Computation of Boolean Matrix Chain Products in 3D ReRAM

Alvaro Velasquez
Department of Computer Science
University of Central Florida, Orlando, FL
Email: velasquez@eecs.ucf.edu

Sumit Kumar Jha
Department of Computer Science
University of Central Florida, Orlando, FL
Email: http://www.sumitkumarrjha.com/contact.html

Abstract—Energy concerns, the infamous memory wall, and the enormous data deluge of the current big-data age have made the integration of processing and memory elements into a very appealing paradigm. In this paper, we focus on a computation-in-memory solution to the problem of multiplying a set of Boolean matrices, also known as Boolean matrix chain multiplication (BMCM). This is a fundamental computational task with applications in graph theory, group testing, data compression, and digital signal processing. In particular, we propose a framework for mapping arbitrary instances of BMCM to a 3-dimensional (3D) crossbar memory architecture consisting of 1-diode 1-resistor (1D1R) structures.

I. INTRODUCTION

In recent years, there has been increased interest in unifying the processor and memory in order to alleviate the performance overheads imposed by the memory-processor bottleneck [1]. In this paper, we leverage this unified model of computation with recent advances in crossbar memory technology in order to tackle the problem of multiplying a chain of Boolean matrices within memory efficiently.

The paper is organized as follows. Section II defines a model for 3D crossbars and provides background information on Boolean matrix chain multiplication (BMCM). In Section III, we propose a mapping of the BMCM problem to a 3D crossbar and establish mathematical guarantees on the space complexity and correctness of the procedure. Section IV follows with experimental results and we finalize with concluding remarks in Section V.

II. BACKGROUND

Traditional Resistive Random Access Memory (ReRAM) architectures are crossbars, or cross-point memories, consisting of two sets of parallel wires, with each wire from one set placed perpendicularly to every wire in the other set. At every junction of wires, there is an interconnect acting as a switch between two wires. These interconnects typically consist of 1-transistor 1-resistor (1T1R) or 1-diode 1-resistor (1D1R) structures. However, it has been shown that in the context of 3-dimensional memory stacking, the usage of 1D1R cells is a more effective [2] and cost-efficient solution [3] than 1T1R. These 3D ReRAM memories are simply crossbars with more than one layer of 1D1R interconnects. An example can be seen in Figure 1. An abstraction of these memories can be found in Definition 1.

Definition 1. 3D CROSSBAR A \( p \times n \times L \) 3D crossbar is a 3-tuple \( \mathcal{C} = (M, R, C) \) where

- \( M = \{M^1, \ldots, M^L\} \) is a set of \( p \times n \) Boolean matrices,
- \( \mathcal{M} = \{M^1, \ldots, M^L\} \) represents a matrix of interconnects with \( p \) rows and \( n \) columns. Each \( m_{ij} \in \{0, 1\} \) denotes the state of the device connecting row \( i \) with column \( j \) in layer \( k \);
- \( \mathbb{R} = \{R^1, \ldots, R^L\} \) is the set of row wire vectors, where \( R^k = \{r^k_1, \ldots, r^k_n\} \) and \( r^k_i \in \{0, 1\} \) provides the same input voltage to every interconnect in row \( i \) of layer \( 2k - 1 \);
- \( \mathbb{C} = \{C^1, \ldots, C^L\} \) is the set of column wire vectors, where \( C^k = \{c^k_1, \ldots, c^k_p\} \) and \( c^k_j \in \{0, 1\} \) provides the same input voltage to every interconnect in column \( j \) of layer \( 2k \).

Definition 1 provides a high-level abstraction that can model an electrical system. For example, \( r_1 = 1 \) denotes that wire \( r_1 \) has a significant flow of electric current while \( r_1 = 0 \) denotes negligible flow. It follows then that the physical meaning of Axiom 1 is that of unidirectional flow of current. This means that the interconnects \( m_{ij} \) act as unidirectional switches, similar to a diode or a rectifying memristor [4] [5]. A survey of such interconnects can be found in [2]. In our abstraction, \( m_{ij} = 1 \) acts like a closed switch which redirects current from one terminal to the other, while \( m_{ij} = 0 \) acts like an open switch that does not allow current to flow from one terminal to the other. For the rest of this paper, we assume square matrices for the sake of simplicity; however, the proposed approach applies to matrices of arbitrary sizes.

Axiom 1 (Unidirectional Flow). Let \( \mathcal{C} = (M, R, C) \) be an \( n \times n \times L \) crossbar. Then \( \forall i, j, k; 1 \leq i, j \leq n; 1 \leq k \leq L - 1 ; \) \( (r^k_i \land m^k_{ij} \land n^{2k-1}_{ij}) \implies c^k_j \land (c^k_j \land m^k_{ij} \land n^{2k-1}_{ij}) \implies r^{k+1}_i \)

While the problem of performing logic computations using crossbars [6], [7], [8], [9], [10], [11], [12], [13] has been extensively studied in the literature, leveraging the structure of a 3-dimensional crossbar for computation has largely gone unexplored. In this paper, we take a step in this direction by computing Boolean matrix chain products within 3D ReRAM. We use the terms 3D ReRAM and crossbar interchangeably.
The Boolean matrix multiplication (BMM) problem may be defined as follows. Given two Boolean matrices \( X^1 \in \{0, 1\}^{n \times n} \) and \( X^2 \in \{0, 1\}^{n \times n} \), we wish to compute their product \( S = X^1 X^2 = (s_{ij}) \), where \( s_{ij} = \bigvee_{k=1}^{n} (x_{ik} \land x_{kj}) \).

\[
X^1 = \begin{pmatrix}
x_{11} & \cdots & x_{1n} \\
\vdots & \ddots & \vdots \\
x_{n1} & \cdots & x_{nn}
\end{pmatrix}, \quad
X^2 = \begin{pmatrix}
x_{11} & \cdots & x_{1n} \\
\vdots & \ddots & \vdots \\
x_{n1} & \cdots & x_{nn}
\end{pmatrix}
\]

\[
S = \begin{pmatrix}
\bigvee_{i=1}^{n} (x_{i1} \land x_{i1}) & \cdots & \bigvee_{i=1}^{n} (x_{i1} \land x_{in}) \\
\vdots & \ddots & \vdots \\
\bigvee_{i=1}^{n} (x_{ni} \land x_{i1}) & \cdots & \bigvee_{i=1}^{n} (x_{ni} \land x_{in})
\end{pmatrix}
\]

Given Boolean matrices \( X^1, \ldots, X^\alpha \), where \( X^k = (x_{ij}^k) \in \{0, 1\}^{n \times n} \), we define the \( k \)-chain product of these matrices as \( S^k = (s_{ij}^k) = X^1 X^2 \cdots X^{k+1} \) as defined by equation (3).

**III. METHODOLOGY**

The basis of our method consists of redirecting the flow of information through the crossbar based on the values of its interconnects. In the case of an electrical system, this can be achieved by applying a high voltage with respect to ground on some row wires in the first layer of the crossbar, grounding all of the bottommost wires in said crossbar, and configuring the interconnects based on variables of the formula \( \phi \) that we wish to compute in such a way that electrical current will flow into the grounded wires if and only if \( \phi = 1 \). In the context of our abstraction, this means that, given a 3D crossbar \( C = \{M, \{R^1, \ldots, R^{L-1}\}, \{C^1, \ldots, C^{L-1}\}\} \), a voltage is applied at some row wires \( r_{i1}^1 = r_{i2}^1 = \cdots = r_{ik}^1 = 1 \), which will generate a flow of current such that, by successively applying

**Axiom 1**, will result in some rows \( r_{i1}^{Ln} = r_{i2}^{Ln} = \cdots = r_{ik}^{Ln} = 1 \) having flow as well (or \( c_{1c}^{Lc} = c_{2c}^{Lc} = \cdots = c_{kc}^{Lc} = 1 \)).

For any layer \( k \), we can define the values of \( s_{ij}^k \) and \( c_{ij}^k \) as specified in (1) and (2), respectively.

\[
s_{ij}^k = \bigvee_{t=1}^{n} (s_{it}^{k-1} \land x_{tj}^{k+1}), \quad s_{1j}^1 = \bigvee_{t=1}^{n} (x_{1t}^1 \land x_{tj}^2)
\]

We can expand equations (1) and (2) to obtain (4) and (5). Note the striking similarity between the structure of these two formulas and (6), which corresponds to the expansion of equation (3). Recall that (3) is an arbitrary entry in the matrix corresponding the product of some Boolean matrices \( X^1, \ldots, X^{k+1} \). This similarity provides some intuition as to the feasibility of mapping the BMCM problem onto a 3-dimensional crossbar. In fact, the mapping is rather simple, as specified in Theorem 1. The crux of the procedure lies in configuring the layers of interconnects according to (7). A pictorial representation can be found in Figure 2.

\[
M^k = \begin{cases}
X^{k+1}, & k \text{ odd} \\
(X^{k+1})^T, & k \text{ even}
\end{cases}
\]

\[
r_1^1 = x_{11}^1, \quad r_1^2 = x_{12}^2, \quad m_1^1 = x_{11}^1
\]

\[
r_4^1 = x_{14}^1, \quad r_4^2 = x_{12}^2, \quad m_4^1 = x_{14}^1
\]

**Theorem 1.** Let \( C = \{M, R, C\} \) be an \( n \times n \times L \) 3D crossbar and let \( X^1, \ldots, X^\alpha \), \( \alpha \geq 2 \), denote a set of Boolean matrices with \( k \)-chain product \( S^k = (s_{ij}^k) = X^1 X^2 \cdots X^{k+1} \). If \( M^k = \begin{cases}
X^{k+1}, & k \text{ odd} \\
(X^{k+1})^T, & k \text{ even}
\end{cases} \), then \( s_{ij}^{\alpha} = s_{ij}^{\alpha-1}, \alpha \) even for any row index \( \gamma \in \{1, \ldots, n\} \).
\[ r_i^k = \bigvee_{j_{k-1}=1}^n \left( \ldots \left( \bigvee_{j_2=1}^n \left( \bigvee_{j_1=1}^n \left( \bigvee_{i_1=1}^n \left( r_{i_1}^1 \land m_{i_1 j_1} \right) \land m_{i_2 j_2} \right) \land m_{i_3 j_3} \right) \ldots \right) \land m_{i_{k-1} j_{k-1}} \right) \right) \] (4)

\[ c_j^k = \bigvee_{i_k=1}^n \left( \ldots \left( \bigvee_{j_2=1}^n \left( \bigvee_{j_1=1}^n \left( \bigvee_{i_1=1}^n \left( r_{i_1}^1 \land m_{i_1 j_1} \right) \land m_{i_2 j_2} \right) \land m_{i_3 j_3} \right) \ldots \right) \land m_{i_{2k-1} j_{2k-1}} \right) \right) \] (5)

\[ s_{\gamma_i j}^k = \bigvee_{i=1}^n \left( \ldots \left( \bigvee_{j_2=1}^n \left( \bigvee_{j_1=1}^n \left( \bigvee_{i_1=1}^n \left( x_{\gamma_i 1} \land x_{i_1 1} \right) \land x_{i_2 1} \right) \land x_{i_3 1} \right) \ldots \right) \right) \] (6)

**Proof.** Let \( r_1^1 = x_1^1 \). The proof is by induction on \( \alpha \), where the notations \( \alpha \) and \( 1 \)H indicate that the result follows from equation (1) and the inductive hypothesis, respectively.

**Base case:** When \( \alpha = 2 \),

\[ c_j^{\alpha/2} = c_j^1 = \bigvee_{i=1}^n \left( r_i^1 \land x_{ij} \right) = \bigvee_{i=1}^n \left( x_{\gamma_i 1} \land x_{ij} \right) = s_{\gamma_i j}^1 \] (3)

**Inductive hypothesis:** Assume that \( c_j^{\beta/2} = s_{\gamma_i j}^{\beta-1} \) for even \( \beta \) and \( r_i^{(\beta+1)/2} = s_{\gamma_i j}^{\beta-1} \) for odd \( \beta \).

**Inductive step:** For \( \alpha = \beta + 1 > 2 \), two cases arise.

- \( \beta + 1 \) even:
  \[ c_j^{\alpha/2} = c_j^{(\beta+1)/2} = \bigvee_{i=1}^n \left( r_i^{(\beta+1)/2} \land m_{ij} \right) \]
  \[ = \bigvee_{i=1}^n \left( \bigvee_{k=1}^n \left( c_k^{(\beta-1)/2} \land m_{ik} \right) \land m_{ij} \right) \]
  \[ = \bigvee_{i=1}^n \left( \bigvee_{k=1}^n \left( c_k^{(\beta-1)/2} \land x_{ki} \right) \land x_{ij} \right) \]
  \[ = 1H \bigvee_{i=1}^n \left( \bigvee_{k=1}^n \left( s_{\gamma_k j}^{\beta-2} \land x_{ki} \right) \land x_{ij} \right) \]
  \[ = \bigvee_{i=1}^n \left( s_{\gamma_i j}^{\beta-1} \land x_{ij} \right) \] (3)

- \( \beta + 1 \) odd:
  \[ r_i^{(\alpha+1)/2} = r_i^{(\beta+2)/2} \]
  \[ = \bigvee_{i=1}^n \left( c_j^{\beta/2} \land m_{ij} \right) \]
  \[ = \bigvee_{i=1}^n \left( \bigvee_{k=1}^n \left( r_k^{\beta/2} \land m_{ik} \right) \land m_{ij} \right) \]
  \[ = \bigvee_{i=1}^n \left( \bigvee_{k=1}^n \left( r_k^{\beta/2} \land x_{ki} \right) \land x_{ij} \right) \]
  \[ = 1H \bigvee_{i=1}^n \left( \bigvee_{k=1}^n \left( s_{\gamma_k j}^{\beta-2} \land x_{ki} \right) \land x_{ij} \right) \]
  \[ = \bigvee_{i=1}^n \left( s_{\gamma_i j}^{\beta-1} \land x_{ij} \right) \] (3)

By applying Theorem 1 on \( X^1, \ldots, X^\alpha \in \{0, 1\}^{n \times n} \) for some row index \( \gamma \), we can compute the entries in the \( \gamma^{th} \) row vector of \( S^{\alpha-1} \). That is, the values of \( s_{\gamma 1}^{\alpha-1}, s_{\gamma 2}^{\alpha-1}, \ldots, s_{\gamma n}^{\alpha-1} \) will be contained in \( r_1^{(\alpha+1)/2}, r_2^{(\alpha+1)/2}, \ldots, r_n^{(\alpha+1)/2} \) when \( \alpha \) is odd and in \( c_1^{(\alpha/2)}, c_2^{(\alpha/2)}, \ldots, c_n^{(\alpha/2)} \) when \( \alpha \) is even. Therefore, we need only repeat this procedure \( n \) times, once for each \( \gamma \in \{1, \ldots, n\} \), in order to compute \( S^{\alpha-1} \). We elucidate this approach with a simple example. Let \( X^1, X^2, X^3 \in \{0, 1\}^{4 \times 4} \) be defined below.

\[
\begin{align*}
X^1 &= \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1
\end{pmatrix}, \\
X^2 &= \begin{pmatrix}
1 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 \\
1 & 1 & 1 & 0
\end{pmatrix}, \\
X^3 &= \begin{pmatrix}
1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1
\end{pmatrix}
\end{align*}
\]

Clearly, \( S^2 = X^1 X^2 X^3 = X^3 \) since \( X^1 \) and \( X^2 \) are identity matrices. This problem instance can be mapped to a 4 \( \times \) 4 crossbar \( C = \{ (M^1, M^2), (R^1, R^2, R^3, R^4), (C^1, C^2, C^3, C^4) \} \) by setting \( M^1 = X^2 \) and \( M^2 = (X^3)^T \). In order to compute \( s_{11}^2, s_{12}^2, s_{13}^2, s_{14}^2 \), let \( r_1^1 = 1, r_2^1 = r_3^1 = r_4^1 = 0 \). From Theorem 1, it follows that \( r_1^2, r_2^2, r_3^2, r_4^2 \) will hold the values of \( s_{11}^2, s_{12}^2, s_{13}^2, s_{14}^2 \). Similarly, the values for the second row of \( S^2 \) \((s_{21}^2, s_{22}^2, s_{23}^2, s_{24}^2) \) are computed by setting \( r_1^2 = 1, r_1^3 = r_3^3 = r_4^3 = 0 \) as can be seen in Figure 3.
IV. EXPERIMENTAL RESULTS

These results have been verified through HSPICE simulations using Schottky diodes and resistors in 1D/1R structures. Interconnects corresponding to variables with value 1 have ON resistances of 10Ω and variables with value 0 have OFF resistances of 100kΩ. A 2V voltage pulse was applied on the topmost row wires in accordance with Theorem I and a resistor-to-ground with resistance 1MΩ was placed on each of the bottommost wires in order to read the outputs, which are shown in the matrix $R_V$ below. Note that the entries of $R_V$ coincide with the entries of $S^2 = X^3 X^2 X^3 = X^3$ as intended. It can be seen that whenever $s_{ij}^2 \in S^2$ evaluates to 1, a voltage value (with respect to ground) of $6-8mV$ is read. Conversely, whenever $s_{ij}^2 \in S^2$ evaluates to 0, a voltage value of approximately $6\mu V$ is measured at the outputs. We have also verified our approach on randomized Boolean matrices using a varying number of layers. While there is a significant read margin between 0 and 1 values when up to 4 layers are used, a substantial signal degradation is observed for 3D crossbars with 8 or more layers.

$$R_V = \begin{pmatrix} 6.4374mV & 6.1352mV & 6.2094mV & 8.4733mV \\ 6.5884mV & 8.4738mV & 6.7009mV & 8.4738mV \\ 6.4275mV & 6.3153mV & 8.4734mV & 6.0864mV \\ 6.4288mV & 8.4746mV & 8.4747mV & 7.8898mV \end{pmatrix}$$

In order to compare the performance of 3D ReRAM against other popular memory architectures, we have simulated a 2 MB memory using the DESTINY memory modeling tool [14]. The results can be seen in Table I. Note that each memory architecture prevails in some aspect of performance. eDRAM has the lowest read and write latency, SRAM requires a low write energy at the cost of area and substantial leakage power, 2D ReRAM has low leakage and read energy, and 3D ReRAM benefits from exceptionally small area. In fact, DESTINY demonstrates that a 32 GB memory would only occupy an area of 33.763 mm² when using 22 nm technology in a 16-layer 3D ReRAM.

<table>
<thead>
<tr>
<th></th>
<th>Read Latency (ns)</th>
<th>Write Latency (ns)</th>
<th>Read Energy (pJ)</th>
<th>Write Energy (pJ)</th>
<th>Leakage (mW)</th>
<th>Area (mm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D ReRAM</td>
<td>124.09</td>
<td>139.51</td>
<td>122.33</td>
<td>129.11</td>
<td>9.852</td>
<td>0.0027</td>
</tr>
<tr>
<td>2D ReRAM</td>
<td>14.325</td>
<td>22.29</td>
<td>5.12</td>
<td>3.842</td>
<td>2.268</td>
<td>0.0065</td>
</tr>
<tr>
<td>SRAM</td>
<td>41.541</td>
<td>41.541</td>
<td>386.6</td>
<td>2.31</td>
<td>1924</td>
<td>1.259</td>
</tr>
<tr>
<td>eDRAM</td>
<td>9.179</td>
<td>9.179</td>
<td>41.558</td>
<td>1009</td>
<td>2.637</td>
<td>0.2764</td>
</tr>
</tbody>
</table>

TABLE I: DESTINY [14] simulations of different memory architectures using 22 nm technology, including the 3D ReRAM used in this paper. In order to avoid bias, we utilize the default parameters included in the simulator.

It is worth noting that the approach proposed in this paper is applicable to matrices whose dimensions are greater than those of the 3D crossbar. This result follows from seminal work found in Cannon’s thesis, where it is shown that matrix multiplication can be carried out by smaller sub-matrix block products [15].

V. CONCLUSION

We have shown how to compute the product of a set of Boolean matrices by mapping it to a 3-dimensional crossbar memory. The correctness of the proposed approach was proven mathematically and a simple example was given to elucidate its effectiveness, with a read margin of three orders of magnitude between the output voltages of 0 and 1 values. This result attempts to ameliorate the divide between the traditional computation model of von Neumann architectures and the memory-processor integration paradigm in 3D ReRAM.

REFERENCES