi, R. Joshi and S. Hamdioui, “A Novel Voltage Controlled Oscillation based ADC Design for Computation-in-Memory Using Emerging ReRAMs,” *ACM Journal on Emerging Technologies in Computing Systems*, vol. 18, pp. 32, 2022.

[37] T. P. Xiao, B. Feinberg, C. H. Bennett, V. Agrawal, P. Saxena, V. Prabhakar, K. Ramkumar, H. Medu, V. Raghavan, R. Chettuvetty, S. Agarwal and M. J. Marinella, “An Accurate, Error-Tolerant, and Energy-Efficient Neural Network Inference Engine Based on SONOS Analog Memory,” *IEEE Transactions on Circuits and Systems I: Regul*, vol. 69, pp. 1480-1493, 2022.

[38] S. Wiefels, “Reliability aspects in resistively switching valence change memory cells,” PhD thesis, RWTH Aachen, 2021.

[39] B. Q. Le, A. Grossi, E. Vianello, T. Wu, G. Lama, E. Beigné, H.-S. P. Wong and S. Mitra, “Resistive RAM With Multiple Bits Per Cell: Array-Level Demonstration of 3 Bits Per Cell,” *IEEE Trans. Electron Devices*, vol. 66, pp. 641-646, 2019.

Christopher Bengel received the B.Sc. and M.Sc. degrees in electrical engineering from RWTH Aachen University, Aachen, Germany, in 2016 and 2018, respectively. He is currently pursuing the Ph.D. degree with the IWE 2 RWTH Aachen, Aachen, Germany, with a focus on modeling resistive switching devices for Computation-in-Memory and neuromorphic computing applications.

Address: RWTH Aachen, Institute of electronic Materials II D-52074 Aachen, E-Mail: bengel@iwe.rwth-aachen.de

Leon Minh Dixius Leon Dixius was born in Trier, Germany, in 1999. He received his B.Sc. degree in electrical engineering from RWTH Aachen University in 2021, where he is currently pursuing his M.Sc. degree in electrical engineering.

Address: RWTH Aachen, Institute of electronic Materials II D-52074 Aachen, E-Mail: leon.dixius@rwth-aachen.de

Prof. Dr. Rainer Waser (M'79) received the Ph.D. degree in physical chemistry from the University of Darmstadt, Darmstadt, Germany, in 1984. In 1992, he joined the Faculty of Electrical Engineering and Information Technology, RWTH Aachen University, Aachen, Germany, as a Professor and the Director of the Institute of Solid State Research, Forschungszentrum Jülich, Jülich, Germany, in 1997. Prof. Waser was a recipient of the prestigious Gottfried Wilhelm Leibniz Preis in 2014.

Address: RWTH Aachen, Institute of Electronic Materials II D-52074 Aachen, & Forschungszentrum Jülich Peter Grünberg Institut 7 & 10 D-52428 Jülich, E-Mail: waser@iwe.rwth-aachen.de

Dr. Dirk J. Wouters received the master’s and Ph.D. degrees in electrical engineering from the University of Leuven, Leuven, Belgium, in 1982 and 1989, respectively. In 2014, he joined the Institute of Electronic Materials, RWTH Aachen University, Aachen, Germany, where he focuses on the research of metal-oxide-based RRAM. He is currently the leader of the neuromorphic computing group.

Address: RWTH Aachen, Institute of Electronic Materials II D-52074 Aachen, E-Mail: wouters@iwe.rwth-aachen.de

Dr. Stephan Menzel (M’12) was born in Bremen, Germany. He received the Diploma degree and the Ph.D. degree (summa cum laude) in electrical engineering from RWTH Aachen University, Aachen, Germany, in 2005 and 2012, respectively. He is currently a Senior Researcher with Peter-Grünberg-Institut 7, Forschungszentrum Jülich GmbH, Jülich, Germany, where he is leading the Simulation Group.

Address: Forschungszentrum Jülich, Peter-Grünberg-Institut 7, D-52428~Jülich, E-Mail: st.menzel@fz-juelich.de