| | |
|---|---|
| United States Patent Application Publication | 20250266414 |
| Kind Code | A1 |
| Publication Date | August 21, 2025 |
| Inventor(s) | ZHANG; Guobiao |

# Discrete Three-Dimensional Processor

## Abstract

A discrete three-dimensional (3-D) processor comprises vertically stacked and communicatively coupled first and second dice. The first die comprises memory arrays, while the second die comprises non-memory circuits and off-die peripheral-circuit components of the memory arrays. The total-thickness difference of the BEOL layers between the memory arrays and the off-die peripheral-circuit components is substantially larger than the total-thickness difference of the BEOL layers between the non-memory circuits and the off-die peripheral-circuit components.

**Inventors:** **ZHANG; Guobiao (Corvallis, OR)**

**Applicant:** **ZHANG; Guobiao** (Corvallis, OR)

**Family ID:** **1000008586809**

**Assignee:** **ShenZhen CunYang Technology Co., Ltd. (ShenZhen, CN)**

**Appl. No.:** **19/175160**

**Filed:** **April 10, 2025**

## Related U.S. Application Data

parent US continuation 17964888 20221012 parent-grant-document US 11695001 child US 18117472
parent US continuation-in-part 18117472 20230305 PENDING child US 19175160

## Publication Classification

**Int. Cl.:** **H01L25/18** (20230101); **G06F9/30** (20180101); **G06F15/80** (20060101); **G06F18/21** (20230101); **G06F21/56** (20130101); **G10L15/183** (20130101); **G10L15/22** (20060101); **H01L25/065** (20230101)

**U.S. Cl.:**

CPC **H01L25/18** (20130101); **G06F9/3001** (20130101); **G06F15/803** (20130101); **G06F18/21** (20230101); **G06F21/561** (20130101); **G10L15/183** (20130101); **G10L15/22** (20130101); **H01L25/0657** (20130101); G06F2221/034 (20130101); H01L2225/06506 (20130101); H01L2225/06513 (20130101); H01L2225/06524 (20130101); H01L2225/06541 (20130101)

---

## Background/Summary

CROSS-REFERENCE TO RELATED APPLICATIONS [0001] This application is a continuation-in-part of U.S. patent application Ser. No. 18/117,472, filed Mar. 5, 2023, which is a continuation of U.S. patent application Ser. No. 17/964,888, filed Oct. 12, 2022, now U.S. Pat. No. 11,695,001, which is a division of U.S. patent application Ser. No. 16/249,021, filed Jan. 16, 2019, now U.S. Pat. No. 11,527,523, which claims priorities from the following Chinese patent applications: [0002] 1) Chinese Patent Application No. 201811506212.1, filed Dec. 10, 2018; [0003] 2) Chinese Patent Application No. 201811508130.0, filed Dec. 11, 2018; [0004] 3) Chinese Patent Application No. 201811520357.7, filed Dec. 12, 2018; [0005] 4) Chinese Patent Application No. 201811527885.5, filed Dec. 13, 2018; [0006] 5) Chinese Patent Application No. 201811527911.4, filed Dec. 13, 2018; [0007] 6) Chinese Patent Application No. 201811528014.5, filed Dec. 14, 2018; [0008] 7) Chinese Patent Application No. 201811546476.X, filed Dec. 15, 2018; [0009] 8) Chinese Patent Application No. 201811546592.1, filed Dec. 15, 2018; [0010] 9) Chinese Patent Application No. 201910002944.5, filed Jan. 2, 2019; [0011] 10) Chinese Patent Application No. 201910029523.1, filed Jan. 13, 2019,
in the State Intellectual Property Office of the People's Republic of China (CN), the disclosures of which are incorporated herein by references in their entireties.

FIELD OF THE INVENTION
[0012] The present invention relates to the field of integrated circuit, and more particularly to a processor.
PRIOR ART
[0013] Processors (including CPU, GPU, FPGA, and others) are extensively used in mathematical computation, computer simulation, configurable gate array, pattern processing and neural network. A conventional processor is based on two-dimensional (2-D) integration, i.e., its logic circuit (a.k.a. processing circuit, e.g., arithmetic logic unit, control unit) and memory circuit (internal memory, including RAM for cache and ROM for look-up table) are disposed on a same plane, i.e., the top surface of a semiconductor substrate. Because the arithmetic logic operations are its primary function, the processor die contains limited amount of internal memory.
[0014] The conventional computer is based on the von Neumann architecture, where processor and memory are physically separated. Most memory takes the form of external memory (e.g., main memory, secondary memory). When it requests a large amount of data, a processor fetches the data from an external memory. Because the processor and the external memory are distant and the system bus between them has a relatively narrow width, data transfer between them has a limited bandwidth. As the amount of data increases, the conventional processor and its associated von Neumann architecture become inefficient.
[0015] The following paragraphs will provide an overview of the fields of applications of the conventional processors and their limitations.
[A] Mathematical Computing
[0016] One important application of processors is mathematical computing, including computing

of mathematical functions and mathematical models. For mathematical computing, the conventional processors use logic-based computation (LBC), which carries out computation primarily with logic circuits (generally known as arithmetic logic unit, or ALU). In fact, the arithmetic operations that can be directly implemented by the ALU consist of addition, subtraction and multiplication. These arithmetic operations are collectively referred to as basic arithmetic operations. The ALU's are suitable for arithmetic functions, but not for non-arithmetic functions. For a processor to compute mathematical functions, an arithmetic function is a mathematical function which can be represented by a combination of its basic arithmetic operations, whereas a non-arithmetic function is a mathematical function which cannot be represented by a combination of its basic arithmetic operations. Exemplary non-arithmetic functions include transcendental functions and special functions. Because it includes more operations than the arithmetic operations provided by the ALU's, a non-arithmetic function cannot be implemented by the ALU's alone. The hardware implementation of the non-arithmetic functions has been a major challenge.

[0017] For the conventional processors, only few basic functions (i.e., single-variable non-arithmetic functions, e.g., basic algebraic functions and basic transcendental functions) are implemented by hardware and they are referred to as built-in functions. These built-in functions are realized by a combination of logic circuits and look-up tables (LUT). In prior art, there are many ways to implement built-in functions. For example, U.S. Pat. No. 5,954,787 issued to Eun on Sep. 21, 1999 taught a method for generating SIN/COS functions using LUT's; U.S. Pat. No. 9,207,910 issued to Azadet et al. on Dec. 8, 2015 taught a method for calculating a power function using LUT's.

[0018] Realization of built-in functions is further illustrated in FIG. **1**AA. A conventional processor **0**X generally comprises a logic circuit **00**L and a memory circuit **00**M. The logic circuit **00**L comprises an ALU for performing arithmetic operations, while the memory circuit **00**M stores an LUT for the built-in function. To obtain a desired precision, the built-in function is approximated to a polynomial of a sufficiently high order. The LUT **00**M stores the coefficients of the polynomial; and the ALU **00**L calculates the polynomial. Because the ALU **00**L and the LUT **00**M are formed side-by-side on a semiconductor substrate **0**, this type of horizontal integration is referred to as two-dimensional (2-D) integration.

[0019] Computation has been developed along the directions of computational density and computational complexity. The computational density is a figure of merit for parallel computation and it refers to the computational power (e.g., the number of floating-point operations per second) per die area. The computational complexity is a figure of merit for scientific computation and it refers to the total number of built-in functions supported by a processor. The 2-D integration severely limits computational density and computational complexity.

[0020] For the 2-D integration, the LUT **00**M would increase the die size of the conventional processor **0**X and lower its computational density. This has an adverse effect on parallel computation. Moreover, as the primary component of the conventional processor **0**X, the ALU **00**L occupies most die area. As a result, the LUT **00**M is left with a small die area which can only accommodate few built-in functions. FIG. **1**AB lists all built-in transcendental functions supported by an Intel IA-64 processor (referring to Harrison et al. “The Computation of Transcendental Functions on the IA-64 Architecture”, Intel Technical Journal, Q4 1999, hereinafter Harrison). The IA-64 processor supports just a total of seven built-in functions. This small set of built-in functions makes mathematical computing difficult, because most mathematical functions need to be decomposed into a combination of built-in functions. The conventional processor **0**X is slow and inefficient for most mathematical computing.

[B] Computer Simulation

[0021] Another important application of processors is computer simulation, which involves computing of mathematical models. As a natural extension of mathematical computing, computer simulation is built upon the small set of built-in functions (about ten types) provided by the

conventional processor. The framework of the conventional computer simulation comprises three layers: a foundation layer, a function layer and a modeling layer. The foundation layer includes built-in functions that can be directly implemented by hardware. The function layer includes mathematical functions that cannot be directly implemented by hardware. The modeling layer includes mathematical models, which are the mathematical descriptions of the behaviors (e.g., input-output characteristics) of a system component.

[0022] The mathematical functions in the function layer and the mathematical models in the modeling layer are implemented by software. As mentioned before, the function layer involves one software-decomposition step. On the other hand, the modeling layer involves two software-decomposition steps: the mathematical models are decomposed into combinations of mathematical functions; before the mathematical functions are decomposed into combinations of built-in functions. Because the mathematical models involve more software-decomposition steps, their implementations are even slower and more inefficient than the mathematical functions.

[0023] To illustrate how computationally intensive a mathematical model could be, FIGS. **1**BA-**1**BB disclose a simple example—the simulation of an amplifier circuit **0**Y. The amplifier circuit **0**Y comprises a transistor **0**T and a resistor **0**R (FIG. **1**BA). All transistor models (e.g., MOS3, BSIM3 V3.2, BSIM4 V3.0, PSP of FIG. **1**BB) model the transistor behaviors based on the small set of built-in functions provided by the conventional processor **0**X. Due to the limited choice of the built-in functions, calculating even a single current-voltage (I-V) point for the transistor **0**T requires a large amount of computation (FIG. **1**BB). As an example, the BSIM4 V3.0 transistor model needs 222 additions, 286 multiplications, 85 divisions, 16 square-root operations, 24 exponential operations, and 19 logarithmic operations.

[0024] The ALU **00**L in the conventional processor **0**X can only realize arithmetic models per se. Typical mathematical models are non-arithmetic models. For a processor to compute mathematical models, an arithmetic model is a mathematical model which can be represented by a combination of its basic arithmetic operations, whereas a non-arithmetic model is a mathematical model which cannot be represented by a combination of its basic arithmetic operations. Because it includes more operations than the arithmetic operations provided by the ALU **00**L, a non-arithmetic model cannot be implemented by the ALU alone. Computation of non-arithmetic models by the conventional processor **0**X is extremely slow and inefficient.

[C] Configurable Gate Array

[0025] A third application of processors is configurable gate array. A configurable gate array is a semi-custom integrated circuit designed to be configured by a customer after manufacturing. It is also referred to as field-programmable gate array (FPGA), complex programmable logic device (CPLD), or other names. U.S. Pat. No. 4,870,302 issued to Freeman on Sep. 26, 1989 (hereinafter referred to as Freeman) discloses a configurable gate array. It contains an array of configurable logic elements (CLE's, also known as configurable logic blocks) and a hierarchy of configurable interconnects (CIT's, also known as programmable interconnects) that allow the CLE's to be wired together per customer's desire. Each CLE in the array is in itself capable of realizing any one of a plurality of logic functions (e.g., shift, logic NOT, logic AND, logic OR, logic NOR, logic NAND, logic XOR, arithmetic addition “+”, arithmetic subtraction “−”, etc.) depending upon a first configuration signal. On the other hand, each CIT can selectively couple or de-couple interconnect lines depending upon a second configuration signal.

[0026] In a configurable gate array, the arithmetic operations (i.e., arithmetic addition and arithmetic subtraction) provided by the CLE are also collectively referred to as basic arithmetic operations. They are fewer than those provided by the conventional processor (i.e., addition, subtraction and multiplication). As used hereinafter, basic arithmetic operations could refer to either those provided by the configurable gate array, or those provided by the conventional processor, depending on the context.

[0027] The configurable gate array can customize logic functions and arithmetic functions, but not

non-arithmetic functions. In the configurable gate array, an arithmetic function is a mathematical function which can be represented by a combination of its basic arithmetic operations; whereas, a non-arithmetic function is a mathematical function which cannot be represented by a combination of its basic arithmetic operations. Because the non-arithmetic functions include more operations than basic arithmetic operations, they cannot be directly implemented by the CLE's alone. It was generally believed that customization of non-arithmetic functions is impossible.

[D] Pattern Processing

[0028] A fourth application of processors is pattern processing. Pattern processing includes pattern matching and pattern recognition, which are the acts of searching a target pattern (i.e., the pattern to be searched) for the presence of the constituents or variants of a search pattern (i.e., the pattern used for searching). The match usually has to be “exact” for pattern matching, whereas it could be “likely to a certain degree” for pattern recognition. As used hereinafter, search patterns and target patterns are collectively referred to as patterns; pattern database refers to a database containing related patterns. Pattern database includes search-pattern database (also known as search-pattern library) and target-pattern database.

[0029] Pattern processing has broad applications. Typical pattern processing includes code matching, string matching, speech recognition and image recognition. Code matching is widely used in information security. Its operations include searching a virus in a network packet or a computer file; or, checking if a network packet or a computer file conforms to a set of rules. String matching, also known as keyword search, is widely used in big-data analytics. Its operations include regular-expression matching. Speech recognition identifies from the audio data the nearest acoustic/language model in an acoustic/language model library. Image recognition identifies from the image data the nearest image model in an image model library.

[0030] The pattern database has become large: the search-pattern library (including related search patterns, e.g., a virus library, a keyword library, an acoustic/language model library, an image model library) is already big; while the target-pattern database (including related target patterns, e.g., computer files on a whole disk drive, a big-data database, an audio archive, an image archive) is even bigger. The conventional processor and its associated von Neumann architecture have great difficulties to perform fast pattern processing on large pattern databases.

[E] Neural Network

[0031] A fifth application of processors is neural network. Neural network is a powerful tool for artificial intelligence (AI). An exemplary neural network is shown in FIG. **1**C. It includes an input layer **32**, a hidden layer **34** and an output layer **36**. The input layer **32** comprises i neurons **33**, whose input data x.sub.1, . . . x.sub.i form an input vector **30***x*. The output layer **36** comprises k neurons **37**, whose output data y.sub.1, y.sub.2, . . . y.sub.k form an output vector **30***y*. The hidden layer **34** is between the input layer **32** and the output layer **36**. It comprises j neurons **35**, each of which is coupled with a first neuron in the input layer **32** as well as a second neuron in the output layer **36**. The couplings between neurons are represented by synaptic weights w.sub.ij and w.sub.jk.

[0032] A machine-learning supercomputer comprising a plurality of accelerator dice **60** is disclosed in prior art (referring to Chen et al. “DaDianNao: A Machine-Learning Supercomputer”, IEEE/ACM International Symposium on Micro-architecture, 5(1), pp. 609-622, 2014). Each accelerator die **60** comprises an array of sixteen tiles **50**, which are connected through a fat tree (FIG. **1**DA). Each tile **50** comprises a neuro-processing unit (NPU) **30** and four eDRAM banks **40** (FIG. **1**DB). The NPU **30** performs neural processing. It comprises 256+32 16-bit multipliers as well as 256+32 16-bit adders. The eDRAM **40** stores the synaptic weights. It has a storage capacity of 2 MB.

[0033] The accelerator **60** has several drawbacks. First of all, because the eDRAM **40** is a volatile memory, the synaptic weights need to be loaded into the eDRAM **40** before operations. This takes time. Secondly, each accelerator die **60** contains 32 MB eDRAM. This number is still quite small for many neural networks. Thirdly, the accelerator **60** adopts an asymmetric architecture where the

tile area is heavily biased towards storage rather than computation. Inside each tile, eDRAM **40** occupies nearly 80% of the area, whereas the NPU **30** only occupies less than 10%. As a result, the computational density is limited.

[0034] With the advent of three-dimensional memory (3D-M), the above difficulties are alleviated (mentioned in applications [A]-[E]). For a 3D-M, its memory cells are disposed in a three-dimensional (3-D) space, i.e., stacked along a direction perpendicular to the top surface of the substrate. U.S. Pat. No. 6,861,715 B2 issued to Zhang on Mar. 1, 2005 discloses an integrated 3-D processor, where a logic circuit is monolithically integrated underneath the 3D-M arrays. The integrated 3-D processor is, by itself, a single 3-D processor die.

[0035] The integrated 3-D processor can be applied to above fields of applications. U.S. patent application Ser. No. 15/487,366, filed Apr. 13, 2017, discloses an integrated 3-D processor for computing mathematical functions and mathematical models; U.S. Pat. No. 9,838,021, issued Dec. 5, 2017, discloses an integrated 3-D processor as a configurable computing array; U.S. patent application Ser. No. 15/452,728, filed Mar. 3, 2017, discloses an integrated 3-D processor as a pattern processor; U.S. patent application Ser. No. 15/464,347, filed Mar. 21, 2017, discloses an integrated 3-D processor as a neuro-processor. The integrated 3-D processor brings about significant advantages in these fields.

[0036] FIGS. **1**EA-**1**EB disclose an integrated 3-D processor die **80**. It comprises at least a 3D-M array **77** and a logic circuit **78** integrated therewith. The 3D-M array **77** stores data, while the logic circuit **78** processes at least a portion of the data stored in the 3D-M array **77**. On the 3-D processor die **80**, the die area occupied by the 3D-M arrays **77** is an array region **70**, whereas the die area outside the 3D-M arrays **77** is a non-array region **71** (FIG. **1**EA). The array region **70** comprises a substrate circuit **0**K and a 3D-M array **77** stacked thereon (FIG. **1**EB). The substrate circuit **0**K includes transistors **0***t* and substrate interconnects **0***i*. The transistors **0***t* are formed in a semiconductor substrate **0** and communicatively coupled by the substrate interconnects **0***i*. The substrate interconnects **0***i* include two interconnect layers **0***m***1**-**0***m***2**, each of which comprises a plurality of interconnects **0***m* on a same physical plane. The 3D-M arrays **77** include four address-line layers **0***a***1**-**0***a***4**. Each address-line layer (e.g., **0***a***1**) comprises a plurality of address lines (e.g., **1***a*) on a same physical plane. The address-line layers **0***a***1**-**0***a***4** form two memory levels **16**A, **16**B, with the memory level **16**A stacked on the substrate circuit **0**K and the memory level **16**B stacked on the memory level **16**A. Memory cells (e.g., **7***aa*) are disposed at the intersections between two address lines (e.g., **1***a*, **2***a*). The memory levels **16**A, **16**B are communicatively coupled with the substrate circuit **0**K through contact vias **1***av*, **3***av.*

[0037] The non-array region **71** also contains a portion of substrate circuit **0**K (FIG. **1**EB). Without the 3D-M array **77**, the non-array region **71** comprises substantially fewer back-end-of-line (BEOL) layers than the array region **70**. As used hereinafter, a BEOL layer is an independent conductive layer (e.g., lines, planes) in the BEOL. In FIG. **1**EB, the array region **70** contains six (6) BEOL layers, including two (2) interconnect layers **0***m***1**-**0***m***2** of the substrate circuit **0**K, and four (4) address-line layers **0***a***1**-**0***a***4** of the 3D-M array **77**, while the non-array region **71** contains only two (2) BEOL layers, including the two (2) interconnect layers **0***m***1**-**0***m***2** of the substrate circuit **0**K. Because it contains neither memory cells nor interconnects, the space **72** above the substrate circuit **0**K is, in fact, wasted.

[0038] The array region **70** contains a plurality of 3D-M arrays **77** and the associated local peripheral circuit **75** and logic circuit **78** (FIG. **1**EA). The local peripheral circuit **75** and the logic circuit **78** are disposed on the substrate **0** and located near the footprint of the 3D-M array **77**. Because they are vertically stacked above the substrate **0** (not formed on the substrate **0**), the 3D-M arrays **77** are drawn by dashed lines. On the other hand, the non-array region **71** contains at least a global peripheral circuit **73** of the 3D-M arrays **77**. They are disposed on the substrate **0** and located outside the footprints of all 3D-M arrays **77**. The local peripheral circuit **75** and global peripheral circuit **73** are collectively referred to peripheral circuits **79**.

[0039] In the 3-D processor die **80**, the non-array region **71** occupies a large die area. At present, the non-array region **71** occupies .sup.˜30% of the total die area. As the storage capacity increases, the number will become larger. Hence, the integrated 3-D processor **80** has a low array efficiency. As used hereinafter, the array efficiency is the ratio of the total footprints of the 3D-M arrays **77** on the substrate **0** and the total die area.

[0040] The prevailing belief in integrated circuits is that integration will lower the cost and improve performance. Accordingly, monolithic integration, which integrates all circuit components into a single die, is generally preferred. Monolithic integration is advantageous to two-dimensional (2-D) circuits, but not so to three-dimensional (3-D) circuits, more particularly when the 3-D circuits (e.g., 3D-M arrays) are mixed with the 2-D circuits. As used hereinafter, a 2-D circuit is a circuit whose active elements (e.g., transistors, memory cells) are disposed on a 2-D plane (e.g., on a top surface of a semiconductor substrate); whereas, a 3-D circuit is a circuit whose active elements (e.g., transistors, memory cells) are disposed in a 3-D space (i.e., stacked along a direction perpendicular to a top surface of a semiconductor substrate).

[0041] Monolithic integration, when applied to the integration of the 3-D circuits and 2-D circuits, has several drawbacks. First of all, because their manufacturing processes are not compatible, integration will force the logic circuit **78** and the peripheral circuit **79** to use the expensive manufacturing process for the 3D-M arrays **77**. Adding the fact that its array efficiency is lower, the overall cost of the integrated 3-D processor die **80** becomes higher.

[0042] To make things worse, because the 3D-M arrays **77** have stringent requirements on manufacturing, the BEOL process of the integrated 3-D processor die **80** is generally optimized for the 3D-M arrays **77**, at the price of worsening the performance of the logic circuit **78** and the peripheral circuit **79**. For the integrated 3-D processor **80**, the logic circuit **78** and the peripheral circuit **79** can only contain a small number of the interconnect layers (as few as two); or, use slower high-temperature interconnect materials (which support the high-temperature manufacturing process for the 3D-M arrays **77**, e.g., tungsten). As a result, the overall performance of the integrated 3-D processor die **80** suffers.

[0043] Finally, with monolithic integration, the die area occupied by the local logic circuit **78** (i.e., the logic circuit under the 3D-M array) is limited by the footprint of the 3D-M array **77**. As a result, the local logic circuit **78** has limited functionalities. Furthermore, because monolithic integration fixedly integrates the 3D-M arrays **77** with the logic circuit **78**, the integrated 3-D processor **80** can only perform fixed functions. To perform another function, the whole 3-D processor **80** needs to be re-designed and re-manufactured, which are time-consuming and expensive.

OBJECT OF THE INVENTION

[0044] It is a principle object of the present invention to provide a 3-D processor with a lower overall cost.

[0045] It is a further object of the present invention to provide a 3-D processor with a better overall performance.

[0046] It is a further object of the present invention to provide a 3-D processor with more processing power and more flexible functionalities.

[0047] It is a further object of the present invention to provide a 3-D processor with more computational density.

[0048] It is a further object of the present invention to provide a 3-D processor with more computational complexity.

[0049] It is a further object of the present invention to improve the speed and efficiency of mathematical computing.

[0050] It is a further object of the present invention to improve the speed and efficiency of computer simulation.

[0051] It is a further object of the present invention to customize non-arithmetic functions.

[0052] It is a further object of the present invention to realize re-configurable computing.

[0053] It is a further object of the present invention to improve the speed and efficiency of pattern processing on large pattern databases.
[0054] It is a further object of the present invention to enhance information security.
[0055] It is a further object of the present invention to improve the speed and efficiency of big-data analytics.
[0056] It is a further object of the present invention to improve the speed and efficiency of speech recognition, as well as enable audio search in an audio archive.
[0057] It is a further object of the present invention to improve the speed and efficiency of image recognition, as well as enable video search in a video archive.
[0058] It is a further object of the present invention to improve the speed and efficiency of neural processing.
[0059] In accordance with these and other objects of the present invention, the present invention discloses a discrete 3-D processor.

SUMMARY OF THE INVENTION

[0060] The present invention follows a design paradigm distinct from any prior arts: partition a processor (logic+memory) based on dimensionality, not functionality. In prior arts, logic and memory are partitioned into different dice based on functionalities (i.e., logic die and memory die). The present invention re-partitions the processor system by dis-integrating 2-D and 3-D circuits. To be more specific, the 2-D circuits and the 3-D circuits are partitioned into different dice as much as possible so that they can be optimized separately. Accordingly, the present invention discloses a discrete 3-D processor, comprising: a plurality of storage-processing units (SPU's), each of said SPU's comprising at least a 3D-M array and a logic circuit; first and second dice, wherein said first die comprises said 3D-M array, said second die comprises at least a portion of said logic circuit and an off-die peripheral-circuit component of said 3D-M array, said first and second dice are communicatively coupled by a plurality of inter-die connections. Simply put, the first die is a memory die with multiple functional physical levels, whereas the second die is a logic die with a single functional physical level.
[0061] Different from the integrated 3-D processor where all peripheral-circuit components are located in the same die as the 3D-M arrays, at least one peripheral-circuit component of the 3D-M arrays is not located in the first die, but located in the second die. Accordingly, this peripheral-circuit component located in the second die is referred to as an off-die peripheral-circuit component. During design, the discrete 3-D processor is partitioned in such a way that the second die comprises as many off-die peripheral-circuit components as possible. Apparently, this partitioning scheme improves the array efficiency of the first die. It should be noted that, although it comprises the 3D-M arrays, the first die per se is not a functional memory die because of the missing peripheral-circuit components. Its performance cannot meet the industrial standards of the memory product of the same type.
[0062] Designed and manufactured separately, the first and second dice have substantially different BEOL structures. Because the BEOL structures of the second die could be independently optimized, the off-die peripheral-circuit components and the logic circuits could have a lower cost and a better performance than their counterparts in the integrated 3-D processor. In the following paragraphs, the discrete 3-D processor is compared with the integrated 3-D processor in several aspects.
[0063] First of all, because it does not include at least a portion of the peripheral circuits and logic circuits, the first die has a better array efficiency. In addition, as a 2-D circuit, the second die comprises substantially fewer BEOL layers than the integrated 3-D processor and can be made with the conventional manufacturing process. Because the wafer cost is roughly proportional to the number of BEOL layers, the second die would have a much lower wafer cost than the integrated 3-D processor. Hence, the total die cost of the discrete 3-D processor (which includes first and second dice) is lower than that of the integrated 3-D processor (which includes a single die). Even with the

extra bonding cost, the discrete 3-D processor still has a lower overall cost than the integrated 3-D processor for a given storage capacity.

[0064] Secondly, because they can be independently optimized, the off-die peripheral-circuit components and the logic circuits of the discrete 3-D processor could have a better performance than their counterparts in the integrated 3-D processor. In one preferred embodiment, the number of the interconnect layers (e.g., four, eight, or even more) in the second die is more than that of the substrate circuits (e.g., two) of the integrated 3-D processor die (or, the first die). In another preferred embodiment, the second die comprises high-speed interconnect materials (e.g., copper), not the high-temperature interconnect materials (e.g., tungsten) used in the integrated 3-D processor (or, the first die), which are generally slower. As a result, the discrete 3-D processor has a better overall performance than the integrated 3-D processor.

[0065] Lastly, in the integrated 3-D processor, the logic circuit can occupy a small die area and has less processing power, because it is disposed in a single die (e.g., within the footprint of the 3D-M array on the substrate). In comparison, in the discrete 3-D processor, the logic circuit can occupy a larger die area and have more processing power, due to the fact that the logic circuit can be disposed in two dice (e.g., a first portion of the logic circuit is disposed in the projection of the 3D-M array on the second die; and, a second portion of the logic circuit is disposed in the second die). In addition, designed and manufactured separately, the second die enjoys more flexibility in design and manufacturing. By combining the same first die with different second dice, different functionalities can be realized for different applications. Better yet, these different functionalities can be implemented in a relatively short time and under a relatively small budget. As a result, the discrete 3-D processor has more processing power and more flexible functionalities than the integrated 3-D processor.

[0066] The following paragraphs provide an overview of the applications of the preferred discrete 3-D processor.

[A] Mathematical Computing

[0067] When applied to the mathematical computing, the preferred discrete 3-D processor computes non-arithmetic functions. It uses memory-based computation (MBC), which carries out computation primarily with the LUT stored in the 3D-M arrays (i.e., 3DM-LUT). Compared with the conventional logic-based computation (LBC), the 3DM-LUT used by the MBC has a much larger capacity than the conventional LUT. For example, a single 3D-XPoint die has a storage capacity of 128 Gb, far larger than any conventional LUT (tens of kb). It can be used to store tens of thousands of non-arithmetic functions, including various types of transcendental functions and special functions. Although arithmetic operations are still performed for most MBC's, using a larger 3DM-LUT as a starting point, the MBC only needs to calculate a polynomial to a smaller order. For the MBC, the fraction of computation done by the memory circuit is significantly more than the logic circuit.

[0068] Accordingly, the present invention discloses a discrete 3-D processor for computing at least a non-arithmetic function, comprising: a plurality of storage-processing units (SPU's), each of said SPU's comprising at least a three-dimensional memory (3D-M) array and an arithmetic logic circuit (ALC), wherein said 3D-M array stores at least a portion of a look-up table (LUT) for said non-arithmetic function, said ALC performs arithmetic operations on selected data from said LUT; first and second dice, wherein said first die comprises said 3D-M array, said second die comprises at least a portion of said ALC and an off-die peripheral-circuit component of said 3D-M array, said first die does not comprise said off-die peripheral-circuit component, said first and second dice are separate dice communicatively coupled by a plurality of inter-die connections; wherein said non-arithmetic function includes more operations than the arithmetic operations provided by said ALC.

[B] Computer Simulation

[0069] When applied to the computer simulation, the preferred discrete 3-D processor computes non-arithmetic models. It still uses the MBC. The MBC brings about significant performance

improvement for computer simulation. With significantly more built-in functions (from about ten types to tens of thousands), the prevailing framework of computer simulation (including the foundation, function and modeling layers) is flattened. The hardware-implemented functions, which were only available to the foundation layer, now become available to the function and modeling layers. Not only mathematical functions in the function layer can be directly realized by hardware, but also mathematical models in the modeling layer. In the function layer, mathematical functions can be computed by a function-by-LUT method, i.e., the function values are calculated by reading the 3DM-LUT plus polynomial interpolation. In the modeling layer, mathematical models can be computed by a model-by-LUT method, i.e., the input-output characteristics of a system component are calculated by reading the 3DM-LUT plus polynomial interpolation. Rapid and efficient computation through 3DM-LUT would lead to a paradigm shift for computer simulation.

[0070] Accordingly, the present invention discloses a discrete 3-D processor for computing at least a non-arithmetic model, comprising: a plurality of storage-processing units (SPU's), each of said SPU's comprising at least a three-dimensional memory (3D-M) array and an arithmetic logic circuit (ALC), wherein said 3D-M array stores at least a portion of a look-up table (LUT) for said non-arithmetic model, said ALC performs arithmetic operations on selected data from said LUT; first and second dice, wherein said first die comprises said 3D-M array, said second die comprises at least a portion of said ALC and an off-die peripheral-circuit component of said 3D-M array, said first die does not comprise said off-die peripheral-circuit component, said first and second dice are separate dice communicatively coupled by a plurality of inter-die connections; wherein said non-arithmetic model includes more operations than the arithmetic operations provided by said ALC.

[C] Configurable Computing Array

[0071] When applied to configurable gate array, the preferred discrete 3-D processor is a discrete 3-D configurable computing array. It can not only customize logic functions and arithmetic functions, but also customize non-arithmetic functions. Accordingly, the present invention discloses a discrete 3-D configurable computing array for customizing at least a non-arithmetic function, comprising: an array of configurable logic elements (CLE's) and/or configurable interconnects (CIT's); an array of configurable computing elements (CCE's) comprising at least a three-dimensional memory (3D-M) array for storing at least a portion of a look-up table (LUT) of said non-arithmetic function; first and second dice, wherein said first die comprises said 3D-M array, said second die comprises at least a portion of said array of CLE's/CIT's and an off-die peripheral-circuit component of said 3D-M array, said first die does not comprise said off-die peripheral-circuit component, said first and second dice are separate dice communicatively coupled by a plurality of inter-die connections; whereby said non-arithmetic function is customized by programming said array of CLE's/CIT's and said array of CCE's; wherein said non-arithmetic function includes more operations than the arithmetic operations provided by said CLE.

[0072] The usage cycle of the CCE comprises two stages: a configuration stage and a computation stage. At the configuration stage, the LUT for a non-arithmetic function is loaded into the 3D-M array. At the computation stage, the values of the non-arithmetic function are read out from the LUT. For an electrically re-programmable 3D-M, different non-arithmetic functions can be realized by loading the LUT's of different non-arithmetic functions into the 3D-M array at different usage cycles. Hence, re-configurable computing can be realized.

[D] Pattern Processing

[0073] When applied to pattern processing, the preferred discrete 3-D processor is a discrete 3-D pattern processor. Its basic functionality is pattern processing. More importantly, the patterns it processes are stored locally. Because the pattern-storage circuit is close to the pattern-processing circuit, it takes a short time to read new patterns. In addition, the preferred 3-D pattern processor comprises tens of thousands of storage-processing units (SPU's). During pattern processing, the input data are sent to all SPU's and processed simultaneously, thus achieving massive parallelism.

The preferred 3-D pattern processor can realize fast and efficient pattern processing for large pattern databases.
[0074] Accordingly, the present invention discloses a discrete 3-D pattern processor, comprising: an input for transferring a first portion of a first pattern; a plurality of storage-processing units (SPU's) communicatively coupled with said input, each of said SPU's comprising at least a three-dimensional memory (3D-M) array and a pattern-processing circuit, wherein said 3D-M array stores at least a second portion of a second pattern, said pattern-processing circuit performs pattern processing for said first and second patterns; first and second dice, wherein said first die comprises said 3D-M array, said second die comprises at least a portion of said pattern-processing circuit and an off-die peripheral-circuit component of said 3D-M array, said first die does not comprise said off-die peripheral-circuit component, said first and second dice are separate dice communicatively coupled by a plurality of inter-die connections.
[E] Neural Processing
[0075] When applied to neural network, the preferred discrete 3-D processor is a discrete 3-D neuro-processor. Its basic functionality is neural processing. More importantly, the synaptic weights required for neural processing are stored locally. Because the memory storing the synaptic weights is close to the neuro-processing circuit, it takes a short time to read synaptic weights. In addition, the preferred 3-D neuro-processor comprises tens of thousands of storage-processing units (SPU's). During neural processing, the input data are sent to all SPU's and processed simultaneously, thus achieving massive parallelism. The preferred 3-D neural process can realize fast and efficient neural processing.
[0076] Accordingly, the present invention discloses a discrete 3-D neuro-processor, comprising: a plurality of storage-processing units (SPU's), each of said SPU's comprising at least a three-dimensional memory (3D-M) array and a neuro-processing circuit, wherein said 3D-M array stores at least a synaptic weight, said neuro-processing circuit performs neural processing with said synaptic weight; first and second dice, wherein said first die comprises said 3D-M array, said second die comprises at least a portion of said neuro-processing circuit and an off-die peripheral-circuit component of said 3D-M array, said first die does not comprise said off-die peripheral-circuit component, said first and second dice are separate dice communicatively coupled by a plurality of inter-die connections.

---

## Description

BRIEF DESCRIPTION OF DRAWINGS
[0077] FIG. **1**AA is a schematic view of a conventional processor (prior art); FIG. **1**AB lists all transcendental functions supported by an Intel IA-64 processor (prior art); FIG. **1**BA is a circuit block diagram of an amplifier circuit; FIG. **1**BB lists number of operations to calculate a current-voltage (I-V) point for various transistor models (prior art); FIG. **1**C is a schematic view of a neural network; FIG. **1**DA shows the tile organization of an accelerator used in a machine-learning supercomputer (prior art); FIG. **1**DB shows a tile architecture of the accelerator of FIG. **1**DA (prior art); FIG. **1**EA is a plan view of an integrated 3-D processor (prior art); FIG. **1**EB is its cross-sectional view;
[0078] FIG. **2**A is a circuit block diagram of a preferred discrete 3-D processor; FIG. **2**B is a circuit block diagram of a preferred storage-processing unit (SPU); FIG. **2**C is a circuit layout view of two dice in the preferred discrete 3-D processor;
[0079] FIGS. **3**A-**3**D are cross-sectional views of two dice in four preferred discrete 3-D processors;
[0080] FIGS. **4**A-**4**E are cross-sectional views of five preferred first dice;
[0081] FIG. **5** is a cross-sectional view of a preferred second die;

[0082] FIG. **6**A is a circuit layout view of a preferred first die; FIGS. **6**BA-**6**BB are circuit layout views of two preferred second dice;
[0083] FIGS. **7**A-**7**C are circuit block diagrams of three preferred SPU's;
[0084] FIGS. **8**A-**8**C are circuit layout views of three preferred SPU's on the first and second dice;
[0085] FIG. **9** is a circuit block diagram of a computing element in a preferred discrete 3-D processor for computing non-arithmetic functions or models;
[0086] FIGS. **10**A-**10**C are the circuit block diagrams of three preferred ALC's;
[0087] FIG. **11**A is a circuit block diagram of a first preferred computing element; FIG. **11**B is a detailed circuit diagram of the first preferred computing element;
[0088] FIG. **12** is a circuit block diagram of a second preferred computing element;
[0089] FIG. **13** is a circuit block diagram of a third preferred computing element;
[0090] FIG. **14**A is a circuit block diagram of a configurable block used in a preferred 3-D configurable computing array; FIG. **14**B lists the functional blocks in the configurable block;
[0091] FIGS. **15**A-**15**B are circuit block diagrams of two preferred configurable computing elements (CCE's);
[0092] FIG. **16** discloses two usage cycles of a preferred re-configurable computing element;
[0093] FIG. **17**A shows an interconnect library supported by a preferred configurable interconnect (CIT); FIG. **17**B shows a logic library supported by a preferred configurable logic element (CLE);
[0094] FIG. **18** is a circuit block diagram of a first preferred 3-D configurable computing array;
[0095] FIG. **19** shows an instantiation of the first preferred 3-D configurable computing array;
[0096] FIG. **20** is a circuit block diagram of a second preferred 3-D configurable computing array;
[0097] FIGS. **21**A-**21**B show two instantiations of the second preferred 3-D configurable computing array;
[0098] FIG. **22** is a circuit block diagram of a preferred discrete 3-D parallel processor;
[0099] FIG. **23** is a circuit block diagram of a SPU in a preferred discrete 3-D pattern processor;
[0100] FIG. **24** is a circuit block diagram of a SPU in a preferred discrete 3-D neuro-processor;
[0101] FIG. **25** is a circuit block diagram of a preferred neuro-processing circuit;
[0102] FIGS. **26**A-**26**B are circuit block diagrams of two preferred computing circuits.
[0103] It should be noted that all the drawings are schematic and not drawn to scale. Relative dimensions and proportions of parts of the device structures in the figures have been shown exaggerated or reduced in size for the sake of clarity and convenience in the drawings. The same reference symbols are generally used to refer to corresponding or similar features in the different embodiments.
[0104] As used hereinafter, the symbol “/” means the relationship of “and” or “or”. The phrase “memory” is used in its broadest sense to mean any semiconductor device, which can store information for short term or long term. The phrase “memory array (e.g., 3D-M array)” is used in its broadest sense to mean a collection of all memory cells sharing at least an address line. The phrase “(data) processing” is used in its broadest sense to mean modification of information in any manner detectable by an external user or a host; whereas, “peripheral circuit (of a memory array)” does not modify information stored herein viewed from an external user or a host. The phrase “circuits on a substrate” is used in its broadest sense to mean that all active elements (e.g., transistors, memory cells) or portions thereof are located in the substrate, even though the interconnects coupling these active elements are located above the substrate. The phrase “circuits above a substrate” is used in its broadest sense to mean that all active elements (e.g., transistors, memory cells) are located above the substrate, not in the substrate. The phrase “communicatively coupled” is used in its broadest sense to mean any coupling whereby electrical signals may be passed from one element to another element. The phrase “look-up table (LUT)” could refer to either the data in the LUT, or the memory circuit storing the LUT (i.e., the LUT memory); the present invention does not differentiate them. The phrase “pattern” could refer to either pattern per se, or the data related to a pattern; the present invention does not differentiate them.

DETAILED DESCRIPTION OF THE INVENTION

[0105] Those of ordinary skills in the art will realize that the following description of the present invention is illustrative only and is not intended to be in any way limiting. Other embodiments of the invention will readily suggest themselves to such skilled persons from an examination of the within disclosure.

[0106] Referring now to FIGS. **2**A-**2**C, an overview of a preferred discrete 3-D processor **100** is disclosed. FIG. **2**A is its circuit block diagram. The preferred discrete 3-D processor **100** not only processes data, but also stores data. More importantly, most data it processes are stored locally and in close proximity. The preferred discrete 3-D processor **100** comprises an array with m rows and n columns (m×n) of storage-processing units (SPU's) **100***aa*-**100***mn*. Using the SPU **100***ij* as an example, it has an input **110** and an output **120**. In general, the preferred discrete 3-D processor **100** comprises thousands to tens of thousands of SPU's **100***aa*-**100***mn* and therefore, it supports massive parallelism.

[0107] FIG. **2**B is a circuit block diagram of a preferred SPU **100***ij*. The SPU **100***ij* comprises a memory circuit **170** and a logic circuit **180**, which are communicatively coupled by inter-die connections **160** (referring to FIGS. **3**A-**3**D). The memory circuit **170** comprises at least a 3D-M array. The 3D-M array **170** stores data, whereas the logic circuit **180** processes at least a portion of these data. Because the 3D-M array **170** is disposed on a different die than the logic circuit **180** (referring to FIG. **2**C), the 3D-M array **170** is drawn by dashed lines.

[0108] FIG. **2**C shows a preferred implementation of the preferred discrete 3-D processor **100**. It comprises a first die (also known as memory die) **100***a* and at least a second die (also known as logic die) **100***b*. The first die **100***a* comprises 3-D circuits, i.e., the 3D-M arrays **170**. The second die **100***b* comprises 2-D circuits, e.g., at least a logic circuit **180** and a peripheral-circuit component **190** of the 3D-M array **170**. The inter-die connections **160** communicatively couple the first and second dice **100***a*, **100***b*. The peripheral-circuit component **190** is a peripheral circuit of the 3D-M array **170**. Since it is disposed on a different die than the 3D-M arrays **170**, the peripheral-circuit component **190** is referred to as an off-die peripheral-circuit component. Note that a portion of the logic circuit may be disposed on the first die **100***a*, e.g., this portion of the logic circuit may be disposed underneath the 3D-M array **170**. For reason of simplicity, unless being pointed out specifically, the logic circuit in this specification refers to the logic circuit **180** on the second die **100***b*.

[0109] The preferred discrete 3-D processor **100** is partitioned in such a way that the second die **100***b* comprises as many off-die peripheral-circuit components **190** as possible. The peripheral-circuit component is an essential circuit without which a memory die (e.g., the first die **100***a*) cannot perform even the basic memory functions (for example, its performance cannot meet the industrial standards of the memory product of the same type). Typical off-die peripheral-circuit component **190** could be an address decoder, a sense amplifier, a programming circuit, a read-voltage generator, a write-voltage generator, a data buffer, or a portion thereof.

[0110] The read/write-voltage generator converts an external power supply into a read/write voltage of the 3D-M array **170**, which generally has a different value than the external power supply. The read/write-voltage generator preferably uses a DC-to-DC converter. It could be a step-up circuit, whose output voltage is higher than the input voltage, or a step-down circuit, whose output voltage is lower than the input voltage. Examples of the step-up circuits include a charge-pump circuit and a boost converter, and examples of the step-down circuits include a low dropout circuit and a buck converter.

[0111] Referring now to FIGS. **3**A-**3**D, four preferred discrete 3-D processors **100** are shown with focus on the implementations of inter-die connections **160**. In these preferred embodiments, the first and second dice **100***a*, **100***b* are vertically stacked. In FIG. **3**A, the first and second dice **100***a*, **100***b* are stacked along the direction perpendicular to the dice **100***a*, **100***b*. Both the first and second dice **100***a*, **100***b* face upward (i.e., along the +z direction). They are communicatively coupled by

the bond wires **160***w*, which realize the inter-die connections **160**.

[0112] In FIG. **3**B, the first and second dice **100***a*, **100***b* are placed face-to-face, i.e., the first die **100***a* faces upward (i.e., along the +z direction), while the second die is flipped so that it faces downward (i.e., along the −z direction). They are communicatively coupled by the micro-bumps **160***x*, which realize the inter-die connections **160**. In this preferred embodiment, the first and second dice **100***a*, **100***b* have same die sizes. All edges of the first and second dice **100***a*, **100***b* are aligned. For example, the left edge of the first die **100***a* is aligned with the left edge of the second die **100***b*; and, the right edge of the first die **100***a* is aligned with the right edge of the second die **100***b*. This is due to the fact that the preferred discrete 3-D processor **100** is diced from bonded wafers (which bond first and second wafers, with the first wafer comprising the first die **100***a* and the second wafer comprising the second dice **100***b*).

[0113] The preferred embodiment of FIG. **3**C comprises two memory dice **100***a***1**, **100***a***2** and a logic die **100***b*. To avoid confusion, the dice **100***a***1**, **100***a***2** are referred to as memory dice instead of first dice; and, the die **100***b* is referred to as the logic die instead of the second die. Each of the memory dice **100***a***1**, **100***a***2** comprises a plurality of 3D-M arrays **170**. The memory dice **100***a***1**, **100***a***2** are vertically stacked and communicatively coupled by the through-silicon vias (TSV's) **160***y*. The stack of the memory dice **100***a***1**, **100***a***2** is communicatively coupled with the logic die **100***b* through the micro-bumps **160***x*. The TSV's **160***y* and the micro-bumps **160***x* realize the inter-die connections **160**. In this preferred embodiment, the logic circuits **180** on the logic die **100***b* process data from both memory dice **100***a***1**, **100***a***2**. Similar to FIG. **3**B, the logic and memory dice **100***b*, **100***a***1**, **100***a***2** have same die sizes; all edges of these dice **100***a***1**, **100***a***2**, and **100***b* are aligned.

[0114] The preferred embodiment in FIG. **3**D uses hybrid bonding. To be more specific, a first dielectric layer **168***a* is deposited on top of the first die **100***a* and first vias **160***za* are etched in the first dielectric layer **168***a*. Then a second dielectric layer **168***b* is deposited on top of the second die **100***b* and second vias **160***zb* are etching in the second dielectric layer **168***b*. After flipping the second die **100***b* and aligning the first and second vias **160***za*, **160***zb*, the first and second dice **100***a*, **100***b* are bonded. Finally, the first and second dice **100***a*, **100***b* are communicatively coupled by the contacted first and second vias **160***za*, **160***zb*, which realizes the inter-die connections **160**. Because they can be made with the standard manufacturing process, the first and second vias **160***za*, **160***zb* are small and numerous. As a result, the inter-die connections **160** have a large bandwidth. In this preferred embodiment, the first and second vias **160***za*, **160***zb* are collectively referred to as vertical interconnect accesses (VIA's). Similar to FIG. **3**B, the first and second dice **100***a*, **100***b* have same die sizes; all edges of the dice **100***a*, **100***b* are aligned.

[0115] In the above embodiments, the memory circuit **170** and the logic circuit **180** are close (relative to the conventional von Neumann architecture). In addition, for the embodiments of FIGS. **3**B-**3**D, more particularly for the embodiments of FIGS. **3**C-**3**D, the inter-die connections (e.g., TSV's, or VIA's) **160** are numerous. They can realize a large bandwidth between the memory circuit **170** and the logic circuit **180**. Adding massive parallelism (FIG. **2**A), the preferred discrete 3-D processor **100** can achieve great performance.

[0116] Referring now to FIGS. **4**A-**4**E, a preferred first dice **100***a* comprising different types of the 3D-M array **170** is shown. In the preferred embodiments of FIGS. **4**A-**4**D, each of the 3D-M arrays **170** uses monolithic integration per se, i.e., the memory cells are vertically stacked without any semiconductor substrate therebetween. In the preferred embodiment of FIG. **4**E, a preferred DRAM die **100***a* comprises 3-D structures.

[0117] Based on its physical structure, a 3D-M can be categorized into horizontal 3D-M (3D-M.sub.H) and vertical 3D-M (3D-M.sub.V). In a 3D-M.sub.H, all address lines are horizontal. The memory cells form a plurality of horizontal memory levels which are vertically stacked above each other. A well-known 3D-M.sub.H is 3D-XPoint. In a 3D-M.sub.V, at least one set of the address lines are vertical. The memory cells form a plurality of vertical memory strings which are placed

side-by-side on/above the substrate. A well-known 3D-M.sub.V is 3D-NAND. In general, the 3D-M.sub.H (e.g., 3D-XPoint) is faster, while the 3D-M.sub.V (e.g., 3D-NAND) is denser.

[0118] 3D-M can be categorized into RAM (random access memory) and ROM (read-only memory). The RAM provides random data access and can be used as cache. Examples of RAM include 3D-SRAM, 3D-DRAM, 3D-RRAM, 3D-MRAM, 3D-FeRAM, and others. The ROM can store data for long term. It is a non-volatile memory (NVM) and may be electrically writable. Examples of ROM include 3D-MPROM, 3D-OTP, 3D-MPT, 3D-EPROM, 3D-EEPROM, 3D-flash, 3D-NOR, 3D-NAND, 3D-XPoint, and others.

[0119] Based on the programming methods, the 3D-M can be categorized into 3-D writable memory (3D-W) and 3-D printed memory (3D-P). The 3D-W cells are electrically programmable. Based on the number of programmings allowed, the 3D-W can be further categorized into three-dimensional one-time-programmable memory (3D-OTP) and three-dimensional multiple-time-programmable memory (3D-MTP, including re-programmable). Common 3D-MTP includes 3D-XPoint and 3D-NAND. Other 3D-MTP's include memristor, resistive random-access memory (RRAM or ReRAM), phase-change memory (PCM), programmable metallization cell (PMC) memory, conductive-bridging random-access memory (CBRAM), and the like.

[0120] For the 3D-P, data are recorded into the 3D-P cells using a printing method during manufacturing. These data are fixedly recorded and cannot be changed after manufacturing. The printing methods include photo-lithography, nano-imprint, e-beam lithography, DUV lithography, and laser-programming, etc. An exemplary 3D-P is three-dimensional mask-programmed read-only memory (3D-MPROM), whose data are recorded by photo-lithography. Because a 3D-P cell does not require electrical programming and can be biased at a larger voltage during read than the 3D-W cell, the 3D-P is faster.

[0121] In FIGS. **4**A-**4**B, the first die **100***a* comprises a substrate circuit **0**Ka and a plurality of 3D-M.sub.H arrays **170** vertically stacked thereon. The substrate circuit **0**Ka includes transistors **0***t* and substrate interconnects **0***ia*. The transistors **0***t* are disposed on a first semiconductor substrate **0***a* and communicatively coupled by the substrate interconnects **0***ia*. The substrate interconnects **0***ia* include two interconnect layers **0***m***1***a*-**0***m***2***a*, each of which comprises a plurality of interconnects **0***m* on a same physical plane. The 3D-M.sub.H array **170** includes four address-line layers **0***a***1***a*-**0***a***4***a*. Each address-line layer (e.g., **0***a***1***a*) comprises a plurality of address lines (e.g., **1***a*) on a same physical plane. These address-line layers **0***a***1***a*-**0***a***4***a* form two memory levels **16**A, **16**B, with the memory level **16**A stacked on the substrate circuit **0**Ka and the memory level **16**B stacked on the memory level **16**A. Memory cells (e.g., **7***aa*) are disposed at the intersections between two address lines (e.g., **1***a*, **2***a*). The memory levels **16**A, **16**B are communicatively coupled with the substrate circuit **0**Ka through contact vias **1***av*, **3***av*, which form intra-die connections **150**. The contact vias **1***av*, **3***av* comprise a plurality of vias, each of which penetrates through a dielectric level and communicatively couples the vias above and below. Note that the substrate circuit **0**Ka comprises at least a portion of a peripheral circuit of the 3D-M arrays **170**, which is referred to as on-die peripheral-circuit component **140**. In some embodiments, the substrate circuit **0**Ka may also comprise a portion of the logic circuit.

[0122] As used hereinafter, the total number of BEOL layers N.sub.A of the 3D-M array **170** counts all BEOL layers within the footprint of the 3D-M array **170** (including the 3D-M array **170** and the substrate circuit **0**Ka underneath the 3D-M array **170**). In other words, N.sub.A is counted from the surface of the substrate **0***a* to the topmost BEOL layer in the region of the 3D-M array **170**. In a simplified counting method, only layers of conductive lines (not layers of vias) are counted. Alternatively, layers of vias are also counted. Except those specifically pointed out, the simplified counting method will be used in this specification. In the examples of FIGS. **4**A-**4**B, N.sub.A is equal to six (6), including two (2) interconnect layers **0***m***1***a*-**0***m***2***a*, and four (4) address-line layers **0***a***1***a*-**0***a***4***a*. On the other hand, the total BEOL thickness T.sub.A of the 3D-M array **170** is a height measured from the surface of the substrate **0***a* to the topmost BEOL layer **0***a***4***a*.

[0123] The 3D-M.sub.H arrays **170** in FIG. **4**A are 3D-W arrays. Its memory cell **7***aa* comprises a programmable layer **5** and a diode layer **6**. The programmable layer **5** could be an antifuse layer (which can be programmed once and used for the 3D-OTP) or a resistive RAM (RRAM) layer (which can be re-programmed and used for the 3D-MTP). The diode layer **6** is broadly interpreted as any layer whose resistance at the read voltage is substantially lower than when the applied voltage has a magnitude smaller than or polarity opposite to that of the read voltage. The diode could be a semiconductor diode (e.g., p-i-n silicon diode), or a metal-oxide (e.g., TiO2) diode.
[0124] The 3D-M.sub.H arrays **170** in FIG. **4**B are 3D-P arrays. It has at least two types of memory cells: a high-resistance memory cell **7***ab*, and a low-resistance memory cell **7***ac*. The low-resistance memory cell **7***ac* comprises a diode layer **6**, which is similar to that in the 3D-W; whereas, the high-resistance memory cell **7***ab* comprises at least a high-resistance layer **9**, which could simply be a layer of insulating dielectric (e.g., silicon oxide, or silicon nitride). It can be physically removed at the location of the low-resistance memory cell **7***ac* during manufacturing.
[0125] In FIGS. **4**C-**4**D, the first die **100***a* comprises a substrate circuit **0**Ka and a plurality of 3D-M.sub.V arrays **170** vertically stacked thereon. The substrate circuit **0**Ka is similar to those in FIGS. **4**A-**4**B. In some embodiments of the 3D-M.sub.V, the substrate circuit **0**Ka (e.g., the on-die peripheral-circuit components **140**) is not disposed under the 3D-M arrays **170**, but disposed next to the 3D-M arrays **170** (as shown in FIG. **4**E). The 3D-M.sub.V array **170** comprises a plurality of vertically stacked horizontal address-line layers **0***a***1***a*-**0***a***8***a*. Each horizontal address-line layer (e.g., **0***a***5***a*) comprises a plurality of horizontal address lines (e.g., **15**) on a same physical plane. The 3D-M.sub.V array **170** also comprises a set of vertical address lines, which are perpendicular to the surface of the substrate **0***a*. The 3D-M.sub.V has the largest storage density among semiconductor memories. For reason of simplicity, the intra-die connections **150** between the 3D-M.sub.V arrays **170** and the substrate circuit **0**Ka are not shown. They are well known to those skilled in the art.
[0126] The preferred 3D-M.sub.V array **170** in FIG. **4**C is based on vertical transistors or transistor-like devices. It comprises a plurality of vertical memory strings **16**X, **16**Y placed side-by-side. Each memory string (e.g., **16**Y) comprises a plurality of vertically stacked memory cells (e.g., **18***ay*-**18***hy*). Each memory cell (e.g., **18***fy*) comprises a vertical transistor, which includes a gate (acts as a horizontal address line) **15**, a storage layer **17**, and a vertical channel (acts as a vertical address line) **19**. The storage layer **17** could comprise oxide-nitride-oxide layers, oxide-poly silicon-oxide layers, or the like. This preferred 3D-M.sub.V array **170** is a 3D-NAND and its manufacturing details are well known to those skilled in the art. In the embodiment of FIG. **4**C, the total number of BEOL layers N.sub.A of the 3D-M.sub.V array **170** is equal to eleven (11), including two (2) interconnect layers **0***m***1***a*-**0***m***2***a*, eight (8) address-line layers **18***ay*-**18***hy*, and one (1) topmost bit-line layer BL**1**; whereas, the total BEOL thickness T.sub.A of the 3D-M.sub.V array **170** is measured from the surface of the substrate **0***a* to the topmost BEOL layer BL**1**.
[0127] The preferred 3D-M.sub.V array **170** in FIG. **4**D is based on vertical diodes or diode-like devices. In this preferred embodiment, the 3D-M.sub.V array comprises a plurality of vertical memory strings **16**U-**16**W placed side-by-side. Each memory string (e.g., **16**U) comprises a plurality of vertically stacked memory cells (e.g., **18***au*-**18***hu*). The 3D-M.sub.V array **170** comprises a plurality of horizontal address lines (word lines) **15** which are vertically stacked above each other. After etching through the horizontal address lines **15** to form a plurality of vertical memory wells **11**, the sidewalls of the memory wells **11** are covered with a programmable layer **13**. The memory wells **11** are then filled with a conductive materials to form vertical address lines (bit lines) **19**. The conductive materials could comprise metallic materials or doped semiconductor materials. The memory cells **18***au*-**18***hu* are formed at the intersections of the word lines **15** and the bit line **19**. The programmable layer **13** could be one-time-programmable (OTP, e.g., an antifuse layer) or multiple-time-programmable (MTP, e.g., an RRAM layer). In the embodiment of FIG. **4**D, the total number of BEOL layers N.sub.A of the 3D-M.sub.V array **170** is equal to ten (10),

including two (2) interconnect layers **0***m***1***a*-**0***m***2***a*, and eight (8) address-line layers **18***au*-**18***hu*; whereas, the total BEOL thickness T.sub.A of the 3D-M.sub.V array **170** is measured from the surface of the substrate **0***a* to the topmost address-line layer **18***hu*.

[0128] To minimize interference between memory cells, a diode is preferably formed between the word line **15** and the bit line **19**. In a first embodiment, this diode is the programmable layer **13** per se, which could have an electrical characteristic of a diode. In a second embodiment, this diode is formed by depositing an extra diode layer on the sidewall of the memory well (not shown in this figure). In a third embodiment, this diode is formed naturally between the word line **15** and the bit line **19**, i.e., to form a built-in junction (e.g., P-N junction, or Schottky junction). More details on the built-in diode are disclosed in U.S. patent application Ser. No. 16/137,512, filed on Sep. 20, 2018.

[0129] Discrete 3-D processor can be extended to other semiconductor memories, e.g., a conventional DRAM. Because the capacitor in the DRAM cell has a 3-D structure (i.e., with a high aspect ratio), DRAM can be viewed as a special type of 3D-M (although the conventional DRAM just comprises a single level of memory cells on the substrate). Accordingly, the 3-D memory (3D-M) in this specification has two meanings: A) it could mean a semiconductor memory with vertically stacked memory cells (FIGS. **4**A-**4**D); B) it could mean a semiconductor memory whose memory cell contains a 3-D structure (FIG. **4**E). Whether a 3D-M refers to meaning A) or B) can be determined from context. Note that the 3-D structure generally refers to an active component (e.g., capacitor, transistor) in a memory cell. It does not refer to interconnects (e.g., metal lines, vias). A typical 3-D structure has a large (e.g., >5) aspect ratio (i.e., the ratio of the vertical dimension and the horizontal dimension).

[0130] FIG. **4**E shows a first die **100***a* containing the conventional DRAM. It comprises a DRAM array **170** and a substrate circuit **0**Ka′. The DRAM array **170** further comprises a plurality of memory cells **20***a*, **20***b* (collectively labeled as **20**). Each memory cell **20** comprises one cell transistor **0***ta* and one cell capacitor **25** (i.e., a 1T1C cell). With the 3-D capacitors **25**, the BEOL manufacturing process of the DRAM array **170** is not compatible with its peripheral circuit **140** (which comprises standard BEOL structures and does not contain any 3-D structures). To be more specific, each memory cell **20** comprises a cell transistor **0***ta*. The gate **0***tg* of the cell transistor **0***ta* is formed in a trench **0***tt* in the substrate **0***a*. The substrate **0***a* functions as a channel of the cell transistor **0***ta*, while the gate **0***tg* functions as a word line of the DRAM array **170**. The gate **0***tg* is separated from the channel **0***a* by the gate dielectric **0***td*. On the other hand, the cell capacitor **25** comprises a bottom capacitor electrode **0***ce***1**, a layer of capacitor dielectric **0***cd*, and a top capacitor electrode **0***ce***2**. In this example, the capacitor **25** takes the shape of a pillar and has a large aspect ratio. Typical aspect ratio is >5 or even higher. The cell capacitor **25** is communicatively coupled with the cell transistor **0***ta* through a first via **0***av***1**, whereas the cell transistor **0***ta* is communicatively coupled with a bit line **0***ab* through a second via **0***v***2**.

[0131] The substrate circuit **0**Ka′ comprises at least an on-die peripheral-circuit component **140** of the DRAM array **170**. Unlike the embodiments shown in FIGS. **4**A-**4**D, the substrate circuit **0**Ka′ is disposed next to the DRAM array **170** (i.e., side by side), but not underneath the DRAM array **170**. Using standard manufacturing process, it comprises standard transistor **0***t* and standard interconnect layers **0***m***1**-**0***m***3**. Note that the topmost interconnect layer **0***m***3** is shared between the DRAM array **170** and the substrate circuit **0**Ka′; while the lower interconnect layers **0***m***1**-**0***m***2** are dedicated to the substrate circuit **0**Ka′ only. A complete manufacturing process of the first die **100***a* includes first steps to form the DRAM array **170**, second steps to form the substrate circuit **0**Ka′, and third steps to form the shared interconnect layer **0***m***3**. Because the footprint of the DRAM array **170** contains the DRAM array **170** only, its total number of BEOL layers N.sub.A is four (4), including: the bit line **0***ab*, the capacitor bottom electrode **0***ce***1**, the capacitor top electrode **0***ce***2**, and the shared interconnect **0***m***3**. The total BEOL thickness T.sub.A is a height measured from the surface of the substrate **0***a* to the shared interconnect **0***m***3**.

[0132] It can be observed from FIG. **4**E that DRAM faces the same challenges as those in FIGS. **4**A-**4**D, i.e., the BEOL structure of the DRAM array **170** is substantially different from that of its peripheral circuit **140**. Even worse, their front-end-of-line (FEOL) structures are also substantially different (trench transistor **0***ta* vs. planar transistors **0***t*). Based on the same reasoning as FIGS. **4**A-**4**D, it is preferred to dis-integrate the DRAM array and its peripheral circuit. In fact, the "dis-integration" scheme disclosed in the present invention can be applied to any memory whose memory array and peripheral circuit have different BEOL structures.

[0133] Referring now to FIG. **5**, a preferred second die **100***b* is disclosed. It comprises a logic circuit **180** and an off-die peripheral-circuit component **190** of the 3D-M array **170** (shown in FIGS. **4**A-**4**E). Both the logic circuit **180** and the off-die peripheral-circuit component **190** are conventional 2-D circuits **0**Kb with standard BEOL structures. Note that the logic circuit **180** and the off-die peripheral-circuit component **190** may have different number of interconnect layers. In this example, the interconnects **0***ib* of the logic circuit **180** comprise five interconnect layers **0***m***1***b*-**0***m***5***b*, and therefore, its total number of BEOL layers is five (5) and its total BEOL thickness is TLG whereas, the interconnects **0***ic* of the off-die peripheral-circuit component **190** comprise four interconnect layers **0***m***1***b*-**0***m***4***b*, and therefore, its total number of BEOL layers is four (4) and its total BEOL thickness is Tp.

[0134] Comparing the first die **100***a* (FIGS. **4**A-**4**E) and the second die **100***b* (FIG. **5**), they have different BEOL structures. To be more specific, the number of the BEOL layers in the first die **100***a* is more than that in the second die **100***b*. For example, the first die **100***a* in FIGS. **4**A-**4**B comprises six BEOL layers, while the first die **100***a* in FIGS. **4**C-**4**D comprises eleven or ten BEOL layers, both of which are larger than the four BEOL layers of the second die **100***b* in FIG. **5**. Even the number of the address-line layers in the first die **100***a* is larger than or equal to the number of BEOL layers (i.e., interconnect layers) of the second die **100***b*. More particularly, for the 3D-M.sub.V arrays **170**, the number of the address-line layers (roughly equal to the number of memory cells on each memory string, is approaching one hundred and increasing) is substantially larger than (e.g., at least twice as much as) the number of interconnects **0***ib* (e.g., four) in the second die **100***b*.

[0135] Besides the BEOL structures of the first and second dice **100***a*, **100***b*, the BEOL structures of three key elements of a 3-D processor (i.e., 3D-M array **170**, the logic circuit **180**, and the off-die peripheral-circuit component **190**) can also be compared. If the BEOL difference between the 3D-M array **170** and the off-die peripheral-circuit component **190** is substantially larger than the BEOL difference between the logic circuit **180** and the off-die peripheral-circuit component **190**, it is of great advantage to re-partition the processor system, i.e., dis-integrate the off-die peripheral-circuit component **190** from the 3D-M array **170** and integrate it to the logic circuit **180**. In one preferred comparing method, the BEOL difference is the difference of the total number of BEOL layers (N). If the difference between N.sub.A (the memory array **170**) and N.sub.P (the off-die peripheral-circuit component **190**) is substantially larger than the difference between N.sub.L (the logic circuit **180**) and N.sub.P, it is of great advantage to re-partition. In another preferred comparing method, the BEOL difference is the difference of the total BEOL thickness (T). If the difference between T.sub.A (the memory array **170**) and T.sub.P (the off-die peripheral-circuit component **190**) is substantially larger than the difference between T.sub.L (the logic circuit **180**) and T.sub.P, it is also of great advantage to re-partition.

[0136] After the system re-partitioning, the second die **100***b* is designed and manufactured independently from the first die **100***a*. As a result, the total number of the interconnect layers N.sub.P in the interconnects **0***ic* of the off-die peripheral-circuit component **190** in the second die **100***b* could be larger than the total number of the interconnect layers N.sub.P′ in the interconnects **0***ia* of the on-die peripheral-circuit component **140** in the first die **100***a*. For example, the off-die peripheral-circuit component **190** in the second die **100***b* of FIG. **5** comprises four interconnect layers (**0***m***1***b*-**0***m***4***b*), larger than the two interconnect layers (**0***m***1***a*-**0***m***2***a*) of the on-die peripheral-

circuit component **140** in the first die **100***a* of FIGS. **4**A-**4**E. Hence, the circuit layout of the off-die peripheral-circuit component **190** on the second die **100***b* is much easier than the on-die peripheral-circuit component **140** on the first die **100***a*. Moreover, the off-die peripheral-circuit component **190** on the second die **100***b* may comprise high-speed interconnect materials (e.g., copper), while at least a portion of the on-die peripheral-circuit component **140** on the first die **100***a* could only use high-temperature interconnect materials (e.g., tungsten), which have a higher resistivity and generally are slower.

[0137] Referring now to FIGS. **6**A-**6**BB, a preferred first die **100***a* and two preferred second dice **100***b* in two preferred 3-D processors **100** are shown. These figures disclose more details than FIG. **2**C. The preferred embodiment in these figures corresponds to that in FIG. **7**A and FIG. **8**A. To those skilled in the art, these layout designs can be easily extended to those in FIG. **7**B and FIG. **8**B, as well as FIG. **7**C And FIG. **8**C.

[0138] In FIG. **6**A, the preferred first die **100***a* comprises a plurality of 3D-M arrays **170***aa*-**170***mn*. In FIG. **6**BA, a preferred second die **100***b* comprises a plurality of logic circuits **180***aa*-**180***mn* and a global peripheral-circuit component **190**G. The global peripheral-circuit component **190**G is located outside the projections of all 3D-M arrays **170***aa*-**170***mn* on the second die **100***b*. The preferred 3-D processor **100** in FIG. **6**A and FIG. **6**BA adopts an “all-aligned” design, i.e., the circuits on the first and second dice **100***a*, **100***b* are designed in such a way that each logic circuit (e.g., **180***ij*) is aligned with at least one 3D-M array (e.g., **170***ij*) when the dice **100***a*, **100***b* are stacked (referring to FIGS. **8**A-**8**C). Because each logic circuit (e.g., **180***ij*) can be aligned and communicatively coupled with multiple 3D-M arrays (e.g., **170***ij*A-**170***ij*D, **170***ij*W-**170***ij*Z) (referring to FIGS. **8**B-**8**C), the pitch of the logic circuit (e.g., **180***ij*) on the second die **100***b* is an integer multiple of that of the 3D-M array (e.g., **170***ij*) on the first die **100***a*.

[0139] In FIG. **6**BB, another preferred second die **100***b* further comprises a plurality of local peripheral-circuit components **190***aa*-**190***mn*. Apparently, the preferred 3-D processor **100** in FIG. **6**A and FIG. **6**BB may also adopt the “all-aligned” design. For example, each local peripheral-circuit component (e.g., **190***ij*) is aligned and communicatively coupled with a 3D-M array (e.g., **170***ij*). Besides the local peripheral-circuit components **190***aa*-**190***mn*, the preferred embodiment of FIG. **6**BB may also comprise a global peripheral-circuit component **190**G (not shown here). As used hereinafter, all local peripheral-circuit components **190***aa*-**190***mn* and the global peripheral-circuit component **190**G are collectively referred to as off-die peripheral-circuit components **190**.

[0140] In the preferred embodiments of FIGS. **6**A-**6**BB, the local peripheral-circuit component (e.g., **190***ij*) generally includes at least a portion of an address decoder, a sense amplifier, a programming circuit, or others. It performs at least a portion of the read/write operation on the memory cells in each 3D-M array (e.g., **170***ij*). The global peripheral-circuit component **190**G generally includes at least a portion of a read-voltage generator, a write-voltage generator, a data buffer, or others. Apparently, there are many ways to partition the local and global peripheral-circuit components. For example, the local peripheral-circuit component may also include at least a portion of the read/write-voltage generator.

[0141] Referring now to FIGS. **7**A-**8**C, three preferred SPU **100***ij* are shown. FIGS. **7**A-**7**C are their circuit block diagrams (for reason of simplicity, the local peripheral-circuit component **190***ij* is not shown in FIGS. **7**A-**7**C) and FIGS. **8**A-**8**C are their circuit layout views. In these preferred embodiments, a logic circuit **180***ij* serves different number of 3D-M arrays **170***ij*.

[0142] In FIG. **7**A, the logic circuit **180***ij* serves one 3D-M array **170***ij*, i.e., it processes the data stored in the 3D-M array **170***ij*. In FIG. **7**B, the logic circuit **180***ij* serves four 3D-M arrays **170***ij*A-**170***ij*D, i.e., it processes the data stored in the 3D-M arrays **170***ij*A-**170***ij*D. In FIG. **7**C, the logic circuit **180***ij* serves eight 3D-M array **170***ij*A-**170***ij*D, **170***ij*W-**170***ij*Z, i.e., it processes the data stored in the 3D-M array **170***ij*A-**170***ij*D, **170***ij*W-**170***ij*Z. As will become apparent in FIGS. **8**A-**8**C, the more 3D-M arrays it serves, a larger footprint and more functionalities the logic circuit **180***ij* will have. In FIGS. **7**A-**7**C, because they are located on a different die **100***a* than the logic circuit

**180***ij* (referring to FIGS. **6**A-**6**BB), the 3D-M arrays **170***ij*-**170***ij*Z are drawn by dashed lines.
[0143] FIGS. **8**A-**8**C disclose the circuit layouts of the second die **100***b*, as well as the projections of the 3D-M arrays **170** (physically located on the first die **100***a*) on the second die **100***b* (drawn by dashed lines). Note that each SPU **100***ij* occupies a first area on the first die **100***a* and a second area on the second die **100***b*. The first and second areas substantially coincide. In other words, the projection of the first area on the second die **100***b* substantially overlaps the second area.
[0144] The embodiment of FIG. **8**A corresponds to that of FIG. **7**A. In this preferred embodiment, the logic circuit **180***ij* and the off-die peripheral-circuit component **190***ij* of the SPU **100***ij* are disposed on the second semiconductor substrate **0***b* of the second die **100***b*. They are at least partially covered by the 3D-M array **170***ij*. In this preferred embodiment, the pitch of the logic circuit **180***ij* is equal to the pitch of the 3D-M array **170***ij*. Because its area is smaller than the footprint of the 3D-M array **170***ij*, the logic circuit **180***ij* has limited functionalities. FIGS. **8**B-**8**C discloses two complex logic circuits **180***ij*.
[0145] The embodiment of FIG. **8**B corresponds to that of FIG. **7**B. In this preferred embodiment, the logic circuit **180***ij* and the off-die peripheral-circuit component **190***ij* are disposed on the second die **100***b*. They are at least partially covered by four 3D-M arrays **170***ij*A-**170***ij*D. Below the four 3D-M arrays **170***ij*A-**170***ij*D, the logic circuit **180***ij* can be laid out freely. Because the pitch of the logic circuit **180***ij* is twice as much as the pitch of the 3D-M arrays **170***ij*A-**170***ij*D, the logic circuit **180***ij* is four times larger than the footprints of the 3D-M arrays **170***ij*A-**170***ij*D and therefore, has more complex functionalities.
[0146] The embodiment of FIG. **8**C corresponds to that of FIG. **7**C. In this preferred embodiment, the logic circuit **180***ij* and the off-die peripheral-circuit component **190***ij* are disposed on the second die **100***b*. These 3D-M arrays **170***ij*A-**170***ij*D, **170***ij*W-**170***ij*Z are divided into two sets: a first set **170***ij*SA includes four 3D-M arrays **170***ij*A-**170***ij*D, and a second set **170***ij*SB includes four 3D-M arrays **170***ij*W-**170***ij*Z. Below the four 3D-M arrays **170***ij*A-**170***ij*D of the first set **170***ij*SA, a first component **180***ij*A of the logic circuit **180***ij* can be laid out freely. Similarly, below the four 3D-M array **170***ij*W-**170***ij*Z of the second set **170***ij*SB, a second component **180***ij*B of the logic circuit **180***ij* can be laid out freely. The first and second components **180***ij*A, **180***ij*B collectively form the logic circuit **180***ij*. In this embodiment, adjacent off-die peripheral-circuit components **190***ij* are separated by physical gaps (e.g., G) for forming the routing channel **182**, **184**, **186**, which provide coupling between different components **180***ij*A, **180***ij*B, or between different logic circuits. Because the pitch of the logic circuit **180***ij* is four times as much as the pitch of the 3D-M arrays **170***ij*A-**170***ij*D, **170***ij*W-**170***ij*Z (along the x direction), the logic circuit **180***ij* is eight times larger than the footprints of the 3D-M arrays **170***ij*A-**170***ij*D, **170***ij*W-**170***ij*Z and therefore, has even more complex functionalities.
[0147] Designed and manufactured separately, the first and second dice **100***a*, **100***b* have substantially different BEOL structures. Because the BEOL structures of the second die **100***b* could be independently optimized, the off-die peripheral-circuit components **190** and the logic circuits **180** could have a lower cost and a better performance than their counterparts in the integrated 3-D processor **80**. In the following paragraphs, the discrete 3-D processor **100** is compared with the integrated 3-D processor **80** in several aspects.
[0148] First of all, because it does not include the off-die peripheral-circuit component **190** and the logic circuit **180**, the first die **100***a* has a better array efficiency. In addition, as a 2-D circuit, the second die **100***b* comprises substantially fewer BEOL layers than the integrated 3-D processor, and can be made with the conventional manufacturing process. Because the wafer cost is roughly proportional to the number of BEOL layers, the second die **100***b* would have a much lower wafer cost than the integrated 3-D processor **80**. Hence, the total die cost of the discrete 3-D processor **100** (which includes first and second dice **100***a*, **100***b*) is lower than that of the integrated 3-D processor **80** (which includes a single die). Even though the extra bonding cost is counted, the discrete 3-D processor **100** still has a lower overall cost than the integrated 3-D processor **80** for a

given storage capacity.
[0149] Secondly, because they can be independently optimized, the off-die peripheral-circuit components **190** and the logic circuits **180** of the preferred discrete 3-D processor **100** have a better performance than their counterparts in the integrated 3-D processor **80**. In one preferred embodiment, the total number of the interconnect layers (e.g., four or even more, FIG. **5**) in the second die **100***b* is more than that of the substrate circuits **0**K (e.g., two, FIG. **1**EB) of the integrated 3-D processor **80** (or, the first die **100***a*). In another preferred embodiment, the second die **100***b* comprises high-speed interconnect materials (e.g., copper), not the high-temperature interconnect materials (e.g., tungsten) used in the integrated 3-D processor **80** (or, the first die **100***a*), which are generally slower. As a result, the discrete 3-D processor **100** has a better overall performance than the integrated 3-D processor **80**.
[0150] Lastly, in the integrated 3-D processor **80**, the logic circuit is smaller and has less processing power, because it is disposed in a single die **80** (e.g., within the footprint of the 3D-M array **77** on the substrate **0** in FIG. **1**EA). In comparison, in the discrete 3-D processor **100**, the logic circuit **180** is larger and has more processing power, because it can be disposed in two dice **100***a*, **100***b* (e.g., a first portion of the logic circuit is disposed under the 3D-M array **170***ij* of the first die **100***a* of FIG. **6**A, like the logic circuit **78** under the 3D-M array **77** of FIG. **1**EA; and, a second portion of the logic circuit is disposed in the second die **100***b* of FIG. **6**BA). In addition, designed and manufactured separately, the second die **100***b* enjoys more flexibility in design and manufacturing. By combining the same first die **100***a* with different second dice **100***b*, different functionalities can be realized for different applications. Better yet, these different functionalities can be implemented in a relatively short time and under a relatively small budget. As a result, the discrete 3-D processor **100** has more processing power and more flexible functionalities than the integrated 3-D processor **80**.
[0151] Both the “discrete 3-D processor (3D-P)” disclosed in the present invention and the “discrete 3-D memory (3D-M)” disclosed by U.S. Pat. No. 8,921,991 are based on dis-integration. However, the discrete 3D-P provides a significant improvement over the discrete 3D-M. First of all, the discrete 3D-P belongs to a processor field, while the discrete 3D-M belongs to a memory field. Applying dis-integration to a processor system leads to system re-partitioning, which cannot be realized by the discrete 3D-M. Furthermore, the discrete 3D-P solves a new problem raised by the discrete 3D-M. This can be understood by reviewing the development from the integrated 3D-M to the discrete 3D-P. Initially, the integrated 3D-M has a low array efficiency of .sup.˜70% (i.e., the 3D-M arrays occupy .sup.˜70% of the die area, while the peripheral circuits occupy .sup.˜30% of the die area). Realizing this and trying to improve the array efficiency, the discrete 3D-M dis-integrates the peripheral circuits (second die) from the 3D-M arrays (first die). Thus, the first die (3D-M arrays) improves its array efficiency to .sup.˜90%. However, the off-die peripheral-circuit components only occupy .sup.˜20% of the second die (as shown in FIGS. **8**A-**8**C). Hence, significant areas on the second die are empty and wasted. By forming logic circuits in these empty areas, the discrete 3D-P makes better use of the second die. Furthermore, by embedding processing power into the 3D-M, the discrete 3D-P can achieve significant performance improvements. All these advantages come at almost no extra cost, as the logic circuit is manufactured at the same time as the off-die peripheral-circuit components on the second die.
[0152] In the following paragraphs, the applications of the preferred discrete 3-D processors **100** will be overviewed.
[A] Mathematical Computing
[0153] When applied to the mathematical computing, the preferred discrete 3-D processor computes non-arithmetic functions. It uses memory-based computation (MBC), which carries out computation primarily with the LUT stored in the 3D-M arrays (i.e., 3DM-LUT). In this field of application, the SPU **100***ij* of FIG. **2**A is referred to as a computing element, where the 3D-M array **170** stores at least a portion of a look-up table (LUT) of a non-arithmetic function while the logic

circuit **180** is an arithmetic logic circuit (ALC).
[0154] Referring now to FIG. **9**, a computing element **100***ij* is disclosed. It includes an input **110**, an output **120**, a 3D-M array **170** and an ALC **180**. The 3D-M array **170** stores at least a portion of the LUT for a non-arithmetic function (or, a non-arithmetic model). The ALC **180** performs arithmetic operations on selected data from the LUT **170**. The 3D-M array **170** and the ALC **180** are communicatively coupled by the inter-die connections **160**. As mentioned before, a non-arithmetic function (or, non-arithmetic model) includes more operations than the arithmetic operations (i.e., addition, subtraction and multiplication) provided by the ALC **180**. As it cannot be represented by a combination of the basic arithmetic operations, a non-arithmetic function (or, non-arithmetic model) cannot be implemented by the ALC **180** alone. It has to be implemented by a combination of the ALC **180** and the LUT **170**.
[0155] Referring now to FIGS. **10**A-**10**C, three preferred ALC **180** are disclosed. The first preferred ALC **180** comprises an adder **180**A, the second preferred ALC **180** comprises a multiplier **180**M, with the third preferred ALC **180** comprising a multiply-accumulator (MAC), which includes an adder **180**A and a multiplier **180**M. The preferred ALC **180** could perform integer arithmetic operations, fixed-point arithmetic operations, or floating-point arithmetic operations.
[0156] Referring now to FIGS. **11**A-**11**B, a first preferred computing element **100***ij* for implementing a non-arithmetic function Y=f(X) is disclosed. It uses the function-by-LUT method. FIG. **11**A is its circuit block diagram. The ALC **180** comprises a pre-processing circuit **180**R, a 3DM-LUT **170**P, and a post-processing circuit **180**T. The pre-processing circuit **180**R converts the input variable (X) **110** into an address (A) of the 3DM-LUT **170**P. After the data (D) at the address (A) is read out from the 3DM-LUT **170**P, the post-processing circuit **180**T converts it into the function value (Y) **120**. A residue (R) of the input variable (X) is fed into the post-processing circuit **180**T to improve the calculation precision.
[0157] FIG. **11**B discloses the first preferred computing element **100***ij* which realizes a single-precision non-arithmetic function Y=f(X). The input variable X **110** has 32 bits (x31 . . . x0). The pre-processing circuit **180**R extracts the higher 16 bits (x31 . . . x16) thereof and sends it as a 16-bit address A to the 3DM-LUT **170**P. The pre-processing circuit **180**R further extracts the lower 16 bits (x15 . . . x0) and sends it as a 16-bit residue R to the post-processing circuit **180**T. The 3DM-LUT **170**P comprises two 3DM-LUT's **170**Q, **170**R. Both 3DM-LUT's **170**Q, **170**R have 2 Mb capacities (16-bit input and 32-bit output): the 3DM-LUT **170**Q stores the functional value D1=f(A), while the 3DM-LUT **170**R stores the first-order derivative value D2=f′(A). The post-processing circuit **180**T comprises a multiplier **180**M and an adder **180**A. The output value (Y) **190** has 32 bits and is calculated from polynomial interpolation. In this case, the polynomial interpolation is a first-order Taylor series: Y(X)=D1+D2*R=f(A)+f′(A)*R. To those skilled in the art, higher-order polynomial interpolation (e.g., higher-order Taylor series) can be used to improve the calculation precision.
[0158] When calculating a non-arithmetic function, combining the LUT with polynomial interpolation can achieve a high precision without using an excessively large LUT. For example, if only LUT (without any polynomial interpolation) is used to realize a single-precision function (32-bit input and 32-bit output), it would have a capacity of 232*32=128 Gb, which is impractical. By including polynomial interpolation, significantly smaller LUT's can be used. In the above embodiment, a single-precision function can be realized using a total of 4 Mb LUT (2 Mb for function values, and 2 Mb for first-derivative values) in conjunction with a first-order Taylor series calculation. This is significantly less than the LUT-only approach (4 Mb vs. 128 Gb).
[0159] Besides elementary functions (including algebraic functions and transcendental functions), the preferred 3-D processor **100** can be used to implement non-elementary functions such as special functions. Special functions can be defined by means of power series, generating functions, infinite products, repeated differentiation, integral representation, differential difference, integral, and functional equations, trigonometric series, or other series in orthogonal functions. Important

examples of special functions are gamma function, beta function, hyper-geometric functions, confluent hyper-geometric functions, Bessel functions, Legrendre functions, parabolic cylinder functions, integral sine, integral cosine, incomplete gamma function, incomplete beta function, probability integrals, various classes of orthogonal polynomials, elliptic functions, elliptic integrals, Lame functions, Mathieu functions, Riemann zeta function, automorphic functions, and others. The 3D-processor will simplify the calculation of special functions and promote their applications in scientific computation.

[0160] Referring now to FIG. **12**, a second preferred computing element **100***ij* for implementing a composite function Y=EXP[K*log(X)]=XK is disclosed. It uses the function-by-LUT method. The preferred computing element **100***ij* comprises two 3DM-LUT's **170**S, **170**T and a multiplier **180**M. The 3DM-LUT **170**S stores the LOG( ) values, while the 3DM-LUT **170**T stores the EXP( ) values. The input variable X is used as an address **110** for the 3DM-LUT **170**S. The output Log(X) **160***s* from the 3DM-LUT **170**S is multiplied by an exponent parameter K at the multiplier **180**M. The multiplication result K*LOG(X) is used as an address **160***t* for the 3DM-LUT **170**T, whose output **120** is Y=XK.

[0161] The functions computed by the computing elements in FIGS. **11**A-**11**B and FIG. **12** are combinational functions. As used hereinafter, a combinational function is a combination of at least two non-arithmetic functions. For example, a single-precision function is a combination of functional values and derivative values; a composite function is a combination of two functions. Accordingly, the present invention discloses a discrete 3-D processor for computing at least a combinational function, comprising: first and second 3D-M arrays, and an arithmetic logic circuit (ALC), wherein said first 3D-M array stores at least a first portion of a first look-up table (LUT) of a first non-arithmetic function, said second 3D-M array stores at least a second portion of a second LUT of a second non-arithmetic function, said ALC performs arithmetic operations on selected data from said first or second LUT's; first and second dice, wherein said first die comprises said first and second 3D-M arrays, said second die comprises at least a portion of said ALC and an off-die peripheral-circuit component of said first or second 3D-M array, said first die does not comprise said off-die peripheral-circuit component, said first and second dice are separate dice communicatively coupled by a plurality of inter-die connections; wherein said combinational function is a combination of said first and second non-arithmetic functions, said first and second non-arithmetic functions include more operations than the arithmetic operations provided by said ALC.

[B] Computer Simulation

[0162] When applied to the computer simulation, the preferred discrete 3-D processor computes non-arithmetic models. It still uses the MBC. The MBC brings about significant performance improvement for computer simulation. In this field of application, the SPU **100***ij* of FIG. **2**A is referred to as a computing element, where the 3D-M array **170** stores the LUT of a non-arithmetic model while the logic circuit is an ALC.

[0163] Referring now to FIG. **13**, a third preferred computing element **100***ij* to simulate the amplifier circuit **0**Y of FIG. **1**BA is disclosed. It uses the model-by-LUT method. The preferred computing element **100***ij* comprises a 3DM-LUT **170**U, an adder **180**A and a multiplier **180**M. The 3DM-LUT **170**U stores the data associated with the behaviors (e.g., input-output characteristics) of the transistor **0**T. By using the input voltage value (VIN) as an address **110** for the 3DM-LUT **170**U, the readout **160***u* of the 3DM-LUT **170**U is the drain-current value (ID). After the ID value is multiplied with the negative resistance value (−R) of the resistor **0**R by the multiplier **180**M, the multiplication result (−R*ID) is added to the VDD value by the adder **180**A to generate the output voltage value (VOUT) **120**.

[0164] The 3DM-LUT **170**U stores different forms of mathematical models. In one case, the mathematical model stored in the 3DM-LUT **170**U is raw measurement data, i.e., the measured input-output characteristics of the transistor **0**T. One example is the measured drain current vs. the

applied gate-source voltage (ID-VGS) characteristics. In another case, the mathematical model stored in the 3DM-LUT **170**U is the smoothed measurement data. The raw measurement data could be smoothed using a purely mathematical method (e.g., a best-fit model). Or, this smoothing process can be aided by a physical transistor model (e.g., a BSIM4 V3.0 transistor model). In a third case, the mathematical data stored in the 3DM-LUT include not only the measured data, but also its derivative values. For example, the 3DM-LUT **170**U stores not only the drain-current values of the transistor **0**T (e.g., the ID-VGS characteristics), but also its transconductance values (e.g., the Gm-VGS characteristics). With derivative values, polynomial interpolation can be used to improve the modeling precision using a reasonable-size 3DM-LUT **170**.

[0165] Model-by-LUT offers many advantages. By skipping two software-decomposition steps (from mathematical models to mathematical functions, and from mathematical functions to built-in functions), it saves substantial modeling time and energy. Model-by-LUT may need less LUT than function-by-LUT. Because a transistor model (e.g., BSIM4 V3.0) has hundreds of model parameters, calculating the intermediate functions of the transistor model requires extremely large LUT's. However, if function-by-LUT is skipped (namely, skipping the transistor models and the associated intermediate functions), the transistor behaviors can be described using only three parameters (including the gate-source voltage VGS, the drain-source voltage VDS, and the body-source voltage VBS). Hence, describing the mathematical models of the transistor **0**T requires relatively small LUT's.

[C] Configurable Computing Array

[0166] When applied to configurable gate array, the preferred discrete 3-D processor is a discrete 3-D configurable computing array. It can not only customize logic functions and arithmetic functions, but also customize non-arithmetic functions. In the preferred 3-D configurable computing array, the SPU **100***ij* of FIG. **2**A is also referred to as a configurable block

[0167] Referring now to FIGS. **14**A-**14**B, a preferred configurable block **100***ij* used in a preferred 3-D configurable computing array is disclosed. The preferred configurable block **100***ij* comprises a 3D-M array **170** and a logic circuit **180** (FIG. **14**A). The 3D-M array **170** stores at least a portion of the LUT of a non-arithmetic function. The logic circuit **180** could be an arithmetic logic circuit (ALC), a configurable logic element (CLE), or a configurable interconnect (CIT). Accordingly, the preferred configurable block **100***ij* includes the following functional blocks (FIG. **14**B): a configurable computing element (CCE) **400** (referring to FIGS. **15**A-**15**B), a CLE **200** (referring to FIG. **17**B) and a CIT **300** (referring to FIG. **17**A). The CCE **400** can realize a non-arithmetic function defined by the LUT; the CLE **200** can realize a selected one of the logic functions from a logic library; and, the CIT **300** can realize a selected one of the interconnects from an interconnect library.

[0168] For the CCE **400**, its input port IN includes input data **410**, the output port OUT includes output data **420**, and the configuration port CFG includes at least a configuration signal **430**. When the configuration signal **430** is “write”, the LUT of a non-arithmetic function is loaded into the CCE **400**; when the configuration signal **430** is “read”, the values of the non-arithmetic function are read out from the CCE **400**. FIGS. **15**A-**15**B disclose two preferred CCE's **400**. In FIG. **15**A, the CCE **400** is simply a 3D-M array **170**, which stores the functional values of the non-arithmetic function. In FIG. **15**B, the CCE **400** is a combination of a 3D-M array **170** and an ALC **180**. The 3D-M array **170** stores the functional values and the derivative values of the non-arithmetic function, while the ALC **180** performs polynomial interpolation with these values.

[0169] Referring now to FIG. **16**, two usage cycles **620**, **660** of a CCE are shown. Because its 3D-M arrays **170** are re-programmable, this configurable computing array can realize re-configurable computing. The first usage cycle **620** comprises two stages: a configuration stage **610** and a computation stage **630**. At the configuration stage **610**, the LUT of a first function is loaded into the 3D-M array **170**. At the computation stage **630**, the values of the first function are read out from the 3D-M array **170**. Similarly, the second usage cycle **660** also comprises a configuration stage **650**

and a computation stage **670**. This preferred embodiment is particularly suitable for single-instruction-multiple-data (SIMD)-type of data processing. Once the LUTs are loaded into the 3D-M array **170** in the configuration stage, a large amount of data can be fed into the CCE **400** and processed in high speed. SIMD has many applications, e.g., vector processing in image processing, massively parallel processing in scientific computing.

[0170] Referring now to FIGS. **17**A-**17**B, an interconnect library and a logic library are shown. FIG. **17**A shows the interconnect library supported by a CIT **300**. An interconnect library is a collection of all interconnects supported by the CIT **300**. This interconnect library includes the followings: a) the interconnects **302**/**304** are coupled, the interconnects **306**/**308** are coupled, but **302**/**304** are not connected with **306**/**308**; b) the interconnects **302**/**304**/**306**/**308** are all coupled; c) the interconnects **306**/**308** are coupled, but the interconnects **302**, **304** are not coupled, neither are **302**, **304** connected with **306**/**308**; d) the interconnects **302**/**304** are coupled, but the interconnects **306**, **308** are not coupled, neither are **306**, **308** connected with **302**/**304**; e) interconnects **302**, **304**, **306**, **308** are not coupled at all. As used hereinafter, the symbol "/" between two interconnects means that these two interconnects are coupled, while the symbol "," between two interconnects means that these two interconnects are not coupled. More details on the CIT's are disclosed in Freeman.

[0171] FIG. **17**B shows the logic library supported by a CLE **200**. A logic library is a collection of all logic functions supported by the CLE **200**. In this preferred embodiment, the inputs A and B are input data **210**, **220**, and the output C is the output data **230**, the logic library includes the followings: C=A, NOT A, A shift by n bits, AND(A,B), OR(A,B), NAND(A,B), NOR(A,B), XOR(A,B), A+B, A−B. To facilitate pipelining, the CLE **200** may comprise sequential logic such as flip-flops and registers. More details on the CLE's are disclosed in Freeman.

[0172] Referring now to FIG. **18**, a first preferred 3-D configurable computing array **100** is disclosed. It comprises first and second configurable slices **100**A, **100**B. Each configurable slice (e.g., **100**A) comprises a first array of CCE's (e.g., **400**AA-**400**AD) and a second array of CLE's (e.g., **200**AA-**200**AD). A configurable channel **320** is placed between the first array of CCE's (e.g., **400**AA-**400**AD) and the second array of CLE's (e.g., **200**AA-**200**AD). The configurable channels **310**, **330**, **350** are also placed between different configurable slices **100**A, **100**B. The configurable channels **310**-**350** comprise an array of CIT's **300**. For those skilled in the art, besides configurable channels, sea-of-gates may also be used.

[0173] Complex functions are common in computing. As used hereinafter, a complex function is a non-arithmetic function with multiple input independent variables (or, arguments); whereas, a basic function is a non-arithmetic function with a single input independent variable. In generally, a complex function can be expressed as a combination of basic functions. The preferred 3-D configurable computing array can customize complex functions, which is unimaginable for prior art. To customize a complex function, the complex function is first decomposed into a number of basic functions. Each basic function is then realized by loading its LUT's into the associated CCE's. Finally, the complex function is realized by programming the corresponding CLE's and CIT's.

[0174] FIG. **19** shows an instantiation of the first preferred 3-D configurable computing array **100** for customizing a complex function e=a.Math.SIN(b)+c.Math.COS(d). The CIT's **300** in the configurable channel **310**-**350** use the same convention as FIG. **17**A: the interconnects with dots at the intersection mean that the interconnects are connected; the interconnects without dots at the intersection mean that the interconnects are not connected; a broken interconnect means that two broken sections are two un-coupled interconnect lines. In this instantiation, the CCE **400**AA is configured to realize the function LOG( ), whose result LOG(a) is sent to a first input of the CLE **200**AA. The CCE **400**AB is configured to realize the function LOG[SIN( )], whose result LOG[SIN(b)] is sent to a second input of the CLE **200**AA. The CLE **200**AA is configured to realize arithmetic addition, whose result LOG(a)+LOG[SIN(b)] is sent the CCE **400**BA. The CCE **400**BA is configured to realize the function EXPO, whose result

EXP{LOG(a)+LOG[SIN(b)]}=a.Math.SIN(b) is sent to a first input of the CLE **200**BA. Similarly, through proper configurations, the results of the CCE's **400**AC, **400**AD, the CLE **200**AC, and the CCE **400**BC can be sent to a second input of the CLE **200**BA. The CLE **200**BA is configured to realize arithmetic addition, whose result a.Math.SIN(b)+c.Math.COS(d) is sent to the output e. Apparently, by changing its configuration, the preferred 3-D configurable computing array **100** can realize other complex functions.

[0175] Accordingly, the present invention discloses a discrete 3-D configurable computing array for customizing a complex function, comprising: a plurality of configurable logic elements (CLE's) and/or configurable interconnects (CIT's); first and second CCE's, wherein said first CCE comprises at least a first three-dimensional memory (3D-M) array for storing at least a first portion of a first look-up table (LUT) of a first non-arithmetic function, said second CCE comprises at least a second 3D-M array for storing at least a second portion of a second LUT of a second non-arithmetic function; first and second dice, wherein said first die comprises said 3D-M array, said second die comprises at least a portion of said array of CCE's/CIT's and an off-die peripheral-circuit component of said first or second 3D-M array, said first die does not comprise said off-die peripheral-circuit component, said first and second dice are separate dice communicatively coupled by a plurality of inter-die connections; whereby said complex function is realized by programming said CLE's/CIT's and said first and second CCE's, wherein said complex function is a combination of said first and second non-arithmetic functions, said first and second non-arithmetic functions include more operations than the arithmetic operations provided by said CLE's.

[0176] Referring now to FIG. **20**, a second preferred 3-D configurable computing array **100** is shown. Besides CCE's **400**A, **400**B and CLE **200**A, this preferred embodiment further comprises a multiplier **500**. With the addition of the multiplier **500**, the preferred 3-D configurable computing array **100** can realize more mathematical functions and its computational power is more powerful.

[0177] FIGS. **21**A-**21**B disclose two instantiations of the second preferred 3-D configurable computing array **100**. In the instantiation of FIG. **21**A, the CCE **400**A is configured to realize the function EXP(f), while the CCE **400**B is configured to realize the function INV(g). The configurable channel **370** is configured in such a way that the outputs of CCE's **400**A, **400**B are fed into the multiplier **500**. The final output is then h=EXP(f)*INV(g). On the other hand, in the instantiation of FIG. **21**B, the CCE **100**A is configured to realize the function SIN(f), while the CCE **400**B is configured to realize the function COS(g). The configurable channel **370** is configured in such a way that the outputs of CCE's **400**A, **400**B are fed into the CLE **200**A, which is configured to realize arithmetic addition. The final output is then h=SIN(f)+COS(g).

[D] Pattern Processing

[0178] When applied to pattern processing, the preferred discrete 3-D processor is a discrete 3-D pattern processor. Its basic functionality is pattern processing. More importantly, the patterns it processes are stored locally.

[0179] FIG. **22** shows the architecture of a preferred discrete 3-D parallel processor **100**. It comprises an array with m rows and n columns (m×n) of storage-processing units (SPU's) **100***aa*-**100***mn*. Each SPU (e.g., **100***ij*) is commutatively coupled with a common input **110** and a common output **120**. The input data are sent through the common input **110** to the SPU's **100***aa*-**100***mn*, where they are processed simultaneously. Because it comprises thousands to tens of thousands of SPU's **100***aa*-**100***mn*, the preferred discrete 3-D parallel processor **100** supports massive parallelism. The preferred discrete 3-D parallel processor **100** may be used for pattern processing, neural processing, and other fields of applications.

[0180] When used for pattern processing, the preferred 3-D parallel processor **100** is a discrete 3-D pattern processor. FIG. **23** shows a SPU **100***ij* used in the preferred 3-D pattern processor **100**. It comprises a pattern-storage circuit **170** and a pattern-processing circuit **180** communicatively coupled by a plurality of inter-die connections **160** (referring to FIGS. **3**A-**3**D). The pattern-storage circuit **170** comprises at least a 3D-M array for storing at least a portion of a pattern, whereas the

pattern-processing circuit **180** performs pattern processing for the pattern

[0181] The preferred discrete 3-D pattern processor **100** can be either processor-like or storage-like. The processor-like 3-D pattern processor **100** acts like a discrete 3-D processor with an embedded search-pattern library. It searches a target pattern from the input **110** against the search-pattern library. To be more specific, the 3D-M array **170** stores at least a portion of the search-pattern library (e.g., a virus library, a keyword library, an acoustic/language model library, an image model library); the input **110** includes a target pattern (e.g., a network packet, a computer file, audio data, or image data); the pattern-processing circuit **180** performs pattern processing on the target pattern with the search pattern. Because a large number of the SPU's **100***ij* (thousands to tens of thousands, referring to FIG. **22**) support massive parallelism and the inter-die connections **160** has a large bandwidth (referring to FIGS. **3**B-**3**D), the preferred 3-D processor with an embedded search-pattern library can achieve fast and efficient search.

[0182] Accordingly, the present invention discloses a discrete 3-D processor with an embedded search-pattern library, comprising: an input for transferring at least a portion of a target pattern; a plurality of storage-processing units (SPU's) communicatively coupled with said input, each of said SPU's comprising at least a three-dimensional memory (3D-M) array and a pattern-processing circuit, wherein said 3D-M array stores at least a portion of a search pattern, said pattern-processing circuit performs pattern processing on said target pattern with said search pattern; first and second dice, wherein said first die comprises said 3D-M array, said second die comprises at least a portion of said pattern-processing circuit and an off-die peripheral-circuit component of said 3D-M array, said first die does not comprise said off-die peripheral-circuit component, said first and second dice are separate dice communicatively coupled by a plurality of inter-die connections.

[0183] The storage-like discrete 3-D pattern processor **100** acts like a 3-D storage with in-situ pattern-processing capabilities. Its primary purpose is to store a target-pattern database, with a secondary purpose of searching the stored target-pattern database for a search pattern from the input **110**. To be more specific, a target-pattern database (e.g., computer files on a whole disk drive, a big-data database, an audio archive, an image archive) is stored and distributed in the 3D-M arrays **170**; the input **110** include at least a search pattern (e.g., a virus signature, a keyword, a model); the pattern-processing circuit **180** performs pattern processing on the target pattern with the search pattern. Because a large number of the SPU's **100***ij* (thousands to tens of thousands, referring to FIG. **22**) support massive parallelism and the inter-die connections **160** has a large bandwidth (referring to FIGS. **3**B-**3**D), the preferred 3-D storage can achieve a fast speed and a good efficiency.

[0184] Like the flash memory, a large number of the preferred discrete 3-D storages **100** can be packaged into a storage card (e.g., an SD card, a TF card) or a solid-state drive (i.e., SSD). These storage cards or SSD can be used to store massive data in the target-pattern database. More importantly, they have in-situ pattern-processing (e.g., searching) capabilities. Because each SPU **100***ij* has its own pattern-processing circuit **180**, it only needs to search the data stored in the local 3D-M array **170** (i.e., in the same SPU **100***ij*). As a result, no matter how large is the capacity of the storage card or the SSD, the processing time for the whole storage card or the whole SSD is similar to that for a single SPU **100***ij*. In other words, the search time for a database is irrelevant to its size, mostly within seconds.

[0185] In comparison, for the conventional von Neumann architecture, the processor (e.g., CPU) and the storage (e.g., HDD) are physically separated. During search, data need to be read out from the storage first. Because of the limited bandwidth between the CPU and the HDD, the search time for a database is limited by the read-out time of the database. As a result, the search time for the database is proportional to its size. In general, the search time ranges from minutes to hours, even longer, depending on the size of the database. Apparently, the preferred 3-D storage with in-situ pattern-processing capabilities **100** has great advantages in database search.

[0186] When a preferred 3-D storage with in-situ pattern-processing capabilities **100** performs

pattern processing for a large database (i.e., target-pattern database), the pattern-processing circuit **180** could just perform partial pattern processing. For example, the pattern-processing circuit **180** only performs a preliminary pattern processing (e.g., code matching, or string matching) on the database. After being filtered by this preliminary pattern-processing step, the remaining data from the database are sent through the output **120** to an external processor (e.g., CPU, GPU) to complete the full pattern processing. Because most data are filtered out by this preliminary pattern-processing step, the data output from the preferred 3-D storage **100** are a small fraction of the whole database. This can substantially alleviate the bandwidth requirement on the output **120**.

[0187] Accordingly, the present invention discloses a discrete 3-D storage with in-situ pattern-processing capabilities, comprising: an input for transferring at least a portion of a search pattern; a plurality of storage-processing units (SPU's) communicatively coupled with said input, each of said SPU's comprising at least a three-dimensional memory (3D-M) array and a pattern-processing circuit, wherein said 3D-M array stores at least a portion of a target pattern, said pattern-processing circuit performs pattern processing on said target pattern with said search pattern; first and second dice, wherein said first die comprises said 3D-M array, said second die comprises at least a portion of said pattern-processing circuit and an off-die peripheral-circuit component of said 3D-M array, said first die does not comprise said off-die peripheral-circuit component, said first and second dice are separate dice communicatively coupled by a plurality of inter-die connections

[0188] In the following paragraphs, applications of the preferred discrete 3-D pattern processor **100** are described. The fields of applications include: A) information security; B) big-data analytics; C) speech recognition; and D) image recognition. Examples of the applications include: a) information-security processor; b) anti-virus storage; c) data-analysis processor; d) searchable storage; e) speech-recognition processor; f) searchable audio storage; g) image-recognition processor; h) searchable image storage.

A) Information Security

[0189] Information security includes network security and computer security. To enhance network security, virus in the network packets needs to be scanned. Similarly, to enhance computer security, virus in the computer files (including computer software) needs to be scanned. Generally speaking, virus (also known as malware) includes network viruses, computer viruses, software that violates network rules, document that violates document rules and others. During virus scan, a network packet or a computer file is compared against the virus patterns (also known as virus signatures) in a virus library. Once a match is found, the portion of the network packet or the computer file which contains the virus is quarantined or removed.

[0190] Nowadays, the virus library has become large. It has reached hundreds of MB. On the other hand, the computer data that require virus scan are even larger, typically on the order of GB or TB, even bigger. On the other hand, each processor core in the conventional processor can typically check a single virus pattern once. With a limited number of cores (e.g., a CPU contains tens of cores; a GPU contains hundreds of cores), the conventional processor can achieve limited parallelism for virus scan. Furthermore, because the processor is physically separated from the storage in the von Neumann architecture, it takes a long time to fetch new virus patterns. As a result, the conventional processor and its associated architecture have a poor performance for information security.

[0191] To enhance information security, the present invention discloses several discrete 3-D pattern processors **100**. It could be processor-like or storage-like. For processor-like, the preferred discrete 3-D pattern processor **100** is an information-security processor, i.e., a processor for enhancing information security; for storage-like, the preferred discrete 3-D pattern processor **100** is an anti-virus storage, i.e., a storage with in-situ anti-virus capabilities.

a) Information-Security Processor

[0192] To enhance information security, the present invention discloses an information-security processor **100**. It searches a network packet or a computer file for various virus patterns in a virus

library. If there is a match with a virus pattern, the network packet or the computer file contains the virus. The preferred information-security processor **100** can be installed as a standalone processor in a network or a computer; or, integrated into a network processor, a computer processor, or a computer storage.

[0193] In the preferred information-security processor **100**, the 3D-M arrays **170** in different SPU **100***ij* stores different virus patterns. In other words, the virus library is stored and distributed in the SPU's **100***ij* of the preferred information-security processor **100**. Once a network packet or a computer file is received at the input **110**, at least a portion thereof is sent to all SPU's **100***ij*. In each SPU **100***ij*, the pattern-processing circuit **180** compares said portion of data against the virus patterns stored in the local 3D-M array **170**. If there is a match with a virus pattern, the network packet or the computer file contains the virus.

[0194] The above virus-scan operations are carried out by all SPU's **100***ij* at the same time. Because it comprises a large number of SPU's **100***ij* (thousands to tens of thousands), the preferred information-security processor **100** achieves massive parallelism for virus scan. Furthermore, because the inter-die connections **160** are numerous and the pattern-processing circuit **180** is physically close to the 3D-M arrays **170** (compared with the conventional von Neumann architecture), the pattern-processing circuit **180** can easily fetch new virus patterns from the local 3D-M array **170**. As a result, the preferred information-security processor **100** can perform fast and efficient virus scan. In this preferred embodiment, the 3D-M arrays **170** storing the virus library could be 3D-P, 3D-OTP or 3D-MTP; and, the pattern-processing circuit **180** is a code-matching circuit.

[0195] Accordingly, the present invention discloses a discrete information-security processor, comprising: an input for transferring at least a portion of data from a network packet or a computer file; a plurality of storage-processing units (SPU's) communicatively coupled with said input, each of said SPU's comprising at least a three-dimensional memory (3D-M) array and a code-matching circuit, wherein said 3D-M array stores at least a portion of a virus pattern, said code-matching circuit searches said virus pattern in said portion of data; first and second dice, wherein said first die comprises said 3D-M array, said second die comprises at least a portion of said code-matching circuit and an off-die peripheral-circuit component of said 3D-M array, said first die does not comprise said off-die peripheral-circuit component, said first and second dice are separate dice communicatively coupled by a plurality of inter-die connections.

b) Anti-Virus Storage

[0196] Whenever a new virus is discovered, the whole disk drive (e.g., hard-disk drive, solid-state drive) of the computer needs to be scanned against the new virus. This full-disk scan process is challenging to the conventional von Neumann architecture. Because a disk drive could store massive data, it takes a long time to even read out all data, let alone scan virus for them. For the conventional von Neumann architecture, the full-disk scan time is proportional to the capacity of the disk drive.

[0197] To shorten the full-disk scan time, the present invention discloses an anti-virus storage. Its primary function is a computer storage, with in-situ virus-scanning capabilities as its secondary function. Like the flash memory, a large number of the preferred anti-virus storage **100** can be packaged into a storage card or a solid-state drive for storing massive data and with in-situ virus-scanning capabilities.

[0198] In the preferred anti-virus storage **100**, the 3D-M arrays **170** in different SPU **100***ij* stores different data. In other words, massive computer files are stored and distributed in the SPU's **100***ij* of the storage card or the solid-state drive. Once a new virus is discovered and a full-disk scan is required, the pattern of the new virus is sent as input **110** to all SPU's **100***ij*, where the pattern-processing circuit **180** compares the data stored in the local 3D-M array **170** against the new virus pattern.

[0199] The above virus-scan operations are carried out by all SPU's **100***ij* at the same time and the

virus-scan time for each SPU **100***ij* is similar. Because of the massive parallelism, no matter how large is the capacity of the storage card or the solid-state drive, the virus-scan time for the whole storage card or the whole solid-state drive is more or less a constant, which is close to the virus-scan time for a single SPU **100***ij* and generally within seconds. On the other hand, the conventional full-disk scan takes minutes to hours, or even longer. In this preferred embodiment, the 3D-M arrays **170** storing massive computer data are preferably 3D-MTP; and, the pattern-processing circuit **180** is a code-matching circuit.

[0200] Accordingly, the present invention discloses a discrete anti-virus storage, comprising: an input for transferring at least a portion of a virus pattern; a plurality of storage-processing units (SPU's) communicatively coupled with said input, each of said SPU's comprising at least a three-dimensional memory (3D-M) array and a code-matching circuit, wherein said 3D-M array stores at least a portion of data, said code-matching circuit searches said virus pattern in said portion of data; first and second dice, wherein said first die comprises said 3D-M array, said second die comprises at least a portion of said code-matching circuit and an off-die peripheral-circuit component of said 3D-M array, said first die does not comprise said off-die peripheral-circuit component, said first and second dice are separate dice communicatively coupled by a plurality of inter-die connections.

B) Big-Data Analytics

[0201] Big data is a term for a large collection of data, with main focus on unstructured and semi-structure data. An important aspect of big-data analytics is keyword search (including string matching, e.g., regular-expression matching). At present, the keyword library becomes large, while the big-data database is even larger. For such large keyword library and big-data database, the conventional processor and its associated architecture can hardly perform fast and efficient keyword search on unstructured or semi-structured data.

[0202] To improve the speed and efficiency of big-data analytics, the present invention discloses several discrete 3-D pattern processors **100**. It could be processor-like or storage-like. For processor-like, the preferred discrete 3-D pattern processor **100** is a data-analysis processor, i.e., a processor for performing analysis on big data; for storage-like, the preferred discrete 3-D pattern processor **100** is a searchable storage, i.e., a storage with in-situ searching capabilities.

c) Data-Analysis Processor

[0203] To perform fast and efficient search on the input data, the present invention discloses a data-analysis processor **100**. It searches the input data for the keywords in a keyword library. In the preferred data-analysis processor **100**, the 3D-M arrays **170** in different SPU **100***ij* stores different keywords. In other words, the keyword library is stored and distributed in the SPU's **100***ij* of the preferred data-analysis processor **100**. Once data are received at the input **110**, at least a portion thereof is sent to all SPU's **100***ij*. In each SPU **100***ij*, the pattern-processing circuit **180** compares said portion of data against various keywords stored in the local 3D-M array **170**.

[0204] The above searching operations are carried out by all SPU's **100***ij* at the same time. Because it comprises a large number of SPU's **100***ij* (thousands to tens of thousands), the preferred data-analysis processor **100** achieves massive parallelism for keyword search. Furthermore, because the inter-die connections **160** are numerous and the pattern-processing circuit **180** is physically close to the 3D-M arrays **170** (compared with the conventional von Neumann architecture), the pattern-processing circuit **180** can easily fetch keywords from the local 3D-M array **170**. As a result, the preferred data-analysis processor **100** can perform fast and efficient search on unstructured data or semi-structured data.

[0205] In this preferred embodiment, the 3D-M arrays **170** storing the keyword library could be 3D-P, 3D-OTP or 3D-MTP; and, the pattern-processing circuit **180** is a string-matching circuit. The string-matching circuit could be implemented by a content-addressable memory (CAM) or a comparator including XOR circuits. Alternatively, keyword can be represented by a regular expression. In this case, the sting-matching circuit **180** can be implemented by a finite-state automata (FSA) circuit.

[0206] Accordingly, the present invention discloses a discrete data-analysis processor, comprising: an input for transferring at least a portion of data; a plurality of storage-processing units (SPU's) communicatively coupled with said input, each of said SPU's comprising at least a three-dimensional memory (3D-M) array and a string-matching circuit, wherein said 3D-M array stores at least a portion of a keyword, said string-matching circuit searches said keyword in said portion of data; first and second dice, wherein said first die comprises said 3D-M array, said second die comprises at least a portion of said string-matching circuit and an off-die peripheral-circuit component of said 3D-M array, said first die does not comprise said off-die peripheral-circuit component, said first and second dice are separate dice communicatively coupled by a plurality of inter-die connections.

d) Searchable Storage

[0207] Big-data analytics often requires full-database search, i.e., to search a whole big-data database for a keyword. The full-database search is challenging to the conventional von Neumann architecture. Because the big-data database is large, with a capacity of GB to TB, or even larger, it takes a long time to even read out all data, let alone analyze them. For the conventional von Neumann architecture, the full-database search time is proportional to the database size.

[0208] To improve the speed and efficiency of full-database search, the present invention discloses a searchable storage. Its primary function is database storage, with in-situ searching capabilities as its secondary function. Like the flash memory, a large number of the preferred searchable storage **100** can be packaged into a storage card or a solid-state drive for storing a big-data database and with in-situ searching capabilities.

[0209] In the preferred searchable storage **100**, the 3D-M arrays **170** in different SPU **100***ij* stores different portions of the big-data database. In other words, the big-data database is stored and distributed in the SPU's **100***ij* of the storage card or the solid-state drive. During search, a keyword is sent as input **110** to all SPU's **100***ij*. In each SPU **100***ij*, the pattern-processing circuit **180** searches the portion of the big-data database stored in the local 3D-M array **170** for the keyword.

[0210] The above searching operations are carried out by all SPU's **100***ij* at the same time and the keyword-search time for each SPU **100***ij* is similar. Because of massive parallelism, no matter how large is the capacity of the storage card or the solid-state drive, the keyword-search time for the whole storage card or the whole solid-state drive is more or less a constant, which is close to the keyword-search time for a single SPU **100***ij* and generally within seconds. On the other hand, the conventional full-database search takes minutes to hours, or even longer. In this preferred embodiment, the 3D-M arrays **170** storing the big-data database are preferably 3D-MTP; and, the pattern-processing circuit **100** is a string-matching circuit.

[0211] Because it has the largest storage density among all semiconductor memories, the 3D-MV is particularly suitable for storing a big-data database. Among all 3D-MV, the 3D-OTP.sub.V has a long data retention time and therefore, is particularly suitable for archiving. Fast searchability is important for archiving. A searchable 3D-OTP.sub.V will provide a large, inexpensive archive with fast searching capabilities.

[0212] Accordingly, the present invention discloses a discrete searchable storage, comprising: an input for transferring at least a portion of a keyword; a plurality of storage-processing units (SPU's) communicatively coupled with said input, each of said SPU's comprising at least a three-dimensional memory (3D-M) array and a string-matching circuit, wherein said 3D-M array stores at least a portion of data, said string-matching circuit searches said keyword in said portion of data; first and second dice, wherein said first die comprises said 3D-M array, said second die comprises at least a portion of said string-matching circuit and an off-die peripheral-circuit component of said 3D-M array, said first die does not comprise said off-die peripheral-circuit component, said first and second dice are separate dice communicatively coupled by a plurality of inter-die connections.

C) Speech Recognition

[0213] Speech recognition enables the recognition and translation of spoken language. It is

primarily implemented through pattern recognition between audio data and an acoustic model/language library, which contains a plurality of acoustic models or language models. During speech recognition, the pattern-processing circuit **180** performs speech recognition to the user's audio data by finding the nearest acoustic/language model in the acoustic/language model library. Because the conventional processor (e.g., CPU, GPU) has a limited number of cores and the acoustic/language model database is stored externally, the conventional processor and the associated architecture have a poor performance in speech recognition.

e) Speech-Recognition Processor

[0214] To improve the performance of speech recognition, the present invention discloses a speech-recognition processor **100**. In the preferred speech-recognition processor **100**, the user's audio data is sent as input **110** to all SPU **100***ij*. The 3D-M arrays **170** store at least a portion of the acoustic/language model. In other words, an acoustic/language model library is stored and distributed in the SPUs **100***ij*. The pattern-processing circuit **180** performs speech recognition on the audio data from the input **110** with the acoustic/language models stored in the 3D-M arrays **170**. In this preferred embodiment, the 3D-M arrays **170** storing the models could be 3D-P, 3D-OTP, or 3D-MTP; and, the pattern-processing circuit **180** is a speech-recognition circuit.

[0215] Accordingly, the present invention discloses a discrete speech-recognition processor, comprising: an input for transferring at least a portion of audio data; a plurality of storage-processing units (SPU's) communicatively coupled with said input, each of said SPU's comprising at least a three-dimensional memory (3D-M) array and a speech-recognition circuit, wherein said 3D-M array stores at least a portion of an acoustic/language model, said speech-recognition circuit performs pattern recognition on said portion of audio data with said acoustic/language model; first and second dice, wherein said first die comprises said 3D-M array, said second die comprises at least a portion of said speech-recognition circuit and an off-die peripheral-circuit component of said 3D-M array, said first die does not comprise said off-die peripheral-circuit component, said first and second dice are separate dice communicatively coupled by a plurality of inter-die connections.

f) Searchable Audio Storage

[0216] To enable audio search in an audio database (e.g., an audio archive), the present invention discloses a searchable audio storage. In the preferred searchable audio storage **100**, an acoustic/language model derived from the audio data to be searched for is sent as input **110** to all SPU **100***ij*. The 3D-M arrays **170** store at least a portion of the user's audio database. In other words, the audio database is stored and distributed in the SPUs **100***ij* of the preferred searching audio storage **100**. The pattern-processing circuit **180** performs speech recognition on the audio data stored in the 3D-M arrays **170** with the acoustic/language model from the input **110**. In this preferred embodiment, the 3D-M arrays **170** storing the audio database are preferably 3D-MTP; and, the pattern-processing circuit **180** is a speech-recognition circuit.

[0217] Accordingly, the present invention discloses a discrete searchable audio storage, comprising: an input for transferring at least a portion of an acoustic/language model; a plurality of storage-processing units (SPU's) communicatively coupled with said input, each of said SPU's comprising at least a three-dimensional memory (3D-M) array and a speech-recognition circuit, wherein said 3D-M array stores at least a portion of audio data, said speech-recognition circuit performs pattern recognition on said portion of audio data with said acoustic/language model; first and second dice, wherein said first die comprises said 3D-M array, said second die comprises at least a portion of said speech-recognition circuit and an off-die peripheral-circuit component of said 3D-M array, said first die does not comprise said off-die peripheral-circuit component, said first and second dice are separate dice communicatively coupled by a plurality of inter-die connections.

D) Image Recognition or Search

[0218] Image recognition enables the recognition of images. It is primarily implemented through

pattern recognition on image data with an image model, which is a part of an image model library. During image recognition, the pattern-processing circuit **180** performs image recognition to the user's image data by finding the nearest image model in the image model library. Because the conventional processor (e.g., CPU, GPU) has a limited number of cores and the image model database is stored externally, the conventional processor and the associated architecture have a poor performance in image recognition.

g) Image-Recognition Processor

[0219] To improve the performance of image recognition, the present invention discloses an image-recognition processor **100**. In the preferred image-recognition processor **100**, the user's image data is sent as input **110** to all SPU **100***ij*. The 3D-M arrays **170** store at least a portion of the image model. In other words, an image model library is stored and distributed in the SPUs **100***ij*. The pattern-processing circuit **180** performs image recognition on the image data from the input **110** with the image models stored in the 3D-M arrays **170**. In this preferred embodiment, the 3D-M arrays **170** storing the models could be 3D-P, 3D-OTP, or 3D-MTP; and, the pattern-processing circuit **180** is an image-recognition circuit.

[0220] Accordingly, the present invention discloses a discrete image-recognition processor, comprising: an input for transferring at least a portion of image data; a plurality of storage-processing units (SPU's) communicatively coupled with said input, each of said SPU's comprising at least a three-dimensional memory (3D-M) array and an image-recognition circuit, wherein said 3D-M array stores at least a portion of an image model, said image-recognition circuit performs pattern recognition on said portion of image data with said image model; first and second dice, wherein said first die comprises said 3D-M array, said second die comprises at least a portion of said image-recognition circuit and an off-die peripheral-circuit component of said 3D-M array, said first die does not comprise said off-die peripheral-circuit component, said first and second dice are separate dice communicatively coupled by a plurality of inter-die connections.

h) Searchable Image Storage

[0221] To enable image search in an image database (e.g., an image archive), the present invention discloses a searchable image storage. In the preferred searchable image storage **100**, an image model derived from the image data to be searched for is sent as input **110** to all SPU **100***ij*. The 3D-M arrays **170** store at least a portion of the user's image database. In other words, the image database is stored and distributed in the SPUs **100***ij* of the preferred searchable image storage **100**. The pattern-processing circuit **180** performs image recognition on the image data stored in the 3D-M arrays **170** with the image model from the input **110**. In this preferred embodiment, the 3D-M arrays **170** storing the image database are preferably 3D-MTP; and, the pattern-processing circuit **180** is an image-recognition circuit.

[0222] Accordingly, the present invention discloses a discrete searchable image storage, comprising: an input for transferring at least a portion of an image model; a plurality of storage-processing units (SPU's) communicatively coupled with said input, each of said SPU's comprising at least a three-dimensional memory (3D-M) array and an image-recognition circuit, wherein said 3D-M array stores at least a portion of image data, said image-recognition circuit performs pattern recognition on said portion of image data with said image model; first and second dice, wherein said first die comprises said 3D-M array, said second die comprises at least a portion of said image-recognition circuit and an off-die peripheral-circuit component of said 3D-M array, said first die does not comprise said off-die peripheral-circuit component, said first and second dice are separate dice communicatively coupled by a plurality of inter-die connections.

[E] Neural Network

[0223] When applied to neural network, the preferred discrete 3-D processor is a discrete 3-D neuro-processor. Its basic functionality is neural processing. More importantly, the synaptic weights required for neural processing are stored locally.

[0224] The preferred discrete 3-D neuro-processor uses the architecture of the preferred discrete 3-

D parallel processor **100** (FIG. **22**). FIG. **24** shows a SPU **100***ij* used in a preferred 3-D neuro-processor **100**. It comprises a neuro-storage circuit **170** and a neuro-processing circuit **180** communicatively coupled by a plurality inter-die connections **160** (referring to FIGS. **3**A-**3**D). The neuro-storage circuit **170** comprises at least a 3D-M array for storing at least a portion of a synaptic weight, whereas the neuro-processing circuit **180** performs neural processing with the synaptic weight.

[0225] Referring now to FIGS. **25**-**26**B, a preferred neuro-processing circuit **180** and its computing circuit **730** are disclosed. In the preferred embodiment of FIG. **25**, the preferred neuro-processing circuit **180** comprises a synaptic-weight (Ws) RAM **740**A, an input-neuron (Nin) RAM **740**B, and a computing circuit **730**. The Ws RAM **740**A is a buffer RAM which temporarily stores the synaptic weights **742** read from the 3D-M array(s) **170**. The Nin RAM **740**B is another buffer RAM which temporarily stores the input data **746** from the input **110**. The computing circuit **730** performs neural processing before generating output data **748**.

[0226] In the preferred embodiment of FIG. **26**A, a first preferred computing circuit **730** comprises a multiplier **732**, an adder **734**, a register **736** and an activation-function circuit **738**. The multiplier **732** multiplies the input data xi with the synaptic weight wij. Then the adder **734** and the register **736** together perform accumulative addition for the multiplication results (i.e., wij*xi). The resulting accumulative sum **737** is passed through the activation-function circuit **738** to generate the output data yj.

[0227] In the preferred embodiment of FIG. **26**B, a second preferred computing circuit **730** comprises a multiply-accumulate circuit (MAC) **732**′ in place of the multiplier **732** of FIG. **26**A. Of course, the MAC **732**′ comprises a multiplier too. The Ws RAM **740**A outputs not only a synaptic weight wij (through port **742***w*), but also a bias bi (through port **742***b*). The MAC **732**′ performs a biased-multiplication (i.e., wij*xi+bj) for the input data xi, the synaptic weight wij and the bias bj.

[0228] The activation function (e.g., a sigmoid function, a signum function, a threshold function, a piecewise-linear function, a step function, a tanh function, etc.) controls the amplitude of its output to be between certain values (e.g., between 0 and 1 or between −1 and 1). It is difficult to realize the activation function in the past. Following the same inventive spirit of the present invention, more particularly that in the section of “mathematical computing”, the logic circuit **180** on the second die **100***b* may comprise a non-volatile memory (NVM) for storing the LUT of the activation function. The NVM is generally a read-only memory (ROM), more particularly a 3-D read-only memory (3D-ROM). The 3D-ROM array can be further stacked above the multiplier/MAC **732** and the adder **734** and overlap them. As a result, the computing circuit **730** becomes quite simple—it only needs to realize multiplication and addition, but not activation function. The computing circuit **730** using the 3D-ROM array to realize the activation functions is small and therefore, has a large computational density.

[0229] While illustrative embodiments have been shown and described, it would be apparent to those skilled in the art that many more modifications than that have been mentioned above are possible without departing from the inventive concepts set forth therein. For example, the preferred 3-D processor could be a micro-controller, a controller, a central processing unit (CPU), a digital signal processor (DSP), a graphic processing unit (GPU), a network-security processor, an encryption/decryption processor, an encoding/decoding processor, a neural-network processor, or an artificial intelligence (AI) processor. These processors can be found in consumer electronic devices (e.g., personal computers, video game machines, smart phones) as well as engineering and scientific workstations and server machines. As another example, the 3D-M arrays in the discrete 3-D processor could be any memory arrays (e.g., RAM, ROM, or NVM arrays). As long as the structures of the memory arrays are substantially different from the structures of their peripheral circuits and the logic circuits (e.g., the BEOL structures of the memory arrays are substantially different from those of their peripheral circuits and the logic circuits), it is advantageous to partition a 3-D processor into two dice: a first die comprises the memory arrays; whereas, a second die

comprises at least a portion of their peripheral circuits and the logic circuits. The invention, therefore, is not to be limited except in the spirit of the appended claims.

## Claims

**1**. A discrete three-dimensional (3-D) processor, comprising: a plurality of storage-processing units (SPU's), wherein each of said plurality of SPU's comprises at least a memory array and a non-memory circuit, wherein said non-memory circuit is not a peripheral circuit of said memory array; a first die comprising the memory arrays of said plurality of SPU's; a second die comprising the non-memory circuits of said plurality of SPU's and at least an off-die peripheral-circuit component, wherein said off-die peripheral-circuit component is a peripheral circuit of the memory arrays of said plurality of SPU's; a plurality of inter-die connections communicatively coupling said first and second dice; wherein a first total-thickness difference of BEOL layers between said memory array and said off-die peripheral-circuit component is substantially larger than a second total-thickness difference of BEOL layers between said non-memory circuit and said off-die peripheral-circuit component.
**2**. The discrete 3-D processor according to claim 1, wherein said first and second dice are vertically stacked and have a same die size.
**3**. The discrete 3-D processor according to claim 2, wherein said non-memory circuit is a logic circuit or a processing circuit.
**4**. The discrete 3-D processor according to claim 2, wherein said memory array is a random-access memory (RAM) array or a non-volatile memory (NVM) array.
**5**. The discrete 3-D processor according to claim 1, wherein said first and second dice are vertically stacked and all edges of said first and second dice are aligned.
**6**. The discrete 3-D processor according to claim 5, wherein said non-memory circuit is a logic circuit or a processing circuit.
**7**. The discrete 3-D processor according to claim 5, wherein said memory array is a random-access memory (RAM) array or a non-volatile memory (NVM) array.
**8**. The discrete 3-D processor according to claim 1, wherein said first and second dice are vertically stacked; said each of said plurality of SPU's occupies a first area on said first die and a second area on said second die; and, said first and second areas substantially coincide.
**9**. The discrete 3-D processor according to claim 8, wherein said non-memory circuit is a logic circuit or a processing circuit.
**10**. The discrete 3-D processor according to claim 8, wherein said memory array is a random-access memory (RAM) array or a non-volatile memory (NVM) array.
**11**. A discrete three-dimensional (3-D) processor, comprising: a plurality of storage-processing units (SPU's), wherein each of said plurality of SPU's comprises a non-memory circuit, at least a memory array and an off-die peripheral-circuit component thereof, wherein said off-die peripheral-circuit component is a peripheral circuit of said memory array; and, said non-memory circuit is not a peripheral circuit of said memory array; a first die comprising the memory arrays of said plurality of SPU's; a second die comprising the non-memory circuits and the off-die peripheral-circuit components of said plurality of SPU's; a plurality of inter-die connections communicatively coupling said first and second dice; wherein a third total-thickness difference of BEOL layers between said memory array and said off-die peripheral-circuit component is substantially larger than a fourth total-thickness difference of BEOL layers between said non-memory circuit and said off-die peripheral-circuit component.
**12**. The discrete 3-D processor according to claim 11, wherein said first and second dice are vertically stacked and have a same die size.
**13**. The discrete 3-D processor according to claim 12, wherein said non-memory circuit is a logic circuit or a processing circuit.

**14**. The discrete 3-D processor according to claim 12, wherein said memory array is a random-access memory (RAM) array or a non-volatile memory (NVM) array.
**15**. The discrete 3-D processor according to claim 11, wherein said first and second dice are vertically stacked and all edges of said first and second dice are aligned.
**16**. The discrete 3-D processor according to claim 15, wherein said non-memory circuit is a logic circuit or a processing circuit.
**17**. The discrete 3-D processor according to claim 15, wherein said memory array is a random-access memory (RAM) array or a non-volatile memory (NVM) array.
**18**. The discrete 3-D processor according to claim 11, wherein said first and second dice are vertically stacked; said each of said plurality of SPU's occupies a first area on said first die and a second area on said second die; and, said first and second areas substantially coincide.
**19**. The discrete 3-D processor according to claim 18, wherein said non-memory circuit is a logic circuit or a processing circuit.
**20**. The discrete 3-D processor according to claim 18, wherein said memory array is a random-access memory (RAM) array or a non-volatile memory (NVM) array.

---