

# In-Memory Binary Vector–Matrix Multiplication Based on Complementary Resistive Switches

Tobias Ziegler, Rainer Waser, Dirk J. Wouters, and Stephan Menzel\*

This work studies a computation in-memory concept for binary multiply-accumulate operations based on complementary resistive switches (CRS). By exploiting the in-memory boolean exclusive OR (XOR) operation of single CRS devices, the Hamming Distance (HD) can be calculated if the center electrodes of multiple CRS cells are connected. This HD is linearly encoded in the voltage drop of the common electrode, and from it the result of a binary multiply-accumulate operation can be calculated. A small-scale demonstration is experimentally realized and the feasibility of the in-memory computation concept is confirmed. A simulation study identifies the low resistance state (LRS) variability as the main reason for the variations in the output voltage. The application as a potential hardware accelerator for the inference step of binary neural networks is investigated. Therefore, a 1-layer fully connected neural network is trained on a binarized version of the MNIST data set and the inference step of the test data set is simulated. The concept achieves a prediction accuracy of approximately 86%.

## 1. Introduction

With artificial neural networks (ANNs) becoming more and more powerful and with the slowdown of complementary metal–oxide–semiconductor (CMOS) scaling, the Von Neumann memory wall is becoming an increasingly prominent problem for ANN hardware systems.<sup>[1,2]</sup> Large neural networks especially suffer from this because not all computational information necessary can be stored in the cache memory, and costly

communication with higher level storage is necessary.<sup>[3]</sup> In software, techniques such as pruning, weight reuse, or reducing the quantization are used to create less complex but still accurate ANNs.<sup>[4]</sup> A promising simplification from the hardware perspective is the aforementioned reduction in quantization due to its reduced computational complexity.

The number of quantization levels can even go down to the bare minimum of two levels and results in binary neural networks (bNNs) which have been heavily studied in recent years.<sup>[5–8]</sup> These networks use the values 1 and –1 to encode their weights and activations during the inference step making them the most efficient ANNs possible to compute in hardware.<sup>[9]</sup> With ongoing improvements in prediction accuracy and the development of new hard-

ware accelerators, these networks are auspicious candidates for computing ANNs on edge devices.

Apart from the use of CMOS accelerators, new emerging technologies based on resistive switching devices can play a crucial role in this advancement.<sup>[10]</sup> These devices can mimic the synaptic weights in ANNs and, configured in a crossbar architecture, enable fast analog computations of vector–matrix multiplications, which are the main operations in ANNs.<sup>[11,12]</sup>


One promising class of resistive switching devices relies on redox reactions and is therefore called redox-based resistive switching devices, also known as redox-based random access memory (ReRAM).<sup>[13]</sup> They are typically based on metal–oxide–metal stacks and can change the conduction through the oxide layer based on electric signals applied to the metal electrodes. These changes in resistance from a high resistive state (HRS) to a low resistance state (LRS) and vice versa originate from ionic movements in the oxide layer and concurrent redox reactions.<sup>[14]</sup>

There are already many realizations of ReRAM crossbar arrays used as accelerators for vector–matrix multiplication.<sup>[15–17]</sup> One common type uses Kirchhoff's current law to do the computation where the calculation result is encoded in the resulting current. To compute this result, the current has to be sensed, which becomes more difficult the lower the current is. This leads to a trade-off in the circuit design as a lower current reduces the energy consumption. Another commonly used architecture builds up a resistive voltage divider between a sense resistance and the resistive crossbar array. In this architecture, the result of the computation is encoded in the voltage drop of the voltage divider.<sup>[18]</sup> One design challenge of this architecture is the

T. Ziegler, Prof. R. Waser, Dr. D. J. Wouters  
JARA-FIT and Institute of Materials in Electrical Engineering and  
Information Technology II  
RWTH Aachen University  
Sommerfeldstraße 24, Aachen 52074, Germany

Prof. R. Waser, Dr. S. Menzel  
JARA-FIT and Peter Grünberg Institute 7  
Forschungszentrum Jülich GmbH  
Wilhelm-Johnen-Straße, Jülich 52428, Germany  
E-mail: st.menzel@fz-juelich.de

Prof. R. Waser  
JARA-FIT and Peter Grünberg Institute 10  
Forschungszentrum Jülich GmbH  
Wilhelm-Johnen-Straße, Jülich 52428, Germany

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.202000134>.

© 2020 The Authors. Published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202000134

nonlinear dependence between the computational result and the voltage drop, which may increase the overhead for the readout circuitry.

The concept explored in this work also uses the voltage divider effect to encode the result of the binary vector–matrix multiplication, but still shows a linear dependence of the output voltage on the computational result. The slope of this linear encoding only depends on the resistance ratio between the HRS and LRS. Thus, ReRAM cells with high LRS states can be used, paving the way for low power applications. With a resistance ratio of approximately 100, nearly the full read voltage is used for the encoding, which helps to separate the computation results from each other. These properties make this concept a promising alternative as an accelerator for binary vector–matrix multiplications and possibly simplifies the design of the peripheral circuitry.

## 2. Concept and Hardware Realization

### 2.1. Concept

ANNs are inspired by biological neural networks, wherein the neurons are connected by synaptic weights. These weights are adjusted during the training procedure of the network to better predict the underlying data used for the training. During the inference step, when a signal propagates through the ANN, the inputs of each neuron are multiplied by the corresponding weights and the results are summed up. This operation is called multiply-accumulate operation and has a large contribution to the energy consumption of ANNs.<sup>[19]</sup>

A binary multiply-accumulate (bMAC) operation of two binary vectors  $\mathbf{x}$  and  $\mathbf{y}$  (with  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  and  $\mathbf{y} = (y_1, y_2, \dots, y_n)$ , where  $x_i, y_i \in [1, -1]$ ) can be computed exploiting boolean logic. To this end, each entry has to be transformed into a boolean value, e.g.,  $x_{i,\text{new}} = (x_{i,\text{old}} + 1)/2$ . Then, a bitwise exclusive OR (XOR) comparison is performed between these two vectors and the resulting vector is accumulated (summation of the “1”-bits). This accumulation calculates the Hamming Distance (HD), which describes how many digits of two binary words are different. The HD needs to be retransformed to receive the same result as in the original bMAC operation by

$\text{bMAC} = n - 2 \times \text{HD}$ , with  $n$  being the length of the compared vectors.

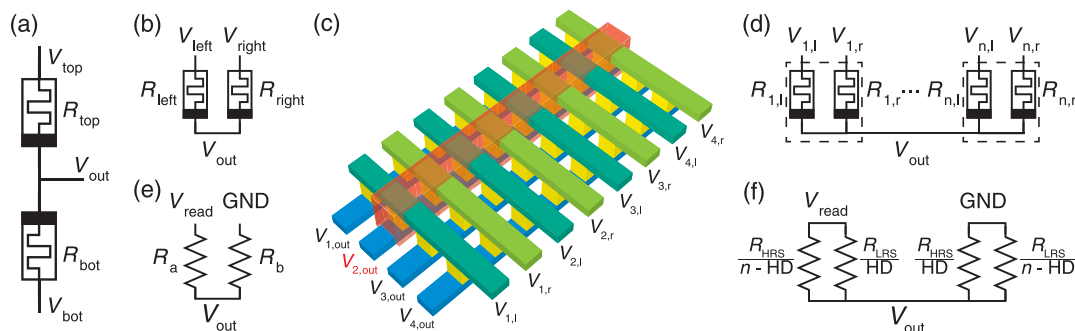
Complementary resistive switch (CRS) cells were introduced by Linn et al. to resolve the sneak path problem in passive crossbar arrays.<sup>[20]</sup> A CRS cell consists of two antiseriably connected ReRAM devices with a complementary encoding where one of them is always in the HRS. Therefore, the total resistance of the CRS cell is always in the HRS state which prevents sneak paths during the destructive readout.<sup>[20]</sup>

It was also shown that CRS cells can perform many logic operations by applying certain voltage patterns.<sup>[21,22]</sup> One promising logic operation is the CRS-based XOR operation enabled by measuring the voltage drop across the center electrode. This logic operation also functions without switching the device state. With a single CRS cell, which is shown in Figure 1a,b in a vertical and horizontal configuration, this operation can be achieved by using the binary encoding ( $b_i, b_s$ ) which is specified in Table 1a,b. The voltage divider formed by the elements of the CRS cell only leads to a relevant voltage output at the shared electrode if the corresponding XOR operation results in a “1.” This XOR operation is also shown in Table 1c.

CRS cells can also be configured in a passive crossbar structure, as shown in Figure 1c. In this configuration, the center electrodes of parallel CRS cells intentionally share one electrode. This disables the intrinsic sneak path prevention of the CRS cell, but does not influence the parallel read operation in the crossbar array. Nevertheless, for programming each cell a selective device is necessary. Typically transistors are used for that and as long as the on-resistance of the transistor is a fraction of the LRS, the influence of it during the read process can be neglected. For the ongoing discussion, a slice of a passive crossbar array (highlighted cells of Figure 1c) will be analyzed. A circuit diagram of such a line of the array is shown in Figure 1d.

With the help of the earlier introduced XOR operation, Figure 1b can be rearranged to Figure 1e. The resistances  $R_a$  and  $R_b$  can then be specified based on the result of the XOR operation of the stored and input bit by  $R_a = R_{\text{LRS}} \times \text{XOR}(b_i, b_s) + R_{\text{HRS}} \times (1 - \text{XOR}(b_i, b_s))$  and  $R_b = R_{\text{LRS}} \times (1 - \text{XOR}(b_i, b_s)) + R_{\text{HRS}} \times \text{XOR}(b_i, b_s)$ .

This rearrangement facilitates creating the equivalent circuit for a line of CRS cells with a common electrode for arbitrary



**Figure 1.** Circuit diagrams. a) Circuit schematic of a CRS cell as developed by Linn et al.<sup>[20]</sup> The voltage drop at the center electrode can be monitored. b) Lateral configuration of a CRS cell. c) 3D illustration of a lateral CRS-based passive crossbar array. The highlighted part corresponds to the circuit diagram of (d). d) Circuit schematic of multiple lateral CRS cells in parallel with a common center electrode. e) Equivalent circuit for one lateral CRS cell with an arbitrary binary input ( $b_i$ ) applied. Using the encoding in Table 1a,b, the resistances  $R_a$  and  $R_b$  can be calculated based on the stored pattern ( $b_s$ ) and the result of XOR( $b_i, b_s$ ). f) Equivalent circuit for an arbitrary input and stored pattern of a shared electrode of multiple CRS cells in a line array. The number of CRS cells connected by the shared electrode is defined by  $n$ .

**Table 1.** XOR encoding for a single CRS cell and constant simulation parameters.

a) Input bit			
$b_i$	$V_{\text{left}}$	$V_{\text{right}}$	
"0"	Ground	$V_{\text{read}}$	
"1"	$V_{\text{read}}$	Ground	
b) Stored bit			
$b_s$	$R_{\text{left}}$	$R_{\text{right}}$	
"0"	$R_{\text{LRS}}$	$R_{\text{HRS}}$	
"1"	$R_{\text{HRS}}$	$R_{\text{LRS}}$	
c) $V_{\text{out}}$ of CRS			
–	$b_s = 0$	$b_s = 1$	
$b_i = 0$	$\approx 0$	$\approx V_{\text{read}}$	
$b_i = 1$	$\approx V_{\text{read}}$	$\approx 0$	
d) Constant simulation parameters			
$A [\mu\text{m}^2]$	$\phi [\text{eV}]$	$m [\text{kg}]$	$R_{\text{series}} [\Omega]$
0.04	0.7	$1.19 \times 9.1 \times 10^{-31}$	1500

input and stored patterns. This equivalent circuit is shown in Figure 1f, and with it an analytical expression for the voltage drop at the common electrode can be derived as follows

$$V_{\text{out}} = \frac{\text{HD} \times R_{\text{HRS}} + (n - \text{HD}) \times R_{\text{LRS}}}{n \times (R_{\text{HRS}} + R_{\text{LRS}})} \times V_{\text{read}} \quad (1)$$

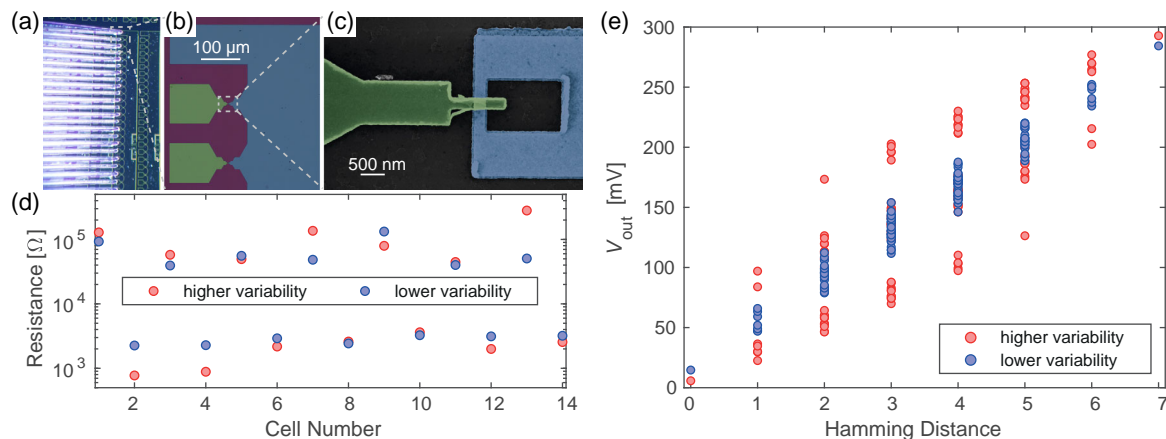
A similar equation can be calculated for the XNOR operation which was done by Chowdhury et al. in their system analysis of an equivalent concept.<sup>[23]</sup> In our approach, the voltage divider effect is used to retrieve the HD from the corresponding voltage

drop of the shared electrode, and from that the bMAC result can easily be computed if needed.

## 2.2. Hardware Realization

To confirm the linear relationship between the output voltage and the HD, a single row of lateral CRS cells, each based on two ReRAM cells, was fabricated.<sup>[24]</sup> The ReRAM devices are based on a platinum, tantalum oxide, tungsten stack capped by another platinum layer. A detailed description of the fabrication process can be found in Section 6. Images of the fabricated sample are shown in Figure 2. Figure 2a shows the sample (background) connected with a probe card (foreground) to the measurement setup. Figure 2b shows a lateral CRS cell (green) and the shared electrode (blue). A scanning electron microscope (SEM) image of a fabricated ReRAM device is shown in Figure 2c. The green-colored part corresponds to the top electrode (bitline), whereas the blue-colored part again shows the shared electrode (wordline). For the measurement, 14 ReRAM cells were combined into 7 lateral CRS cells and an exemplary "1111111" pattern was stored in these devices.

The following procedure was used to program the corresponding resistance states. In the first series of measurements, the LRS (HRS) of the corresponding device was programmed by a positive (negative) triangular voltage sweep with a maximum (minimum) voltage of 3.0 V (–1.5 V) applied to the bitline and a sweep rate of  $500 \text{ V s}^{-1}$ . The wordline was connected to a virtual ground and the current was measured. For the transition to the LRS, a series resistance of  $10 \text{ k}\Omega$  was included to limit the current. For the LRS (HRS), a programmed resistance of  $2.5 \text{ k}\Omega$  ( $90 \text{ k}\Omega$ ) was targeted around which the measured resistances are fluctuating due to the intrinsic resistance variability. The resistance of each device was measured with a read voltage of  $0.3 \text{ V}$  applied to the bitline



**Figure 2.** Measurement results of the fabricated sample. a) The image shows the fabricated sample (background) connected by a probe card (metal needles in the foreground) to the measurement system. b) The center image depicts a lateral CRS cell (green) to which an input bit is applied. The output voltage is measured at the shared electrode (blue). c) The SEM image on the right shows one resistive switch with an area of  $200 \text{ nm} \times 200 \text{ nm}$ . The green-colored part is the top electrode and the blue-colored part is the shared electrode. d) Resistance value of each resistive switch read at a voltage of  $0.3 \text{ V}$  applied to the top electrode. Two neighboring cells are combined to one lateral CRS cell to store an exemplary "1111111" pattern. The red dots represent the resistance states with higher intrinsic variability and the blue dots represent the resistance states with the manually reduced variability. e) Voltage drop at the shared electrode for each input pattern from "0000000" to "1111111" on the y-axis. On the x-axis, the HD between the input and the stored pattern is shown. This measurement reveals that the variability from the resistance states has a significant influence on the variability of the output voltage.

while the wordline was grounded again. The measured results are visualized by the red dots of Figure 2d. In the second series of measurements, the same sweep rate was used and the maximum and minimum voltage was manually adjusted to reduce the deviation from the target resistance. The resistances were again read with a voltage of 0.3 V applied to the bitline while grounding the wordline and are represented by the blue dots in Figure 2d.

For the computation of the HD, a pair of two bitlines is used to encode one bit of the input pattern. All 128 possible input patterns from “0000000” to “1111111” were encoded in voltages as shown in Table 1a and applied to the line array. The voltage drop on the wordline was measured and the results are shown in Figure 2e. The HD between the input pattern and the stored pattern is displayed on the x-axis. The voltage drop of the shared electrode of the CRS cells is shown on the y-axis. The voltage response of the stored resistance states with the higher variability is represented by the red dots and the voltage response of the resistance states with the lower variability is shown by the blue dots. The linear dependence between the HD and the output voltage is visible in Figure 2e. The measurements also show that the resistance variability has a significant influence on the variability of the output voltage. To better understand this influence, a simulation study will be discussed in the following section.

### 3. Simulation Study on the Influence of Resistance Variability

#### 3.1. Modeling of ReRAM Cell Conduction in LRS and HRS

As the HRS of the ReRAM cells shows a nonlinear behavior with respect to the voltage, a single ohmic resistance cannot be used to describe the device behavior properly. Instead, measured  $I$ - $V$  sweeps for each programmed resistance state in the voltage regime from  $-0.3$  to  $0.3$  V in steps of  $0.01$  V reveal this

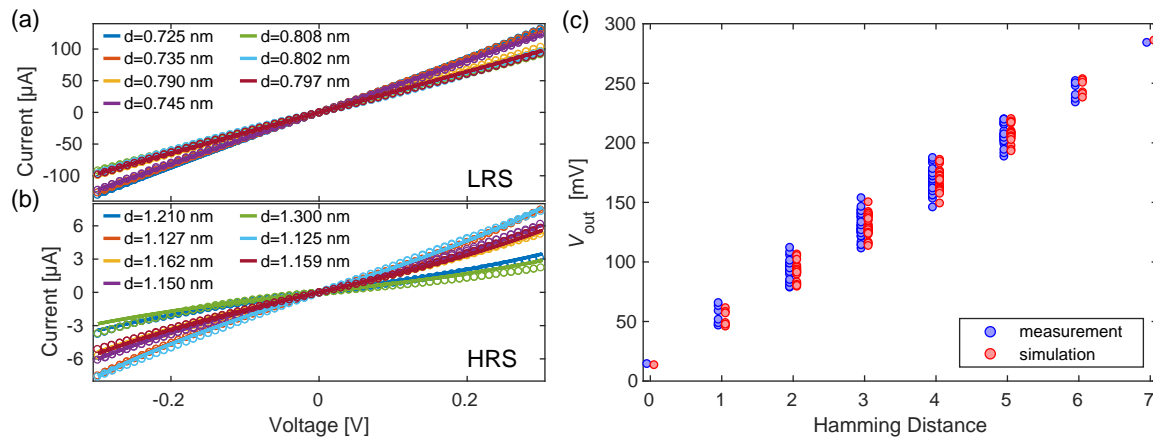
nonlinear behavior with a slight asymmetry with respect to the voltage polarity (cf. Figure 3). To model this  $I$ - $V$  characteristic, the conduction is described by assuming a tunneling process at the platinum interface and an ohmic resistance in the bulk of the oxide. The tunneling process is described by the so-called intermediate current-voltage relationship derived by Simmons<sup>[25]</sup>

$$I = \frac{eA}{2\pi\hbar d^2} \left[ \left( \phi - \frac{eV}{2} \right) \exp \left( -\frac{4\pi d\sqrt{2m}}{\hbar} \sqrt{\phi - \frac{eV}{2}} \right) - \left( \phi + \frac{eV}{2} \right) \exp \left( -\frac{4\pi d\sqrt{2m}}{\hbar} \sqrt{\phi + \frac{eV}{2}} \right) \right] \quad (2)$$

Here,  $e$  is the electron charge,  $A$  is the device area,  $\hbar$  is Planck's constant,  $V$  is the applied voltage,  $m$  is the effective electron mass, and  $\phi$  is the tunneling barrier height. To fit the measured resistance states, the only parameter adjusted is the effective tunnel barrier thickness  $d$ . The constant simulation parameters are shown in Table 1d. The Simmons equation has been used previously to describe the electron transport in ReRAM devices.<sup>[26,27]</sup>

The measured  $I$ - $V$  sweeps for each cell in the LRS are shown in Figure 3a by the colored circles. The simulated sweep based on the described model is visualized by the lines using the same color coding. The LRS is well described by that model only by adjusting the effective tunnel barrier thickness  $d$ . The measured  $I$ - $V$  sweeps for the HRS are shown in Figure 3b. Again the measurement is represented by the circles and the simulation by the accordingly colored lines. The HRS can be fairly well described by the model, and the deviations from the real nonlinearity and asymmetry are small.

Using this conduction model and the fitted data, the measurement results can be simulated with the Spectre Simulation Platform of Cadence. In Spectre, seven lateral CRS cells are configured in the experimental “1111111” configuration based on the fitted model parameters and all possible input patterns are applied.



**Figure 3.** Fitted measurement results described by a series combination of an ohmic resistor and a metal-insulator-metal tunnel barrier defined by Equation (2). a) The circles represent measured  $I$ - $V$  sweeps of the LRS with manually adjusted variability as described in Section 6. The voltage is applied to the top electrode and ranges from  $-0.3$  to  $0.3$  V with a voltage step of  $0.01$  V. For the model, the constant parameters described in Table 1d are used and the effective tunnel barrier  $d$  is adjusted to fit the measurement results which is shown by the accordingly colored lines. b) The circles show the same measurement results for the HRS as in (a) and the accordingly colored lines represent the characteristic of the simulation model. c) The model parameters fitted to the measured data in (a) and (b) are used to simulate the output voltage for the stored “1111111” pattern and all possible input patterns. These simulation results are represented by the red circles and compared with the measured results from Figure 2e. The simulated circuit is in good agreement with the measured data.

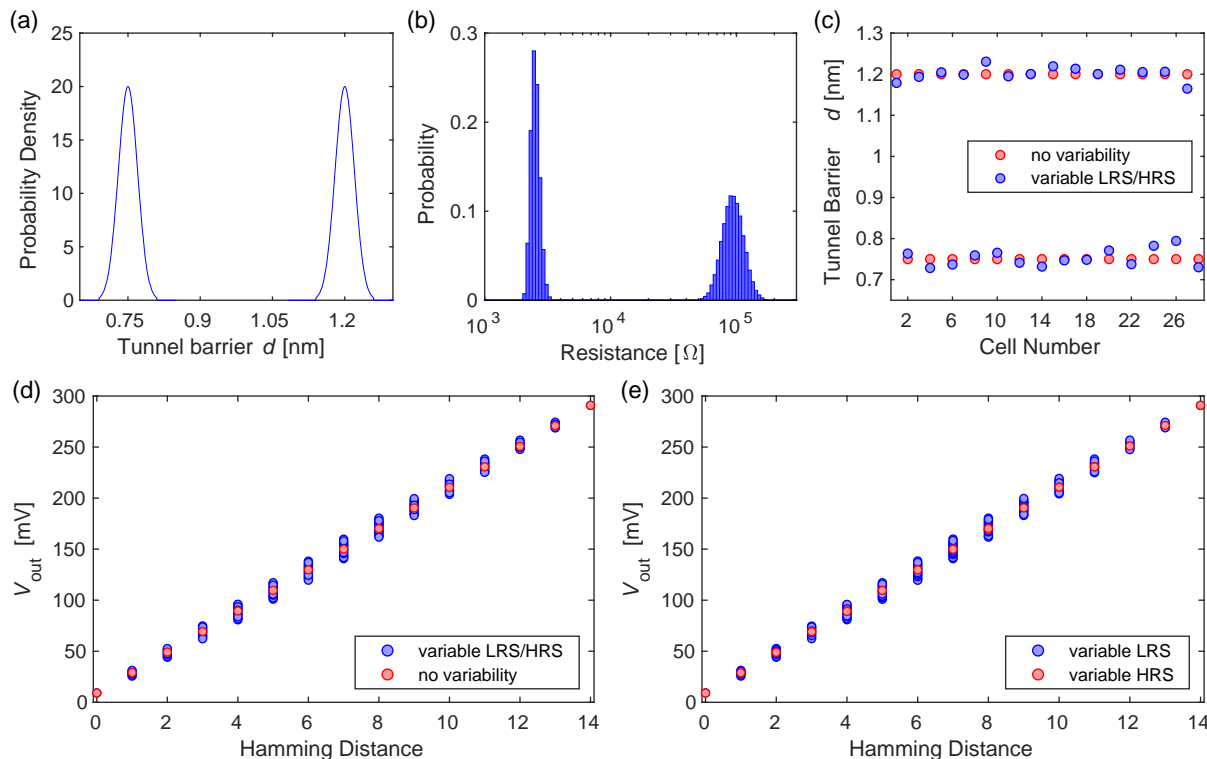
The obtained results are shown in Figure 3c where the lower variability data of Figure 2e are visualized by the blue circles, whereas the simulation data are represented by the red circles. The data points are slightly shifted apart from each other for a better visualization. This direct comparison of the measured and simulated circuit behavior also supports the agreement between the device properties and the utilized conduction model.

### 3.2. Modeling of the Resistance Variability

To better understand the origin of the variability in Figure 2e, a simulation study with changing variability contributions is performed. Two truncated normal distributions for the tunnel thickness  $d$  are assumed from which values are randomly taken for the simulation. The LRS is drawn from a truncated normal distribution with a mean value of  $d_{\text{LRS}} = 0.75$  nm and a standard deviation of  $\sigma_{\text{LRS}} = 0.02$  nm. The distribution is truncated after an interval of  $3\sigma$ . The HRS is drawn from a distribution that has the same standard deviation, is equally truncated, and has a mean value of  $d_{\text{HRS}} = 1.2$  nm. The utilized distributions for the tunnel thickness are shown in **Figure 4a**. To confirm that the assumed distributions correspond to reasonable resistance

variations, the coefficient of variation ( $\sigma/\mu$ ) of the resulting LRS and HRS distributions is compared with the measured data by Sheng et al.<sup>[28]</sup> For this purpose, 100 000 values are randomly drawn from each tunnel barrier distribution and the model resistance at a read voltage of  $-0.11$  V is calculated to match the experimental data. The simulated resistance distributions are summarized in two histograms which are shown in Figure 4b. The resulting LRS distribution has a mean value of  $2.53$  k $\Omega$  with a standard deviation of  $206.8$  k $\Omega$  and the HRS distribution has a mean value of  $95.33$  k $\Omega$  with a standard deviation of  $18.51$  k $\Omega$ . The coefficient of variation is  $0.19$  for the HRS distribution, which is similar to the one measured by Sheng et al. for that resistance range.<sup>[28]</sup> The coefficient of variation for the LRS distribution is  $0.08$ , which is roughly one order of magnitude higher compared with the results of Sheng et al. This deviation is intentionally chosen to be higher to attribute for the missing transistor in our demonstrator. Having a transistor in series to the ReRAM cell enables a better control of the LRS, which would result in a lower coefficient of variation.<sup>[28,29]</sup>

For the simulation study, 14 CRS devices are simulated with the Spectre Simulation Platform of Cadence. To this end, 14 tunnel barriers are drawn from each distribution and stored in a “11111111111111” configuration in the CRS cells.



**Figure 4.** Simulation study to investigate the main contribution to the variability of the output voltage. a) Assumed truncated normal distributions for the tunnel barrier  $d$ . The mean values are  $0.75$  and  $1.2$  nm and the standard deviation for both distributions is  $0.02$  nm. After 3 standard deviations, both distributions are truncated. b) Histograms of the resulting resistances from the model read at a voltage of  $-0.11$  V for 100 000 randomly drawn tunnel thicknesses from each distribution in (a). The LRS mean is  $2.53$  k $\Omega$  with a standard deviation of  $206.8$  k $\Omega$ . The HRS mean is  $95.33$  k $\Omega$  with a standard deviation of  $18.51$  k $\Omega$ . c) To represent the all “1” pattern for 14 CRS cells, 14 tunnel thicknesses had to be drawn from each distribution from (a) for the variable case (blue dots). For the case without variability, the mean values of the distributions are used (red dots). d) Simulated output voltage for all possible input patterns for tunnel barriers with variability (blue dots) and without (red dots). e) Simulated behavior of the output voltage with either LRS tunnel barrier variability (blue dots) or HRS tunnel barrier variability (red dots) enabled. From (d) and (e) it can be deduced that mainly the LRS variability causes the variation of the output voltage.



In Figure 4c, each cell is shown with its corresponding tunnel barrier drawn from the distribution visualized by the blue circles. To be able to vary the amount of variability in the simulation, each cell also has a constant tunnel barrier thickness assigned to it. This value is defined by the mean value of the corresponding distribution and is indicated by the red circles of Figure 4c.

In the first simulation, the influence of the variability on the output voltage is studied. The ideal case, which is represented by the red dots in Figure 4d, has no variability on the used tunnel thicknesses and follows the linear relation of Equation (1). In contrast, adding variability to the HRS and LRS leads to a dispersion of the output voltage around the ideal output voltage (blue dots).

To investigate which variability is mainly contributing to the output voltage, the next simulation enables each variability separately. The blue and red dots in Figure 4e correspond to only LRS and only HRS variability, respectively. As shown by the red dots, the HRS variability only has a minor influence on the output voltage. Assuming only LRS variability, in contrast, results in a higher variability of the output voltage. Thus, the main reason for the output variability is the variability of the LRS. This is a considerable result as it is the LRS variability which can be controlled significantly better.<sup>[28,29]</sup>

## 4. Potential as bNN Inference Accelerator

### 4.1. Simulation of the Inference Accuracy

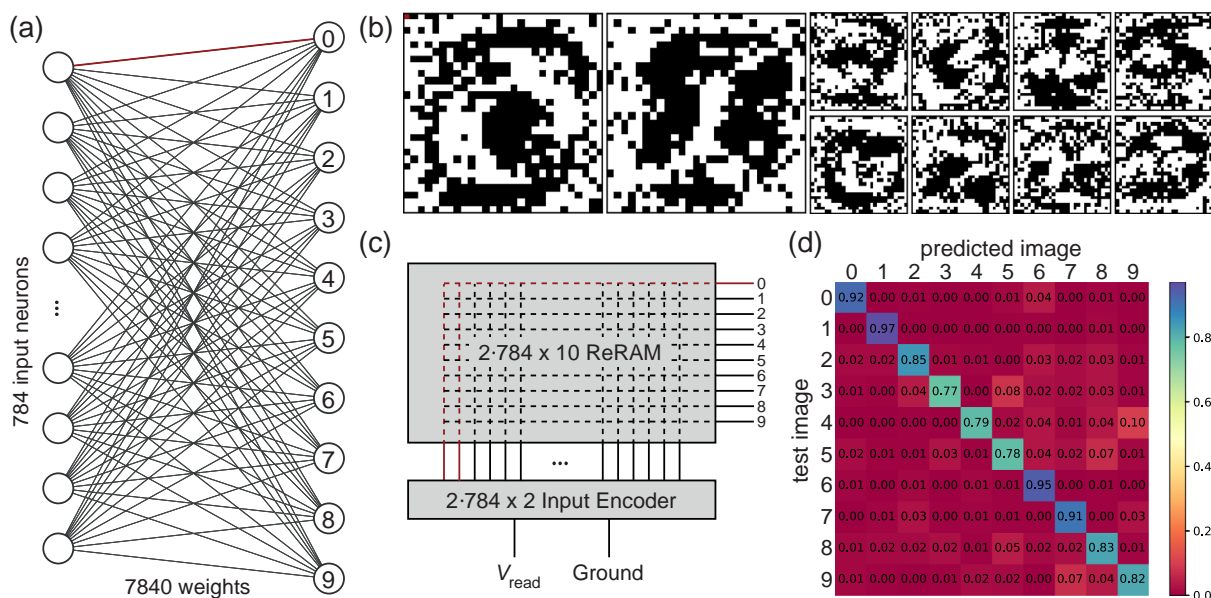
To investigate the potential of the concept as a hardware accelerator for binary vector–matrix multiplications, the inference step

of the MNIST data set was simulated with the Spectre Simulation Platform of Cadence.

As architecture, a 1-layer fully connected neural network with 784 input neurons and 10 output neurons was used. A representation of this network is shown in Figure 5a. Each output neuron is representing the prediction of the bNN for its corresponding number from 0 to 9. No activation functions are used because no hidden layers are implemented.

The bNN was trained in software on the MNIST data set for which an adapted version of the binarynet source code of the nn\_playground project of the user DingKe was used.<sup>[30,31]</sup> For the training in software, full precision weights are used which are binarized for the inference step. During the training, the prediction results of each training batch are used to calculate an error function. This error is backpropagated using the adam optimizer and the straight through estimator as approximation for the gradient. With this process, the full precision weights are updated for each training batch.<sup>[9,32]</sup>

The data set had to be preprocessed because it originally consists of grayscale images with 256 quantization levels. Therefore, each pixel was binarized by using the following equation  $x_b = 2 \times \text{round}(x/128) - 1$ . The trained weights of each output are visualized in Figure 5b. Black pixels correspond to a weight of  $-1$  and white pixels to a weight of  $1$ . For a better visualization, the trained weights are reshaped into a  $28 \text{ pixel} \times 28 \text{ pixel}$  image, which corresponds to the original shape of the MNIST data set. These 7840 weights are then transferred to the corresponding 7840 CRS cells by drawing tunnel barriers from the LRS and HRS distributions shown in Figure 4a. A weight of value “1” (“ $-1$ ”) is encoded according to a “1” (“0”) as described in



**Figure 5.** A 1-layer fully connected bNN was trained on the MNIST data set published by LeCun et al.<sup>[30]</sup> a) Software architecture of the trained bNN. It has 784 input neurons, one for each pixel of an image in the MNIST data set and 7840 weights which can be trained to be either 1 or  $-1$ . This image is adapted from an image created by a web application of LeNail.<sup>[36]</sup> b) The trained weights between the input neurons and each output neuron are shown. Black pixels represent a weight of  $-1$  and white pixels a weight of  $1$ . c) A block diagram of the simulated hardware architecture is shown. Two resistive switches form one of the 7840 CRS cells which represent a single weight. For a prediction of the hardware accelerator, an input encoder encodes the input image into a voltage pattern which is applied to the crossbar array. The lowest voltage drop at the outputs represents the prediction of the network. d) Confusion matrix of all predictions and their actual values of the simulated hardware accelerator.

Table 1b. A block diagram of the simulated hardware implementation is shown in Figure 5c. In this case, line resistances are neglected to only show a proof of concept simulation. A more detailed discussion of these effects is included in the supporting information.

For estimating the accuracy of the accelerator, each image of the test data set is binarized, flattened, and transformed to boolean values “1” (“−1”) to “1” (“0”). The resulting vector is encoded into a voltage pattern based on the encoding described in Table 1a and applied to the crossbar array. The voltage drops at each output line are compared with each other. The lowest voltage drop is used as the prediction of the hardware accelerator. In the simulation, the network can predict the correct handwritten number with an accuracy of  $\approx 86\%$ . This result is promising, as it is comparable with the 1-layer neural network with grayscale inputs and analog weights by LeCun et al., which achieved an accuracy of 88%.<sup>[30]</sup> The simulated hardware accelerator not only has to deal with binarization of the weights but also with the variability of the resistance states and therefore suffers from this accuracy loss. To understand where the hardware realization is mainly doing false predictions, a visualization of the confusion matrix, which compares the prediction result with the actual number, is shown in Figure 5d. This image conveys that the network mostly confuses the numbers 4 with 9.

#### 4.2. Design Considerations

For deriving some design considerations, it is helpful to introduce the resistance ratio  $r = R_{\text{HRS}}/R_{\text{LRS}}$  and rearrange Equation (1) to

$$V_{\text{out}} = \left( \frac{1}{1+r} + \frac{r-1}{r+1} \times \frac{\text{HD}}{n} \right) \times V_{\text{read}} \quad (3)$$

With Equation (3), the theoretical possible voltage window for a specific resistance ratio can be calculated by  $(r-1)/(r+1)$ . For filamentary ReRAM cells, realistic resistance ratios lie between 10 and 100 and will lead to a voltage window between 81% and 98% of the applied read voltage.<sup>[33]</sup> Increasing this resistance ratio further will always improve the voltage window but quickly slow down and stop having a significant influence on it.

Another consideration can be concluded from the fact that Equation (3) only depends on the resistance ratio and not on any resistance state itself. Thus, technologies with high LRS states and a reasonable resistance ratio benefit the most from this concept. An increased LRS lowers the current for each calculation and therefore makes the calculation more energy efficient.

This is confirmed by calculating the worst-case current for one operation. The current through the crossbar array depends on the HD between the stored pattern and the input pattern and is the most if the  $\text{HD} = n/2$ . With the help of the equivalent circuit in Figure 1f, the equation for the worst-case current can be derived to

$$I_{\text{worst-case}} = \frac{n}{4} \times \frac{1+r}{r} \times \frac{V_{\text{read}}}{R_{\text{LRS}}} \quad (4)$$

Apart from the energy considerations, the simulation study has shown that the concept is intrinsically resistant to HRS

variability, so the main challenge is the control of the LRS variability. This variability can be well controlled in integrated circuits by using a 1T1R structure or introducing write-verify schemes to reprogram the resistance if it is outside specified boundaries.<sup>[28,34]</sup>

#### 5. Conclusion

This work presented a computing in-memory concept based on a bitwise XOR operation for CRS cells. The center electrode of multiple CRS cells is connected to perform the accumulation of each cell's boolean logic operation. A demonstrator of this concept was fabricated and the measurement results were presented. The intrinsic variability of the programmed resistance states led to a significant variation of the output voltage. For understanding the underlying mechanism, a simulation study was performed to separate the LRS and HRS variability contribution. From this, the conclusion that the majority of the output voltage variability stems from the variations in the LRS could be derived. To show that real-world problems can be tackled by the studied concept, a 1-layer fully connected bNN was trained on the MNIST data set and the inference step was simulated. In this simulation, the hardware accelerator achieved an accuracy of around 86%.

#### 6. Experimental Section

**Fabrication:** For a reduction in processing steps, the CRS cells were fabricated in a lateral configuration. A thermally oxidized Si piece ( $\approx 430$  nm silicon oxide) was covered with  $\approx 5$  nm titanium as adhesive and  $\approx 30$  nm platinum as bottom electrode by sputter deposition. Then, diluted AZ nLOF 2020 photoresist was spin-coated and patterned by electron beam lithography. Reactive ion beam etching was used to create the shared electrode of the parallel CRS cells. After a resist removal process, the whole sample was covered with a stack of  $\approx 9$  nm tantalum oxide,  $\approx 16$  nm tungsten, and  $\approx 20$  nm platinum by sputter deposition. Again, diluted AZ nLOF 2020 photoresist was spin-coated and patterned by electron beam lithography. The excess material was removed by reactive ion beam etching and the leftover mask was cleaned. With this process, lateral CRS cells with a common electrode down to a device size of  $200 \text{ nm} \times 200 \text{ nm}$  could be realized.

**Measurement Setup:** The  $\mu$ Controller module platform and a  $4 \times 32$  switch matrix by aixACCT systems were the essential components that were used to apply binary encoded patterns to the CRS cells. A Picoscope 5444D MSO was used for measuring the voltage drop at the shared electrode (wordline) of the CRS cells. Two of the inputs of the switch matrix were connected to the voltage source of the  $\mu$ Controller module, where one of them included an ohmic series resistance of  $10 \text{ k}\Omega$ . Another input was connected to the ground of the  $\mu$ Controller module. This port could also measure the current. The last input of the switch matrix was connected to a channel of the Picoscope to monitor the voltage drop across the shared electrode. The 32 outputs of the switch matrix were connected to a probe card that connected the measurement system to the sample.

**Initial Measurements:** For the initial measurements, the measurement signal was always applied to the top electrode (bitline) of the devices. The shared electrode (wordline) was connected to ground. After fabrication, a triangular forming voltage sweep up to  $4 \text{ V}$  and down to  $-1.8 \text{ V}$  was applied. The sweep rate was set to  $500 \text{ V s}^{-1}$  and a series resistance of  $10 \text{ k}\Omega$  was included only during the positive cycle to limit the current through the devices after the forming process. After that, the devices were switched 5 times between the HRS and LRS to establish a stable switching behavior. To this end, a triangular sweep with the same sweep rate up to  $3 \text{ V}$  and down to  $-1.5 \text{ V}$  was applied. Again, a series resistance of  $10 \text{ k}\Omega$  was only included during the transition to the LRS.

## Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

## Acknowledgements

This work was mainly supported by the Federal Ministry of Education and Research (BMBF, Germany) within the NEUROTEC project (project numbers 16ES1134 and 16ES1133K). The Helmholtz Association Initiative and Networking Fund under project number SO-092 partly supported this research within the advanced computing architectures (ACA) project. This work is based on the Jülich Aachen Research Alliance (JARA-FIT). The authors want to acknowledge the Helmholtz Nano Facility and René Borowski for supporting the device fabrication.<sup>[35]</sup>

## Conflict of Interest

The authors declare no conflict of interest.

## Keywords

binary neural networks, complementary resistive switches, computation in-memory, neuromorphic computing, vector–matrix multiplication

Received: June 16, 2020

Revised: July 24, 2020

Published online: August 17, 2020

- [1] N. Z. Haron, S. Hamdioui, in *2008 3rd Int. Design and Test Workshop*, IEEE, Monastir, Tunisia **2008**, pp. 98–103.
- [2] S. Bianco, R. Cadene, L. Celona, P. Napoletano, *IEEE Access* **2018**, 6, 64270.
- [3] T. Yang, Y. Chen, J. Emer, V. Sze, in *2017 51st Asilomar Conf. on Signals, Systems, and Computers*, IEEE, Pacific Grove, CA **2017**, pp. 1916–1920.
- [4] S. Han, H. Mao, W. J. Dally, arXiv preprint arXiv:1510.00149, **2015**.
- [5] T. Simons, D.-J. Lee, *Electronics* **2019**, 8, 661.
- [6] W. Tang, G. Hua, L. Wang, <http://www.ganghua.org/publication/AAAI17.pdf> (accessed: August 2020).
- [7] S. Darabi, M. Belbahri, M. Courbariaux, V. Partovi Nia, *arXiv e-prints* **2018**, arXiv:1812.11800.
- [8] *Computer Vision ECCV 2016* (Eds: M. Rastegari, V. Ordonez, J. Redmon, A. Farhadi, In B. Leibe, J. Matas, N. Sebe, M. Welling), Springer International Publishing, Cham, **2016**, pp. 525–542.
- [9] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, Y. Bengio, *arXiv e-prints* **2016**, arXiv:1602.02830.
- [10] H. Tsai, S. Ambrogio, P. Narayanan, R. M. Shelby, G. W. Burr, *J. Phys. D: Appl. Phys.* **2018**, 51, 283001.
- [11] F. Cüppers, S. Menzel, C. Bengel, A. Hardtdegen, M. von Witzleben, U. Böttger, R. Waser, S. Hoffmann-Eifert, *APL Mater.* **2019**, 7, 091105.
- [12] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, W. Lu, *Nano Lett.* **2010**, 10, 1297.
- [13] R. Waser, R. Dittmann, G. Staikov, K. Szot, *Adv. Mater.* **2009**, 21, 2632.
- [14] R. Waser, M. Aono, *Nat. Mater.* **2007**, 6, 833.
- [15] M. Hu, C. E. Graves, C. Li, Y. Li, N. Ge, E. Montgomery, N. Davila, H. Jiang, R. S. Williams, J. J. Yang, Q. Xia, J. P. Strachan, *Adv. Mater.* **2018**, 30, 1705914.
- [16] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, D. B. Strukov, *Nature* **2015**, 521, 61.
- [17] M. Bocquet, T. Hirtzlin, J. Klein, E. Nowak, E. Vianello, J. Portal, D. Querlioz, in *2018 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco, CA **2018**, pp. 20.6.1–20.6.4.
- [18] S. Yin, Y. Kim, X. Han, H. Barnaby, S. Yu, Y. Luo, W. He, X. Sun, J. Kim, J. Seo, *IEEE Micro* **2019**, 39, 54.
- [19] V. Sze, Y.-H. Chen, T.-J. Yang, J. Emer, *Proc. IEEE* **2017**, 105, 2295.
- [20] E. Linn, R. Rosezin, C. Kügeler, R. Waser, *Nat. Mater.* **2010**, 9, 403.
- [21] E. Linn, R. Rosezin, S. Tappertzhofen, U. Böttger, R. Waser, *Nanotechnology* **2012**, 23, 305205.
- [22] Y. Zhou, Y. Li, L. Xu, S. Zhong, H. Sun, X. Miao, *Appl. Phys. Lett.* **2015**, 106, 233502.
- [23] A. Pal Chowdhury, P. Kulkarni, M. Nazm Bojnordi, *J. Low Power Electron. Appl.* **2018**, 8, 4.
- [24] A. Kindsmüller, C. Schmitz, C. Wiemann, K. Skaja, D. J. Wouters, R. Waser, C. M. Schneider, R. Dittmann, *APL Mater.* **2018**, 6, 046106.
- [25] J. G. Simmons, *J. Appl. Phys.* **1963**, 34, 1793.
- [26] H. Zhang, S. Yoo, S. Menzel, C. Funck, F. Cüppers, D. J. Wouters, C. S. Hwang, R. Waser, S. Hoffmann-Eifert, *ACS Appl. Mater. Interfaces* **2018**, 10, 29766.
- [27] A. Herpers, C. Lenser, C. Park, F. Offi, F. Borgatti, G. Panaccione, S. Menzel, R. Waser, R. Dittmann, *Adv. Mater.* **2014**, 26, 2730.
- [28] X. Sheng, C. E. Graves, S. Kumar, X. Li, B. Buchanan, L. Zheng, S. Lam, C. Li, J. P. Strachan, *Adv. Electron. Mater.* **2019**, 5, 1800876.
- [29] A. Fantini, L. Goux, R. Degraeve, D. J. Wouters, N. Raghavan, G. Kar, A. Belmonte, Y. Chen, B. Govoreanu, M. Jurczak, in *2013 5th IEEE Int. Memory Workshop*, IEEE, Monterey, CA **2013**, pp. 30–33.
- [30] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, *Proc. IEEE* **1998**, 86, 2278.
- [31] DingKe, nn Playground – Binarynet, [https://github.com/DingKe/nnet\\_playground/tree/master/binarynet](https://github.com/DingKe/nnet_playground/tree/master/binarynet) (accessed: April 2020).
- [32] D. P. Kingma, J. Ba, arXiv preprint arXiv:1412.6980, **2014**.
- [33] W. Kim, S. Menzel, D. J. Wouters, R. Waser, V. Rana, *IEEE Electron Device Lett.* **2016**, 37, 564.
- [34] F. Alibart, L. Gao, B. D. Hoskins, D. B. Strukov, *Nanotechnology* **2012**, 23, 075201.
- [35] W. Albrecht, J. Moers, B. Hermanns, Hnf – Helmholtz Nano Facility, <https://doi.org/10.17815/jlsrf-3-158> (accessed: June 2020).
- [36] A. LeNail, *J. Open Source Softw.* **2019**, 4, 747.