

ASSEMBLY HAVING A FACE-TO-FACE BONDED CHIPLET**CROSS-REFERENCE TO RELATED APPLICATIONS**

[0001] The present application claims the benefit of and priority to U.S. Provisional Patent Application No. 63/518,988, filed on August 11, 2023, entitled "INTEGRATED CIRCUIT HAVING MEMORIES AND A SHARED WRITE PORT", identified by Docket Number P23-133-US-PSP, the entire contents of which is incorporated herein by reference in its entirety.

[0002] The present application claims the benefit of and priority to U.S. Provisional Patent Application No. 63/602,733, filed on November 27, 2023, entitled "METHOD AND SYSTEM FOR KNOWN-GOOD-DIE TESTABILITY OF FACE-TO-FACE BONDED CHIPLETS", the entire contents of which is incorporated herein by reference in its entirety.

[0003] The present application claims the benefit of and priority to U.S. Provisional Patent Application No. 63/602,737, filed on November 27, 2023, entitled "SYSTEM AND METHOD FOR HAVING CORRECT-BY-CONSTRUCTION TIMING CLOSURE", the entire contents of which is incorporated herein by reference in its entirety.

[0004] The present application claims the benefit of and priority to U.S. Provisional Patent Application No. 63/567,649, filed on March 20, 2024, entitled "ASSEMBLY HAVING A FACE-TO-FACE BONDED CHIPLET", identified by Docket Number P24-052-US-PSP, the entire contents of which is incorporated herein by reference in its entirety.

[0005] The present application claims the benefit and priority to U.S. Provisional Patent Application No. 63/637,742, filed on April 23, 2024, entitled "INTEGRATED CIRCUIT HAVING MICROVAULT MEMORIES", identified by Docket Number P24-081-US-PSP, the entire contents of which is incorporated herein by reference in its entirety.

[0006] The present application claims the benefit of and priority to U.S. Provisional Patent Application No. 63/637,764, filed on April 23, 2024, entitled "FEFET STRUCTURES ON INTEGRATED CIRCUITS", identified by Docket Number P24-082-US-PSP, the entire contents of which is incorporated herein by reference in its entirety.

[0007] The present application claims the benefit of and priority to U.S. Provisional Patent Application No. 63/674,471, filed on July 23, 2024, entitled "SYSTEM, METHOD, AND APPARATUS FOR WAFER-SCALE MEMORY", identified by Docket Number P24-135-US-PSP, the entire contents of which is incorporated herein by reference in its entirety.

BACKGROUND*Relevant Field*

[0008] The present disclosure relates to integrated circuits. More particularly, the present disclosure relates to integrated circuits that form an assembly of face-to-face bonding of chiplets.

Description of Related Art

[0009] Chiplets refer to miniature chips that are designed to work as a single entity while using advanced packaging technology. These miniaturized chips are created by dividing the larger chip into several smaller chips, each with its own function or capability. The concept originated from the semiconductor industry's need to overcome the physical restrictions of traditional monolithic chip designs and achieve higher levels of integration. The idea behind chiplets is to create a modular system of interconnected and interchangeable chips that can be combined in different configurations to create advanced computing systems with improved performance, power efficiency, and functionality.

[0010] Chiplets can be based on different architectures, such as CPU, GPU, memory, or IO, and can be assembled and stacked in a variety of ways, depending on the specific application requirements. One of the advantages of the chiplet approach is the ability to mix and match different chiplets from different manufacturers to create custom solutions that meet specific computing needs. This approach also allows for faster time to market, reduced development costs, and increased flexibility, as chiplets can be upgraded or replaced without the need for a complete system redesign.

[0011] SUMMARY

[0012] An embodiment of an integrated circuit may comprise a first semiconductor device that includes a first programmable gate array. This array may be coupled with a variety of interface logics, among which is a first interface logic. Additionally, a first memory port may be present, connected to a first set of bonds on the surface of the first semiconductor device. The first programmable gate array is operatively connected to the first interface logic, allowing for communication through the first memory port.

[0013] In some embodiments, the first interface logic of the integrated circuit could be constituted from the first programmable gate array. The integrated circuit may feature the first interface logic distributed in a spaced pattern within the first programmable gate array. An embodiment of the first semiconductor device within the integrated circuit may include

multiple memory ports, inclusive of the first memory port, where these ports form a spaced pattern on the semiconductor device.

[0014] Additionally, the integrated circuit might also comprise multiple sets of bonds, including the first set of bonds, with these sets distributed in a spaced pattern on the surface of the first semiconductor device. In some embodiments, the first semiconductor device of the integrated circuit may include a plurality of memory ports, such as the first memory port. Here, each interface logic from the plurality of interface logics can be coupled to a corresponding memory port from the plurality of memory ports.

[0015] In certain embodiments, the first semiconductor device may also include multiple programmable gate arrays, encompassing the first programmable gate array. Each gate array from this plurality may be coupled to a respective interface logic from the plurality of interface logics. The integrated circuit could further comprise a number of cores, with each core from the plurality of cores being coupled to a respective interface logic of the plurality of interface logics. An embodiment of the integrated circuit may include various sets of bonds on the surface of the first semiconductor device. In this configuration, each memory port from the plurality of memory ports is coupled to a corresponding set of bonds from the plurality of sets of bonds, which includes the first set of bonds.

[0016] In some embodiments, the first set of bonds might be composed of metallic pads. The first semiconductor device of the integrated circuit may optionally include a network-on-chip. This network is configured to enable communication among at least two programmable gate arrays from the plurality of programmable gate arrays. Optionally, the first semiconductor device may comprise at least one network-on-chip communication system. This system connects at least two processing elements implemented on the first programmable gate array, which is operatively coupled to the plurality of interface logics for communication via the memory port. In certain embodiments of the integrated circuit, the first memory port may function as a read port. The first memory port, when acting as a read port, may also be a multi-cycle port in some embodiments. Alternatively, the first memory port could serve as a write port in the integrated circuit. Similar to the read port configuration, the write port may also be designed as a multi-cycle port in certain embodiments.

[0017] In some configurations, the first memory port may be a combined read/write port. When functioning as a read/write port, it may also be configured as a multi-cycle port in some embodiments. The first programmable gate array within the integrated circuit could be a field programmable gate array in certain embodiments. Alternatively, the first programmable gate array may be a reconfigurable gate array. In some cases, this reconfigurable gate array

may be dynamically programmable. Alternatively, the reconfigurable gate array might be one-time programmable in an embodiment.

[0018] The first semiconductor device in the integrated circuit may include a network-on-chip fabric. Within the integrated circuit, the programmable gate array may include an array of embedded programmable gate array cores. These cores are configured to communicate with each other via the network-on-chip fabric. In some embodiments, the network-on-chip fabric is formed by the first programmable gate array. An embodiment of the first programmable gate array within the integrated circuit may include an array of embedded programmable gate array cores. Each core from this array may be operatively coupled with a respective interface logic from the plurality of interface logics. In the integrated circuit, the first programmable gate array may involve at least two embedded programmable gate array cores. Each core from the at least two is operatively coupled with a respective interface logic from the plurality of interface logics.

[0019] The first programmable gate array may include an array of embedded programmable gate array cores. Each core from this array is operatively coupled with at least one of the plurality of interface logics. Alternatively, each core from the array of embedded programmable gate array cores may be operatively coupled with a single one of the plurality of interface logics. The integrated circuit may further comprise a second semiconductor device. This device includes a plurality of memory modules, among which is a first memory module. This module is coupled to a second set of bonds on the surface of the second semiconductor device, with the first and second set of bonds configured to interface with each other when bonded together. In some embodiments, the first memory port is particularly a write memory port. The first memory port may be configured to write to all of the plurality of memory modules. The first semiconductor device may further include a plurality of read memory ports. The plurality of interface logics includes a number of read interface logics, with each read interface logic operatively coupled to a respective one of the read memory ports. Each read interface logic within the plurality can interface with a respective memory module from the plurality of memory modules.

[0020] The plurality of memory modules within the second semiconductor device may include an array of SRAMs. Alternatively, the memory modules might include an array of non-volatile memories. In some embodiments, the memory modules could be EEPROMs. Or, the memory modules might be ROMs. Optionally, the memory modules may include read-optimized non-volatile memory. In the integrated circuit, the programmable gate array may utilize antifuse, SRAM, or flash memory technology for its programming. The programmable

gate array cores might be connected in a mesh topology by the network-on-chip fabric within the integrated circuit. The interface logics of the integrated circuit may include voltage level shifters. These shifters are designed to convert signals of the programmable gate array to and from the voltage levels of the first memory port.

[0021] An embodiment may consist of a method for operating an integrated circuit, which involves the utilization of a first semiconductor device equipped with a first programmable gate array. This embodiment may include the employment of a number of interface logics, of which one is a first interface logic. It may also involve connecting a first memory port to a set of bonds located on the surface of the first semiconductor device. Additionally, this method may include the operative coupling of the first programmable gate array with the first interface logic, which enables communication through the first memory port. In some embodiments, the employment of the first interface logic may involve forming it from the first programmable gate array. In an embodiment, employing the first interface logic may include distributing this logic in a spaced pattern within the first programmable gate array. An embodiment may further include a plurality of memory ports, such as the first memory port, and may involve forming a spaced pattern on the surface of the first semiconductor device with these ports. In an embodiment, the method may further comprise the distribution of multiple sets of bonds, including the first set, in a spaced pattern on the surface of the first semiconductor device.

[0022] An embodiment of the method may utilize a number of memory ports, including the first memory port. Here, each interface logic from the plurality of interface logics may be coupled to a respective memory port from the multiple memory ports. In some embodiments, the method may include employing a number of programmable gate arrays, such as the first programmable gate array, wherein each gate array is coupled to a respective interface logic from the plurality of interface logics. An embodiment may involve coupling a number of cores, with each core from the plurality of cores being coupled to a respective interface logic from the plurality of interface logics. In another embodiment, the method may involve coupling a number of sets of bonds on the surface of the first semiconductor device. Here, each memory port from the plurality of memory ports may be coupled to a respective set of bonds from the multiple sets of bonds, including the first set.

[0023] In some embodiments, utilizing the first set of bonds may involve employing metallic pads. An embodiment of the method may involve employing a number of programmable gate arrays, including the first one, and configuring a network-on-chip to enable communication among at least two of the programmable gate arrays. In an embodiment, the

method may further comprise implementing at least one network-on-chip communication system to connect at least two processing elements on the first programmable gate array, and operatively coupling the first gate array to the multiple interface logics to enable communication through the memory port. In some embodiments, utilizing the first memory port may involve employing it as a read port. Wherein employing the first memory port as a read port, in some embodiments, may also include using it as a multi-cycle port.

[0024] An embodiment may involve utilizing the first memory port as a write port. In an embodiment, employing the first memory port as a write port may also include its use as a multi-cycle port. In certain embodiments, the method may involve employing the first memory port as a read/write port. Wherein employing the first memory port as a read/write port, in some embodiments, may also include using it as a multi-cycle port. In an embodiment, utilizing the first programmable gate array may include employing a field programmable gate array. An embodiment may involve utilizing the first programmable gate array as a reconfigurable gate array. Wherein employing a reconfigurable gate array, in some embodiments, may further include dynamically programming the gate array.

[0025] Alternatively, in an embodiment, employing a reconfigurable gate array may include one-time programming of the gate array. An embodiment of the method may further comprise including a network-on-chip fabric in the first semiconductor device. Wherein utilizing the network-on-chip fabric, in some embodiments, may include employing an array of embedded programmable gate array cores configured to communicate via the network-on-chip fabric. In an embodiment, forming the network-on-chip fabric may be accomplished by the first programmable gate array. An embodiment may include utilizing the first programmable gate array with an array of embedded programmable gate array cores, each operatively coupled with a respective interface logic from the plurality of interface logics. In some embodiments, utilizing the first programmable gate array may include employing at least two embedded programmable gate array cores, each operatively coupled with a respective interface logic from the plurality of interface logics. An embodiment may involve using an array of embedded programmable gate array cores, each operatively coupled with at least one of the multiple interface logics.

[0026] Alternatively, in an embodiment, utilizing the first programmable gate array may include employing an array of embedded programmable gate array cores, each operatively coupled with a single one of the multiple interface logics. An embodiment may further involve utilizing a second semiconductor device comprising a number of memory modules, including a first memory module, and coupling this module to a second set of bonds on the surface of the

second semiconductor device. The first and second sets of bonds may be configured to interface with each other when the first and second semiconductor devices are bonded together. In some embodiments, utilizing the first memory port may involve employing it as a write memory port. Wherein employing the first memory port as a write memory port, in an embodiment, may include configuring it to write to all of the plurality of memory modules. An embodiment may further include a number of read memory ports within the first semiconductor device, with each read interface logic from the plurality of read interface logics operatively coupled to a respective one of the multiple read memory ports. In some embodiments, each of the plurality of read interface logics may interface with a respective memory module from the plurality of memory modules.

[0027] An embodiment may involve employing the plurality of memory modules as an array of SRAMs. Alternatively, in an embodiment, the plurality of memory modules may be utilized as an array of non-volatile memories. In some embodiments, employing the plurality of memory modules may include using an EEPROM. In another embodiment, the plurality of memory modules may be employed as a ROM. An embodiment may involve using the plurality of memory modules as read-optimized non-volatile memory. In some embodiments, utilizing the programmable gate array may involve using antifuse, SRAM, or flash memory technology for programming. An embodiment may involve connecting the programmable gate array cores by employing a mesh topology through the network-on-chip fabric. In an embodiment, employing the interface logics may include using voltage level shifters to convert signals of the programmable gate array to and from voltage levels of the first memory port.

[0028] BRIEF DESCRIPTION OF THE DRAWINGS

[0029] These and other aspects will become more apparent from the following detailed description of the various embodiments of the present disclosure with reference to the drawings wherein:

[0030] Fig. 1 is a block diagram of an integrated circuit that may be part of a semiconductor device such as a chiplet in accordance with an embodiment of the present disclosure;

[0031] Fig. 2 shows a perspective view of an assembly having the integrated circuit of Fig. 1 implemented on a semiconductor device that is electrically connected to another device to form the assembly in accordance with an embodiment of the present disclosure;

[0032] Fig. 3A shows a perspective view of an assembly having the integrated circuit of Fig. 1 implemented on a semiconductor device that is electrically connected to another

device having a field programmable gate array fabric to form the assembly in accordance with an embodiment of the present disclosure;

[0033] Fig. 3B shows a perspective view of the assembly of Fig. 3A to illustrate the bonds of the surfaces of the semiconductor devices in accordance with an embodiment of the present disclosure;

[0034] Fig. 4. shows a perspective view of an assembly having the integrated circuit of Fig. 1 implemented on a semiconductor device that is electrically connected to another device having field programmable gate array cores interconnected by a network to form the assembly in accordance with an embodiment of the present disclosure;

[0035] Fig. 5 shows a block diagram illustrating the memory address space of the integrated circuit of Fig. 1 in accordance with an embodiment of the present disclosure;

[0036] Fig. 6 shows a block diagram illustrating the memory address space with the signal interfaces of the integrated circuit of Fig. 1 in accordance with an embodiment of the present disclosure;

[0037] Fig. 7 shows a diagram of two semiconductor devices, such as two chiplets, that automatically tests the connectivity of read interconnects within a semiconductor device and the integrated connectivity in accordance with an embodiment of the present disclosure; and

[0038] Fig. 8 shows a diagram of two semiconductor devices, such as two chiplets, that automatically tests the connectivity of write interconnects within a semiconductor device and the integrated connectivity in accordance with an embodiment of the present disclosure; and

[0039] Figs. 9A-9B show a block diagram of a system employing correct-by-construction timing closure for an application chiplet having read and write registers and a memory chiplet in accordance with an embodiment of the present disclosure.

[0040] DETAILED DESCRIPTION

[0041] Fig. 1 shows a block diagram of an integrated circuit 100 that may be packaged as a bondable chiplet (e.g., face-to-face chiplet bondable) in accordance with an embodiment of the present disclosure. The integrated circuit (IC) 100 includes a modules group 106 consisting of modules 108, 110, 112, and 114. The IC 100 also features a shared write port 102, configured to write to the modules group 106 using a write peripheral 104. Additionally, it includes read peripherals 116, 118, 120, and 122 and read ports 124, 126, 128, and 130, configured to read from the modules 108, 110, 112, 114.

[0042] The write port 102 may be configured to provide a single write address space for all of the modules group 106 where each of the modules 108, 110, 112, 114 has a dedicated read port 124, 126, 128, 130 respectively. The integrated circuit 100 may be packaged as part

of a chiplet configured to be electrically connected to another integrated circuit device (e.g., another chiplet, or IC package, with or without electrical contacts, electrical bumps, etc.). The chiplet may be electrically connected to another device including, for example, by bonding, soldering, wafer-to-wafer bonding, face-to-face chiplet bonding, chiplet-to-wafer bonding, chiplet-to-interposer bonding, and/or may be connected together with an interposer or other interfacing technology. None, one, or more of interposers may be used or other interfacing technologies that are common to heterogeneous 3D system-in-package solutions may be utilized in electrically connecting a chiplet to another device.

[0043] Each read port (124, 126, 128, 130) in the chiplet may feature electrical contacts on a side of the chiplet (e.g., top side or bottom side) or on multiple sides of the chiplet. The read ports 124, 126, 128, 130 may use multi-cycle pipelined circuitry. Upon bonding to another device (e.g., wafer, chiplet, chip, SOC, package, FPGA, etc.), the electrical contacts may line up in a manner that provides dedicated access to specific modules of the modules 108, 110, 112, 114. For instance, a processing/computing element may have exclusive access to module 108 via the read port 124, which may contain the neural network weights in a registry file. Similarly, a different processing/computing element may have exclusive read access to module 110 via the read port 126, which includes a different registry file. In this specific embodiment, this arrangement of the electrical contacts ensures that each computing/processing element has the dedicated access it needs to carry out its specific computation efficiently thereby providing a compact, modular, and scalable system that allows different processing elements to maintain dedicated access to specific modules 108, 110, 112, 114. Without dedicated access, different processing elements might have to queue up to use the same resource which would slow down overall processing speed. By providing dedicated access, the proposed chiplet ensures that each processing element can operate at its maximum capability without interference from other computing elements in this specific embodiment.

[0044] The write peripheral 104 is a peripheral circuitry responsible for processing and writing data into the memory cells found within the modules 108, 110, 112, 114. The write peripheral 104 may include dedicated contacts so that a chip electrically connected (e.g., bonded) to a chiplet of the integrated circuit, such that the write port 102 is accessible via a shared write logic system that involves utilizing a shift register-based, different voltage design, preferably high voltage design, that has a shared write address and data components. This shared write logic system is designed to be accessed via a bonded chip, another bonded chiplet, and/or via other circuitry in the same package as the integrated circuit 100. A shift register could allow the system to move data through a series of stages, with each subsequent stage

receiving the data from the previous stage. By utilizing a shift register, the system can increase the data throughput while maintaining a low rate of data transfers. The shared write address space refers to the location where data is written in the chiplet.

[0045] In another embodiment, an interlock 132 may disable the read ports 124, 126, 128, 130 while data is being written to the modules group 106 via the write port 102. Likewise, the interlock 132 may disable the write port 102 when read operations are being carried out on the read ports 124, 126, 128, 130. The written data can later be accessed concurrently by all processing elements that need to read the data via a respective one of the read ports 124, 126, 128, 130. This ensures that all processing elements have the most commonly used data available to them without regard to other reads being concurrently carried out by other processing elements.

[0046] The write peripheral 104 circuit includes a write driver. This unit receives the data to be written and converts it into suitable signals that can change the state of the memory cells. Depending on the type of memory technology used, these signals could involve voltage levels, current pulses, or other types of energy. The shared write logic system may be high voltage due to the specific voltage requirements of the chiplet. The write driver must provide enough power to reliably change the state of the memory cells, but it must also operate within suitable parameters to avoid causing damage or unnecessary wear.

[0047] The write peripheral 104 circuit may also feature a data buffer or write buffer. This component temporarily stores the data to be written, allowing the write operation to be performed at an predetermined pace. By balancing the speed of incoming data with the speed at which the memory cells can be written, the write buffer helps prevent data loss and optimizes system performance.

[0048] The write peripheral 104 may also include, in some embodiments, a write control unit that orchestrates the sequence of operations in the write process. It generates control signals to activate the write driver at the appropriate times, controls the flow of data from the write buffer, and coordinates the timing of the write operations. By synchronizing these various activities, the write control unit ensures efficient and reliable write operations.

[0049] The write peripheral 104 may also include data encoding mechanisms to improve reliability and data integrity. For example, before the data is written to the memory cells, these mechanisms encode it in a way that allows potential errors to be detected, and in some cases, corrected when the data is later read. This can be helpful in systems where data integrity has a higher priority, such as in servers or scientific research devices.

[0050] The write peripheral 104 may also include a timing unit that serves as the system's heartbeat, supplying clock signals that synchronize the operation of the system's various components. In some systems, it may include components like oscillators, clock generators, or phase-locked loops. The timing unit may ensure that all operations occur at the suitable time relative to each other.

[0051] The IC 100 may be implemented as a face-to-face bonded chiplet, with modules 108, 110, 112, and 114 formed from a non-volatile memory. In some specific embodiments, the IC 100 may also feature a dynamic allocation circuitry to allocate memory blocks to the modules group 106 based on the usage of the modules group 106 (e.g., each module 108 may include dynamic allocation circuitry for dynamically allocating a range of read locations for a respective processing element).

[0052] The IC 100 features a plurality of clocks, with each clock of the plurality of clocks feeding a respective module of the plurality of modules, providing each respective module with decoupled timing relative to the other modules of the plurality of modules. The modules group 106 may be arranged in any topology known to one of ordinary skill in the relevant art. Bit-cell density can be up to 10 times more dense than embedded SRAM cells in the modules group 106.

[0053] The IC 100 may be formed on a chiplet that includes a first side and a second side, with the second side configured for bonding to a second semiconductor device. The IC 100 may include a high voltage write logic adjacent to the first side of the chiplet. A decoder circuitry, a driver circuitry, and a register circuitry may be formed on the silicon substrate portion of the chiplet, while the modules group 106 is formed on a second layer portion of the chiplet. The second semiconductor device may comprise a plurality of processing elements. Each processing element includes a respective interface to communicate with a respective module of the plurality of modules on the modules group 106 when the second semiconductor device is bonded to the chiplet.

[0054] The silicon substrate traditionally serves as the initial stage of IC fabrication, focusing on the creation of active components, particularly transistors. Techniques like diffusion, ion implantation, oxidation, and material deposition are employed to fashion the intricate structures of transistors. These processes operate at small scales. The application of photolithography, etching, and implantation techniques enables the definition of transistor structures with precision. The silicon substrate's significance lies in its ability to establish the fundamental building blocks necessary for signal processing, amplification, and control within the IC. This layer is sometimes called Front-End-Of-The-Line ("FEOL").

[0055] Next in the manufacturing process, a second layer may be added that traditionally takes on the role of interconnect fabrication, facilitating the electrical connections between various IC components. The interconnects may be, but are not limited to, wires, conductive paths, waveguides, signal paths, logical paths, digital paths, buses, ports, etc. This phase traditionally focused on the creation of passive components, including interconnects, vias, and metal-insulator-metal (MIM) capacitors. The second layer processes typically differ from the processes used on the silicon substrate in terms of precision and scale. The interconnects are formed by depositing and patterning metal layers, typically aluminum or copper, to construct the wiring network. Dielectric layers, such as silicon dioxide or low-k dielectrics, are introduced to insulate the interconnects and prevent signal interference between different wiring layers. The second layer's traditional function is to establish the necessary interconnections that enable the routing and distribution of electrical signals throughout the IC. However, as described herein, circuitry may be utilized within this second layer (sometimes referred to as Back-End-Of-The-Line ("BEOL")).

[0056] Alternate embodiments of the IC 100 may be implemented as a stacked die and/or a monolithic design where electrical conductivity by TSVs and/or silicon through vias is used. In a stacked die design, several dies may be stacked on top of each other, with each die performing different functions, such as memory and processing. The stacked die may communicate through wire bonds, microbumps, or bump-less bonds. In a monolithic design, the various functions and modules of the IC 100 may be integrated onto a single die, forming a more compact and power-efficient design.

[0057] Additionally, the IC 100 may include one or more interlocks 132 to manage conflicts in reading and writing data. The modules group 106 may be formed from a variety of non-volatile or semi-volatile (e.g., very long refresh periods) memory technologies, such as Static Random-Access Memory (SRAM), Ferroelectric Field Effect Transistor (FeFET), Ferroelectric Random Access Memory (FeRAM), Resistive Random Access Memory (ReRAM), Spin-Orbit Torque (SOT) Memory, Spin Transfer Torque (STT) Memory, charge trap, floating gate memories, and/or Schottky diodes.

[0058] The modules group 106 may utilize a Static Random-Access Memory (SRAM) Topology. The SRAM topology may employ a cross-coupled flip-flop structure (e.g., latching flip-flops), ensuring the stored data remains intact as long as power is supplied. Thus, in some specific embodiments, the modules group 106 may utilize heterogeneous types of memory including volatile and non-volatile memory types.

[0059] The modules group 106 may utilize a Flash Memory Topology. The Flash memory is a non-volatile memory technology used in applications where data persistence is needed, such as solid-state drives (SSDs) and USB flash drives. The flash memory topology disclosed herein features a matrix of memory cells, each consisting of a floating-gate transistor or charge trap device. The modules group 106 may also use wear-leveling techniques to prolong the lifespan of the memory cells.

[0060] The modules group 106 may utilize a Ferroelectric Random-Access Memory (FeRAM) Topology. The FeRAM topology utilizes a ferroelectric material capable of retaining polarization states. One such memory topology may, in specific embodiments, utilize a FeFET to retain state information and program the ferroelectric material. These ferroelectric materials may be used to retain state information and act as a memory bit cell.

[0061] The modules group 106 may utilize a Phase Change Memory (PCM) Topology, which is a non-volatile memory technology that utilizes reversible phase changes in materials to store data. The PCM topology may include any phase change material, for example a chalcogenide alloy or a chalcogenide glass housed within a memory cell.

[0062] The modules group 106 may utilize a Resistive Random-Access Memory (ReRAM) Topology, which is a non-volatile memory technology based on resistive switching phenomena. The ReRAM topology may utilize a thin-film material that exhibits reversible changes in resistance upon the application of electrical stimuli.

[0063] The modules group 106 may utilize a Spin-Orbit Torque (SOT) Magnetic Random-Access Memory Topology. SOT-MRAM is a type of non-volatile memory that utilizes spin-orbit torque to switch the magnetic state of a storage element. The SOT-MRAM topology may incorporate a magnetic tunnel junction (MTJ) structure and leverages the spin-orbit coupling effect to write and read data. The magnetic tunnel junction may have a dielectric layer between a magnetic fixed layer and a magnetic free layer. Writing may be done by switching magnetization of the free magnetic layer by injecting an in-plane current in an adjacent SOT layer. Reading may be done by putting current into the magnetic tunnel junction. The SOT-MRAM can optimize the spin-orbit materials by using current-driven switching schemes while minimizing write energy consumption, in some specific embodiments.

[0064] The modules group 106 may utilize a Spin Transfer Torque (STT) Magnetic Random-Access Memory Topology. The STT-MRAM is another type of non-volatile memory that relies on spin transfer torque to manipulate the magnetic state of a storage element. The STT-MRAM topology can use a magnetic tunnel junction (MTJ) structure, where the magnetization orientation determines the stored data. Additionally, the orientation of a

magnetic layer in a magnetic tunnel junction or spin valve can be changed using a spin-polarized current, for example.

[0065] The IC 100 may include a single write peripheral 104 with a dedicated clock, or each module 108, 110, 112, 114 may have its own dedicated write peripheral utilizing a shared clock (not shown in Fig. 1). Additionally, the modules group 106 may be organized into separate partitions, each with a dedicated read peripheral 116, 118, 120, 122 having an independent clock.

[0066] Another possible embodiment of the IC 100 includes an interface (e.g., the same, different, higher or lower voltage) to enable data transfer external to the packaging of the IC 100. The IC 100 may also include an integrated microcontroller unit (MCU) or a digital signal processor (DSP) for processing data within the IC in yet additional specific embodiments.

[0067] Fig. 2 shows a perspective view an assembly 200 of the integrated circuit 212 of Fig. 1 implemented on a chiplet 230 that is bonded to a second device 226 in accordance with an embodiment of the present disclosure. The integrated circuit 212 is the circuitry within the chiplet 230. The second device 226 may be a chiplet, semiconductor wafer, semiconductor package, encased circuitry, etc. For example, the second device 226 may be an AI accelerator such that each processing unit has read access to one module (or a predetermined set) of the modules group 236. In yet another embodiment, the second device 226 may be a network controller where there is an offload circuit to read the data from each of the modules to processing incoming/outgoing packets, etc. The assembly 200 includes a modules group 236 having a plurality of modules, including a first module 232 and a second module 234. Fig. 2 shows several modules, however, for clarity, only modules 232, 234 have reference numbers. The integrated circuit 212 further comprises a shared write port 222. The shared write port 222 interfaces into the write peripheral 202.

[0068] Although the second device 226 may use the shared write port 222 via an address & data bus with a clock and a enable signal to write data to any modules within the modules group 236, other ways of writing data may be considered. For example, serial connections, parallel connections, various buses, or ports, may be used, such as a DDR (Double Data Rate) Interface, a SRAM (Static Random-Access Memory) Interface, a NAND Flash Memory Interface, a NOR Flash Memory Interface, a HBM (High Bandwidth Memory) Interface, a GDDR (Graphics Double Data Rate) Interface, a NVMe (Non-Volatile Memory Express) Interface, SPI, IC2, etc. Each of the modules has a read port with a read address 218

(to send an address to a module 234) and read data 214 (which is the data read from the module 232).

[0069] The modules group 236 is formed on a chiplet 230 having two sides including a surface 228 that can be bonded to and complement a second device 226. The chiplet 230 may be formed by forming circuitry on a silicon substrate 204 and then by adding a second layer 206. In other embodiments, these layers may be reversed and/or other layers may be added, removed, etc. The read address 218 and read data 220 are used for reading the module 232.

[0070] Although the second device 226 may use an address & data bus with a clock and a enable signal to read data from the module 232, other ways of reading data may be considered. For example, serial connections, parallel connections, various buses, or ports, may be used, such as a DDR (Double Data Rate) Interface, a SRAM (Static Random-Access Memory) Interface, a NAND Flash Memory Interface, a NOR Flash Memory Interface, a HBM (High Bandwidth Memory) Interface, a GDDR (Graphics Double Data Rate) Interface, a NVMe (Non-Volatile Memory Express) Interface, SPI, IC2, etc.

[0071] All of the read ports (e.g., 218 and 222) are configured to be inactive when a write operation is applied to the shared write port 222. The read ports may also be configured to process reads concurrently with each other. The shared write port 222 is configured to write to an address space, where the shared write port 222 is configured to write to the first module 232 via a first portion of the address space and write to the second module 234 via a second portion of the address space. Each module of the plurality of modules 236 includes an independent read port for concurrent reading via a respective independent read port of any of the plurality of modules.

[0072] Each read port for a respective module may include contacts for circuitry found within the second device 226 to interface via metallic contacts. Thus, there may be metallic contacts on the top layer 208 that are configured to interface with metallic contacts on the surface 228 of the chiplet 230 such that the metallic contacts allow for a read space that is coextensive with a read space of a module of the modules group 236. The read spaces of the modules group 236 may all be coextensive with each other (as is described with reference to Figs. 3 and 4).

[0073] In one embodiment, the read peripheral for the first module 232 is implemented on a silicon substrate 204 (sometimes referred to as a Front-end-of-the-line). The second layer 206 (sometimes called the Back-end-of-the-line) may be built next in the manufacturing process on top of the silicon substrate 204 (and any circuitry) and may contain the respective

memory bit cells. In an alternative embodiment, the read peripheral for the first module 232 is implemented in the second layer 206 and is disposed between the modules group 236 and the surface 228 of the chiplet 230. In some embodiments, there is a read peripheral for each of the modules of the modules group 236.

[0074] The modules group 236 may be configured to process write commands only during reset. The write commands may be “slow write” commands. That is, the modules group 236 may have very low write speeds relative to its read speed. The write logic may be frozen (or disabled) when the modules group 236 are used for reading data. In some specific embodiments, the integrated circuit 212 provides functionality to allocate memory blocks to the modules group 236 based on the usage of the modules group 236. In other embodiments, the memory addresses are fixed along with the allocation. The integrated circuit 212 may be implemented as a face-to-face bonded chiplet 230. The face-to face bonding may be bump-less wafer bonding.

[0075] The modules group 236 can have a single write peripheral 202. In other embodiments, each module of the modules group 236 may have a dedicated write peripheral that utilizes a shared clock. In yet other embodiments, the modules group 236 may also be organized into separate partitions each with partition having a dedicated read peripheral, where each dedicated read peripheral has an independent clock. The partitions may be one, two, or more modules of the modules group 236.

[0076] The write peripheral 202 circuitry's overall architecture may include a series of different components, including write driver, address decoders, sense amplifiers, data input latches, data bus, etc. and/or some combination thereof. Write drivers or write buffers, may be tasked with transferring data onto the memory cell. They may enhance the input signal to achieve a level appropriate for the memory cell. Address decoders may be used to interpret the memory address that is fed as an input where the data needs to be written. By activating the specific row and column of the memory array linked to that address, they may be used to select the target memory cell. Sense amplifiers may be used to identify and boost the signal from the memory cells during reading operations, also participate in refreshing the memory cell post data write in write operations. The write operation is instigated by a write enable signal. When a write command is initiated, this signal propels the write drivers and decoders into the writing process. Data input latches may be used as temporary storage units, retaining the data set to be written into the memory until the write operation is implemented. A data bus with a transmission route, can be used to facilitate the movement of data from the data input latches to the memory cells.

[0077] A write operation to the modules group may be performed through a priority arbitration circuit that facilitates the modules to be accessed in a predetermined order, and the shared write port 222 may be configured to write to a virtual address space that is mapped onto a physical memory space. The integrated circuit 212 may include a high voltage write logic used within the write peripheral 202, and the second semiconductor device 226 may comprise a plurality of processing elements, whereby each processing element includes a respective interface to communicate with a respective module of the modules group 236. Furthermore, the chiplet 230 may include an interface to the shared write port 222 on the second side to thereby interface with a complementary interface on the second semiconductor device 226.

[0078] The integrated circuit 212 may also include a power gating circuitry that selectively powers down a module of the modules 236 when not in use. Additionally, the integrated circuit 212 may have a write peripheral 202 of the modules group 236 connected to a dedicated I/O pad to enable data transfer external to the package of the integrated circuit.

[0079] The integrated circuit 212 may utilize multiple modules of the modules group 234 grouped together. These modules may be synchronized with one another in specific embodiments. In some cases, all the modules are synchronized, while in other instances, only specific modules are to be synchronized. For instance, the circuit on a second device 226 synchronize with a specific module when reading data from one of the modules in the module group 236.

[0080] To synchronize the modules, the integrated circuit 212 may use various timing technologies. In some cases, a plurality of clocks may feed each respective module of the modules group 236, thereby allowing each module to have decoupled timing relative to the other modules in the group. This decoupling ensures that any delay in one module will not affect the functioning of other modules. It is worth noting that the clocks used may or may not need to be synchronized. In some cases, a common clock can be used to synchronize the modules. In yet other embodiments, the clock signal or signals may be provided by the second device 226.

[0081] In alternative embodiments, other synchronization techniques can be used, such as phase comparison of the clock signals or a phase-locked loop (PLL) synchronization method. Another embodiment for synchronizing the modules in the IC could use delay-locked loop (DLL) synchronization. In this method, a delay element is added to the clock signal path, and the output is compared to the input clock signal. The feedback loop adjusts the delay element until the output of the DLL matches the input, resulting in synchronization of the clock signals.

[0082] In another embodiment, the integrated circuit 212 could use a combination of different synchronization techniques to achieve synchronization between the modules. For example, some modules may use PLL synchronization while others use clock delay lines or DLL synchronization, depending on their specific requirements. Additionally, the integrated circuit 212 can also use redundant synchronization techniques to ensure reliability and redundancy in case one method fails. For example, the integrated circuit 212 could use both PLL synchronization and DLL synchronization simultaneously, so that if one method fails, the other can still maintain synchronization.

[0083] Fig. 3A shows a perspective view of an assembly 240 having the integrated circuit of Fig. 1 implemented on a memory-module semiconductor device 243 that is electrically connected to FPGA semiconductor device 245 having a field programmable gate array fabric 246 to form the assembly 240 in accordance with an embodiment of the present disclosure. The memory-module semiconductor device 243 has a plurality of memory modules 253 that can be written to by a shared write peripheral 241. The memory modules 253 may be in a spaced pattern (e.g., a grid) in some embodiments. Each of the memory modules 253 includes a read peripheral 242 also on the memory-module semiconductor device 243. A 3D memory structure 244 may be used to form the memory modules 253. Write data can be received via a shared write port 251 that is coupled to write interface logic 252 of the FPGA semiconductor device 245. The FPGA semiconductor device 245 has a programmable gate array fabric 246, that may be an FPGA. The FPGA semiconductor device 245 includes interface logic that may include a read address interface logic 249 and a read data interface logic 250. A read data port 248 can communicate data from the memory-module semiconductor device 243 to the FPGA semiconductor device 245. A read address port 247 can communicate a read address from the FPGA semiconductor device 245 to the memory-module semiconductor device 243.

[0084] The assembly 240 includes the memory-module semiconductor device 243 bonded to the FPGA semiconductor device 245. The memory-module semiconductor device 243 contains a plurality of memory modules 253 that can store data. The memory modules 253 are formed using a 3D memory structure 244 to enable high density data storage.

[0085] The memory modules 253 can be accessed independently via read peripherals 242 located within the memory-module semiconductor device 243. Each read peripheral 242 is connected to a respective memory module 253. For example, read peripheral A is connected to memory module A, read peripheral B is connected to memory module B, and so on. This independent connectivity allows different parts of the FPGA semiconductor device 245 to

concurrently access different memory modules 253 via the separate read ports 247, 248 without interference. The read peripherals 242 may be positioned in a spaced pattern (e.g., a grid).

[0086] The memory-module semiconductor device 243 contains a plurality of memory modules 253 that are organized into a 3D memory structure 244 which may be arranged to maximize memory density in some embodiments. The memory modules 253 can store data that is accessed by different processing elements within the FPGA semiconductor device 245.

[0087] The read peripherals 242 provide access to the memory modules 253 via read address ports 247 and read data ports 248. The read address ports 247, which each interface with a read peripheral 242, receive read addresses specified by processing elements in the FPGA semiconductor device 245. These addresses are used by the read peripherals 242 to query the appropriate location within the corresponding memory module 253. The read data ports 248 then transmit the retrieved data from the memory modules 253 to interface logic in the FPGA semiconductor device 245.

[0088] All read operations via the read address ports 247 and read data ports 248 may occur independently without interfering with each other. This ensures that multiple processing elements can concurrently read data from different memory modules 253 without delays. The independent read access also allows processing elements (in some embodiments) to efficiently access required data stored in a dedicated memory module 253 without queuing or arbitration between read requests.

[0089] A shared write peripheral 241 facilitates writing to the memory modules 253. The shared write peripheral 241 receives a shared write address and write data via a shared write port 251. Using the shared write address, the shared write peripheral 241 writes the received data to the specified memory location within any memory module 253. During write operations, the read ports 247/248 may be disabled via an interlock (not shown) to avoid contention with reads. This shared write access allows efficient single-cycle writing of updated data to the memory modules 253.

[0090] The FPGA semiconductor device 245 contains a programmable gate array fabric 246, which may also include various processing elements. An interface logic that may include read address interface logic 249 and read data interface logic 250 may connect logic and/or processing elements to the memory interconnects. The read address interface logic 249 interfaces with the read address ports 247, and the read data interface logic 250 interfaces with the read data ports 248. A shared write port 251 connects to write interface logic 252, which interfaces with the shared write peripheral 241 via the shared write port 251 to facilitate write access to the memory modules 253.

[0091] The FPGA semiconductor device 245 includes a field programmable gate array fabric 246 that provides programmable logic and interconnect resources. The field programmable gate array fabric 246 contains an array of configurable logic blocks (CLBs) arranged in a two-dimensional grid layout. Each CLB includes programmable lookup tables (LUTs), flip-flops, and programmable interconnects to implement user logic functions and connections between functions.

[0092] The field programmable gate array fabric 246 may also include a hierarchical programmable interconnect structure with multiple types of interconnect wires, including short wires connecting neighboring CLBs, longer wires connecting more distant CLBs, and still longer wires spanning across the entire FPGA semiconductor device 245. The various interconnect wires may be programmably connected at programmable switch boxes to provide configurable routing of signals between the different logic blocks.

[0093] The interconnect structure within the field programmable gate array fabric 246 may utilize passive routing switches, such as antifuses or SRAM, to programmably connect the different logic and interconnect resources. The passive routing switches can be configured to implement a user design by loading a configuration bitstream that sets the states of the switches and programs the functions implemented by the LUTs and flip-flops.

[0094] In addition to the programmable logic blocks and interconnect, the FPGA semiconductor device 245 may include configurable input/output blocks (IOBs) arranged around the peripheral of the field programmable gate array fabric 246. The IOBs include programmable power buffers, registers, and high-speed serial transceivers to interface between the programmable fabric and external package pins.

[0095] The FPGA semiconductor device 245 may also include dedicated memory interface circuits integrated into some regions of the IOBs or dispersed programmably throughout the CLBs. These memory interface circuits provide ports compatible with common memory standards to interface with off-chip memory or between on-chip memory interfaces of other devices in the assembly. In this specific embodiment, the FPGA semiconductor device 245 may include a shared write peripheral 241 integrated into one region. The shared write peripheral 241 may contain circuitry to implement the protocols required to write data via the shared write port 251 to any of the memory modules 253 within the memory-module semiconductor device 243. The shared write peripheral 241 may utilize the programmable resources of the field programmable gate array fabric 246 and is connected to the shared write port 251 through dedicated I/O pins or routed programmably throughout the FPGA.

[0096] The FPGA semiconductor device 245 also includes interface logic in the form of the read address interface logic 249 and read data interface logic 250. These interface logics may be dispersed programmably throughout the CLBs of the FPGA to interface between the programmable fabric and the read address port 247 and read data port 248, respectively. The interface logics may contain registers, buffers, and other ancillary circuits to meet the timing requirements for high-speed memory read transfers.

[0097] The field programmable gate array fabric 246, including the CLBs, IOBs, and programmable interconnect, can in some embodiments be programmed and reprogrammed in-system multiple times using a configuration bitstream loaded through an external active serial or parallel configuration interface. This facilitates modifying the user design implemented in the FPGA semiconductor device 245 over the lifetime of the system.

[0098] The field programmable gate array fabric 246 is located on the FPGA semiconductor device 245. The field programmable gate array fabric 246 may provide programmable logic and routing resources. The field programmable gate array fabric 246 can include an array of programmable logic blocks that can be configured to perform basic logic functions, such as AND, OR, NOT, NAND, NOR, and XOR. The programmable logic blocks may be interconnected via a configurable routing structure that allows for customizable logic signals to be routed between the blocks. This programmable interconnection of logic and routing resources can allow for different combinational or sequential functions to be implemented within the field programmable gate array fabric 246 through a process known as hardware description language programming.

[0099] The field programmable gate array fabric 246 may utilize non-volatile memory elements, such as static RAM (SRAM) cells, antifuses, or flash memory, to store the configuration data for the programmable logic blocks and switches. Each memory cell corresponds to a programmable resource within the field programmable gate array fabric 246, such as a lookup table, routing switch, or logic gate. The state of each memory cell determines whether the associated logic block or interconnect is active and its specific configuration.

[00100] The field programmable gate array fabric 246 may include interface logic that provides connections to external circuitry integrated on the FPGA semiconductor device 245. In particular, the field programmable gate array fabric 246 may include a read address interface logic 249 coupled to the read address port 247 and a read data interface logic 250 coupled to the read data port 248. These interface logics facilitate communication between the programmable logic and routing resources of the field programmable gate array fabric 246 and

other components located on the FPGA semiconductor device 245, such as memory modules on a bonded memory semiconductor device.

[00101] In summary, the field programmable gate array fabric 246 provides the configurable logic and routing functions for the FPGA semiconductor device 245. The field programmable gate array fabric 246 can be programmed to implement various digital circuits through loading of configuration data, allowing flexible and reconfigurable utilization in various system applications and designs.

[00102] The interface logic located within the FPGA semiconductor device 245 includes a read address interface logic 249 and a read data interface logic 250. The read address interface logic 249 and read data interface logic 250 facilitate communication between the programmable gate array fabric 246 and the memory modules 253 on the memory-module semiconductor device 243.

[00103] The read address interface logic 249 may contain circuitry for transmitting read address signals from the programmable gate array fabric 246 to the memory-module semiconductor device 243. The read address interface logic 249 may include a plurality of read address interconnect lines that form a path for read address signals from the gate array fabric 246 to exit the FPGA semiconductor device 245. These read address interconnect lines are routed to terminate at a set of read address bonds located on the surface of the FPGA semiconductor device 245. The read addresses may be raw addresses or may be transformed in some embodiments by the read address interface logic 249.

[00104] The read address bonds are configured to electrically interconnect with a complementary set of read address bonds on the memory-module semiconductor device 243 when the two devices are bonded together. With this electrical coupling, read address signals output by the read address interface logic 249 can propagate from the FPGA semiconductor device 245 to the memory-module semiconductor device 243 to access specific memory locations.

[00105] The read address interface logic 249 may also include read address register circuitry for registering read address signals from the programmable gate array fabric 246 before transmitting them onto the read address interconnect lines and bonds. This read address register may act as an interface between the timing domains of the gate array fabric 246 and the memory-module semiconductor device 243. The read address register may sample read address signals on a register clock edge and then launch the sampled read address onto the interconnect lines based on timing constraints to ensure proper setup and hold times are met.

[00106] The read address interface logic 249 may, in some specific embodiments, include address decoder circuitry for decoding portion of the transmitted read address signals on the memory-module semiconductor device 243 side to selectively access the appropriate memory module 253. The address decoder circuitry translates a portion of the transmitted read address into decode signals that enable specific rows, columns or other access lines of a given memory module 253. In other embodiments, the read address interface logic 249 passes the address to the semiconductor device 243 without decoding.

[00107] The read data interface logic 250 may contain circuitry for receiving read data returned from the memory modules 253 on the memory-module semiconductor device 243. The read data interface logic 250 can comprise read data interconnect lines forming a path for received read data to enter the FPGA semiconductor device 245 from the memory-module semiconductor device 243. These read data interconnect lines terminate at a set of read data bonds on the FPGA semiconductor device surface.

[00108] Similar to the read address bonds, the read data bonds may be configured to electrically couple with a complementary set of read data bonds on the memory-module semiconductor device 243. Through this coupling, read data signals can propagate from the accessed memory location on a memory module 253, onto the memory-module semiconductor device 243, and then across to the FPGA semiconductor device 245 via the read data interconnect paths and bonds.

[00109] The read data interface logic 250 may include read data register circuitry for registering received read data signals before transmitting them onto internal read data interconnect lines coupling to the programmable gate array fabric 246. The read data register may act as an interfacing element between timing domains and may latch read data signals using a register clock edge and then launch the latched read data signals based on defined setup/hold constraints.

[00110] The read data interface logic 250 may contain latch or decoder circuitry on the FPGA semiconductor device 245 side for accessing the read data signals in a format suitable for the gate array fabric 246. The read data interface logic 250 coordinates read data transfer between the memory-module semiconductor device 243 and the internal resources of the programmable gate array fabric 246.

[00111] Together, the read address interface logic 249 and read data interface logic 250 may facilitate the programmable gate array fabric 246 to interface with the memory modules 253 on the memory-module semiconductor device 243 via a suitable timing sequence. The interface logics 249, 250 can facilitate the transmission and reception of address, data and

control signals across the bonded semiconductor devices in a manner that is correct-by-construction to meet all timing constraints.

[00112] In some embodiments, the interface logics 249, 250 may also include test circuitry such as boundary scan cells or logic for built-in self-test of the interconnect paths and bonding between the bonded semiconductor devices, as mentioned herein. The boundary scan cells can be connected in daisy-chain configuration to test interconnects prior to semiconductor device bonding. Overall, the read address interface logic 249 and read data interface logic 250 may provide an interface between the programmable logic resources of the FPGA and memory on the memory chiplet.

[00113] As previously mentioned, the read address interface logic 249 is part of the interface logic that facilitates communication between the FPGA semiconductor device 245 and the memory-module semiconductor device 243. The read address interface logic 249 includes circuitry for transmitting read address signals from the FPGA semiconductor device 245 to the memory-module semiconductor device 243 to select specific memory locations during memory read operations.

[00114] The read address interface logic 249 resides within the integrated circuit formed by the face-to-face bonding between the FPGA semiconductor device 245 and the memory-module semiconductor device 243. The read address interface logic 249 is positioned between the field programmable gate array fabric 246 and the read address port 247. This allows the read address interface logic 249 to receive read address signals from the field programmable gate array fabric 246 and transmit the corresponding read address signals to the memory-module semiconductor device 243 through the read address port 247.

[00115] The read address interface logic 249 includes input/output interfaces to operatively couple to the field programmable gate array fabric 246 and the read address port 247. The interface to the field programmable gate array fabric 246 receives read address signals originating from the programmable logic array cores or other processing elements within the FPGA semiconductor device 245. The interface to the read address port 247 transmits the appropriate read address signals for receipt by the memory-module semiconductor device 243.

[00116] In addition to signal interfaces and as previously mentioned, the read address interface logic 249 may include register circuits, decoding logic, and transmission driver circuitry. The register circuits latch read address signals from the field programmable gate array fabric 246 in preparation for transmission. The decoding logic decodes address information and selects the appropriate read address line or lines. The transmission driver circuitry

amplifies the read address signals to the voltage levels required for reliable transmission through the read address port 247.

[00117] The read address interface logic 249 may contain circuitry to support multiple embodiments. In one embodiment, the read address interface logic 249 supports a single read address line to address individual memory locations. In another embodiment, the read address interface logic 249 supports multiple parallel read address lines to allow higher memory bandwidth through wider addressing. The read address interface logic 249 may also incorporate error detection or correction mechanisms to promote reliable transmission of address signals.

[00118] The read address signals transmitted by the read address interface logic 249 are received by the memory-module semiconductor device 243 through the read address port 247, e.g., via an array of metallic contacts on the surface of the memory-module semiconductor device 243. These metallic contacts align with corresponding contacts on the bonded surface of the FPGA semiconductor device 245, enabling direct electrical connections when the two semiconductor devices are face-to-face bonded. Once received, the read address signals are used by the read peripherals within the memory-module semiconductor device 243 to access specific memory locations based on the given read address.

[00119] In this way, the read address interface logic 249 allows programming logic and processing elements within the FPGA semiconductor device 245 to indirectly access contents of the 3D memory structure 244 through transmitting the appropriate read address signals to select targeted memory locations, providing access to the integrated memory. The read address interface logic 249 provides an interface facilitating reliable read operations between the logically-programmable FPGA semiconductor device 245 and the large-capacity, directly-addressable memory enabled by the 3D stacked memory structure 244.

[00120] Also as previously mentioned, the read data interface logic 250 is responsible for transferring read data from the memory modules 253 to the FPGA semiconductor device 245. The read data interface logic 250 may include read data registers that are used to capture and buffer read data received from the memory modules 253. These read data registers may provide temporary storage of the read data, allowing time for the read data to be driven onto the read data interconnects while also accounting for any timing delays in the read data propagation. The read data registers may be configured to provide data integrity and prevent data loss during the read operation.

[00121] The read data interface logic 250 may also include read data interface circuits connected to the read data registers. These read data interface circuits drive the read data buffered in the registers onto the read data interconnects 218. The driver strength of the read

data interface circuits can be set to safely transfer the read data signals considering factors such as interconnect length and load. In some embodiments, the read data interface circuits may provide additional amplification or buffering of the read data signals.

[00122] The read data interconnects 218 form part of the read data interface logic 250 and may be used to transmit the read data from the registers to the bonds 248 on the surface of the memory-module semiconductor device 243. These read data interconnects may extend from the location of the read data registers to the bonds. In some embodiments, the read data interconnects may utilize low-resistance metal such as copper to minimize signal delay.

[00123] The bonds may be part of the read data interface logic 250 that provide electrical connections of the read data interconnects to the surface of the memory-module semiconductor device 243. These bonds allow transmission of the read data signals to the FPGA semiconductor device 245 once bonding brings the devices 243, 245 into electrical communication. In some embodiments, the bonds 248 may be metallic solder balls, copper pillars, metallic pads, etc.

[00124] Together, the read data registers, read data interface circuits, read data interconnects, and bonds of the read data interface logic 250 may function to transfer read data from the memory modules 253, capture and buffer the read data signals, drive the read data onto the interconnects, and transmit the read data signals to the bonded FPGA semiconductor device 245. The read data interface logic 250 thus enables reliable transfer of read data from the memory modules 253 for use by the FPGA cores.

[00125] In some embodiments, the read data interface logic 250 may include additional components like data encryption/decryption circuitry or error detection and correction circuitry to ensure security and reliability of the read data transfer between the memory modules 253 and FPGA semiconductor device 245. The read data interface logic 250 provides an interface between the memory device and programmable logic that seamlessly transfers read data while meeting timing and signal integrity constraints.

[00126] The write interface logic 252 provides an interface between the FPGA semiconductor device 245 and the shared write port 251 to facilitate the transmission of write address, write data, and write control signals from the FPGA semiconductor device 245 to the shared write port 251.

[00127] The write interface logic 252 may include a write address register that receives a write address from address generation circuitry within the programmable gate array fabric 246 of the FPGA semiconductor device 245. The write address register holds the write address to be transmitted to the shared write port 251. The write interface logic 252 may also include

write address interconnect wiring that transmits the write address from the output of the write address register to the plurality of write address bonds 750 on the surface of the FPGA semiconductor device 245. These write address bonds 750 provide an interface between the write address interconnect wiring and the shared write port 251 when the FPGA semiconductor device 245 is bonded to the memory-module semiconductor device 243.

[00128] In addition to the write address register and interconnect wiring, the write interface logic 252 includes a write data register that receives write data from circuits within the programmable gate array fabric 246. For example, the write data may include weights, instructions, or other data to be written to the memory modules 253. The write data register holds the write data to be transmitted to the shared write port 251. The write interface logic 252 further includes write data interconnect wiring coupled to the output of the write data register. This write data interconnect wiring transmits the write data to a plurality of write data bonds on the surface of the FPGA semiconductor device 245, which interface with the shared write port 251 when bonding occurs.

[00129] The write interface logic 252 additionally includes a write control register that receives write control signals, such as a write enable signal, from control circuits within the programmable gate array fabric 246. For example, the write control signals may include a write enable signal that initiates the write operation when activated. The write control register holds the write control signals to be transmitted to the shared write port 251. The write interface logic 252 also includes write control signal interconnect wiring coupled to the output of the write control register. This write control signal interconnect wiring transmits the write control signals, such as a write enable signal, to a write control signal bond 754 on the surface of the FPGA semiconductor device 245, which interfaces with the shared write port 251.

[00130] The write interface logic 252 further includes a write clock pad that receives a write clock signal. This write clock pad may be coupled to write clock interconnect wiring that distributes the write clock signal to the write address register, write data register, and write control register within the write interface logic 252. The write clock signal synchronizes the operation of these registers and timing of the write operation. The write interface logic 252 may optionally include frequency division circuitry to generate a divided write clock signal to be transmitted to the shared write port 251 via a write clock bond 756 on the surface of the FPGA semiconductor device 245.

[00131] The write interface logic 252 may provide write timing control signals via a timing controller circuit within the write interface logic 252. These timing control signals are transmitted via timing control signal bonds on the surface of the FPGA semiconductor device

245 to interface with the shared write port 251. The timing control signals synchronize operation of the write address generation, data transmission, and other timing-critical circuits associated with the write operation.

[00132] In an alternative embodiment, the write interface logic 252 may include level shifter circuitry to adjust the voltage levels of the write signals to match the voltage requirements and specifications of the shared write port 251. For example, the shared write logic system may employ a high voltage design for writing reliability and endurance. The level shifter circuitry helps ensure compatibility of the write interface logic 252 and shared write port 251.

[00133] The write address bonds 750, write data bonds 752, write control signal bond 754, write clock bond 756, and timing control bonds 758 formed on the surface of the FPGA semiconductor device 245 are configured to physically, electrically, and signal interface with the shared write port 251 when the FPGA semiconductor device 245 is bonded to the memory-module semiconductor device 243, thereby enabling transmission of write address, write data, and control signals from the FPGA semiconductor device 245 to the shared write port 251 and memory modules 253 during a write operation.

[00134] In summary, the write interface logic 252 provides an interface between the programmable gate array fabric 246 of the FPGA semiconductor device 245 and the shared write port 251 to transmit write address, write data, and control signals with appropriate timing during a write operation to the memory modules 253, thereby facilitating writing of data to the memory modules 253 from circuits within the programmable gate array fabric 246. The write interface logic 252 may handle signal transmission, timing control, voltage level translation, and physical interfacing between the two semiconductor devices 245, 243 during writes to the memory modules 253.

[00135] The memory-module semiconductor device 243 includes a 3D memory structure 244 that forms a plurality of memory modules 253. The 3D memory structure 244 utilizes a vertical stacking of memory cell layers separated by inter-layer dielectric layers to significantly increase memory density. This vertical integration enables the formation of numerous memory cells within a limited chip surface area.

[00136] Each memory module 253 may include an array of non-volatile memory cells that can retain data when power is turned off. The memory cells may utilize flash, resistive random-access, spin-torque transfer, phase-change memory technologies, etc.

[00137] The memory modules 253 may be independently accessible via read peripherals 242 included within the memory-module semiconductor device 243. This independent

connectivity allows different parts of the FPGA semiconductor device 245 to concurrently access different memory modules 253 via separate read address ports 247 and read data ports 248 without interference.

[00138] Each read peripheral 242 may include circuitry to interface with its corresponding memory module 253. This may include row and column address decoders to select memory cells based on supplied addresses, sense amplifiers to detect and amplify signal levels during read operations, I/O circuits to transmit data to and from the memory module 253, and control logic to orchestrate read operations in response to control signals.

[00139] The memory modules 253 can be written to via a shared write peripheral 241. The shared write peripheral 241 includes circuitry such as a write address decoder, write drivers, data buffers, and control logic to perform write operations to any memory location within the plurality of memory modules 253.

[00140] When power is off or the memory-module semiconductor device 243 is not being written to, the memory modules 253 can maintain their data retention for an extended period without the need for power, in some specific embodiments. This non-volatility enables data to persist even when electrical power is removed, allowing the memory to function as permanent or semi-permanent storage, in some embodiments.

[00141] The independent read ports provided by each read peripheral 242, combined with the high density enabled by 3D integration, allows the memory-module semiconductor device 243 to provide concurrent accesses to independent memory spaces by different processing elements or logic within the FPGA semiconductor device 245. This may provide efficient data access for various memory-intensive applications, such as those needed by AI accelerators.

[00142] Each memory module 253 may be formed on the memory-module semiconductor device 243 in a second layer above a silicon substrate. The memory modules 253 may be arranged in an array structure within the 3D memory structure 244 utilizing vertically-stacked memory cells. The memory cells within each memory module 253 may utilize a variety of memory technologies including SRAM, DRAM, flash memory, resistive RAM (ReRAM), magnetoresistive RAM (MRAM), phase-change memory (PCM), etc.

[00143] The memory cells may be arranged in a matrix of rows and columns with word lines and bit lines used to access individual memory cells based on a supplied memory address. Sense amplifiers may be used to detect and amplify signals from the bit lines during read operations. Voltage generation circuitry supplies the necessary voltages for programming, erasing, or reading memory cells based on the memory technology employed. Additional

circuitry such as address decoders, input/output circuits and control logic are used to orchestrate memory operations in response to supplied control signals and commands.

[00144] Error correction circuitry such as ECC (Error Correction Code) encoding/decoding blocks may be included within each memory module 253 or implemented at a higher level to safeguard data integrity. Circuit techniques like wear leveling and bad block management may be used to increase memory endurance over many program/erase cycles. On-chip voltage regulators may be used to ensure a stable power supply for reliable memory operations. A finite state machine or control processor manages the overall sequencing of memory operations, in some specific embodiments.

[00145] Each memory module 253 has a dedicated read peripheral 242 which includes registering and buffering circuitry associated with read operations from that memory module 253. For example, read peripheral 242A is dedicated to memory module 253A. The read peripheral 242 is responsible for receiving and latching the read address from the associated read address interconnect 247. It then decodes and activates the appropriate word and bit lines to access the addressed memory location. Data is sensed and amplified then latched and buffered in the read peripheral 242 before being output on the associated read data interconnect 248.

[00146] The read peripheral 242 may include control circuitry such as address decoders, data sense amplifiers, input/output buffers, and latches to reliably interface between the associated memory module 253 and the shared read data interconnects 248. It may operate based on control signals from a clock supplied by the read clock interconnect 249.

[00147] In this way, each memory module 253 can be independently and concurrently accessed through a dedicated read peripheral 242 for maximum memory throughput when servicing multiple read requests from different processing elements in the FPGA 245. The read peripheral 242 may provide buffering and control to interface between the potentially higher-speed FPGA 245 and lower-speed memory technologies in memory modules 253, in some specific embodiments.

[00148] All the memory modules 253 may share the same write peripheral 241 to receive write addresses and data via the shared write port 251 under control of write interface logic 252. The write peripheral 241 may include addressing circuits, data input/output circuits, and control logic to program memory cells in any memory module 253 across a common write address space based on the address values supplied.

[00149] In summary, each memory module 253 within the 3D memory structure 244 provides independent data storage that can be randomly accessed through the dedicated read

peripheral 242. Concurrent and independent read access to different memory modules 253 maximizes memory bandwidth when servicing multiple read requests, while a shared write peripheral 241 handles memory programming efficiently via the shared write port 251.

[00150] The shared write port 251 is configured to write data to the plurality of memory modules 253 via the shared write peripheral 241 located on the memory-module semiconductor device 243. The shared write peripheral 241 receives write data, a write address, and a write clock signal from the FPGA semiconductor device 245. Specifically, write data is received from the write interface logic 252 of the FPGA semiconductor device 245 via the shared write port 251, which is coupled to the shared write peripheral 241.

[00151] The shared write peripheral 241 includes various components and circuitry to perform write operations to the memory modules 253. It may include a write driver that receives write data from the shared write port 251 and converts it into suitable signals, such as voltage levels or current pulses, that can change the state of memory cells within the memory modules 253. The shared write peripheral 241 may also include a write buffer or data buffer that temporarily stores write data received from the shared write port 251. This allows write operations to the memory modules 253 to be performed at a controlled pace that balances the speed of incoming write data with the write speed capabilities of the memory modules 253.

[00152] The shared write peripheral 241 may further includes a write control unit that orchestrates the sequence of write operations. It may generate control signals to activate the write driver at appropriate times, controls the flow of write data from the write buffer, and coordinates the timing of write operations. By synchronizing these write operation activities, the write control unit may facilitate efficient and reliable writing of data to the addressed locations within the memory modules 253. The shared write peripheral 241 may also include error detection and correction mechanisms to improve the reliability and integrity of written data, in some specific embodiments.

[00153] In some specific embodiments, the shared write peripheral 241 may be a single peripheral circuit shared by all of the memory modules 253. It receives write data, addresses, and control/timing signals via the shared write port 251 and controls/orchestrates the writing of that data to the targeted memory location(s) within the addressed memory module(s) 253 using the components and techniques described above. By using a shared write peripheral 241 and shared write port 251, write operations can target any memory module 253 in an efficient manner through a single write access point.

[00154] Each read peripheral 242 may include read address interface logic 249 to receive and decode a read address sent from the FPGA semiconductor device 245. The read address

interface logic 249 translates the received read address to determine which memory location within the corresponding memory module 253 is being requested.

[00155] In addition, each read peripheral 242 may include read data interface logic 250 to interface with the corresponding memory module 253 and send read data to the FPGA semiconductor device 245. The read data interface logic 250 may contain circuitry to access the corresponding memory module 253 based on the decoded read address. This may include enabling appropriate wordlines and bitlines within the 3D memory structure 244 to access the requested memory cell/location. The read data interface logic 250 also contains output drivers, registers, and/or latches to temporarily store the read data and transmit it onto the read data port 248 at an appropriate time based on the read clock signal.

[00156] Each read peripheral 242 may be capable of operating independently and concurrently with the other read peripherals to allow different processing modules or cores within the FPGA 245 to simultaneously read non-overlapping portions of the address space. The read address interface logic 249 and read data interface logic 250 within each read peripheral 242 contain logic and circuits to avoid interference or collisions between concurrent read operations.

[00157] In some embodiments, each read peripheral 242 may contain additional circuitry such as decoders, latches, multiplexers, and synchronous or asynchronous control logic to manage the read operation flow and coordinate interaction with the corresponding memory module 253. Error correction code (ECC) encoding and decoding circuits may also be included in some embodiments.

[00158] Timing circuits such as programmable delay lines, clocks, and phase-locked loops (PLLs) may be distributed throughout the read peripherals 242 to synchronize operations between peripherals and across the memory busses. Alternatively, an independent timing source such as a ring oscillator embedded within each read peripheral 242 provides a local clock isolated from other peripherals.

[00159] The read peripheral circuitry may be implemented using standard digital and memory interface circuits comprising logic gates, registers, and other basic elements constructed depending on the memory technology and interface requirements. In some embodiments, the read peripheral logic incorporates any suitable adaptive techniques such as power gating, voltage/frequency scaling, or adaptive circuit structures which can modify gate sizing or interconnection schemes to optimize operations.

[00160] In an alternative embodiment, any of the components, circuits, and techniques described above in reference to the read peripheral 242 may be partially or wholly contained

within the interface logic block 247 itself rather than distributed among the individual read peripheral circuits. One of ordinary skill would understand the various architectures applicable.

[00161] The 3D memory structure 244 may be formed on the memory-module semiconductor device 243 and may be used to implement the plurality of memory modules 253. The 3D memory structure 244 may employ a three-dimensional cross-point resistive memory array, such as a three-dimensional resistive random access memory (ReRAM) array, to increase the memory density of the memory modules 253, in some specific embodiments.

[00162] The 3D memory structure 244 may include vertically stacked memory layers, with each layer including a regular arrangement of metal lines that serve as electrodes to access memory elements within the layer. The lines are arranged orthogonally in different layers so that the intersections of lines between layers form memory elements accessible via their respective electrodes. For example, the lines in a first layer may run in a lateral direction, while lines in directly adjacent second and third layers run orthogonally in vertical and lateral directions, respectively. This orthogonal line arrangement at each layer intersection creates a three-dimensional cross-point node that incorporates a memory element, such as an oxide-based reversible resistance-switching element.

[00163] The memory elements within the 3D memory structure 244 may utilize any resistance switching material suitable for non-volatile memory applications. In one embodiment, the memory elements use transition metal oxide materials, such as hafnium oxide (HfOx), titanium oxide (TiOx), tantalum oxide (TaOx), aluminum oxide (AlOx), zinc oxide (ZnOx), and/or niobium oxide (NbOx), as the reversible resistance-switching layer between the electrodes. Other candidate materials for the memory elements include chalcogenide materials, such as germanium-antimony-tellurium (GST), as well as various solid electrolytes.

[00164] Regardless of the specific materials, each memory element may be switchable between a high resistance reset state and a low resistance set state by applying suitable write voltages of different polarities across the element via its respective top and bottom electrodes. The state of each memory element can be read in a non-destructive manner by applying a lower read voltage and sensing the resulting read current or resistance. This allows each three-dimensional cross-point junction to operate as a single memory bit, thereby increasing the density of memory bits per unit area over conventional two-dimensional memory device implementations in some specific embodiments.

[00165] Each memory module 253 contains a three-dimensional block portion of the overall 3D memory structure 244. The block size may be configured based on factors such as the desired module capacity and manufacturing constraints. Each block is addressable via row

and column decode circuits associated with the respective read peripheral 242. Word lines and bit lines are connected to the rows and columns, respectively, within each block to allow selective access of individual memory elements via their cross-point intersections. Sense amplifiers may also be associated with each read peripheral 242 to detect the resistance state of addressed memory elements during read operations.

[00166] Each row of memory elements within a block may be implemented as a vertically extending metal nanowire electrode, with orthogonally running bitlines as lateral thin metal layers serving as the other electrode of each three-dimensional memory junction. Additional thin dielectric layers are interleaved between the nanowire and bitline layers to provide electrical isolation at non-intersection regions. The vertical nanowires may be formed using a top-down nanofabrication process involving deposition and patterning. The bitlines can be deposited and patterned using a damascene process common in back-end-of-line (BEOL) semiconductor manufacturing.

[00167] To access the full three-dimensional memory density, multiple blocks within each memory module 253 may be vertically stacked. In one approach, the vertical stacking may involve iteratively depositing alternative horizontal and vertical thin film layers to increasingly build up a monolithic three-dimensional structure. In another approach, pre-formed vertical blocks may be horizontally integrated in a chip-stacking methodology using a combination of lithography, thin film deposition and planarization techniques to progressively build up the full multi-block structure. Interconnects are routed within the 3D memory structure 244 and to the read peripherals 242 to address individual blocks as well as the entire three-dimensional memory array within each memory module 253.

[00168] Each memory module 253 may interact with its respective read peripheral 242, regardless of the internal three-dimensional structure used. The read peripheral 242 receives read address signals from the FPGA device 245 to access corresponding memory locations within the block-level 3D memory structure 244 of the associated memory module 253. Read data is then returned from the selected memory location to the FPGA device 245 through the read peripheral 242. Additional control and power delivery circuitry as needed may also be included within the memory module 253 or its constituent 3D memory structure 244 to enable proper memory operation and support writing, reading and erasing of the memory elements at a module or system level.

[00169] Additional components may optionally be included in the 3D memory structure 244 to support various operations and functions. For example, redundant rows and columns may be added to replace defective regions and improve manufacturing yields. Voltage

generators, addressing decoders and other analog control circuitry may be placed within the structure where needed. Temperature sensors and heat removal structures may be integrated for thermal monitoring and management. Furthermore, a dynamic partitioning and wear-leveling scheme may be implemented within the 3D memory structure 244 to balance write and erase cycles across its elements.

[00170] In summary, the 3D memory structure 244 may use three-dimensional stacking of thin films and cross-point resistive memory elements to increase the density of non-volatile data storage within each memory module 253 relative to conventional two-dimensional memory implementations. This storage density enhancement enables high-capacity memory modules 253 to interface individually with processing cores through the respective read peripherals 242 in a way that provides dedicated high-speed access to data and instructions for each core or processor within an overall system.

[00171] The read address port 247 provides an interface for transmitting read address signals from the FPGA semiconductor device 245 to the memory-module semiconductor device 243. The read address port 247 includes a plurality of electrically conductive contacts or pads formed on a surface of the memory-module semiconductor device 243. These contacts are arranged such that they will line up and bond with a complementary set of contacts on the FPGA semiconductor device 245 when the two semiconductor devices 243, 245 are bonded together.

[00172] The read address port 247 may be coupled to receive read address signals from a read address interface logic 249 located within the interface logic of the FPGA semiconductor device 245. The read address interface logic 249 contains circuitry for receiving read address signals from other components within the FPGA 245, such as programmable logic circuits or processor cores, and conditioning the signals appropriately for transmission via the read address port 247.

[00173] The read address interface logic 249 contains read address buffer and register circuits that temporarily store incoming read address signals and synchronize the signals to a read clock signal. This ensures the read address signals are transmitted to the read address port 247 at the appropriate point in the read clock cycle. The buffers and registers allow read address signals to be pipelined and transmitted continuously, maximizing throughput and efficiency.

[00174] The read address port 247 includes a plurality of electrically conductive lines or traces formed within the memory-module semiconductor device 243 that electrically connect the pads on the surface of the device to circuitry internal to the device. In some embodiments, the conductive lines may be formed within a back-end-of-line metal stack on top of the silicon

substrate of the device 243. In other embodiments, the conductive lines may be routed along or within the silicon substrate itself.

[00175] The internal read address lines are routed to connect the pads of the read address port 247 to read address decoder and distribution circuits associated with each individual memory module 253 within the memory modules group. These decoder circuits receive the read address signals, decode the value, and selectively activate a particular memory module 253 corresponding to that address value.

[00176] Additionally, the read address port 247 may include signal conditioning and amplification circuitry to ensure the read address signals can be transmitted at sufficient voltage levels, timing margins, etc. to reliably activate the appropriate memory modules 253 once received. In some embodiments, the read address port 247 circuitry may include level shifter or translator elements to convert the voltage signaling levels from those compatible with the interface logic 249 to those compatible with the memory modules 253.

[00177] In some embodiments, the read address port 247 may contain additional error detection and correction circuitry, such as parity bits or ECC bits, to ensure read addresses can be transmitted and received reliably. The port 247 thereby provides an integrated interface for robust and reliable transmission of read address signals from the FPGA 245 to the exact memory location within the memory modules 253 specified by that address.

[00178] The read address port 247 thereby provides an electrical interface between the FPGA semiconductor device 245 and the memory module semiconductor device 243 to facilitate the transmission of read address signals from the FPGA 245 for selecting target memory locations within the memory modules 253 during memory read operations. The port 247 contains all necessary conductive contacting structures, internal routing, and signal conditioning circuitry to perform this function effectively and reliably.

[00179] The read data port 248 provides an interface for reading data from the memory modules 253. The read data port 248 includes a plurality of read data interconnects that terminate at a set of read data bonds on the surface of the memory-module semiconductor device 243. The read data interconnects are laid out in a direction perpendicular to the surface of the memory-module semiconductor device 243 to minimize path lengths and ensure consistent timing.

[00180] The read data interconnects may be formed using multiple metal layers during the back-end-of-line processing of the memory-module semiconductor device 243. A low-k dielectric material is used between the metal layers to reduce parasitic capacitance. Wider metal

lines with lower resistivity, such as aluminum or copper, are used for the read data interconnects to minimize resistive-capacitive delay.

[00181] The read data interconnects incorporate serializer/deserializer circuitry to transmit read data from the memory modules 253 to the read data bonds in a serialized format. This enables the transmission of multiple parallel read data bits using fewer interconnects. The serializer/deserializer circuitry includes latch circuits that sequentially capture read data bits in response to non-overlapping clock phases. Transmission gates controlled by the clock phases sequentially route the latched read data bits along the read data interconnects to the read data bonds.

[00182] The read data bonds may be arranged in a layout to match the arrangement of pins or contacts on the FPGA semiconductor device 245 for bonding. The read data bonds may comprise metallization pads of differing sizes, shapes and pitches optimized for the bonding technology used, such as solder bumps or micro-bumps. Matching anti-pads are positioned underneath the read data bonds to prevent short circuits.

[00183] Interface circuitry inside the memory modules 253 may buffer and synchronizes the read data to the relevant clock signal before transmission on the read data interconnects. The interface circuitry also applies error correcting codes to the read data to enable detection and correction of transmission errors.

[00184] The read data port 248 may provide an interface for off-chip communication of read data from the memory modules 253. Its physical design and circuit techniques may be used to optimize timing closure, signal integrity, and error resilience for demanding memory applications, in some specific embodiments.

[00185] Fig. 3B shows a perspective view of the assembly 240 of Fig. 3A to illustrate the bonds of the surfaces of the semiconductor devices in accordance with an embodiment of the present disclosure. That is, Fig. 3B is to illustrate the ports and bonds as described as follows. Although a single write interface logic, read interface logic, and memory module are described in Fig. 3B, one of ordinary skill in the relevant art will appreciate how the pattern can be repeated for all memory modules shown herein. The write interface logic 252 is coupled to a shared write port 257, which is coupled to a set of bonds 258. The set of bonds 258 may include complementary connections on the FPGA semiconductor device 245 vis-à-vis the semiconductor device 243. The semiconductor device 243 includes a shared write port 251 coupled to the share write peripheral 241.

[00186] The read interface logic 259 may comprise the read data interface logic 250 and the read address interface logic 249 as shown in Fig. 3A. The read interface logic 259 is

coupled to a read address port 247 of the FPGA semiconductor device 254 which is coupled to a set of bonds 260, which are also coupled to the read address port 262 of the semiconductor device 243. The set of bonds 260 may include complementary connections on the FPGA semiconductor device 245 vis-à-vis the semiconductor device 243. The read address is sent to a memory module 263 as shown in Fig. 3B. Data is returned via a read data port 261 of the semiconductor device 243, which are coupled to the bonds 260, which are in turn coupled to the read data port 248 of the FPGA semiconductor device.

[00187] Fig. 4. shows a perspective view of an assembly 264 having the integrated circuit of Fig. 1 implemented on a semiconductor device that is electrically connected to another device having field programmable gate array cores interconnected by a network to form the assembly in accordance with an embodiment of the present disclosure. Although the bonds are not shown in Fig. 4, it will be appreciated by one of ordinary skill in the relevant art how the bonds may be arranged with reference to Fig. 3B and the accompanying description.

[00188] The assembly 264 allows communication between the semiconductor device 256 containing FPGA cores like 254 and the memory-module semiconductor device 243. In particular, the assembly 264 includes a memory module semiconductor device 243. This device 243 contains non-volatile memory storage in the form of memory modules, such as memory module 253. These memory modules are arranged in a grid pattern across the device 243 and can be implemented using memory technologies like SRAM, FeRAM, ReRAM, or flash memory.

[00189] The memory capacity and performance characteristics of the semiconductor device 243 may be optimized for mass storage of data that needs to be accessed by the FPGA cores on device 256. This could include weights for neural networks, lookup tables, or other application data.

[00190] The assembly 264 uses a shared write peripheral 241 on the semiconductor device 243. This peripheral 241 connects to a shared write port and allows the FPGA cores, like 254, to write data to one or all of the memory modules 253 simultaneously. The shared write peripheral 241 contains the necessary write circuitry like write drivers, registers, and control logic.

[00191] For reading data, each memory module 253 has its own dedicated read peripheral 242 located on device 243. This allows concurrent read access to different memory modules 253 by different FPGA cores 254. Each read peripheral 242 features independent ports for read addresses 247 and read data 248.

[00192] The assembly 264 also includes the semiconductor device 256 which houses the FPGA cores like 254. These FPGA cores 254 communicate with the read and write ports of device 243 to access the data in the memory modules 253. The FPGA cores 254 connect to the memory ports via interconnects and bonds between the two devices 243 and 256. Additionally, the FPGA cores 254 can communicate with each other using an on-chip network 255. This network-on-chip (NoC) 255 allows the cores 254 to coordinate memory accesses and exchange data as needed.

[00193] The semiconductor device 256 includes FPGA cores 254 that can access and utilize the memory modules 253 on the memory-module semiconductor device 243.

[00194] The semiconductor device 256 may be implemented as an integrated circuit chip, containing multiple FPGA cores 254 fabricated on its silicon substrate. The FPGA cores 254 are programmable logic blocks that can be configured to implement custom logic functions and circuits. Each FPGA core 254 consists of an array of programmable logic elements such as lookup tables, registers, digital signal processing blocks, I/O elements, and interconnects. The functionality of each FPGA core 254 is determined by configuration data loaded into the programmable elements.

[00195] The FPGA cores 254 can be implemented using SRAM, flash, or antifuse technology. SRAM-based FPGA cores provide dynamic reconfigurability, allowing the logic functionality to be reprogrammed repeatedly at runtime. Flash or antifuse-based FPGA cores are one-time programmable, with the logic configuration set during manufacturing or system startup.

[00196] The FPGA cores 254 on the semiconductor device 256 are arranged in a distributed pattern, spaced evenly across the chip area. One purpose of this distribution is to enable parallel data processing by the multiple FPGA cores 254, leveraging spatial architecture advantages.

[00197] The FPGA cores 254 may use features tailored for data-centric workloads. This may include digital signal processing blocks for math-intensive algorithms, high-speed I/O elements, and abundant interconnects for massive data throughput. The FPGA cores 254 are optimized to provide the parallel processing capabilities needed for compute-intensive applications. Communication between the FPGA cores 254 is enabled using an on-chip network 255. This network-on-chip (NoC) 255 allows the FPGA cores 254 to coordinate memory accesses, exchange data, and synchronize operations.

[00198] To interface with the memory-module semiconductor device 243, each FPGA core 254 connects to a dedicated interface logic circuit. The interface logic handles the signal

conversion between the FPGA cores 254 and the memory read/write ports. This provides each FPGA core 254 with independent access to the memory modules 253, enabling parallel memory operations.

[00199] The FPGA core 254 is one of multiple FPGA cores implemented on the semiconductor device 256. The semiconductor device 256 contains an array of FPGA cores including FPGA core 254 distributed in a spaced pattern across its surface.

[00200] The FPGA cores like 254 are programmable logic blocks that can be configured to implement desired logic functions and circuits. Each FPGA core 254 consists of an interconnected array of fundamental programmable elements such as lookup tables, registers, digital signal processing blocks, input/output elements, and programmable interconnects.

[00201] The specific logic implemented by each FPGA core 254 is determined by configuration data loaded into the programmable elements. The configuration data defines the logic functions carried out by the LUTs, the connections between logic blocks and I/O established by the programmable interconnects, the modes of operation of the DSP blocks, and other aspects of the FPGA core's functionality.

[00202] There are several possible technologies that can be utilized to implement the FPGA cores like 254. Some options include SRAM-based FPGAs, antifuse-based FPGAs, and flash-based FPGAs. SRAM-based FPGA cores provide dynamic reconfigurability, allowing the logic configuration to be reprogrammed repeatedly at runtime. Antifuse and flash-based FPGA cores are one-time programmable, with their logic configuration set during manufacturing or system startup.

[00203] As previously mentioned, the FPGA core 254 may contain high-performance components tailored for data-centric workloads. This includes abundant DSP blocks for math-intensive algorithms, high-speed I/O elements, and dense programmable interconnects for massive data throughput. The FPGA core 254 hardware architecture is optimized to enable the high throughput and parallel processing capabilities required by compute-intensive applications.

[00204] To facilitate communication between FPGA cores like 254, the semiconductor device 256 implements an on-chip network 255. This network-on-chip (NoC) 255 allows the FPGA cores 254 to coordinate memory accesses, exchange data, and synchronize operations. The NoC 255 provides high bandwidth and low latency connectivity between the distributed FPGA cores 254.

[00205] The FPGA core 254 interfaces with the memory modules 253 via dedicated connectivity components. These include interface logic circuits that handle signal conversion

between the FPGA core 254 and the memory read/write ports. The interface logic enables FPGA core 254 to independently access the memory modules 253 for parallel data transfers.

[00206] As previously mentioned, the network on chip (NoC) 255 enables communication between the FPGA cores 254 implemented on the semiconductor device 256. The NoC 255 acts as an interconnect fabric, facilitating data transfer between the distributed FPGA cores 254 across the semiconductor device 256.

[00207] The NoC 255 may utilize a mesh topology, with network switches and links interconnecting the FPGA cores 254 in a grid-like pattern. This provides multiple redundant paths between any two FPGA cores 254, enhancing overall network resilience and performance.

[00208] In some embodiments, the NoC 255 may use wormhole or cut-through switching to reduce latency. As packets traverse the NoC 255, they are progressively forwarded in a pipeline fashion through the network switches along the route to the destination without waiting for the full packet to arrive before starting transmission. This allows lower latency compared to store-and-forward techniques.

[00209] The links interconnecting the network switches and FPGA cores 254 in the NoC 255 consist of dedicated communication wires and circuits optimized for data transmission. These links provide high bandwidth connectivity between FPGA cores 254 and may support various signaling standards such as differential signaling or low-voltage signaling to enable high data rates.

[00210] Optionally, the NoC 255 may utilize multiple virtual channels on each physical link to avoid protocol deadlock scenarios and enable quality-of-service traffic differentiation. Traffic from different applications or with different latency requirements can be assigned to separate virtual channels.

[00211] For power efficiency, the NoC 255 may leverage clock gating or power gating techniques. Inactive network switches and links can be powered down when not in use. The NoC 255 may also implement adaptive link width and voltage scaling based on the current traffic load.

[00212] To coordinate access to the NoC 255 and avoid collisions, various arbitration schemes can be employed. A possible approach is round-robin arbitration, where each FPGA core 254 takes turns transmitting in a cyclic order. Alternatively, priority-based schemes can provide differentiated quality of service.

[00213] Error detection and recovery capabilities may also be incorporated in the NoC 255. Each packet can include a cyclic redundancy check (CRC) to detect transmission errors. Corrupted packets are discarded and retries requested to handle errors.

[00214] In summary, the NoC 255 provides a flexible on-chip communication infrastructure to match the distributed parallel nature of computations across multiple FPGA cores 254. The redundant connectivity, advanced switching modes, and QoS capabilities of the NoC 255 enable efficient data exchange and coordination between the FPGA cores. This allows the overall assembly 264 to meet high throughput and low latency requirements of data-intensive applications mapped across the FPGA cores.

[00215] Fig. 5 shows a block diagram 300 illustrating the memory address space of the integrated circuit of Fig. 1 in accordance with an embodiment of the present disclosure. The memory address space includes a write address space 316 and read data address spaces 310, 312, 314.

[00216] The write address space 316 consists of various units where data, e.g., weights, and/or instructions can be stored. These units are referred to as memory addresses. The module group 302 includes multiple memory modules 304, 306, 308. The write address space 316 may be distributed among the memory modules 304, 306, 308 such that the write address space 316 spans from 0 to $N*M-1$. As shown in Fig. 5, the modules group 302 has N memory modules 304, 306, 308, where N is a positive integer, and each module has a memory size of M . The total number of specific write memory addresses in the write address space will be $N*M$, which can be referenced by an integer from 0 to $N*M-1$.

[00217] Starting at 0, memory addresses of the write address space 316 are ordered sequentially up to $N*M-1$. In other words, the first address is 0 and the final address is $N*M-1$, encompassing a total of $N*M$ addresses. This ordering can be linear (each address increases by one) or some other specified pattern depending.

[00218] The write memory addressing can be implemented in a variety of ways based on the system architecture. One method used in a specific embodiment is to use the base and limit registers. In one specific embodiment, the base register holds the smallest legal physical write memory address, and the limit register specifies the size of the range. Therefore, to generate a logical address, you would add the base to the relative address. In other embodiments, a memory addressing scheme may be used where the base used is set to be 0. Yet additional write addressing techniques will be appreciated by one of ordinary skill in the relevant art.

[00219] For any device that writes to the modules group 302, each memory module can possess a specific set of write memory addresses such all memory addresses within the modules group 302 is specific with respect to writing data. e.g., the first module starting at 0 and the last one ending at $N*M-1$. This allocation, in some embodiments, may be dependent on the memory management system of the device writing data to the modules 304, 306, 308, which could range from simple fixed partitioning schemes to more complex dynamic partitioning models.

[00220] For instance, in a straightforward linear model where each module (304, 306, or 308) has an equal size of M addresses, the first module 304 would possess write addresses 0 to $M-1$, the second module would have write addresses M to $2M-1$, the third module would have write addresses $2M$ to $3*M-1$, and so forth. The N th module 308, therefore, would possess write addresses from $(N-1)*M$ to $N*M-1$.

[00221] It is contemplated that one of ordinary skill in the relevant art may use other implementations of write memory addresses from 0 to $N*M-1$ that depends on various factors such as the hardware architecture, operating system, memory management schemes, and the nature of the programs being run on the system, etc.

[00222] The modules group 302 has different read data address spaces 310, 312, 314. These read address spaces 310, 312, 314 may have overlapping addresses spaces, may have contiguous address spaces, or may have coextensive address spaces. The read address spaces 310, 312, 314 may be independent relative to each other. The system includes three independent read address spaces, labeled as read address spaces 310, 312, and 314. Each of these read address spaces is distinct from the others, meaning that reads can be performed in each space without affecting the others.

[00223] The read address spaces 310, 312, 314 may be defined as contiguous blocks of memory addresses, each with its own starting address and ending address. In modules group 302, each read address space 310, 312, 314 may have a range of addresses that corresponds to values from 0 to $M-1$, where M is a maximum value determined by the size of the modules 304, 306, 308 being used.

[00224] In one embodiment, allowing one processing unit to interface with each read address space 310, 312, 314, the concurrent reads may be implemented as described herein. The independence of the read address spaces 310, 312, 314 ensures that each processing unit can access its desired data without causing any interference or conflict with other processing units.

[00225] Fig. 6 shows a block diagram illustrating the memory address space with the signal interfaces of the integrated circuit of Fig. 1 in accordance with an embodiment of the

present disclosure. The signals used in Fig. 6 may be used with any embodiment described herein. However, one of ordinary skill in the relevant art will appreciate that different signaling schemes may be used.

[00226] The modules group 402 includes modules 404, 406, 408 that share a common write peripheral 411. The write peripheral 411 includes a write address bus that includes the address of the data being written, a write data bus that includes the data, a write clock cause the writes to occur (e.g., either on a leading or trailing edge of the clock signal, etc.). The writes only occur if the write enable signal indicates a write should occur. Any logic may be used, e.g., high voltage may correspond to 1 and a low voltage may correspond to 0, or vice versa. In some embodiments, the write peripheral 411 may be on the chiplet 230 and in other embodiments, the write peripheral 411 is on the second device 226.

[00227] The modules group 402 has modules 404, 406, 408 where each has a respective read peripheral 410, 412, 414. Each of the read peripheral 410, 412, 414 has a read address bus to send an address for reading, a read data bus to receive the data, a read clock which is the clock used to control the timing of the output of the digital data, and an output enable that is a precondition to outputting data. Any logic may be used, e.g., high voltage may correspond to 1 and a low voltage may correspond to 0, or vice versa. In yet additional embodiments, multi-bit or analog data storage may be used. In some embodiments, one or more of the read peripherals 410, 412, 414 may be on the chiplet 230 and in other embodiments, one or more of the read peripherals 410, 412, 414 are on the second device 226.

[00228] FIG. 7 illustrates a stack 500 consisting of two semiconductor devices 562, 502 that can be bonded together. The first semiconductor device 562 may be an application chiplet, while the second semiconductor device 502 may be a memory chiplet. The application chiplet 562 includes circuitry to facilitate interconnect testing prior to bonding by using interconnect loopbacks 522, 532, 542. The first semiconductor device 562 has several test signal connections 580, 582, 550 which can be used to test a path by examining how the signal is returned via a respective one of a return signal connection 558, 554, 584. These connections may be coupled to boundary scan cells and/or external circuitry that is external to the interface logic 548. A tri-state buffer 544 may have an enable connection 586 that can control whether or not the tri-state buffer 544 is in an active state or an inactive state.

[00229] The stack 500 can facilitate the design and integration of a second semiconductor device 502 (e.g., system-on-chip (SoC) or application chiplet) with a first semiconductor device 562 (e.g., a co-chiplet) and more particularly, to a known-good-die application design method to facilitate independent testing.

[00230] In some embodiments, the first and second semiconductor devices 562, 502 are bonded together at the final stages. Thus, to achieve independently verified chiplets, the disclosed method provides a thin interface module (e.g., the interface logic 548) on the first semiconductor device 562 (e.g., an application SoC base-die) wherein the inputs and outputs may be registered on a clock edge. By doing this, the application designer can focus on ensuring independent validity of each semiconductor device.

[00231] With the thin interface logic 548 in place, the application designer's responsibility is primarily to interface with the thin interface logic 548 according to specifications. By following this approach, the design complexities associated with co-chiplet integration can be effectively managed, leading to improved efficiency and reliability.

[00232] Independent testing of the semiconductor devices 562, 502 can be used to minimize yield loss during post-integration stages. For example, in a 3D Memory Fab, each memory chiplet may be fabricated and individually tested to identify known-good dies, which are then inventoried. The testing process utilizes a serial scan-based built-in self-test (BIST) technique to ensure thorough testing and verification of the memory chiplets. During the place and route stage, the interface logic 548 is placed by the placement tool, while custom scripts can facilitate vertical routing of the loopbacks 522, 532, 542 and bonds 518, 524, 536 etc. The rest of the design may then be fully routed to establish complete connectivity. Subsequently, the application wafer for the first semiconductor device 562 may be fabricated with top metal bonds 518, 524, 536.

[00233] The first semiconductor device 562 (e.g., as a fully fabricated application wafer) may be subjected to testing using pass-thru interface logic, allowing comprehensive evaluation of the application device's functionality and performance. Optionally, testing at the wafer level may also be carried out to further ensure the quality and reliability of the first semiconductor device 562 (e.g., an application chiplet). Throughout this testing phase, the aim is to identify any defects or issues that may affect the device's functionality or integration.

[00234] In the hybrid package integration phase, the known good second semiconductor device 502 (e.g., a memory chiplet), obtained from previous testing, may be face-to-face bonded with the known good first semiconductor device 562 (e.g., the application chiplet). This bonding process can facilitate a secure and reliable connection between the semiconductor devices 502, 562 (e.g., a memory chiplet and an application chiplet). Moreover, by optionally performing the semiconductor device bonding at the wafer level, post-packaging yield can be enhanced, thereby reducing potential yield losses and improving overall production efficiency.

[00235] The first semiconductor device 562 includes an interface logic 548 that facilitates communication to external circuitry and external semiconductor devices. The interface logic 548 also includes interconnect loopbacks 522, 532, 542 to test the connectivity between the interface logic 548 and conductive pads (or bonds) 518, 526, 536 found on the surface of the first semiconductor device 562. These connections may be tested prior to bonding with another device (or in some embodiments, after bonding). The conductive pad 518 of the first semiconductor device 562 may be electrically coupled to the conductive pad 516 of the second semiconductor device 502. Additionally, the conductive pads 526 of the first semiconductor device 562 may be electrically coupled to the conductive pads 528 of the second semiconductor device 502. And, the conductive pads 526 of the first semiconductor device 562 may be electrically coupled to the conductive pads 538 of the second semiconductor device 502.

[00236] The interface logic 548 comprises several input or output interfaces to other circuitry (not shown) within the first semiconductor device 562, including a read address input 564, a read clock input 560, and a read data output 546 (each of which may have boundary scan cells to facilitate boundary scan testing). The read clock input 560 is connected to a read clock interconnect 520, which extends to the surface of the first semiconductor device 562 and terminates at a conductive pad 518. Additionally, the conductive pad 518 is connected to an interconnect loopback 522 that is coupled to a buffer 556. The buffer 556 is further connected to the read clock output 558, which should have the same value as the read clock input 560 of the interface logic 548 if the integrity of the circuit is sound. Thus, the buffer 556 can couple and/or amplify a signal from the interconnect loopback 522 such that when external circuitry applies a clock signal to the read clock input 560 the value of the output to the read clock output 558 can be checked to test the integrity of the path from the interface logic 548 to the conductive pad 518 and finally back to the read clock output 558.

[00237] Similarly, the interface logic 548 comprises a read address interconnect 524 connected to the read address input 564. The read address interconnect 524 extends to the surface of the first semiconductor device 562 and connects to conductive pads 526. An interconnect loopback 532 is coupled to the conductive pads 526 and linked to a buffer 522, allowing for testing of the read address output 554 by external test circuitry. Although only one connection is shown, one of ordinary skill in the art would know how to extend this to parallel data lines (e.g., 8 interconnects, with 8 conductive pads, 8 interconnect loopbacks, 8 buffers, etc.) to form an 8-bit address space. Any other number of bits or address size may be used.

[00238] And for the read address interconnect 524, consider that the test signal is a read address, and the return signal is a return read address obtained from the interconnect loopback 522. The return read address 532 may be applied to a read data output bus 546 of the interface logic 548, amplified or coupled by a buffer 552, tested, etc.

[00239] The interconnect loopbacks 522, 532, 542 in the interface logic 548 may include additional components such as comparators and delay circuits to ensure signal integrity and test timing characteristics. A test may include dynamically adjusting the test signal based on predefined test patterns, allowing comprehensive testing and validation of the interconnects.

[00240] For the clock interconnect 520, the test signal is a clock signal, and the return signal is a return clock signal received from the interconnect loopback 532. Similarly, the return clock signal can be applied to a read data output bus of the interface logic 548, buffered in a register, tested, etc.

[00241] For the read data interconnect 534, the test signal represents test data that is applied via a read data input connection 550 which can be amplified by a tri-state buffer 544, and the return signal is a return test data received via the read data output connection 584 from test data applied to the interconnect loopback 542. The test data may be received and/or outputted to a bus of the interface logic 548, buffered in a register, tested, etc.

[00242] When the first semiconductor device 562 and second semiconductor device 502 are bonded together, their respective conductive pads (518, 516), (526, 528), and (536, 538) are electrically coupled together. This enables the transfer of electrical signals between the first and second semiconductor devices 562, 502.

[00243] The second semiconductor device 502 includes an interface logic 568 consisting of various interconnects (each of which may include a boundary scan cell for testing). This interface logic 568 comprises a read clock interconnect 514 receiving a read clock signal from the first semiconductor device 562, a read address interconnect 530 receiving a read address from the first semiconductor device 562, and a read data interconnect 540 transmitting data from the memory cells via a read data output 504 to the first semiconductor device 562. That is, the read data from the memory cells 504 is passed through the interface logic 568 to the read data interconnect 540. Similarly, the read address interconnect 530 is passed through the interface logic 568 to the read address 508, which is used for accessing the memory cells. The read clock interconnect 514 is the clock that is sent to the memory cells and/or internal logic so that the interface logic can buffer the read data as presented to the read data interconnect 540.

[00244] The interface logic 568 can also utilize a serial test data input 510, a test clock 512, and a serial test data output 566 for performing boundary scanning. These components

enable the interface logic 568 to test the connectivity of interconnects and verify signal integrity. The interface logic 548 also includes boundary scan cells including a serial test data in 570 and a serial test data out 574. Optionally, a separate clock, e.g., test clock 574, can be used instead of the read clock 560. However, in some embodiments, the boundary scan cells may use the read clock 560. Thus, the connectivity from the interface logic 548 of the first semiconductor device 562 to the second semiconductor device 502 may be tested to make sure that the integrity of the connectivity therebetween is suitable.

[00245] A boundary scan is a testing technique used to verify the connectivity and integrity of interconnects within semiconductor devices. Thus, a boundary scan in testing the connectivity between the interface logic 548 of the first semiconductor device 562 and the interface logic 568 of second semiconductor device 502, may be used to ensure the suitability of the connectivity. Any, all, or some of the inputs and/or outputs for the interface logics 548, 568 may include boundary scan cells to control, modify, or test any, all, or some of the inputs and/or output.

[00246] The boundary scan test can be done by forming a boundary scan chain, connecting the boundary scan cells within each interface logic 548, 568 in a daisy-chain configuration. This creates a serial shift register arrangement, enabling controlled shifting of test data and return data through the boundary scan chain.

[00247] The boundary scan cells within the interface logics 548, 568 provide the control and capture capabilities to manipulate and observe the test data and return data within the boundary scan chain via a connected device. This can ensure reliable testing of the connectivity between the interface logic 548 of the first semiconductor device 562 and the interface logic 568 of the second semiconductor device 502. Thus, through the boundary scan test, the interconnects between the two interface logics 548, 568 can be thoroughly examined and validated to ensure that the connectivity is suitable and functioning as intended. The boundary scan cells within each interface logic 548, 568, can enable precise control over the test data and return data, allowing for comprehensive analysis and assessment of the connectivity between the two semiconductor devices.

[00248] During testing, various test patterns and data can be loaded into the boundary scan cells through the serial test data input 570 of the first semiconductor device's 562 interface logic 548 and/or the serial test data input 510 of the second semiconductor's 502 interface logic 568. These test patterns simulate different input scenarios and conditions, allowing for the examination of various connectivity scenarios between the two interface logics. The loaded test data is then shifted through the boundary scan chain using the test clocks 512, 574. Please note

that the clocks may be tied together, synchronized, and/or other clocks may be used. With each clock cycle, the test data propagates through the serially connected boundary scan cells, advancing to the subsequent cells one by one. This shifting process allows the test data to traverse the interconnects between the two interface logics 548, 568, verifying the connectivity and integrity of the interconnects. This allows for subsequent analysis and comparison with the expected return data. By comparing the sampled return data with the expected values, the integrity of the connectivity between the two interface logics 548, 568 can be accurately assessed.

[00249] Also, at specific points within the boundary scan chain, the return data from the first semiconductor device's 562 interface logic 548 and/or the serial test data input 510 of the second semiconductor's 502 interface logic 568 is sampled into additional boundary scan cells. These boundary scan cells capture and retain the return data for further analysis and comparison.

[00250] Fig. 8 shows a diagram of two semiconductor devices 622, 602, such as two chiplets, that automatically tests the connectivity of write interconnects 620, 624, 634 within a semiconductor device and the integrated connectivity in accordance with an embodiment of the present disclosure. The first semiconductor device 662 has several test signal connections 680, 682, 684 which can be used to test a path by examining how the signal is returned via a respective one of a return signal connections 668, 664, 660. In some embodiments, the write interconnects 620, 624, 634 may be integrated with the read interconnects 520, 524, 534 of Fig. 7 on the same interface logic.

[00251] As shown in Fig. 8, the stack 600 of two semiconductor devices 662, 602: an application chiplet (first semiconductor device 662) and a memory chiplet (second semiconductor device 602). The application chiplet 662 includes interconnect loopbacks 622, 632, 642 for interconnect testing before bonding. The interface logic 648 within the application chiplet facilitates communication with external circuitry and includes interconnect loopbacks 622, 632, 642 to test connectivity with conductive pads 618, 626, 636.

[00252] The interface logic 648 includes input/output interfaces for connectivity to the write address 624, the write clock 620, and the write data 634 of the first semiconductor device 662. These interfaces, along with interconnect loopbacks 622, 632, 642, enable testing of the connectivity between the interface logic 648 and the conductive pads (or bonds) 618, 626, 636 prior to bonding. Additionally, the conductive pads 618, 626, 636 of the first semiconductor device 662 may be electrically coupled to the conductive pads 616, 628, 638 of the second

semiconductor device 602 such that the interface logic 648 can provide connectivity to write functionally of the memory found within the second semiconductor device 602.

[00253] The interface logic 648 consists of interconnects such as the write clock interconnect 620, connected to a write clock input 660, and terminating at conductive pad 618. It also includes interconnect loopback 622 and a buffer 656. The integrity of the circuit can be tested by applying a clock signal to the write clock input 660 and comparing it with the write clock output 658.

[00254] Similarly, the interface logic 648 includes a write address interconnect 624 connected to the write address input 664 and terminated at conductive pads 626. An interconnect loopback 632 coupled with buffer 622 allows for testing the write address output 664.

[00255] The interconnect loopbacks 622, 632, 642 may include additional components such as comparators and delay circuits to ensure signal integrity and to test timing characteristics. The test signals and return signals for testing may include a write address 624, a write data 634, and a write clock signal 620. By applying these signals through interconnect loopbacks and comparing them with the expected values, the connectivity and integrity of the interconnects can be validated.

[00256] When the first semiconductor device 662 and second semiconductor device 602 are bonded together, their respective conductive pads (618, 616), (626, 628), (636, 638) are electrically coupled, enabling the transfer of electrical signals between them.

[00257] The second semiconductor device 602 has its own interface logic 668, which includes interconnects for a write clock 614, a write address 630, and write data 640. The write data applied to the write data 634 interconnect, is passed through the interface logic 668 to the write data 604, which is used to write to the memory cells addressed by the write address 608 on a write clock 606. The write clock interconnect 614 can receive the clock signal to control the writing to the memory cells and internal logic.

[00258] The interface logic 668 also incorporates components for boundary scanning, such as a serial test data input 610, a test clock 612, and a serial test data output 666. Boundary scanning allows for testing the connectivity of interconnects and verifying signal integrity between the interface logics 648 and 668 of the two semiconductor devices.

[00259] The boundary scan test involves forming a boundary scan chain by connecting boundary scan cells in a daisy-chain configuration within each interface logic 648, 668. This enables controlled shifting of test data and return data through the boundary scan chain. The boundary scan cells provide control and capture capabilities to manipulate and observe the test

data and return data within the boundary scan chain. This enables reliable testing of the connectivity between the interface logics 648 and 668. Some or all of the inputs and/or outputs of the interface logics 648, 668 may include boundary scan cells to facilitate the reading, writing, or recoding of test data as known to one of ordinary skill in the relevant art.

[00260] During testing, test patterns and data are loaded into the boundary scan cells through the serial test data input 610, 670 of the interface logics 668, 648. These test patterns simulate different input/output scenarios, allowing examination of various connectivity scenarios between the two interface logics 668, 648. The loaded test data is then shifted through the boundary scan chain using the test clocks 612, 660. With each clock cycle, the test data propagates through the boundary scan cells, verifying the connectivity and integrity of the interconnects.

[00261] Return data from the interface logics is sampled into additional boundary scan cells within the boundary scan chain for analysis and comparison. By comparing the sampled return data with the expected values, the integrity of the connectivity between the two interface logics can be accurately assessed.

[00262] Figs. 9A-9B shows a block diagram of a system 700 employing correct-by-construction timing closure for an application chiplet 702 with an interface circuit 701 having read registers 726a, 726b and a write register 728, and a memory chiplet 704 in accordance with an embodiment of the present disclosure. Thus, the system 700 may include two separate semiconductor devices, e.g., the application chiplet 702 and the memory chiplet 704 that may have been formed on two separate dies. The interface circuit 702 (which can be referred to as interface module) can include registers 762a, 762b, 728 to allow the application chiplet 702 to have a correct-by-construction timing closure with the group of memories 706 on the memory chiplet 704.

[00263] The memory chiplet 704 may include a group of modules 706 having modules 708a, 708b and 708c that are independently accessible via read peripherals 722a, 722b, and 722c, respectively. The memory chiplet 704 may also include a shared write peripheral 710 where data may be written to a memory location within the group of modules 706. The interconnects found within the memory chiplet 704 may be coupled to the surface via bonds to facilitate communication between the semiconductor devices 702, 704, for example when they are bonded together in a stack configuration in one specific embodiment. A designer of the application chiplet 702 may place the interface module 701 using a netlist on a location within a predetermined distance from a surface of the application chiplet 702. That is, the signal time and characteristic may be predefined to work the memory of the memory chiplet 704. For

example, the travel time of a signal from a read register 726 may be less than a predetermined time.

[00264] The interface module 701 may include one or more read registers 726a, 726b and a write register 728 that may communicate with the memory chiplet 704 via several bonds as shown in Figs. 9A-9B.

[00265] The read register 726a can interface and communicate with other circuitry within the application chiplet 702 via a read application programming interface (“API”) 714a. That is, the read API 714a may be a bus where data may be requested by external circuitry (e.g., via a CPU) to provide data from the memory chiplet 704 to the other circuitry (e.g., the exemplary CPU).

[00266] The read register 726a may be controlled via a read clock that is received via a read-clock interconnect 712a. The read-clock interconnect 712a may also be coupled to a read-clock bond 738a which can be connected to the read peripheral 722a on the memory chiplet 704 to provide clocking for the memory contained therein. The interconnects may be connected together by the read-clock bond 738a on the surface of the application chiplet 702 and a respective bond (not explicitly shown in Figs. 9A-9B) on the memory chiplet 704.

[00267] The read register 726a may be coupled to a read-address interconnect 716a (which may include multiple parallel connections) that communicates a read address that has been loaded into the read register 726a. The read-address interconnect 716a is coupled to a plurality of read-address bonds 740a so that the read address can be received by the memory chiplet 704. The read address is a value that the read peripheral 722a can translate to query a location within the module 708a.

[00268] The read register 726a also includes a read-data interconnect 718a that can receive the data from the module 708 via a plurality of read-data bonds 742a. The data may be held by the register 726a for communication to other circuitry via the read API 714a.

[00269] The read register 726a may also include a read-data enable interconnect 720a to enable the output of data from the module 708a. The read-data enable interconnect 720a is coupled to read-data enable bond 744a so that when stacked, the chiplets 702, 704 are in electrical communication with each other.

[00270] A read register 726b may be similar or identical to the read register 726a on the application chiplet 702 that can communicate with other circuitry through a read application programming interface (API) 714b. The read API 714b acts as a bus where external circuitry, such as a CPU, can request data from the memory chiplet 704.

[00271] The read registry 726b is controlled by a read clock, which is received through a read-clock interconnect 712b. The read-clock interconnect 712b is also connected to a read-clock bond 738b, which provides clocking for the memory chiplet 704. The read-clock bond 738b connects the interconnects on the application chiplet 702 and the memory chiplet 704.

[00272] The read register 726b is connected to a read-address interconnect 716b, which transmits a loaded read address. The read-address interconnect 716b is connected to multiple read-address bonds 740b, allowing the memory chiplet 704 to receive the read address. The read address is used by the read peripheral 722b to query a location within the module 708b.

[00273] The read register 726b also has a read-data interconnect 718b, which receives data from the module 708. The data is stored in the register 726b and can be communicated to other circuitry through the read API 714b.

[00274] Additionally, the read register 726b includes a read-data enable interconnect 720b, which enables the output of data from the module 708b. The read-data enable interconnect 720b is connected to a read-data enable bond 744b, ensuring electrical communication between the stacked chiplets 702 and 704.

[00275] The interface module 701 also includes a write-interface register 728 that receives write data and a write address via a write application programming interface 724 to write the write data to the group of modules 706. The write-interface register 728 receives a write clock via a write write-clock interconnect 730 that is also coupled to the surface of the semiconductor device via write-clock bond 746. The write-interface register 728 also has coupled to it write-address interconnects 732 to provide a write address from the write-interface register 728 to a plurality of write-address bonds 748 on the surface. The write-interface register 728 is also coupled to write-data interconnect 734 to send data via a plurality of write-data bonds 750. The write-interface register 728 is also coupled to a write-data enable interconnect 736 which send a write enable signal via a write-enable bond 752.

[00276] Various alternatives and modifications can be devised by those skilled in the art without departing from the disclosure. Accordingly, the present disclosure is intended to embrace all such alternatives, modifications, and variances. Additionally, while several embodiments of the present disclosure have been shown in the drawings and/or discussed herein, it is not intended that the disclosure be limited thereto, as it is intended that the disclosure be as broad in scope as the art will allow and that the specification be read likewise. Therefore, the above description should not be construed as limiting, but merely as exemplifications of particular embodiments. And those skilled in the art will envision other modifications within the scope and spirit of the claims appended hereto. Other elements, steps,

methods, and techniques that are insubstantially different from those described above and/or in the appended claims are also intended to be within the scope of the disclosure.

[00277] The embodiments shown in the drawings are presented only to demonstrate certain examples of the disclosure. And the drawings described are only illustrative and are non-limiting. In the drawings, for illustrative purposes, the size of some of the elements may be exaggerated and not drawn to a particular scale. Additionally, elements shown within the drawings that have the same numbers may be identical elements or may be similar elements, depending on the context.

[00278] Where the term "comprising" is used in the present description and claims, it does not exclude other elements or steps. Where an indefinite or definite article is used when referring to a singular noun, e.g., "a," "an," or "the," this includes a plural of that noun unless something otherwise is specifically stated. Hence, the term "comprising" should not be interpreted as being restricted to the items listed thereafter; it does not exclude other elements or steps, and so the scope of the expression "a device comprising items A and B" should not be limited to devices consisting only of components A and B. This expression signifies that, with respect to the present disclosure, the only relevant components of the device are A and B.

[00279] The term "stack" as used herein may mean any such coupling, bonding, securing, gluing, electrically coupling, physically coupling, signally coupling, optically coupling, or otherwise interfacing one or more devices together in any orientation such that they are secured together on any heterogeneous or homogenous surfaces between each other.

[00280] Furthermore, the terms "first," "second," "third," and the like, whether used in the description or in the claims, are provided for distinguishing between similar elements and not necessarily for describing a sequential or chronological order. It is to be understood that the terms so used are interchangeable under appropriate circumstances (unless clearly disclosed otherwise and that the embodiments of the disclosure described herein are capable of operation in other sequences and/or arrangements than are described or illustrated herein).

[00281] Each of the characteristics and examples described herein, and combination thereof, may be said to be encompassed by the present disclosure. The present disclosure is thus drawn to the following non-limiting numbered aspects:

[00282] 1. An integrated circuit, comprising: a first semiconductor device, comprising: a first programmable gate array; and a plurality of interface logics including a first interface logic; and a first memory port connected to a first set of bonds on a surface of the first semiconductor device, the first programmable gate array operatively coupled to the first interface logic to communicate via the first memory port.

- [00283] 2. The integrated circuit according to aspect 1, wherein the first interface logic is formed from the first programmable gate array.
- [00284] 3. The integrated circuit according to aspect 1, wherein the first interface logic is distributed in a spaced pattern in the first programmable gate array.
- [00285] 4. The integrated circuit according to aspect 1, wherein the first semiconductor device includes a plurality of memory ports including the first memory port, wherein the plurality of memory ports forms a spaced pattern on the first semiconductor device.
- [00286] 5. The integrated circuit according to aspect 1, further comprising a plurality of sets of bonds including the first set of bonds, wherein the plurality of sets of bonds is distributed in a spaced pattern on the surface of the first semiconductor device.
- [00287] 6. The integrated circuit according to aspect 1, the first semiconductor device further comprising a plurality of memory ports including the first memory port, wherein each interface logic of the plurality of interface logics is coupled to a respective memory port of the plurality of memory ports.
- [00288] 7. The integrated circuit according to aspect 6, the first semiconductor device further comprising a plurality of programmable gate arrays including the first programmable gate array, wherein each of the plurality of programmable gate arrays is coupled to a respective interface logic of the plurality of interface logics.
- [00289] 8. The integrated circuit according to aspect 6, further comprising a plurality of cores, wherein each core of the plurality of cores is coupled to a respective interface logic of the plurality of interface logics.
- [00290] 9. The integrated circuit according to aspect 6, further comprising a plurality of sets of bonds on the surface of the first semiconductor device, wherein each memory port of the plurality of memory ports is coupled to a respective set of bonds of the plurality of sets of bonds, wherein the plurality of sets of bonds includes the first set of bonds.
- [00291] 10. The integrated circuit according to aspect 1, wherein the first set of bonds are metallic pads.
- [00292] 11. The integrated circuit according to aspect 1, wherein the first semiconductor device further comprises: a plurality of programmable gate arrays including the first programmable gate array; and a network-on-chip configured to enable communication among at least two programmable gate arrays of the plurality of programmable gate arrays.
- [00293] 12. The integrated circuit according to aspect 1, wherein the first semiconductor device further comprises: at least one network-on-chip communication system connecting at least two processing elements implemented on the first programmable gate array,

the first programmable gate array operatively coupled to the plurality of interface logics to communicate via the memory port.

[00294] 13. The integrated circuit according to aspect 1, wherein the first memory port is a read port.

[00295] 14. The integrated circuit according to aspect 13, wherein the first memory port is a multi-cycle port.

[00296] 15. The integrated circuit according to aspect 1, wherein the first memory port is a write port.

[00297] 16. The integrated circuit according to aspect 15, wherein the first memory port is a multi-cycle port.

[00298] 17. The integrated circuit according to aspect 1, wherein the first memory port is a read/write port.

[00299] 18. The integrated circuit according to aspect 17, wherein the first memory port is a multi-cycle port.

[00300] 19. The integrated circuit according to aspect 1, wherein the first programmable gate array is a field programmable gate array.

[00301] 20. The integrated circuit according to aspect 1, wherein the first programmable gate array is a reconfigurable gate array.

[00302] 21. The integrated circuit according to aspect 20, wherein the reconfigurable gate array is dynamically programmable.

[00303] 22. The integrated circuit according to aspect 20, wherein the reconfigurable gate array is one-time programmable.

[00304] 23. The integrated circuit according to aspect 1, wherein the first semiconductor device includes a network-on-chip fabric.

[00305] 24. The integrated circuit according to aspect 23, wherein the programmable gate array includes an array of embedded programmable gate array cores configured to communicate with each other via the network-on-chip fabric.

[00306] 25. The integrated circuit according to aspect 23, wherein the network-on-chip fabric is formed by the first programmable gate array.

[00307] 26. The integrated circuit according to aspect 1, wherein the first programmable gate array includes an array of embedded programmable gate array cores, wherein each of the array of embedded programmable gate array cores is operatively coupled with a respective interface logic of the plurality of interface logics.

[00308] 27. The integrated circuit according to aspect 1, wherein the first programmable gate array includes at least two embedded programmable gate array cores, wherein each of the at least two embedded programmable gate array cores is operatively coupled with a respective interface logic of the plurality of interface logics.

[00309] 28. The integrated circuit according to aspect 1, wherein the first programmable gate array includes an array of embedded programmable gate array cores, wherein each of the array of embedded programmable gate array cores is operatively coupled with at least one of the plurality of interface logics.

[00310] 29. The integrated circuit according to aspect 1, wherein the first programmable gate array includes an array of embedded programmable gate array cores, wherein each of the array of embedded programmable gate array cores is operatively coupled with a single one of the plurality of interface logics.

[00311] 30. The integrated circuit according to aspect 1, further comprising: a second semiconductor device comprising: a plurality of memory modules including a first memory module, the first memory module coupled a second set of bonds on the surface of the second semiconductor device, wherein the first set of bonds and the second set of bonds are configured to interface with each other when the first and second semiconductor devices are bonded together.

[00312] 31. The integrated circuit according to aspect 30, wherein the first memory port is a write memory port.

[00313] 32. The integrated circuit according to aspect 31, wherein the first memory port is configured to write to all of the plurality of memory modules.

[00314] 33. The integrated circuit according to aspect 31 or 32, the first semiconductor device further comprising: a plurality of read memory ports, wherein the plurality of interface logics includes a plurality of read interface logics, wherein each read interface logic of the plurality of read interface logics is operatively coupled to a respective one of the plurality of read memory ports.

[00315] 34. The integrated circuit according to aspect 33, wherein each of the plurality of read interface logics interfaces with a respective memory module of the plurality of memory modules.

[00316] 35. The integrated circuit according to aspect 30, wherein the plurality of memory modules includes an array of SRAMS.

[00317] 36. The integrated circuit according to aspect 30, wherein the plurality of memory modules includes an array of non-volatile memories.

- [00318] 37. The integrated circuit according to aspect 30, wherein the plurality of memory modules includes an EEPROM.
- [00319] 38. The integrated circuit according to aspect 30, wherein the plurality of memory modules includes a ROM.
- [00320] 39. The integrated circuit according to aspect 30, wherein the plurality of memory modules includes read-optimized none-volatile memory.
- [00321] 40. The integrated circuit of aspect 1, wherein the programmable gate array utilizes antifuse, SRAM, or flash memory technology for programming.
- [00322] 41. The integrated circuit of aspect 1, wherein programmable gate array cores are connected in a mesh topology by the network-on-chip fabric.
- [00323] 42. The integrated circuit of aspect 1, wherein the interface logics include voltage level shifters to convert signals of the programmable gate array to and from voltage levels of the first memory port.
- [00324] 43. A method of operating an integrated circuit, the method comprising: utilizing a first semiconductor device that includes a first programmable gate array; employing a plurality of interface logics, including a first interface logic; connecting a first memory port to a first set of bonds on a surface of the first semiconductor device; and operatively coupling the first programmable gate array to the first interface logic to enable communication via the first memory port.
- [00325] 44. The method according to aspect 43, wherein employing the first interface logic comprises forming the first interface logic from the first programmable gate array.
- [00326] 45. The method according to aspect 43, wherein employing the first interface logic includes distributing the first interface logic in a spaced pattern within the first programmable gate array.
- [00327] 46. The method according to aspect 43, further comprising including a plurality of memory ports, such as the first memory port, wherein forming a spaced pattern on the first semiconductor device with the plurality of memory ports.
- [00328] 47. The method according to aspect 43, further comprising distributing a plurality of sets of bonds, including the first set of bonds, in a spaced pattern on the surface of the first semiconductor device.
- [00329] 48. The method according to aspect 43, further comprising utilizing a plurality of memory ports including the first memory port, wherein each interface logic of the

plurality of interface logics is coupled to a respective memory port of the plurality of memory ports.

[00330] 49. The method according to aspect 48, further comprising employing a plurality of programmable gate arrays, including the first programmable gate array, wherein each of the plurality of programmable gate arrays is coupled to a respective interface logic of the plurality of interface logics.

[00331] 50. The method according to aspect 48, further comprising coupling a plurality of cores, wherein each core of the plurality of cores is coupled to a respective interface logic of the plurality of interface logics.

[00332] 51. The method according to aspect 48, further comprising coupling a plurality of sets of bonds on the surface of the first semiconductor device, wherein each memory port of the plurality of memory ports is coupled to a respective set of bonds of the plurality of sets of bonds, including the first set of bonds.

[00333] 52. The method according to aspect 43, wherein utilizing the first set of bonds involves employing metallic pads.

[00334] 53. The method according to aspect 43, further comprising employing a plurality of programmable gate arrays, including the first programmable gate array, and configuring a network-on-chip to enable communication among at least two programmable gate arrays of the plurality.

[00335] 54. The method according to aspect 43, further comprising implementing at least one network-on-chip communication system to connect at least two processing elements on the first programmable gate array, and operatively coupling the first programmable gate array to the plurality of interface logics to communicate via the memory port.

[00336] 55. The method according to aspect 43, wherein utilizing the first memory port comprises employing the first memory port as a read port.

[00337] 56. The method according to aspect 55, wherein employing the first memory port as a read port further includes using the first memory port as a multi-cycle port.

[00338] 57. The method according to aspect 43, wherein utilizing the first memory port comprises employing the first memory port as a write port.

[00339] 58. The method according to aspect 57, wherein employing the first memory port as a write port further includes using the first memory port as a multi-cycle port.

[00340] 59. The method according to aspect 43, wherein utilizing the first memory port comprises employing the first memory port as a read/write port.

- [00341] 60. The method according to aspect 59, wherein employing the first memory port as a read/write port further includes using the first memory port as a multi-cycle port.
- [00342] 61. The method according to aspect 43, wherein utilizing the first programmable gate array includes employing a field programmable gate array.
- [00343] 62. The method according to aspect 43, wherein utilizing the first programmable gate array includes employing a reconfigurable gate array.
- [00344] 63. The method according to aspect 62, wherein employing a reconfigurable gate array further includes dynamically programming the reconfigurable gate array.
- [00345] 64. The method according to aspect 62, wherein employing a reconfigurable gate array further includes one-time programming the reconfigurable gate array.
- [00346] 65. The method according to aspect 43, further comprising including a network-on-chip fabric in the first semiconductor device.
- [00347] 66. The method according to aspect 65, wherein utilizing the network-on-chip fabric includes employing an array of embedded programmable gate array cores configured to communicate with each other via the network-on-chip fabric.
- [00348] 67. The method according to aspect 65, wherein forming the network-on-chip fabric is accomplished by the first programmable gate array.
- [00349] 68. The method according to aspect 43, wherein utilizing the first programmable gate array includes employing an array of embedded programmable gate array cores, each operatively coupled with a respective interface logic of the plurality of interface logics.
- [00350] 69. The method according to aspect 43, wherein utilizing the first programmable gate array includes employing at least two embedded programmable gate array cores, each operatively coupled with a respective interface logic of the plurality of interface logics.
- [00351] 70. The method according to aspect 43, wherein utilizing the first programmable gate array includes employing an array of embedded programmable gate array cores, each operatively coupled with at least one of the plurality of interface logics.
- [00352] 71. The method according to aspect 43, wherein utilizing the first programmable gate array includes employing an array of embedded programmable gate array cores, each operatively coupled with a single one of the plurality of interface logics.
- [00353] 72. The method according to aspect 43, further comprising: utilizing a second semiconductor device comprising a plurality of memory modules including a first memory module; and coupling the first memory module to a second set of bonds on the surface

of the second semiconductor device, wherein the first set of bonds and the second set of bonds are configured to interface with each other when the first and second semiconductor devices are bonded together.

[00354] 73. The method according to aspect 72, wherein utilizing the first memory port comprises employing the first memory port as a write memory port.

[00355] 74. The method according to aspect 73, wherein employing the first memory port as a write memory port includes configuring the first memory port to write to all of the plurality of memory modules.

[00356] 75. The method according to aspect 73 or 74, further comprising: including a plurality of read memory ports within the first semiconductor device; and operatively coupling each read interface logic of the plurality of read interface logics to a respective one of the plurality of read memory ports.

[00357] 76. The method according to aspect 75, wherein each of the plurality of read interface logics interfaces with a respective memory module of the plurality of memory modules.

[00358] 77. The method according to aspect 72, wherein employing the plurality of memory modules includes utilizing an array of SRAMs.

[00359] 78. The method according to aspect 72, wherein employing the plurality of memory modules includes utilizing an array of non-volatile memories.

[00360] 79. The method according to aspect 72, wherein employing the plurality of memory modules includes using an EEPROM.

[00361] 80. The method according to aspect 72, wherein employing the plurality of memory modules includes using a ROM.

[00362] 81. The method according to aspect 72, wherein employing the plurality of memory modules includes using read-optimized non-volatile memory.

[00363] 82. The method according to aspect 43, wherein utilizing the programmable gate array includes using antifuse, SRAM, or flash memory technology for programming.

[00364] 83. The method according to aspect 43, further comprising connecting programmable gate array cores and employing a mesh topology by the network-on-chip fabric.

[00365] 84. The method according to aspect 43, wherein employing the interface logics includes using voltage level shifters to convert signals of the programmable gate array to and from voltage levels of the first memory port.

What is claimed is:

1. An integrated circuit, comprising:
a first semiconductor device, comprising:
a first programmable gate array; and
a plurality of interface logics including a first interface logic; and
a first memory port connected to a first set of bonds on a surface of the first semiconductor device, the first programmable gate array operatively coupled to the first interface logic to communicate via the first memory port.
2. The integrated circuit according to claim 1, wherein the first interface logic is formed from the first programmable gate array.
3. The integrated circuit according to claim 1, wherein the first interface logic is distributed in a spaced pattern in the first programmable gate array.
4. The integrated circuit according to claim 1, wherein the first semiconductor device includes a plurality of memory ports including the first memory port, wherein the plurality of memory ports forms a spaced pattern on the first semiconductor device.
5. The integrated circuit according to claim 1, further comprising a plurality of sets of bonds including the first set of bonds, wherein the plurality of sets of bonds is distributed in a spaced pattern on the surface of the first semiconductor device.
6. The integrated circuit according to claim 1, the first semiconductor device further comprising a plurality of memory ports including the first memory port, wherein each interface logic of the plurality of interface logics is coupled to a respective memory port of the plurality of memory ports.
7. The integrated circuit according to claim 6, the first semiconductor device further comprising a plurality of programmable gate arrays including the first programmable gate array, wherein each of the plurality of programmable gate arrays is coupled to a respective interface logic of the plurality of interface logics.
8. The integrated circuit according to claim 6, further comprising a plurality of cores, wherein each core of the plurality of cores is coupled to a respective interface logic of the plurality of interface logics.
9. The integrated circuit according to claim 6, further comprising a plurality of sets of bonds on the surface of the first semiconductor device, wherein each memory port of the plurality of memory ports is coupled to a respective set of bonds of the plurality of sets of bonds, wherein the plurality of sets of bonds includes the first set of bonds.

10. The integrated circuit according to claim 1, wherein the first set of bonds are metallic pads.
11. The integrated circuit according to claim 1, wherein the first semiconductor device further comprises:
 - a plurality of programmable gate arrays including the first programmable gate array; and
 - a network-on-chip configured to enable communication among at least two programmable gate arrays of the plurality of programmable gate arrays.
12. The integrated circuit according to claim 1, wherein the first semiconductor device further comprises:
 - at least one network-on-chip communication system connecting at least two processing elements implemented on the first programable gate array, the first programmable gate array operatively coupled to the plurality of interface logics to communicate via the memory port.
13. The integrated circuit according to claim 1, wherein the first memory port is a read port.
14. The integrated circuit according to claim 13, wherein the first memory port is a multi-cycle port.
15. The integrated circuit according to claim 1, wherein the first memory port is a write port.
16. The integrated circuit according to claim 15, wherein the first memory port is a multi-cycle port.
17. The integrated circuit according to claim 1, wherein the first memory port is a read/write port.
18. The integrated circuit according to claim 17, wherein the first memory port is a multi-cycle port.
19. The integrated circuit according to claim 1, wherein the first programmable gate array is a field programmable gate array.
20. The integrated circuit according to claim 1, wherein the first programmable gate array is a reconfigurable gate array.
21. The integrated circuit according to claim 20, wherein the reconfigurable gate array is dynamically programmable.

22. The integrated circuit according to claim 20, wherein the reconfigurable gate array is one-time programmable.
23. The integrated circuit according to claim 1, wherein the first semiconductor device includes a network-on-chip fabric.
24. The integrated circuit according to claim 23, wherein the programmable gate array includes an array of embedded programmable gate array cores configured to communicate with each other via the network-on-chip fabric.
25. The integrated circuit according to claim 23, wherein the network-on-chip fabric is formed by the first programmable gate array.
26. The integrated circuit according to claim 1, wherein the first programmable gate array includes an array of embedded programmable gate array cores, wherein each of the array of embedded programmable gate array cores is operatively coupled with a respective interface logic of the plurality of interface logics.
27. The integrated circuit according to claim 1, wherein the first programmable gate array includes at least two embedded programmable gate array cores, wherein each of the at least two embedded programmable gate array cores is operatively coupled with a respective interface logic of the plurality of interface logics.
28. The integrated circuit according to claim 1, wherein the first programmable gate array includes an array of embedded programmable gate array cores, wherein each of the array of embedded programmable gate array cores is operatively coupled with at least one of the plurality of interface logics.
29. The integrated circuit according to claim 1, wherein the first programmable gate array includes an array of embedded programmable gate array cores, wherein each of the array of embedded programmable gate array cores is operatively coupled with a single one of the plurality of interface logics.
30. The integrated circuit according to claim 1, further comprising:
a second semiconductor device comprising:
a plurality of memory modules including a first memory module, the first memory module coupled a second set of bonds on the surface of the second semiconductor device, wherein the first set of bonds and the second set of bonds are configured to interface with each other when the first and second semiconductor devices are bonded together.
31. A method of operating an integrated circuit, the method comprising:

utilizing a first semiconductor device that includes a first programmable gate array;
employing a plurality of interface logics, including a first interface logic;
connecting a first memory port to a first set of bonds on a surface of the first semiconductor device; and
operatively coupling the first programmable gate array to the first interface logic to enable communication via the first memory port.

32. The method according to claim 31, wherein employing the first interface logic comprises forming the first interface logic from the first programmable gate array.

33. The method according to claim 31, wherein employing the first interface logic includes distributing the first interface logic in a spaced pattern within the first programmable gate array.

34. The method according to claim 31, further comprising including a plurality of memory ports, such as the first memory port, wherein forming a spaced pattern on the first semiconductor device with the plurality of memory ports.

35. The method according to claim 31, further comprising distributing a plurality of sets of bonds, including the first set of bonds, in a spaced pattern on the surface of the first semiconductor device.