(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
8 September 2017 (08.09.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/151223 A1

(51) International Patent Classification:
G06F 12/02 (2006.01) *G06F 11/22* (2006.01)

(21) International Application Number:
PCT/US2017/012857

(22) International Filing Date:
10 January 2017 (10.01.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/057,685 1 March 2016 (01.03.2016) US

(71) Applicant: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95054 (US).

(72) Inventors: **PAPPU, Lakshminarayana**; 1480 Leonard Court, Folsom, California 95630 (US). **CALLAHAN, Timothy J.**; 2501 NW 229th Avenue, Mailstop: RA1-331, Hillsboro, Oregon 97124 (US). **LEVY, Tomer**; 10 Uri Zvi Greenberg, 6937916 Tel Aviv (IL).

(74) Agents: **MALLIE, Michael J.** et al.; BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP, 1279 Oakmead Parkway, Sunnyvale, California 94085 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) Title: LOW COST INBUILT DETERMINISTIC TESTER FOR SOC TESTING

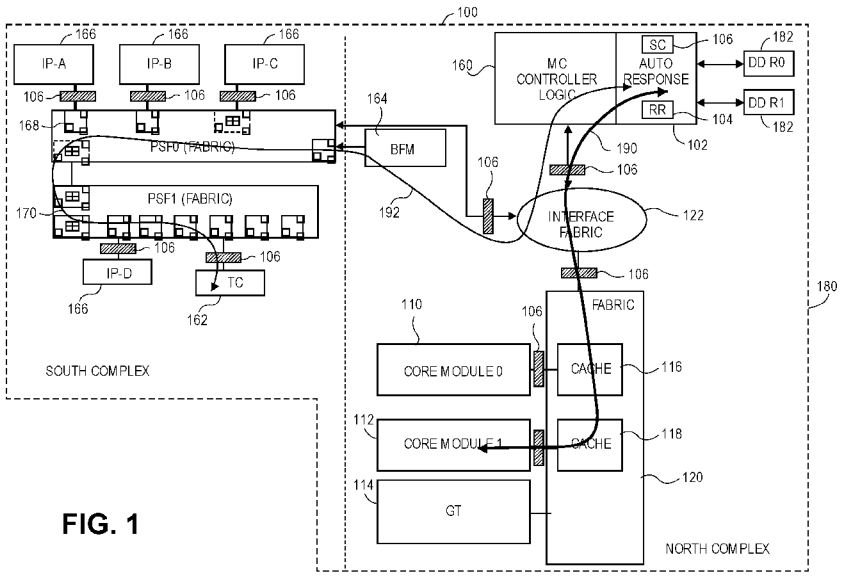


FIG. 1

(57) Abstract: A method and system for high speed on chip testing for quality assurance. A multi-core system on a chip has a plurality of processing cores. The cores act as transaction agents with an auto-response unit fabricated on the chip at a chip boundary, the auto-response unit to provide a deterministic return value based on a logical address of a received read request.

WO 2017/151223 A1

LOW COST INBUILT DETERMINISTIC TESTER FOR SOC TESTING

FIELD

Embodiments of the invention relate to system on a chip testing. More specifically, embodiments relate to use of an auto-response unit fabricated on a chip to perform on-chip testing.

BACKGROUND

Testing of systems on a chip is essentially for quality assurance. As systems have become more complex and faster, effective testing has become more difficult and more expensive. Generally, testing involves using external testers to analyze data exchanges with an external memory. Unfortunately, because of the increasing disparity between tester speeds and double data rate (DDR) memory exchanges, existing techniques fail to provide the desired quality assurance.

Stretch mode, one traditional technique in which the data strobe time is stretched with a ratio equivalent to the speed ratio of the DDR to the tester speed, become entirely impractical as speeds increase. The testing interface has a much slower speed than is desired, often with a stretch factor of four or more. As the stretch factor increases, so does the command bubble duration. The logic to handle the DDR input/output (I/O) is both complex and intrusive. Moreover, for stretch mode to work for design for manufacturability (DFX) tests, enormous tuning of the tester is required. Such tuning has been found impractical to real-world applications.

A second historical testing option is write data always response (WDAR) mode. This mechanism requires 128 cache lines of additional memory for data. Often, this amount of memory does not exist in products to be tested. Moreover, WDAR has a huge time impact resulting from the need to fill the memory with known data before starting the test. Finally, it is not possible to calculate the expected response on the fly. Therefore, the WDAR test writer must perform pre-silicon simulation and compare with post-silicon results. Often, mismatches occur between the signatures calculated from the simulation and those that exist in the actual silicon.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention are illustrated by way of example and not by way of limitation in the figures of the accompanying drawings in which like references indicate similar elements. It should be noted that different references to "an" or "one" embodiment in this disclosure are not necessarily to the same embodiment, and such references mean at least one.

Figure 1 is a block diagram a system according to one embodiment of the invention.

Figure 2 is a more detailed block diagram of the auto-response unit according to one embodiment of the invention.

Figure 3 is a flow diagram of operation of a system according to one embodiment of the invention.

Figure 4 is a block diagram of a system that may incorporate one embodiment of the invention.

DETAILED DESCRIPTION

Figure 1 is a block diagram a system according to one embodiment of the invention. Auto-response unit 102 is fabricated on a chip 100 having a chip boundary 180. Auto-response unit 102 is preferably fabricated near the chip boundary 180 to provide effective testing for chip devices further from the boundary than itself. As explained below, the auto responder effectively tests all components of the chip in the signal path of the read and write transactions issued during a test regimen. In one embodiment, auto-response unit 102 is coupled between a memory controller 160 and the memory ports 182 through which the memory controller 160 accesses off-chip memory (not shown). In this arrangement the auto response unit can provide effective testing of the memory controller 160 in addition to other chip components.

Auto-response unit 102 includes a signature collector 106 that non-deterministically collects signatures of transactions received at the auto-response unit 102. In some embodiments, signature collector 106 may collect only signatures of write transactions. In other embodiments, signature collector 106 may collect signatures of both read and write transactions. Auto-response unit 102 also includes a read-response module 104 to provide an automatic deterministic response to read requests received by the auto-response unit 102. Auto-response unit 102 is discussed more detail below with reference to Figure 2.

Chip 100 has fabricated thereon multiple core modules 110, 112. While two core modules are shown, an arbitrarily large number of core modules may be fabricated on a chip. As such, an arbitrary number of core modules are deemed within the scope and contemplation of various embodiments of the invention. Each core module is associated with a cache memory. Here, core module 110 is associated with cache 116, and core module 112 is associated with cache 118. If more cores are present correspondingly more caches will be present. A Graphic technologies module 114 may also be fabricated on chip 100. Graphic technology module 114 may include a graphics processing unit (GPU) and any other components desired for graphics support within the system. In some embodiments an analogous auto response unit may be use to test the GPU subsystem by fabrication it at the chip boundary between a graphics controller (not shown) and the graphics input/output interface (not shown).

Chip 100 provides a fabric 120 through which the core modules 110, 112 communicate and can send read and write requests via an interface fabric 122 to memory controller 160. A bus functional module 164 is provided to facilitate communication between the north complex of chip 100 and the south complex. The south complex may include a plurality of fabrics 168 and 170 that provide network connectivity through IP ports 166. A test controller 162 may also be fabricated in the south complex. Test controller 162 can act as a transaction agent, sending read and write transactions through the local fabrics 168, 170 and the interface fabric 122 to memory controller 160.

The memory controller 160 receives a transaction either from one of the core modules, e.g., transaction 190, or from the test controller, e.g., transaction 192, and translates the logical address specified in the transaction to a physical address, which is then forwarded through the auto-response unit 102. If the auto-response unit 102 is not in a test mode, it takes no action and the transaction is conducted in the normal manner of the DDR interface 182. But, if the auto-response unit is in a test mode, it will handle the transaction as discussed more fully with reference to Figure 2. Notably, signature collectors 106 can be distributed at different branch points around the system to improve localization of faults identified in the course of a test.

Figure 2 is a more detailed block diagram of the auto-response unit according to one embodiment of the invention. As noted with reference to Figure 1, auto-response unit 102 includes a signature collector 106. In one embodiment, signature collector 106 uses AND logic with a global trigger to ensure valid data has passed to a compression unit 222. A summer 224 adds compressed data is to any previously accumulated signatures creating a running non-deterministic checksum. An accumulator 228 feeds back the result to the summer for a subsequent transaction while latching the present value and also storing it in a register storage 230. This construction results in a non-deterministic signature accumulation, which is agnostic to the order of the transaction's arrival, high speed, and provides an easy checksum for comparison with an expect value for an easy pass fail determination. When a test is complete, the testing transaction agent need only read register file 230 and compare it with its expected value to determine whether the test has passed or failed. In one embodiment, signature collection is only performed for write transactions. In other embodiments, signature collection is performed on both read and write transactions. Generally, the physical address for the transaction forms part of the checksum and during the posttest comparison the transactions agent use an expected value based on the physical rather than the logical addresses. The use of the physical address is driven by the desire auto response unit as deep in the signal path as possible to the greatest coverage during testing. The logical to physical address conversion by the

memory controller occurs quite close to the chip boundary, by placing the auto response unit later in the path, excellent chip coverage is achieved.

Auto-response unit 102 includes read-response unit 104, which receives read requests from a memory controller when the system is under test. In one embodiment, read-response unit 104 includes an address conversion module 240 that converts the physical address supplied by the memory controller to a logical address that would have been sent by the transaction agent. An example of a logical address conversion is shown in Table 1 below.

DRAM to system address conversion assuming DDR4 16GB, x8 full occupancy (all ranks in all channels), no interleaving and no remap	
DRAM Address	System Address
Row[16:0]	`36:20
Bank[1]	19
Rank[1]	18
Bank[0]	17
Rank[0]	16
Group[1]	15
Column[9:3]	`14:8
Group[0]	7
Channel	6
Column[2:0]	`5:3

TABLE 1

This conversion is based on a system using DDR 4 16GB X 8 full occupancy (all ranks and all channels) with no interleaving, no remap and no error correction codes. Appropriate mapping may be employed for whatever system memory population is intended. This conversion is required because the memory controller convert the logical address receive with the transaction to a physical address and since the auto response unit is after the memory controller in the signal path, it only receives the physical address. Some embodiments of the invention could place the auto response unit before the memory controller in the signal path. Those embodiments would avoid the need for physical to logical address conversion in the auto response unit, but would not provide effective testing of the memory controller.

The converted address (that is, the logical address) is then fed into an encoder 242, which applies a set of equations to generate a deterministic value corresponding to that logical address. In one embodiment, the encoder 242 generates 128 bits of data response to be returned to the transaction agent. Each data bit generation involves a unique logic equation. For example:

Data_response[0] = f(address bits 18, 25, 11)

Data_response[1] = f(address bits 12, 23, 11)

Data_response[2] = f(address bits 19, 37, 11)

Data_response[3] = f(address bits 30, 15, 9)

Data_response[4] = f(address bits 13, 25, 7)

Data_response[5] = f(address bits 28, 16, 10)

Data_response[6] = f(address bits 20, 6, 11)

Data_response[7] = f(address bits 33, 19, 10)

Data_response[8] = f(address bits 16, 32, 11)

Data_response[9] = f(address bits 21, 38, 9)

Data_response[10] = f(address bits 36, 14, 6)

Data_response[11] = f(address bits 22, 18, 10)

Data_response[12] = f(address bits 29, 13, 8)

Data_response[13] = f(address bits 20, 29, 6)

Data_response[14] = f(address bits 23, 17, 7)

Data_response[15] = f(address bits 17, 33, 6)

Data_response[16] = f(address bits 12, 26, 9)

Data_response[17] = f(address bits 26, 11, 6)

Data_response[18] = f(address bits 21, 38, 7)

Data_response[19] = f(address bits 15, 28, 10)

Data_response[20] = f(address bits 31, 16, 10)

Data_response[21] = f(address bits 26, 21, 6)

Data_response[22] = f(address bits 15, 30, 7)

Data_response[23] = f(address bits 35, 20, 7)

Data_response[24] = f(address bits 18, 27, 9)

Data_response[25] = f(address bits 14, 26, 10)

Data_response[26] = f(address bits 15, 13, 8)

Data_response[27] = f(address bits 35, 12, 8)

Data_response[28] = f(address bits 14, 24, 9)

Data_response[29] = f(address bits 17, 28, 9)

Data_response[30] = f(address bits 34, 9, 10)

Data_response[31] = f(address bits 18, 22, 6)

Data_response[32] = f(address bits 28, 17, 8)

Data_response[33] = f(address bits 36, 15, 11)

Data_response[34] = f(address bits 27, 14, 10)

Data_response[35] = f(address bits 32, 9, 10)

Data_response[36] = f(address bits 19, 22, 9)

Data_response[37] = f(address bits 18, 38, 7)

Data_response[38] = f(address bits 26, 18, 11)

Data_response[39] = f(address bits 31, 19, 6)

Data_response[40] = f(address bits 14, 37, 8)

Data_response[41] = f(address bits 29, 12, 7)

Data_response[42] = f(address bits 38, 13, 9)

Data_response[43] = f(address bits 18, 30, 6)

Data_response[44] = f(address bits 13, 29, 8)

Data_response[45] = f(address bits 28, 16, 8)

Data_response[46] = f(address bits 16, 28, 8)

Data_response[47] = f(address bits 25, 21, 11)

Data_response[48] = f(address bits 21, 13, 6)

Data_response[49] = f(address bits 18, 25, 7)

Data_response[50] = f(address bits 23, 21, 8)

Data_response[51] = f(address bits 31, 20, 9)

Data_response[52] = f(address bits 15, 32, 10)

Data_response[53] = f(address bits 17, 36, 11)

Data_response[54] = f(address bits 20, 35, 8)

Data_response[55] = f(address bits 14, 22, 11)

Data_response[56] = f(address bits 21, 34, 9)

Data_response[57] = f(address bits 32, 19, 10)

Data_response[58] = f(address bits 38, 20, 10)

Data_response[59] = f(address bits 24, 16, 8)

Data_response[60] = f(address bits 37, 13, 10)

Data_response[61] = f(address bits 29, 6, 11)

Data_response[62] = f(address bits 15, 31, 6)

Data_response[63] = f(address bits 17, 33, 9)

Data_response[64] = f(address bits 22, 18, 10)

Data_response[65] = f(address bits 14, 32, 8)

Data_response[66] = f(address bits 32, 16, 7)

Data_response[67] = f(address bits 37, 11, 7)

Data_response[68] = f(address bits 31, 7, 8)

Data_response[69] = f(address bits 17, 34, 9)

Data_response[70] = f(address bits 22, 20, 6)

Data_response[71] = f(address bits 31, 15, 11)

Data_response[72] = f(address bits 13, 36, 11)

Data_response[73] = f(address bits 25, 18, 7)

Data_response[74] = f(address bits 14, 28, 6)

Data_response[75] = f(address bits 21, 26, 10)

Data_response[76] = f(address bits 27, 13, 8)

Data_response[77] = f(address bits 24, 12, 6)

Data_response[78] = f(address bits 22, 16, 9)

Data_response[79] = f(address bits 29, 7, 8)

Data_response[80] = f(address bits 20, 30, 9)

Data_response[81] = f(address bits 23, 14, 6)

Data_response[82] = f(address bits 34, 17, 11)

Data_response[83] = f(address bits 38, 19, 8)

Data_response[84] = f(address bits 27, 21, 10)

Data_response[85] = f(address bits 38, 12, 6)

Data_response[86] = f(address bits 14, 33, 8)

Data_response[87] = f(address bits 27, 21, 7)

Data_response[88] = f(address bits 15, 37, 7)

Data_response[89] = f(address bits 12, 29, 11)

Data_response[90] = f(address bits 35, 6, 8)

Data_response[91] = f(address bits 18, 24, 10)

Data_response[92] = f(address bits 35, 20, 7)

Data_response[93] = f(address bits 33, 16, 9)

Data_response[94] = f(address bits 19, 21, 10)

Data_response[95] = f(address bits 15, 33, 10)

Data_response[96] = f(address bits 15, 23, 11)

Data_response[97] = f(address bits 12, 24, 10)

Data_response[98] = f(address bits 16, 27, 8)

Data_response[99] = f(address bits 19, 17, 11)

Data_response[100] = f(address bits 17, 33, 8)

Data_response[101] = f(address bits 27, 8, 7)

Data_response[102] = f(address bits 37, 7, 8)

Data_response[103] = f(address bits 38, 18, 6)

Data_response[104] = f(address bits 18, 37, 8)

Data_response[105] = f(address bits 12, 14, 7)

Data_response[106] = f(address bits 34, 17, 11)

Data_response[107] = f(address bits 21, 34, 6)

Data_response[108] = f(address bits 36, 20, 10)

Data_response[109] = f(address bits 32, 19, 6)

Data_response[110] = f(address bits 23, 19, 9)

Data_response[111] = f(address bits 18, 24, 11)

Data_response[112] = f(address bits 34, 14, 9)

Data_response[113] = f(address bits 35, 15, 6)

Data_response[114] = f(address bits 25, 18, 10)

Data_response[115] = f(address bits 13, 22, 6)

Data_response[116] = f(address bits 16, 37, 10)

Data_response[117] = f(address bits 20, 29, 8)

Data_response[118] = f(address bits 19, 20, 8)

Data_response[119] = f(address bits 26, 13, 10)

Data_response[120] = f(address bits 35, 12, 7)

Data_response[121] = f(address bits 16, 38, 10)

Data_response[122] = f(address bits 14, 31, 11)

Data_response[123] = f(address bits 28, 11, 9)

Data_response[124] = f(address bits 36, 21, 11)

Data_response[125] = f(address bits 27, 13, 10)

Data_response[126] = f(address bits 17, 30, 9)

Data_response[127] = f(address bits 30, 15, 6)

In one embodiment, function f could be an XOR function. However, more complex equations may alternatively be used. The above example is only one possible function set; generally, it is desirable to choose the function f to minimize the aliasing between generated data bits. Simulations may be conducted to ensure sufficient deterministic randomness in the return data. Any set of functions that provide suitable deterministic randomness could be used. The deterministic value generated by the encoder 242 may then be fed into a masking unit 246, which may apply a mask to change or otherwise ensure that one or more bits of the deterministic value resolve to a known value. This may be important where a particular bit has been found to be the source of errors and it is desirable to test if errors exist elsewhere in the system. A register file 248 may hold various masks for this purpose. It is of course possible that the masking logic 246

will leave the original deterministic value unchanged. The masked or unmasked value is then returned to the transaction agent. The auto-response unit 102 includes a mode selector 208 which ensures that the deterministic value is fed back only when the system is in test mode and real data, that is, functional data from the memory is returned from read requests when test mode is not selected.

Referring again to Figure 1, after a series of transactions e.g. 190, 192 conducted as part of a test, the transaction agent 110, 162 respectively, verifies the check sum of the collected signatures and confirms that the read response values have match expected values. It both are true, the logic in the signal path of the transaction is deem to be validated and the chip can be accepted as meeting specifications.

Figure 3 is a flow diagram of operation in a system according to one embodiment of the invention. At block 302, a test routine is loaded into the cache of a transaction agent. This can be done with low speed low cost test equipment. At block 304, the agent directs the auto-response unit 102 to transition into test mode. The transaction agent configures any desired masking in the auto-response unit 102 at block 306. At block 308, the transaction agent sends a transaction directed towards the off chip memory. At block 310, a determination is made whether the transaction is a read. If a read has been received, the auto-response unit 102 converts the physical address to a logical address (that is, the physical address received from the memory controller into a logical address that is used by the transaction agent). At block 314, the auto-response unit 102 encodes the logical address to yield a deterministic value. A determination is then made if masking is desired at block 316. If masking is desired, a specified mask is applied at block 318. The resultant value, with or without masking, is then returned to the transaction agent at block 320.

Generally, the transaction agent maintains a precomputed expected value for the transaction. A determination is made by the transaction agent whether the returned value is equal to the expected value at block 322. If it is not, the read test for that logical address is deemed to have failed at block 324. If at block 310, the transaction is not a read transaction, a determination is made at block 326 if a write transaction has been received. If a write transaction has been received, the signature of that transaction is accumulated as a checksum at block 328. Then, at block 330, a determination is made whether the test transactions are complete. If they are not, further transactions are sent from the agent, and the process continues. If all transactions are complete, the checksum of the accumulated values is acquired and compared with the expected value at block 332. A determination is made whether those values match at block 334. If the checksum of accumulated values matches the expected value, the

write test is deemed to pass. Then, a determination is made if there have been any read test failures at block 236. If there have been no read test failures, the test passes at block 240. If not, the test fails at block 238. The test will also fail at block 238 if there is no match of the checksum from the accumulation of signatures.

Figure 4 is a block diagram of a system according to one embodiment of the present invention. The computer system 1000 (also referred to as the electronic system 1000) as depicted can employ the auto response testing, according to any of the several disclosed embodiments and their equivalents as set forth in this disclosure. The computer system 1000 may be a mobile device such as a netbook computer. The computer system 1000 may be a mobile device such as a wireless smart phone. The computer system 1000 may be a desktop computer. The computer system 1000 may be a hand-held reader. The computer system 1000 may be a server system. The computer system 1000 may be a supercomputer or high-performance computing system.

In an embodiment, the electronic system 1000 is a computer system that includes a system bus 1020 to electrically couple the various components of the electronic system 1000. The system bus 1020 is a single bus or any combination of busses according to various embodiments. The electronic system 1000 includes a voltage source 1030 that provides power to the integrated circuit 1010. In some embodiments, the voltage source 1030 supplies current to the integrated circuit 1010 through the system bus 1020.

The integrated circuit 1010 is electrically coupled to the system bus 1020 and includes any circuit, or combination of circuits according to an embodiment. In an embodiment, the integrated circuit 1010 includes a processor 1012 that can be of any type. As used herein, the processor 1012 may mean any type of circuit such as, but not limited to, a microprocessor, a microcontroller, a graphics processor, a digital signal processor, or another processor. In an embodiment, the processor 1012 includes, or is coupled with, a SIP as disclosed herein. In an embodiment, SRAM embodiments are found in memory caches of the processor. Other types of circuits that can be included in the integrated circuit 1010 are a custom circuit or an application-specific integrated circuit (ASIC), such as a communications circuit 1014 for use in wireless devices such as cellular telephones, smart phones, pagers, portable computers, two-way radios, and similar electronic systems, or a communications circuit for servers. In an embodiment, the integrated circuit 1010 includes on-die memory 1016 such as static random-access memory (SRAM). In an embodiment, the integrated circuit 1010 includes embedded on-die memory 1016 such as embedded dynamic random-access memory (eDRAM).

In an embodiment, the integrated circuit 1010 is complemented with a subsequent integrated circuit 1011. Useful embodiments include a dual processor 1013 and a dual communications circuit 1015 and dual on-die memory 1017 such as SRAM. In an embodiment, the dual integrated circuit 1010 includes embedded on-die memory 1017 such as eDRAM.

In an embodiment, the electronic system 1000 also includes an external memory 1040 that in turn may include one or more memory elements suitable to the particular application, such as a main memory 1042 in the form of RAM, one or more hard drives 1044, and/or one or more drives that handle removable media 1046, such as diskettes, compact disks (CDs), digital variable disks (DVDs), flash memory drives, and other removable media known in the art. The external memory 1040 may also be embedded memory 1048 such as the first die in a die stack, according to an embodiment.

In an embodiment, the electronic system 1000 also includes a display device 1050, an audio output 1060. In an embodiment, the electronic system 1000 includes an input device such as a controller 1070 that may be a keyboard, mouse, trackball, game controller, microphone, voice-recognition device, or any other input device that inputs information into the electronic system 1000. In an embodiment, an input device 1070 is a camera. In an embodiment, an input device 1070 is a digital sound recorder. In an embodiment, an input device 1070 is a camera and a digital sound recorder. In an embodiment, an input device is a touch screen that may be part of display device 1050.

As shown herein, the integrated circuit 1010 can be implemented in a number of different embodiments, including a system on a chip (SOC), according to any of the several disclosed embodiments and their equivalents, an electronic system, a computer system, one or more methods of fabricating an integrated circuit, and one or more methods of fabricating an electronic assembly that includes a SOC, according to any of the several disclosed embodiments as set forth herein in the various embodiments and their art-recognized equivalents. The elements, materials, geometries, dimensions, and sequence of operations can all be varied to suit particular I/O coupling requirements including array contact count, array contact configuration for a microelectronic die embedded in a processor mounting substrate according to any of the several disclosed SIP embodiments and their equivalents. A foundation substrate may be included, as represented by the dashed line of Figure 4. Passive devices may also be included, as is also depicted in Figure 4.

The following examples pertain to further embodiments. The various features of the different embodiments may be variously combined with some features included and others excluded to suit a variety of different applications. Some embodiments pertain multi-core system

on a chip having at least a first processing core and a second processing fabricated on a chip.

An auto-response unit is also fabricated on the chip at a chip boundary. The auto-response unit is to provide a deterministic return value based on a logical address of a received read request.

In further embodiments, the multi-core system has a memory controller fabricated on the chip and the auto-response unit resides in a signal path between the memory controller and the chip boundary.

In further embodiments, the auto-response unit includes a converter to convert a physical address to a logical address and an encoder to deterministically encode the return value based on the logical address.

In further embodiments, multi-core system has an accumulator to non-deterministically accumulate signatures of transactions received by the auto-response unit.

In further embodiments, multi-core system has a switch fabric on the chip and a controller coupled to the switch fabric. The controller acts as a transaction agent to which the auto-response unit is to respond to a read transaction with a deterministic return value.

In further embodiments, the auto-response unit has a mode selector to select between a test mode and a functional mode. The auto-response unit provides a deterministic return value only when the test mode is selected.

In further embodiments, the auto-response unit includes masking logic to mask a bit of the return value responsive to an instruction from a transaction agent.

Some embodiments pertain to a method of testing an integrated circuit by transitioning an auto-response module into a test mode. Once in test mode a read transaction is sent from a transaction agent. The read transaction is intercepted in the auto-response module. The auto response module sends a deterministic response to the transaction agent based on the read transaction.

In further embodiments, the method includes sending a write transaction from the processing agent, intercepting the write transaction in the auto-response module; and accumulating a non-deterministic signature of the write transaction.

In further embodiments, the method includes translating a physical address of the read transaction into a logical address of the read transaction. A set of equations are applied to the logical address to generate the deterministic response.

In further embodiments, the auto-response module resides between a memory controller and a chip boundary.

In further embodiments, the method includes applying a mask to the deterministic response to force one or more bits of the deterministic response to a known value.

Some embodiments pertain to an on-chip test component with a mode selector to select between a test mode and a functional mode. The test component also has an encoder to encode a logical address into a deterministic return value and masking logic to mask one or more bits of the return value responsive to a signal from a transaction agent.

In further embodiments, the on-chip test component has a signature collector to store signatures of write transactions received at the test component.

In further embodiments, the masking logic comprises a register file to retain one or more masks.

In further embodiments, the on-chip test component includes response logic to send the deterministic return value to an agent that is the source of the read request.

In further embodiments, the signatures collected are non-deterministic.

In further embodiments, the signature collector includes a data compressor, an accumulator, and a register file.

Some embodiments pertain to a chip having on-chip test functionality including a transaction agent fabricated on the chip. The chip also includes means for accumulating non-deterministic signatures of write transactions from the transaction agent and means for automatically responding with a deterministic return value when a read transaction is received from the transaction agent.

In further embodiments, the chip includes means for selecting between a test mode and a functional mode.

In further embodiments, the means for automatically responding has means for encoding a logical address into a deterministic value and means for masking bits of the deterministic value.

While embodiments of the invention are discussed above in the context of flow diagrams reflecting a particular linear order, this is for convenience only. In some cases, various operations may be performed in a different order than shown or various operations may occur in parallel. It should also be recognized that some operations described with respect to one embodiment may be advantageously incorporated into another embodiment. Such incorporation is expressly contemplated.

In the foregoing specification, the invention has been described with reference to the specific embodiments thereof. It will, however, be evident that various modifications and changes can be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A multi-core system on a chip comprising:

a first processing core fabricated on a chip;

a second processing core fabricated on the chip; and

an auto-response unit fabricated on the chip at a chip boundary, the auto-response unit to provide a deterministic return value based on a logical address of a received read request.

2. The multi-core system of claim 1, further comprising:

a memory controller fabricated on the chip,

wherein the auto-response unit resides in a signal path between the memory controller and the chip boundary.

3. The multi-core system of claims 1 or 2, wherein the auto-response unit comprises:

a converter to convert a physical address to a logical address; and

an encoder to deterministically encode the return value based on the logical address.

4. The multi-core system of claims 1 or 2, further comprising:

an accumulator to non-deterministically accumulate signatures of transactions received by the auto-response unit.

5. The multi-core system of claim 1 or 2, further comprising:

a switch fabric on the chip;

a controller coupled to the switch fabric;

wherein the auto-response unit is to respond to a read transaction from the controller with a deterministic return value.

6. The multi-core system of claims 1 or 2, wherein the auto-response unit comprises:

a mode selector to select between a test mode and a functional mode;

wherein the auto-response unit provides a deterministic return value only when the test mode is selected.

7. The multi-core system of claim 1 or 2, wherein the auto-response unit further comprises:

masking logic to mask a bit of the return value responsive to an instruction from a transaction agent.

8. A method of testing an integrated circuit, comprising:

transitioning an auto-response module into a test mode;

sending a read transaction from a transaction agent;

intercepting the read transaction in the auto-response module; and

sending a deterministic response to the transaction agent based on the read transaction.

9. The method of claim 8, further comprising:

sending a write transaction from the processing agent;

intercepting the write transaction in the auto-response module; and

accumulating a non-deterministic signature of the write transaction.

10. The method of claims 8 or 9, further comprising:

translating a physical address of the read transaction into a logical address of the read transaction; and

applying a set of equations to the logical address to generate the deterministic response.

11. The method of claims 8 or 9, wherein the auto-response module resides between a memory controller and a chip boundary.

12. The method of claim 10, further comprising:

applying a mask to the deterministic response to force one or more bits of the deterministic response to a known value.

13. An on-chip test component, comprising:

a mode selector to select between a test mode and a functional mode;

an encoder to encode a logical address into a deterministic return value; and

masking logic to mask one or more bits of the return value responsive to a signal from a transaction agent.

14. The on-chip test component of claim 13, further comprising a signature collector to store signatures of write transactions received at the test component.

15. The on-chip test component of claims 13 or 14, wherein the masking logic comprises a register file to retain one or more masks.

16. The on-chip test component of claims 13 or 14, further comprising:

response logic to send the deterministic return value to an agent that is the source of the read request.

17. The on-chip test component of claims 13 or 14, wherein the signatures collected are non-deterministic.

18. The on-chip test component of claims 13 or 14, wherein the signature collector comprises:

a data compressor;

an accumulator; and

a register file.

19. A chip having on-chip test functionality comprising:

a transaction agent fabricated on the chip;

means for accumulating non-deterministic signatures of write transactions from the transaction agent; and

means for automatically responding with a deterministic return value when a read transaction is received from the transaction agent.

20. The chip of claim 19, further comprising:

means for selecting between a test mode and a functional mode.

21. The chip of claims 19 or 20, wherein the means for automatically responding comprises:

means for encoding a logical address into a deterministic value; and

means for masking bits of the deterministic value.

22. The chip of claims 19 or 20, further comprising:

a switch fabric on the chip; and

a controller coupled to the switch fabric that act as the transaction agent.

23. The chip of claims 19 or 20 wherein the chip comprises:

a plurality of processing cores any of which can function as the transaction agent.

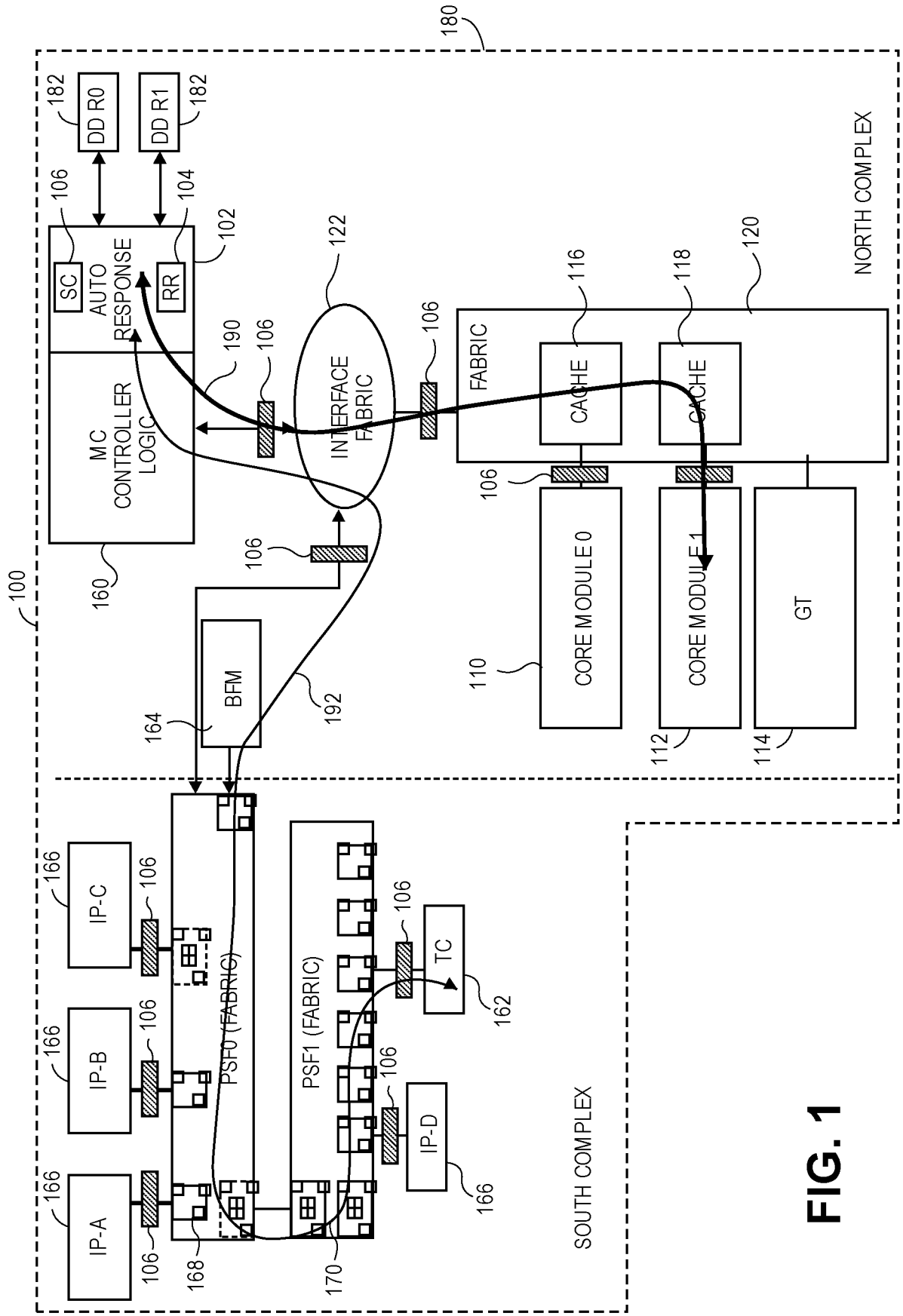

180
NORTH COMPLEX
182
DD R0
182
DD R1
106
104
102
SC
AUTO RESPONSE
RR
MC CONTROLLER LOGIC
160
190
106
122
INTERFACE FABRIC
106
FABRIC
116
118
120
CACHE
CACHE
106
106
106
CORE MODULE 0
CORE MODULE 1
GT
110
112
114
100
BFM
164
192
SOUTH COMPLEX
166
IP-C
106
166
IP-B
106
166
IP-A
106
PSF0 (FABRIC)
PSF1 (FABRIC)
106
TC
162
106
IP-D
166
168
170

FIG. 1

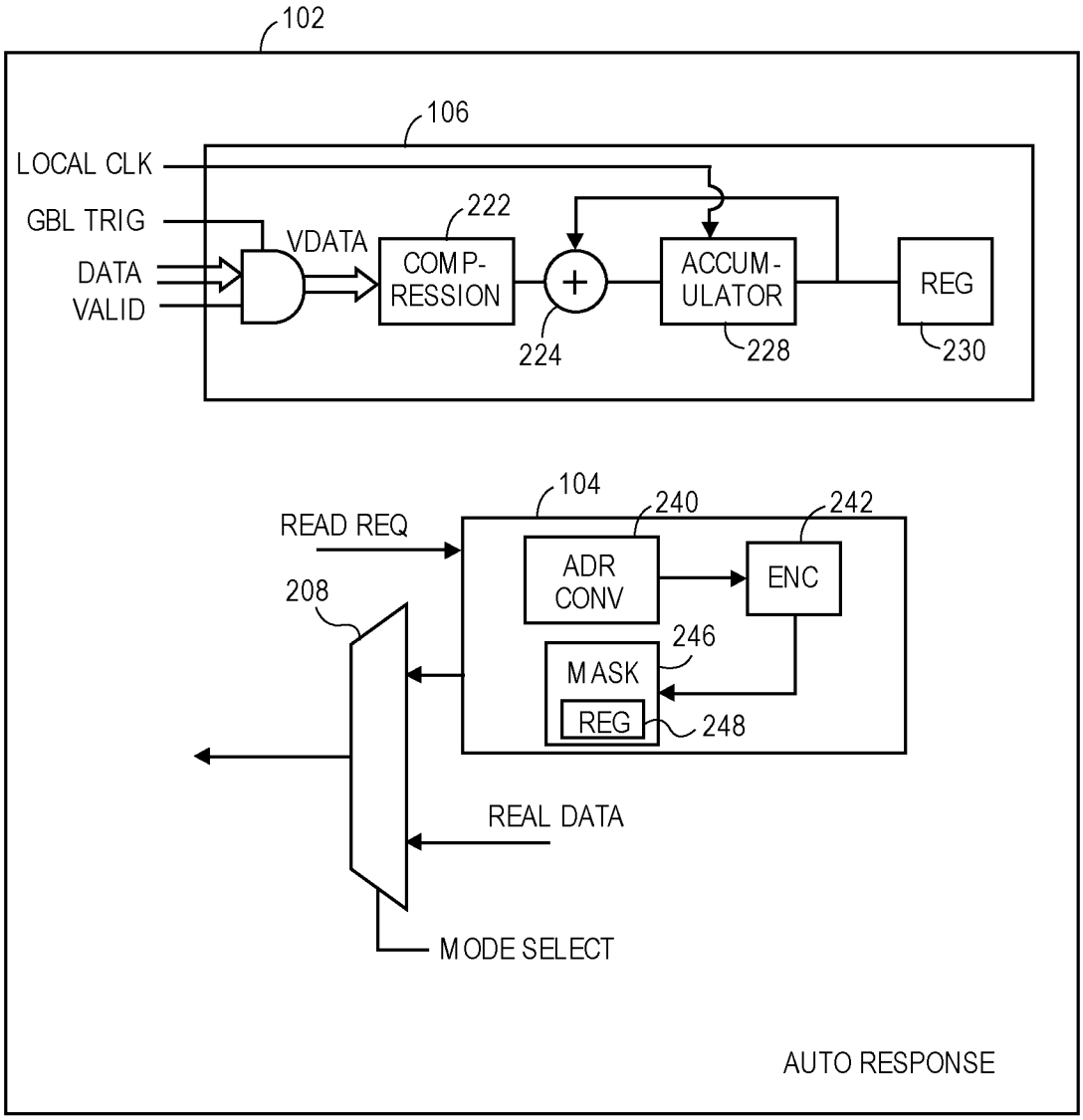

102
106
LOCAL CLK
GBL TRIG
DATA
VALID
VDATA
222
COMP-
RESSION
224
ACCUM-
ULATOR
228
REG
230
104
240
242
READ REQ
ADR
CONV
ENC
208
246
MASK
REG
248
REAL DATA
MODE SELECT
AUTO RESPONSE

FIG. 2

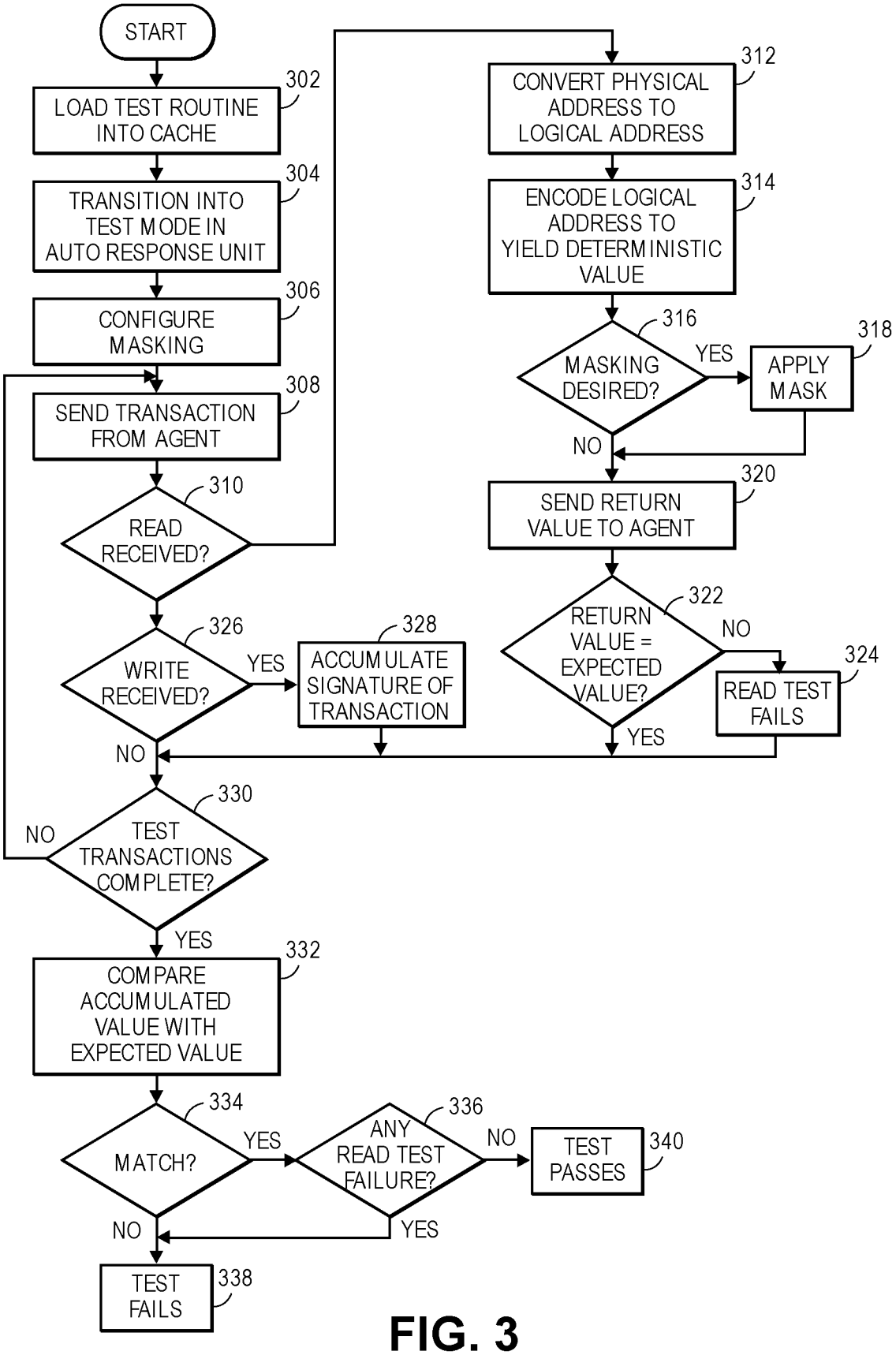

START
LOAD TEST ROUTINE INTO CACHE
302
TRANSITION INTO TEST MODE IN AUTO RESPONSE UNIT
304
CONFIGURE MASKING
306
SEND TRANSACTION FROM AGENT
308
READ RECEIVED?
310
WRITE RECEIVED?
326
YES
ACCUMULATE SIGNATURE OF TRANSACTION
328
NO
TEST TRANSACTIONS COMPLETE?
330
NO
YES
COMPARE ACCUMULATED VALUE WITH EXPECTED VALUE
332
MATCH?
334
YES
ANY READ TEST FAILURE?
336
NO
TEST PASSES
340
YES
NO
TEST FAILS
338
CONVERT PHYSICAL ADDRESS TO LOGICAL ADDRESS
312
ENCODE LOGICAL ADDRESS TO YIELD DETERMINISTIC VALUE
314
MASKING DESIRED?
316
YES
APPLY MASK
318
NO
SEND RETURN VALUE TO AGENT
320
RETURN VALUE = EXPECTED VALUE?
322
NO
READ TEST FAILS
324
YES

FIG. 3

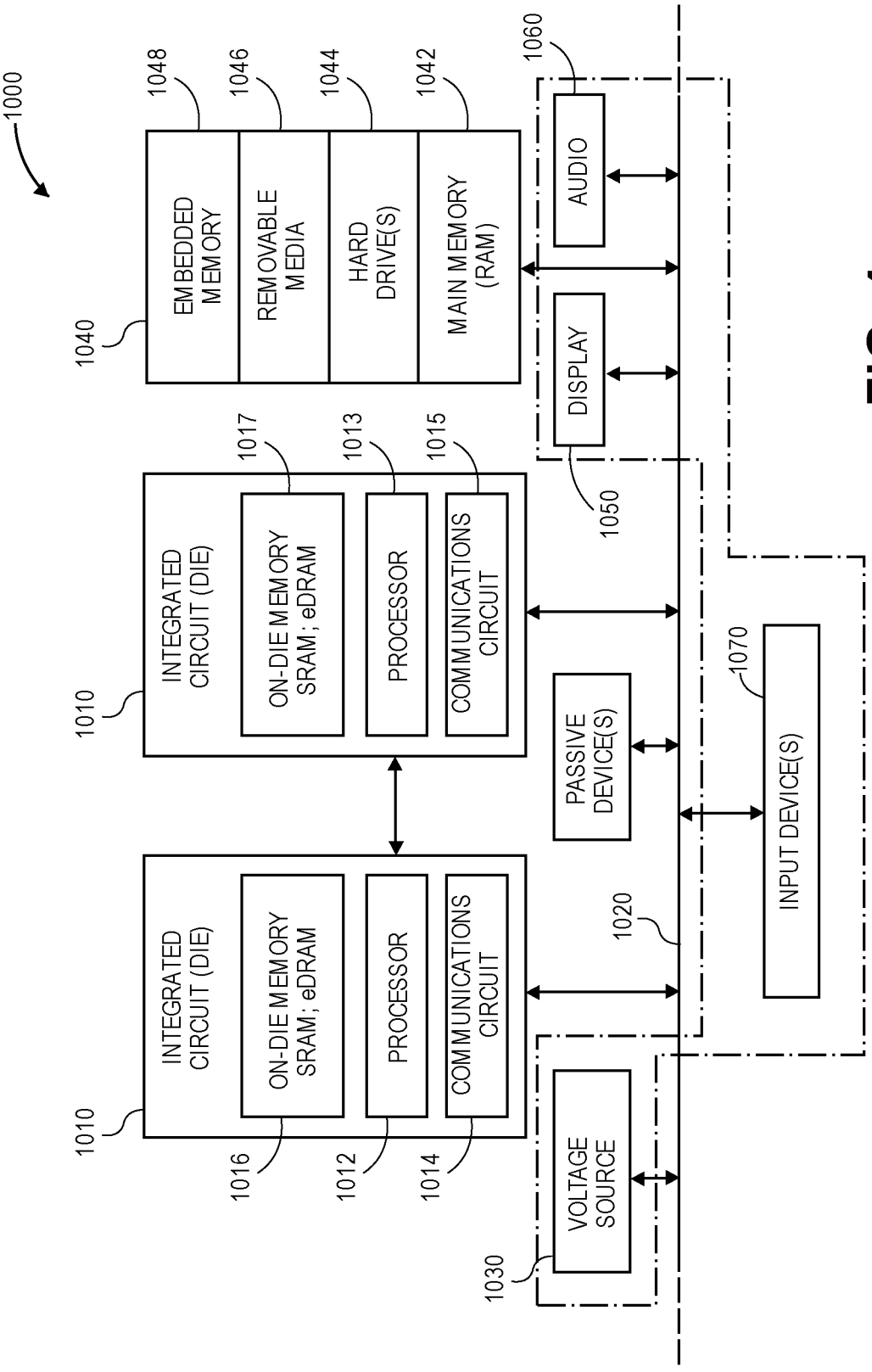

1000
1040
EMBEDDED MEMORY
1048
REMOVABLE MEDIA
1046
HARD DRIVE(S)
1044
MAIN MEMORY (RAM)
1042
1060
AUDIO
DISPLAY
1050
1010
INTEGRATED CIRCUIT (DIE)
ON-DIE MEMORY SRAM; eDRAM
1017
PROCESSOR
1013
COMMUNICATIONS CIRCUIT
1015
PASSIVE DEVICE(S)
1070
INPUT DEVICE(S)
1020
1010
INTEGRATED CIRCUIT (DIE)
ON-DIE MEMORY SRAM; eDRAM
1016
PROCESSOR
1012
COMMUNICATIONS CIRCUIT
1014
VOLTAGE SOURCE
1030

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2017/012857

A. CLASSIFICATION OF SUBJECT MATTER

G06F 12/02(2006.01)i, G06F 11/22(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 12/02; G06F 11/25; G01R 31/3177; H04L 1/22; G01R 31/28; H02H 3/05; G06F 11/22

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: muti-core, system, on-chip, test, auto-response, unit, fabricated, logical address, boundary

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2015-0316605 A1 (DUKE UNIVERSITY) 05 November 2015 See paragraphs [0005], [0027], [0031], [0036], [0040]-[0055], [0067], [0078], [0081], [0088], [0091], [0107]; claims 7, 17, 23; and figure 4.	8-18
Y		1-7,19-23
Y	US 2009-0083594 A1 (NATHAN P. CHELSTROM et al.) 26 March 2009 See paragraphs [0002], [0009], [0058], [0067]; and figure 4.	1-7,19-23
A	US 2007-0168796 A1 (CHRIS MARTIN et al.) 19 July 2007 See paragraphs [0034]-[0035]; and figure 7.	1-23
A	US 2003-0005380 A1 (HANG T. NGUYEN et al.) 02 January 2003 See paragraphs [0006], [0017]-[0022]; and figure 1.	1-23
A	KR 10-2000-0029368 A (ADVANTEST CORPORATION) 25 May 2000 See pages 2-3; and figure 1.	1-23

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
18 April 2017 (18.04.2017)	**19 April 2017 (19.04.2017)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea
Facsimile No. +82-42-481-8578

Authorized officer
CHIN, Sang Bum
Telephone No. +82-42-481-8398

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/US2017/012857

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2015-0316605 A1	05/11/2015	None	
US 2009-0083594 A1	26/03/2009	US 2007-0266284 A1	15/11/2007
		US 7478300 B2	13/01/2009
		US 7908536 B2	15/03/2011
US 2007-0168796 A1	19/07/2007	US 2006-0236170 A1	19/10/2006
		US 2007-0168795 A1	19/07/2007
		US 7251762 B2	31/07/2007
		US 7404124 B2	22/07/2008
		US 7412634 B2	12/08/2008
US 2003-0005380 A1	02/01/2003	None	
KR 10-2000-0029368 A	25/05/2000	DE 19952272 A1	04/05/2000
		DE 19952272 B4	31/05/2007
		JP 2000-137061 A	16/05/2000
		KR 10-0502128 B1	19/07/2005
		TW 533313 A	21/05/2003
		US 6249893 B1	19/06/2001