3D computational architectures and applications

George Zhang - February 26, 2018

3D offers a new approach to meet the challenges faced by today's integrated circuits (IC) and information technology (IT). A 3D-ROM uses a 3D physical structure to extend the Moore's Law and is now in mass production. As an extension of the 3D-ROM, a 3D computation device (3D-COM) uses a 3D computer architecture to satisfy the critical requirements for high-performance computing (HPC) tasks ranging from artificial intelligence to super-computing.

Moore's Law [1] and von Neumann architecture [2] are cornerstones of IC and IT. Transistors and computers have been enjoying exponential growth since their inception. However, after over half a century, their rates of growth have started to slow down. As the size of transistors is being reduced to that of tens of silicon atoms, doubling the transistor count every two years is not sustainable, as long as IC remains 2D. On the other hand, hindered by the von Neumann bottleneck, computer performance slowly levels off and most HPC tasks need to be carried out by hundreds, and even thousands of computers.

Foreseeing these challenges, the author invented a 3D ROM (3D-ROM) in 1996 [3] and more recently a 3D computation (3D-COM). The 3D-ROM is already in mass production. Matrix Semiconductor delivered the first 3D-ROM product in industry - a 3D-OTP - in 2003 [4]. In March 2017, Intel and Micron started to ship a product known as 3D-XPoint [5]. The 3D-COM, although in its infancy, is expected to produce orders of magnitude improvement in performance over the von Neumann architecture for many HPC tasks.

This article will first review the foundation of the 3D-COM – a 3D-ROM with a 3-D physical structure, then disclose a 3-D computer architecture for the 3D-COM, before discussing applications. Throughout this article, within the context of system components, memory refers to temporary information store (i.e., cache) and generally comprises a plurality of RAM chips; and, storage refers to long-term information store and typically comprises a plurality of ROM chips (for SSD) or disc platters (for HDD). Within the context of memory IC, memory refers to either RAM or ROM.

3D structures

A traditional IC chip has a planar 2D structure, since all active elements (e.g. transistors) are formed on the surface of a semiconductor substrate. Over twenty years ago, the author started to explore the third dimension by proposing a 3D-ROM [3]. The idea before this structure was that the traditional IC chip is like a flat (Figure 1A), while the 3D-ROM is like a skyscraper (Figure 1B).

The 3D-ROM was so named because its memory cells are distributed in a 3D space. In the exemplary 3D-ROM die of Figure 2, memory cells form memory levels 1 and 2 which are then stacked on top of each other. Within each memory level, address lines form a cross-point array and the memory cells
are located at their intersections. If it has a low resistance, the memory cell represents '1'; otherwise it represents '0'. The bottom of the 3D-ROM die is known as substrate and it contains transistor circuits for reading/programming the memory cells. The memory levels are coupled to the substrate circuit through thousands of inter-level vias. A typical 3D-ROM die consists of thousands (or, even tens of thousands) of 3D-ROM arrays. Since it is the primary form of 3D memory, 3D-ROM has become synonymous with 3D memory.

Figure 1 Evolution from 2D to 3D physical structures: (A) a traditional IC chip is like a flat; (B) a 3D ROM (3D-ROM) die is like a skyscraper whose floors are all residence; (C) a 3D computation (3D-COM) die is like a modified skyscraper whose ground floor is a shop and floors above remain residence.

Figure 2 A 3D-ROM die consists of thousands (or, even tens of thousands) of 3D-ROM arrays. Each 3D-ROM array comprises a plurality of vertically stacked memory cells for storage [3]. The 3D structure of the 3D-ROM die leads to a larger storage density.

During the past decades, several 3D-ROMs went into mass production. The first 3D-ROM product was a 3D-OTP (one-time-programmable) developed by Matrix Semiconductor. In 2005, Matrix released a 1Gb 3D-OTP at the 0.13-micrometer (um) node. Consisting of four vertically stacked memory levels, it had the world's smallest 1Gb die size (31mm$^2$) at the time [4]. The second and more well-known 3D-ROM product is a 3D-XPoint (multiple-time-programmable) device jointly developed by Intel and Micron (Figure 3), which has been lauded as "a major breakthrough in memory process technology, the first new memory category since the introduction of NAND flash in 1989." In 2017,
Intel/Micron released a 128Gb 3D-XPoint at the 20-nanometer (nm) node. The 3D-XPoint consists of two vertically stacked memory levels and has a die size of 241mm². It was claimed that the 3D-XPoint was ~1000x faster than the NAND flash [5].

Both the 3D-OTP and the 3D-XPoint use a narrow-line 3D structure [6], which is quite different from a standard CMOS. In the standard CMOS, the minimum feature size of the backend process (e.g. metal interconnects) is larger than that of the front-end (e.g. gate length) (Figure 4A). On the contrary, in the narrow-line 3D structure, the minimum feature size of the backend process (i.e. the memory diodes) is smaller than that of the front-end (i.e. the substrate transistors) (as shown in Figure 4B). This is because diodes have a simpler structure than transistors and can be scaled faster. The narrow-line 3D structure is conspicuous in the 3D-XPoint, where the minimum feature sizes of the memory diodes and the substrate transistors are 20nm and 200nm, respectively (Figure 3) [7].

Figure 3 A cross-sectional view of the 3D-XPoint [7]. It has a narrow-line 3D structure. (image courtesy of TechInsights)
So far, industry has used the 3D-ROM as a storage device only, and not as a computing device. The author and co-workers have extended the idea of the 3D-ROM into the field of computing by inventing a new computational method - 3D-COM. The 3D-COM uses a 3-D computer architecture, which can be understood using the same skyscraper analogy as shown in Figure 1C (with more details disclosed in Figures 6C-6D). By turning its ground floor into a shop, residents of the skyscraper do not have to go outside the building to shop and thus, save time and enjoy greater convenience. A 3D-COM die is like a city with thousands of such skyscrapers.

In a 3D-ROM die, because the storage is above the substrate, not inside the substrate, the substrate could be fairly empty. This is confirmed in Figure 3 by the large size (200nm) of the substrate transistors of the 3D-XPoint. Were the 20nm transistors used instead, ~80% substrate area of the 3D-XPoint would become empty. The 3D-COM takes advantage of this fact by integrating a computing element (referred to as a micro-core) under each 3D-ROM array. The 3D-ROM array and the micro-core together will now form a storage-computing unit (SCU). Because each 3D-ROM array gives rise to an SCU, a 3D-COM die comprises thousands (or, even tens of thousands) of SCUs (as shown in Figure 5).

Figure 4 A standard CMOS \( f > F \) (A) vs. a narrow-line 3-D structure \( f < F \) [6] (B).

**3D architectures**

A 3D-COM die consists of thousands (or, even tens of thousands) of SCUs. Each SCU integrates a micro-core under the 3D-ROM array. In such closeness (3-D stacking) and at such a fine scale (at the array level), storage and computing are highly fused.
The 3D integration used by the 3D-COM die is unique. First of all, only the substrate level is used for computing. This is because the substrate, containing expensive single-crystalline semiconductor, is best used for computing. This arrangement also helps to alleviate the heating issues. Secondly, integrated in such closeness (3-D stacking) and at such a fine scale (at the array level), storage and computing are highly fused. This ensures massive parallelism, as thousands (or, even tens of thousands) of SCUs can perform computations simultaneously.

The inter-level vias provide storage-computing interface (Figure 5). Because a 3D-COM die has thousands (or, even tens of thousands) of SCUs with each SCU having thousands of inter-level vias, the number of inter-level vias in a 3D-COM die will reach millions (or, even tens of millions). In other words, the storage-computing interface in the 3D-COM die would be million-bit wide, a giant leap from a modern computer whose storage-computing interface is only 64-bit wide. Thus, the storage-computing bandwidth in a 3D-COM die is orders of magnitude larger than any modern computer. This tremendous increase in the storage-computing bandwidth will reshape the landscape of computer architecture.

Figure 6 Evolution from 2D to 3D computer architectures: (A) von Neumann and (B) Harvard architectures are 2D; (C) instruction-light/data-heavy and (D) instruction-heavy/data-light architectures are 3D. The areal interface in the 3D architecture has a much larger bandwidth than the linear interface in the 2D architecture. The arrow width indicates relative bandwidth. Purple represents data; blue represents instruction.

The conventional computer architectures such as the von Neumann architecture (Figure 6A) and Harvard architecture (Figure 6B) are 2D. They work fine for moderately-sized data and instructions. Once the amounts of data or instructions become too big for the internal memory, the computer performance degrades because it has to frequently access an external storage. This scenario can be avoided with the emergence of a 3D architecture which provides one extra dimension for interfacing. Whereas a 2D architecture has a linear interface along an edge of the die, a 3D architecture has an areal interface over the whole die surface. Depending on their relative size, data and instructions can be stored at different locations: for a large amount of data but moderate amount of instructions (instruction-light/data-heavy), the data are stored in the 3D-ROM arrays, while the instructions are stored externally (as in Figure 6C); for a large amount of instructions but moderate amount of data (instruction-heavy/data-light), the instructions are stored in the 3D-ROM
arrays, while the data are stored externally (as in Figure 6D).

**Applications**

**Applications**

Based on mature technologies (3D-OTP and 3D-XPoint), the 3D-COM is ready to move from idea to reality. By sheer volume of storage and sheer scale of parallelism, the 3D-COM is far superior to any conventional computational method. Table 1 lists several HPC tasks where the von Neumann architecture struggles but the 3D architecture excels. They include smart storage for big-data analytics, rule enforcement (RE) and anti-virus scan (AV) for network security, artificial intelligence (AI), super-computing (SC), and field-programmable computing array (FPCA), among others. All these applications have high market demand but currently are critical problems areas. Realizing even a single one of these HPC tasks would justify our efforts on the 3D-COM.

**Table 1** Several HPC tasks where the 3D-COM excels.

<table>
<thead>
<tr>
<th>HPC Task</th>
<th>Application Category</th>
<th>Information stored in 3D-ROM array</th>
<th>Micro-Core function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Big-Data Analytics</td>
<td>Storage</td>
<td>Data</td>
<td>Text Matching</td>
</tr>
<tr>
<td>Network Security</td>
<td>Processor</td>
<td>Rule/Virus Database</td>
<td>Code Matching</td>
</tr>
<tr>
<td>Artificial Intelligence</td>
<td></td>
<td>Synaptic Weights</td>
<td>Convolution</td>
</tr>
<tr>
<td>Super-Computing</td>
<td></td>
<td>Look-up Tables</td>
<td>Interpolation</td>
</tr>
<tr>
<td>Field-Programmable Computing Array</td>
<td></td>
<td>Look-up Tables</td>
<td>Configurable Logic/Interconnect</td>
</tr>
</tbody>
</table>

1. **Smart storage for big data**

In a conventional computer, all computing powers are consolidated into the processor, away from the memory and the storage. The memory and the storage can be considered “dumb” since they do not have any data-analyzing capabilities per se. For big-data analytics, TBs to PBs of data are transferred from the storage to the processor. During this process, these data have to go through two von Neumann bottlenecks: the 1st one is from the storage to the memory; and, the 2nd one is from the memory to the processor (Figure 7A). In 1990s, an intelligent RAM (iRAM) [8], and more recently, an Automata [9], were developed to eliminate the 1st von Neumann bottleneck by integrating at least a portion of the processor into the memory RAM (as shown in Figure 7B). However, the 2nd von Neumann bottleneck remains.
**Figure 7** Evolution of system partition in a computer: (A) a conventional computer suffers from two von Neumann bottlenecks (in red); (B) an intelligent RAM (iRAM) [8] eliminates the 1st von Neumann bottleneck; (C) an intelligent ROM (iROM) [10] eliminates both von Neumann bottlenecks. The arrow width indicates relative bandwidth.

**Figure 8** An intelligent ROM (iROM) die consists of thousands (or, even tens of thousands) of SCUs. Within each SCU, the 3D-ROM array stores data, while the micro-core performs text matching [10].
Based on massively distributed processing, the string-searching time remains relatively constant regardless of the data volume. The addition of the string-search capabilities adds little or no cost to the storage die.

The concept of the iRAM can be extended into storage. By integrating at least a portion of the processor (i.e., CPU B) into the storage ROM to form an intelligent ROM (iROM), the 2nd von Neumann bottleneck can now be eliminated (as shown in Figure 7C) [10]. The iROM is intelligent because it has in-situ string-searching capabilities, which is a key part of big-data analytics. The iROM adopts a 3D architecture: within each SCU of the iROM die, the 3D-ROM array stores data, while the micro-core performs text matching (Figure 8). To analyze unstructured or semi-structured data, a search string is sent to all SCUs where it is compared with the data stored in the associated 3D-ROM arrays. Only the data that match the search string are sent to an external CPU (i.e., CPU A) for further analysis (as shown in Figure 7C). To store big data, multiple iROM dice are packaged together to form a smart storage card or a smart SSD. They are “smarter” than a traditional storage card or a traditional SSD because of their in-situ string-searching capabilities.

An exciting advantage of the smart storage is that its string-searching time does not increase with the total storage capacity. This is because each SCU in an iROM die has its own text-matching circuit. This text-matching circuit needs only to process the data stored in a single 3D-ROM array. As a result, no matter how large the total storage capacity, the string-searching time for the entire storage is the same as that of a single SCU, i.e., the time to search ~1Mb of data. This is much faster than the von Neumann architecture where the string-searching time increases linearly with the storage capacity.

Another advantage of the smart storage is that the additional string-searching capabilities come with little additional cost. With the 3D integration, adding text-matching circuits into a 3D-COM die does not increase the die size since these circuits are fabricated under the 3D-ROM array. Better yet, because the peripheral circuits of the 3D-ROM array need to be fabricated on the substrate anyway, fabricating the text-matching circuits along with the peripheral circuits does not change the manufacturing process and causes minimal increase in wafer cost. For a given storage capacity, a "smart" storage, with in-situ string-searching capabilities, costs almost the same as a conventional "dumb" storage.

II. Rule-enforcement (RE) and anti-virus scan (AV) for network security

Rule enforcement (RE) and anti-virus scan (AV) are key features of network security. Their operations involve code matching each incoming network packet with each rule/virus pattern in a rule/virus database. The rule/virus database has become quite large nowadays: the number of network rules has reached hundreds of thousands, while the number of computer viruses has reached millions. Typical processors (CPU, GPU, etc.) can only screen a limited number (tens to hundreds) of rules/viruses simultaneously. As a result, a conventional computer cannot efficiently handle code matching for a large rule/virus database. Furthermore, the Neumann bottleneck would cause a long delay when the processor fetches rule/virus patterns from an external storage. Consequently, the conventional network processor has a poor RE/AV performance.

To improve the RE/AV efficiency, a new type of 3D-COM, namely, a 3D anti-virus processor (to be abbreviated as 3D-AV, with the understanding that its functions include RE), was invented [10]. Within each SCU of the 3D-AV die, the 3D-ROM array permanently stores the rule/virus patterns, while the micro-core performs code matching (Figure 9). During the RE/AV operations, a search string from the input is sent to all SCUs, where it is compared with the rule/virus patterns stored in the associated 3D-ROM arrays. Because it contains thousands of SCUs, a 3D-AV die offers massive
parallelism by screening thousands of rules/virus at the same time. In addition, the permanent and in-situ nature of the rule/virus storage makes the rule/virus access very fast since no external access is needed. With massive parallelism and fast storage access, the 3D-AV processor can significantly improve the RE/AV performance.

**Figure 9** A 3D anti-virus processor (3D-AV) die consists of thousands of SCUs. Within each SCU, the 3D-ROM array stores rule/virus patterns, while the micro-core performs code matching [10]. Through massive parallelism and permanent in-situ rule/virus storage, substantial RE/AV performance gains can be achieved.

### III.  Artificial intelligence (AI)

Artificial intelligence (AI) has become the next wave in computing. Neural network is a powerful AI tool. General neural-network processing involves convolution with synaptic weights. A conventional neuro-processor consists of RAM blocks for storing synaptic weights and convolution circuits for performing convolution with synaptic weights (**Figure 10A**) [11]. With the 2D integration, the conventional neuro-processor has a poor computational density, because a substantial amount (~80%) of its die area is used for the RAM blocks and only a small fraction (~10%) is used for the convolution circuits [11]. Furthermore, there is a clear trend towards increasingly larger neural networks. Most neural networks use one billion to ten billion synaptic weights. To store them, a large number (tens to hundreds) of neuro-processor dice are needed. This is not desirable for mobile applications.

To improve the AI efficiency, another type of 3D-COM, known as 3D artificial intelligence processor (or, 3D-AI), has been invented [12]. Within each SCU of the 3D-AI die, the 3D-ROM array stores synaptic weights while the micro-core performs convolution (**Figure 10B**). Unlike the conventional neuro-processor where synaptic weights are stored in the RAM blocks located side-by-side with the convolution circuits (**Figure 10A**), synaptic weights are stored in the 3D-ROM arrays located above the convolution circuits. With the 3D integration, not only a substantial amount (~90%) of die area is devoted to storage, but a similar substantial amount (~90%) of die area can be used for computing. This nearly ten-fold increase in computational density will significantly improve the AI performance. In addition, because the storage capacity of its 3D-ROM arrays can reach >100Gb, a single 3D-COM die can easily store all synaptic weights (billions to tens of billions) of a neural network. This means a “single-chip” AI solution will become possible, a great news for mobile applications.
Figure 10 Comparison of 2D and 3D artificial-intelligence (AI) circuits: (A) on a 2D-AI die, computing (e.g. convolution circuit) only occupies ~10% of the die; (B) on a 3D-AI die, computing (e.g. convolution circuit) occupies ~90% of the die [12]. The 3D-AI die increases the computational density nearly ten-fold.

IV. Super-computing (SC)

Super-computing (SC) involves computations of extremely complex functions, solutions of a huge number of equations, or calculations based on models involving a large number of parameters. Surprisingly, high-end CPUs used for SC support only two built-in arithmetic operations (i.e. addition and multiplication) and a small set of nine built-in basic elementary functions (i.e. inv, log, exp, sin, cos, tan, atan, sqrt, cbrt) [13]. These are the only mathematical tools available to scientists and engineers. All SC, no matter how complicated the problem is, has to be decomposed through software into combinations of these built-in functions. Lengthy software decomposition and limited choice of built-in functions make computations both time and energy consuming.

The above computational paradigm has been in the mainstream for years. It stemmed from an original CPU design, which adopted a logic-based computation (LBC). For the LBC, logic circuits are the primary computing force. The logic circuits, when combined with an excessive amount of look-up tables (LUT) under a 2D integration, would lead to large die size and high die cost (as shown in Figure 11A). Thus in a conventional CPU, very little (<1%) of the die area is allocated to the LUTs, as most die area is used for logic circuits (Figure 11A). Overall, the amount of the LUTs in a conventional CPU is <100kb.

With the drastic increase in storage capacity, a memory-based computation (MBC) is developed to complement the LBC [14]. For the MBC, the memory circuits (for storing the LUTs) are the primary computing force. This leads to another type of 3D-COM, which has been labeled as 3D SC (or, 3D-SC) processor [14]. Within each SCU of the 3D-SC die, the 3D-ROM array stores the LUTs while the micro-core performs interpolations (Figure 11B). By moving the LUTs from “aside” to “above” (relative to the logic circuits), a substantial amount (~90%) of the die area becomes available for the LUTs. In fact, a 3D-SC die can store >100Gb of LUTs, more than six orders of magnitude larger than a conventional CPU, which stores <100kb of LUTs. This large amount of LUTs can be used to store values and derivatives of complex functions, the solutions of a huge number of equations, and the numerical results of large-scale models. For the MBC, computations are simplified into two steps: reading numbers from an LUT and then performing an interpolation to arrive at answers.
Figure 11 Comparison of 2D and 3D super-computing (SC) circuits: (A) on a 2D-SC die, the look-up tables (LUT) occupy <1% of the die; (B) on a 3D-SC die, the LUTs cover ~90% of the die [14]. The LUTs stored in a 3D-SC die are significantly larger than those stored in a 2D-SC die (>100Gb vs. <100kb).

With significantly more LUTs (>100Gb), the 3D-COM can lead to a paradigm shift for scientific computation. Scientific computation uses advanced computing capabilities to advance human knowledge and to solve engineering problems. The prevailing framework of scientific computation consists of four layers: foundation, library, model and simulation layers (as shown in Figure 12). The foundation layer realizes traditional built-in operations and functions [13]. The library layer includes a mathematical library collecting a significant portion of known special mathematical functions such as Bessel function and Chebyshev function. The model layer contains established models for describing individual system components, e.g. the analytical models of individual transistors. The simulation layer achieves the final goal of scientific computation by calculating the overall system behavior, e.g. the I-V behavior of an amplifier circuit comprising at least a transistor.

Figure 12 Paradigm shift in scientific computation. As scientific computation evolves from conventional (A) to 3D (B & C), larger proportion of computation is implemented by hardware through memory-based computation (MBC), with less proportion implemented by software through logic-based computation (LBC).

Conventionally, only the foundation layer is implemented by hardware through the MBC, while all other layers (i.e. the library, model and simulation layers) are implemented by software through the LBC (Figure 12A). As scientific computation evolves from conventional to 3D, larger proportion of computation is implemented by hardware through the MBC, with less proportion implemented by...
software through the LBC. For example, for a moderate paradigm shift (as shown in Figure 12B), the foundation and library layers are implemented by the MBC, while the model and simulation layers are implemented by the LBC. On the other hand, for an aggressive paradigm shift (as shown in Figure 12C), all layers except the simulation layer are implemented by the MBC. Because the MBC yields the results by simply reading numbers from the LUTs, more MBC leads to a faster simulation speed.

Interestingly, to perform the same simulation, the aggressive paradigm shift of Figure 12C may require less amount of LUTs than the moderate paradigm shift of Figure 12B. Take, for example, simulation of an amplifier circuit containing a transistor. For the moderate paradigm shift, the transistor’s I-V characteristics are modeled by an analytical model. Since a transistor model involves over one hundred parameters, an impractical amount of LUTs is required. However, for the aggressive paradigm shift, the transistor’s I-V characteristics are read out from the LUTs. Because a transistor has only six input variables ($V_{GS}$, $V_{DS}$, $V_{BS}$, $W$, $L$ and N/P-type), more reasonable LUT capacity (10Gb-scale) is required, which can be easily stored in a 3D-COM die.

V. Field-programmable computing array (FPCA)

Since the implementations of computations become similar based on the concept of the MBC, it is feasible to use the same set of hardware to implement different mathematical functions. This idea is similar to that of the FPGA, which starts with the same set of hardware to implement different logic functions [15]. The FPCA extends the concept of the FPGA into the field of computing. Besides configurable logic elements and configurable interconnects, the FPCA further introduces configurable computing elements (Figure 13) [14]. Each configurable computing element consists of at least one 3D writable memory array, which is electrically programmable and can be loaded with an LUT of a desired mathematical function (Figure 11). Within each SCU, the 3D-ROM arrays storing the LUT are stacked above the configurable logic elements and the configurable interconnects.

![Figure 13](image)

**Figure 13** A field-programmable computing array (FPCA) consists of a plurality of configurable slices, with each slice comprising arrays of configurable computing elements, configurable logic elements and configurable interconnects [14].
An implementation of the FPCA in Figure 13 for calculating a mathematical function: 
\[ e = a \sin(b) + c \cos(d) \] [14].

**Figure 14** is an example of how the FPCA in **Figure 13** can be programmed to calculate a mathematical function: \( e = a \sin(b) + c \cos(d) \) [14]. By re-programming configurable computing elements, configurable logic elements and configurable interconnects, other mathematical functions can be calculated. The usage cycle of the configurable computing element includes two stages: a configuration stage and a computing stage. In the configuration stage, the LUT for a desired mathematical function is loaded into the 3D writable memory array. In the computing stage, the values of the desired mathematical function are read out from the LUT.

**Outlook**

In an EDN cover story published over 16 years ago [16], editor **Brian Dipert**, after surveying a number of exotic memories including the 3D-ROM, asked: “Newfangled memory technologies come and go, but once in a while, one makes it out of the laboratory and into your next design. All of these candidates aspires to be the next semiconductor success. Which ones’ dream will come true?” Now, 3D seems to emerge as that successful candidate. **Table 2** shows that nearly all components in a computer will be turned into 3D in the future: for processors, certain CPU functions will be implemented by the 3D-COM; for internal memory, the 3D-XPoint (from Intel/Micron) [5] will partially replace DRAM; for external storage, the 3D-NAND (from Samsung and other manufacturers) is replacing HDD; for archiving, the 3D-OTP (from Matrix) [4] will replace tape.

**Table 2** Components in a computer are becoming 3D.

<table>
<thead>
<tr>
<th>Components</th>
<th>2D Era</th>
<th>3D Era</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>CPU</td>
<td>3D-COM</td>
</tr>
<tr>
<td>Memory</td>
<td>DRAM</td>
<td>3D-XPoint</td>
</tr>
<tr>
<td>Storage</td>
<td>HDD</td>
<td>3D-NAND</td>
</tr>
<tr>
<td>Archive</td>
<td>Tape</td>
<td>3D-OTP</td>
</tr>
</tbody>
</table>
Related articles:

- The future of IC design
- The year of 3D memory

Guobiao (George) Zhang has been working in 3D memory and related applications for over twenty years. He is now chief scientist at 3D-ROM Inc. in the U.S. and HaiCun IP Group in China. He holds over one hundred U.S. and Chinese patents.

The author would like to thank Dr. Chen Shen, Prof. Hongyu Yu and Prof. Bin Yu of South Univ. of Science and Technology (ShenZhen, China) for contribution and support. The author would also like to thank Prof. Peter Yu of Univ. of Calif. at Berkeley for carefully reviewing the manuscript.

References:
