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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 March 2004 (25.03.2004)

PCT

(10) International Publication Number
WO 2004/025407 A2

(51) International Patent Classification[7]: G06F

(21) International Application Number:
PCT/US2003/028356

(22) International Filing Date:
9 September 2003 (09.09.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/241,511 10 September 2002 (10.09.2002) US

(71) Applicant: QUICKSILVER TECHNOLOGY, INC. [US/US]; 6640 Via Del Oro, San Jose, CA 95119 (US).

(72) Inventor: SCHEUERMANN, W., James; 21485 Saratoga Hills Road, Saratoga, CA 95070 (US).

(74) Agents: SAWYER, Joseph, A., Jr. et al.; Sawyer Law Group LLP, P.O. Box 51418, Palo Alto, CA 94303 (US).

(81) Designated States *(national)*: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 2004/025407 A2

(54) Title: METHOD AND SYSTEM FOR AN INTERCONNECTION NETWORK TO SUPPORT COMMUNICATIONS AMONG A PLURALITY OF HETEROGENEOUS PROCESSING ELEMENTS

(57) Abstract: Aspects of a method and system for supporting communication among a plurality of heterogeneous processing elements of a processing system are described. The aspects include an interconnection network that supports services between any two processing nodes within a plurality of processing nodes. A predefined data word format is utilized for communication among the plurality of processing nodes on the interconnection network, the predefined data word format indicating a desired service and desired routing. The desired routing is utilized to allow for the broadcasting of real time inputs. Further, look-ahead logic in the network is used to maximize throughput for the network by each processing node. Finally, a security field is utilized to limit peek-poke privileges for a particular node. With the aspects of the present invention, multiple processing elements are networked in an arrangement that allows fair and efficient communication in a point-to-point manner to achieve an efficient and effective system.

METHOD AND SYSTEM FOR AN INTERCONNECTION NETWORK TO SUPPORT COMMUNICATIONS AMONG A PLURALITY OF HETEROGENEOUS PROCESSING ELEMENTS

RELATED APLICATION

The present application is a continuation in part of application Serial No. 09/898,350, filed on July 3, 2001, and entitled "Method and System for an Interconnection Network to Support Communications Among a Plurality of Heterogeneous Processing Elements."

FIELD OF THE INVENTION

The present invention relates to communications among a plurality of processing elements and an interconnection network to support such communications.

BACKGROUND OF THE INVENTION

The electronics industry has become increasingly driven to meet the demands of high-volume consumer applications, which comprise a majority of the embedded systems market. Embedded systems face challenges in producing performance with minimal delay, minimal power consumption, and at minimal cost. As the numbers and types of consumer applications where embedded systems are employed increases, these challenges become even more pressing. Examples of consumer applications where embedded systems are employed include handheld devices, such as cell phones, personal digital assistants (PDAs), global positioning system (GPS) receivers, digital cameras, etc. By their nature, these devices are required to be small, low-power, light-weight, and feature-rich.

In the challenge of providing feature-rich performance, the ability to produce efficient utilization of the hardware resources available in the devices becomes paramount. As in most every processing environment that employs multiple processing elements, whether these elements take the form of processors, memory, register files, etc., of particular concern is coordinating the interactions of the multiple processing elements. Accordingly, what is needed is a manner of networking multiple processing elements in an arrangement that allows fair and efficient communication in a point-to-point fashion to achieve an efficient and effective system. The present invention addresses such a need.

SUMMARY OF THE INVENTION

Aspects of a method and system for supporting communication among a plurality of heterogeneous processing elements of a processing system are described. The aspects include an interconnection network that supports services between any two processing nodes within a plurality of processing nodes. A predefined data word format is utilized for communication among the plurality of processing nodes on the interconnection network, the predefined data word format indicating a desired service and desired routing. The desired routing is utilized to allow for the broadcasting of real time inputs. Further, look-ahead logic in the network is used to maximize throughput for the network by each processing node. Finally, a security field is utilized to limit peek-poke privileges for a particular node.

With the aspects of the present invention, multiple processing elements are networked in an arrangement that allows fair and efficient communication in a point-to-point manner to achieve an efficient and effective system.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram illustrating an adaptive computing engine.

Figure 2 illustrates a network architecture in accordance with the present invention.

Figure 3 illustrates a data structure utilized to support the communications among the nodes via the MIN.

Figure 4 illustrates a block diagram of logic included in the interconnection network to support communications among the nodes in accordance with a preferred embodiment of the present invention.

Figure 5 illustrates an instance of look-ahead logic for a 64 node system.

Figure 6 illustrates an interconnection diagram for ACM core.

Figure 7 illustrates a minimum system with 1-4 cores plus system resources, booting from system memory.

Figure 8 illustrates systems with 1-4 cores, local external memory, and System Bus I/O.

Figure 9 illustrates systems with 1-4 cores, local external memory/memories, a single system interface, and separate real time I/O.

DETAILED DESCRIPTION OF THE INVENTION

The present invention relates to communications support among a plurality of processing elements in a processing system. The following description is presented to enable one of ordinary skill in the art to make and use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred

embodiment and the generic principles and features described herein will be readily apparent to those skilled in the art. Thus, the present invention is not intended to be limited to the embodiment shown but is to be accorded the widest scope consistent with the principles and features described herein.

In a preferred embodiment, the aspects of the present invention are provided in the context of an adaptable computing engine in accordance with the description in co-pending U.S. Patent application, serial no. 09/815,122, entitled "Adaptive Integrated Circuitry with Heterogeneous and Reconfigurable Matrices of Diverse and Adaptive Computational Units Having Fixed, Application-Specific Computational Elements," assigned to the assignee of the present invention and incorporated by reference in its entirety herein. Portions of that description are reproduced herein below for clarity of presentation of the aspects of the present invention.

Referring to Figure 1, a block diagram illustrates an adaptive computing engine ("ACE") 100, which is preferably embodied as an integrated circuit, or as a portion of an integrated circuit having other, additional components. In the preferred embodiment, and as discussed in greater detail below, the ACE 100 includes a controller 120, one or more reconfigurable matrices 150, such as matrices 150A through 150N as illustrated, a matrix interconnection network 110, and preferably also includes a memory 140.

The controller 120 is preferably implemented as a reduced instruction set ("RISC") processor, controller or other device or IC capable of performing the two types of functionality. The first control functionality, referred to as "kernal" control, is illustrated as

kernal controller ("KARC") 125, and the second control functionality, referred to as "matrix" control, is illustrated as matrix controller ("MARC") 130.

The various matrices 150 are reconfigurable and heterogeneous, namely, in general, and depending upon the desired configuration: reconfigurable matrix 150A is generally different from reconfigurable matrices 150B through 150N; reconfigurable matrix 150B is generally different from reconfigurable matrices 150A and 150C through 150N; reconfigurable matrix 150C is generally different from reconfigurable matrices 150A, 150B and 150D through 150N, and so on. The various reconfigurable matrices 150 each generally contain a different or varied mix of computation units, which in turn generally contain a different or varied mix of fixed, application specific computational elements, which may be connected, configured and reconfigured in various ways to perform varied functions, through the interconnection networks. In addition to varied internal configurations and reconfigurations, the various matrices 150 may be connected, configured and reconfigured at a higher level, with respect to each of the other matrices 150, through the matrix interconnection network (MIN) 110.

In accordance with the present invention, the MIN 110 provides a foundation that allows a plurality of heterogeneous processing nodes, e.g., matrices 150, to communicate by providing a single set of wires as a homogeneous network to support plural services, these services including DMA (direct memory access) services, e.g., Host DMA (between the host processor and a node), and Node DMA (between two nodes), and read/write services, e.g., Host Peek/Poke (between the host processor and a node), and Node Peek/Poke (between two nodes). In a preferred embodiment, the plurality of heterogeneous nodes is organized in a

manner that allows scalability and locality of reference while being fully connected via the MIN 110. U.S. patent application serial number 09/898,350 entitled Method and System for an Interconnection Network to Support Communications Among a Plurality of Heterogeneous Processing Elements filed on July 3, 2001, discusses an interconnection network to support a plurality of processing elements and is incorporated by reference herein. This network is enhanced by a plurality of features which are described herein below.

Figure 2 illustrates a network architecture 200 in accordance with the present invention. In this embodiment there are four groupings 210-280 of nodes. As is seen, grouping 210-240 can communicate with MIN 272 and groupings 250-280 communicate with MIN 274. MINs 272 and 274 communicate with the network root 252. A MIN 110 further supports communication between nodes in each grouping and a processing entity external to the grouping 210, via a network root 252. The network root 250 is coupled to a K-Node 254, network input and output I/O blocks 256 and 258, system interface I/O blocks 261, a SRAM memory controller 262, and an on/chip bulk RAM/bulk memory 264. In a preferred embodiment, the organization of nodes as a grouping 210-280 can be altered to include a different number of nodes and can be duplicated as desired to interconnect multiple sets of groupings, e.g., groupings 230, 240, and 250, where each set of nodes communicates within their grouping and among the sets of groupings via the MIN 110.

In a preferred embodiment, a data structure as shown in Figure 3 is utilized to support the communications among the nodes 200 via the MIN 110. The data structure preferably comprises a multi-bit data word 300, e.g., a 30 bit data word, that includes a

service field 310 (e.g., a 4-bit field), a node identifier field 320 (e.g., a 6-bit field), a data/payload field 340 (e.g., a 32-bit data field), a routing field 342, and a security field 344 as shown. Thus, the data word 300 specifies the type of operation desired, e.g., a node write operation, the destination node of the operation, e.g., the node whose memory is to be written to, a specific entity within the node, e.g., the input channel being written to, and the data, e.g., the information to be written in the input channel of the specified node. The MIN 110 exists to support the services indicated by the data word 300 by carrying the information under the direction, e.g., "traffic cop", of arbiters at each point in the network of nodes.

For an instruction in a source node, a request for connection to a destination node is generated via generation of a data word. Referring now to Figure 4, for each node 200 in a grouping 210, a token-based, round robin arbiter 410 is implemented to grant the connection to the requesting node 200. The token-based, round robin nature of arbiter 410 enforces fair, efficient, and contention-free arbitration as priority of network access is transferred among the nodes, as is standardly understood by those skilled in the art. Of course, the priority of access can also be tailored to allow specific services or nodes to receive higher priority in the arbitration logic, if desired. For the quad node embodiment, the arbiter 410 provides one-of-four selection logic, where three of the four inputs to the arbiter 410 accommodate the three peer nodes 200 in the arbitrating node's quad, while the fourth input is provided from a common input with arbiter and decoder logic 420.

The common input logic 420 connects the grouping 210 to inputs from external processing nodes. Correspondingly, for the grouping 210 illustrated, its common output arbiter and decoder logic 430 would provide an input to another grouping's common input

logic 420. It should be appreciated that although single, double-headed arrows are shown for the interconnections among the elements in Figure 4, these arrows suitably represent request/grant pairs to/from the arbiters between the elements, as is well appreciated by those skilled in the art.

A feature of the present invention is a broadcast mode to allow for the routing of real time-inputs (RTIs). The details of the implementation of such a feature are described in detail herein below.

Broadcast mode

For the real time inputs that are to be routed to multiple nodes, the routing field 342 of Figure 3 will be encoded with the broadcast information. Coding for a 8-bit routing field [7:0] is shown below:

[7:6]

0 0 chip 0

0 1 chip 1

1 0 chip 2

1 1 chip 3

[5]

0 nodes

1 root

when bit [5] = 0, bits [4:0] indicate one of 32 nodes; when bit [5] = 1, bits [4:0] indicate a root resource: knode, bulk memory, external memory, and so on.

For real time input data, each one of the eight routing field bits directs (or does not direct) the data at one of eight quads. Note that countless additional combinations are possible when one selects a set that includes the intended nodes and, at the unintended nodes within the set, silently discards the data. Of course, this has the potential of denying other transfers to the unintended nodes during real time input data transfers.

Security Field 344

The security field 344 has been added to restrict Peek/Poke privileges within the network to the K-node, which runs the OS, or to a host with K-node permission.

A bit in the security field 344 is set to 'b1' for K node transfers and for system (host) transfers given the K-node's permission. In this embodiment, the K-node writes a "Permissions Register" to control which system transfers propagate beyond the system input port and which system transfers are silently discarded. The 14-bit Permissions Register preferably is located in the Network's system_out module. The K-node writes this register by placing the following in its node output register:

ROUTE[7:0] = 0x3C;

SERV[3:0] = 0x0;

AUX[5:0] = 0x00;

DATA[31:0] = 18 b'000000000000000, Perm_Reg[13:0];

.Perm_Reg[11:0] (enable_knode_access, enable_at_knode_access, enable_rto_access, enable_bulk_memory_access, enable_SDRAM_access, enable_node_access, enable_point_to_point_access, enable_peek_poke_access, enable_memory_random_access);

```
if      ((target_is_knode and enable_knode_access) or
         (target_is_at_knode and enable_at_knode_access) or
         (target_is_rto and enable_rto_access) or
         (target_is_bulk_memory and enable_bulk_memory_access) or
         (target_is_SDRAM and enable_SDRAM_access) or
         (target_is_node and enable_node_access)
and
        ((service_is_point_to_point and enable_point_to_point_access) or
         (service_is_peek_poke and enable_peek_poke_access) or
         (service_is_dmaand enable_dma_access) or
         (service_is_message and enable_message_access) or
         (service_is_rti and enable_rti_access) or
         (service_is_memory_random_access and
enable_memory_random_access) ) )

        transfer data from system to destination

else silently discard data from system ;
```

Perm_Reg[13:2] are used to encode the number of cores that are interconnected in the system:

- b'00 one core
- b'01 two cores
- b'10 three cores
- b'11 four cores

To eliminate endless recirculation of non-existent-destination network traffic, the core with chip_id=b'00 will silently discard such network traffic.

Look-ahead Logic for Maximizing Network Throughput

Full look-ahead logic is utilized across the entire network of MINS to maximize network throughput. The network moves data from one pipeline register of one MIN to another pipeline register of another MIN when the latter is "available". A pipeline register is "available" if:

(1) The register is empty.

(2) The register is full, but its contents will be transferred at the next network clock tick.

The look-ahead logic allows for the second (2) of the above two conditions.

Figure 5 illustrates an instance 500 of look-ahead logic. A flip-flop 502 signals that a register 504 is full. A decoder 506 requests access to one of four possible destinations. This register 504 is "available" for new data when:

(1) the register is empty, or

(2) the register is full, but its contents will be transferred at the next network clock tick, as indicated by a grant signal from an arbiter 501 at one of four possible destinations.

Connecting Multiple ACMs

The network also has been enhanced to allow the interconnection of multiple ACMs. This requires adding in one embodiment two bits to the routing field 342 of the network data structure. In this embodiment, the core will have a 51 bit data structure as shown below. With it, up to four cores can be interconnecting to realize more powerful systems than could be achieved with a single core. Figure 6 illustrates an interconnection diagram for ACM core. The ACM 600 receives signals from a memory 602, high bandwidth real time input 604, and high bandwidth real time outputs 606. ACM 600 also communicates with a host bridge 610 and I/O interfaces 612. Figure 7 illustrates a minimum system with four serial connected ACMs 600. Figure 8 illustrates a series of 4 ACMs which includes a local external memory 702.

Figure 9 illustrates a series of four local external memory / memories, a single system interface, and separate real time I/O with ACMs. Each ACM that processes RTI data must have a MUX at its "net_in" port. Each ACM that does not process RTI data connects its "net_in" port directly to the "net_out" port of its neighbor.

The interconnections among the elements are realized utilizing a straightforward and effective point-to-point network, allowing any node to communicate with any other node efficiently. In addition, for *n* nodes, the system supports *n* simultaneous transfers. A common data structure and use of arbitration logic provides consistency and order to the communications on the network.

From the foregoing, it will be observed that numerous variations and modifications may be effected without departing from the spirit and scope of the novel concept of the invention. It is to be understood that no limitation with respect to the specific methods and apparatus illustrated herein is intended or should be inferred. It is, of course, intended to cover by the appended claims all such modifications as fall within the scope of the claims.

CLAIMS

What is claimed is:

1. A method for supporting communication among a plurality of heterogeneous processing elements of a processing system, the method comprising:

forming an interconnection network to support services between any two processing nodes within a plurality of processing nodes;

utilizing a predefined data word format for communication among the plurality of processing nodes on the interconnection network, the predefined data word format indicating a desired service and a desired routing; and

utilizing the desired routing to allow for the broadcasting of real time inputs.

2. The method of claim 1 wherein forming an interconnection network further comprises forming connections between each node in a grouping of nodes and between each of a plurality of groupings.

3. The method of claim 2 wherein the grouping of nodes further comprises a grouping of four nodes.

4. The method of claim 3 further comprising utilizing a matrix element as a processing node.

5. The method of claim 1 wherein forming an interconnection network further comprises forming a network of connections to support services in a point-to-point manner.

6. The method of claim1 further comprising utilizing the interconnection network to support services between a node and a host processor external to the plurality of processing nodes.

7. The method of claim 1 wherein utilizing a predefined data word format further comprises utilizing a data word format that includes a service field, a node field, a tag field, a routing field, a security field and a data field.

8. The method of claim 7 wherein the data word format further comprises a 51-bit data structure.

9. The method of claim 1 which includes the step of:

utilizing a look-ahead logic across the interconnection network to maximize network throughput.

10. The method of claim 1 which includes the step of utilizing the security field in the data structure to limit PEEK-POKE privileges for a particular node.

11. A system for supporting communication among a plurality of processing elements, the system comprising

a plurality of heterogeneous processing nodes organized as a plurality of groupings;

an interconnection network for supporting data services within and among the plurality of groupings as indicated by a data word sent from one processing node to another; and

a look-ahead logic on the interconnection network to maximize throughput for the interconnection network by the plurality of heterogeneous processing nodes.

12. The method of claim 11 wherein each grouping in the plurality of groupings further comprises four processing nodes.

13. The system of claim 11 wherein a plurality of arbiters provide arbitration within and among each grouping in a token-based, round robin manner.

14. The system of claim 11 further comprising a matrix as a processing node type.

15. The system of claim 11 further comprising a host processor coupled to the plurality of heterogeneous processing nodes via the interconnection network.

16. The system of claim 11 wherein the data word further comprises a plurality of bits organized as a services field, a node identification field, a tag field, routing field, security field and a data field.

17. The system of claim 16 wherein the routing field is utilized to allow for the broadcasting of real time inputs.

18. The system of claim 16 wherein the security field is utilized to limit peek/poke privileges for a particular node.

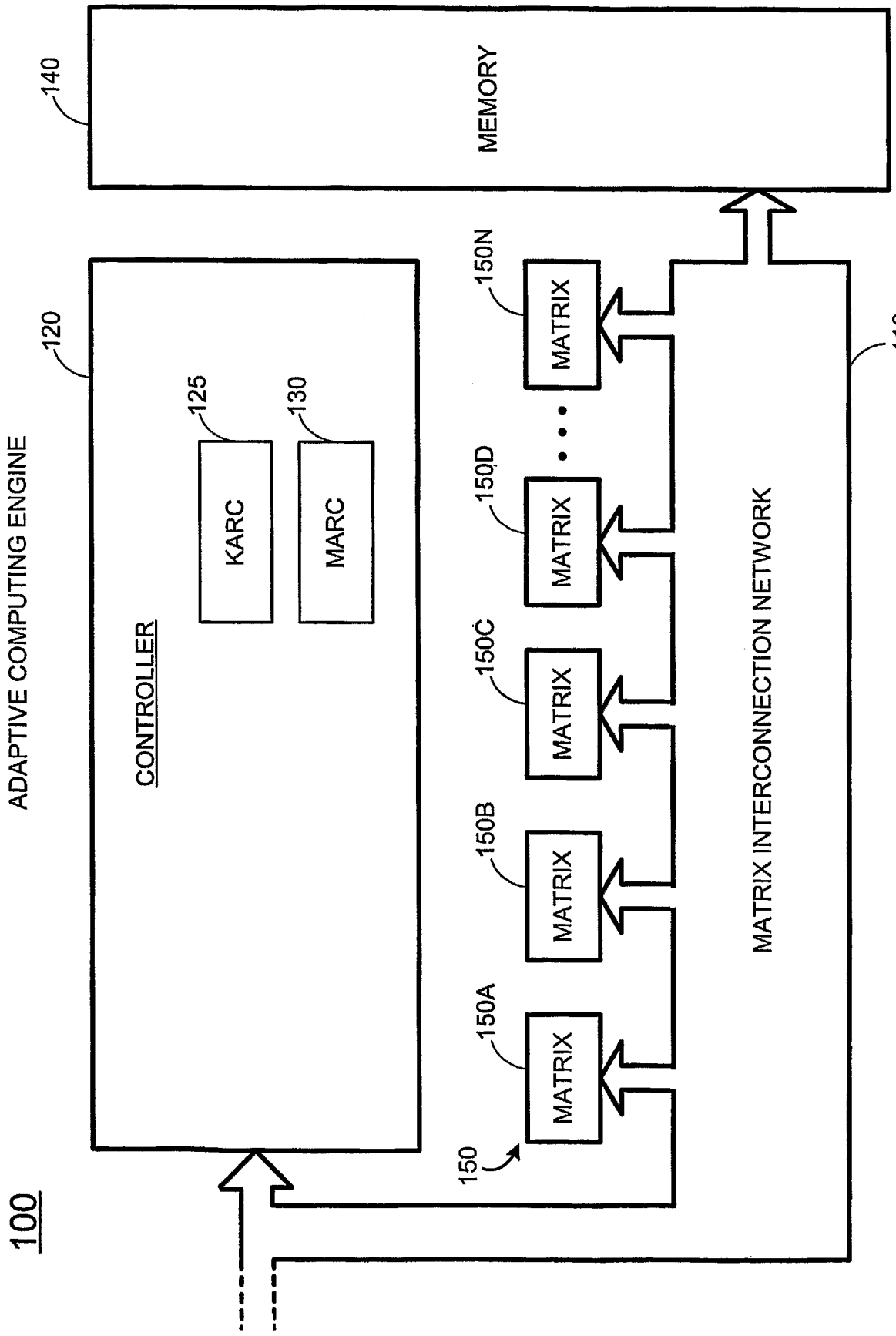

100
ADAPTIVE COMPUTING ENGINE
120
CONTROLLER
125
KARC
130
MARC
140
MEMORY
150
150A
MATRIX
150B
MATRIX
150C
MATRIX
150D
MATRIX
150N
MATRIX
110
MATRIX INTERCONNECTION NETWORK

FIG. 1

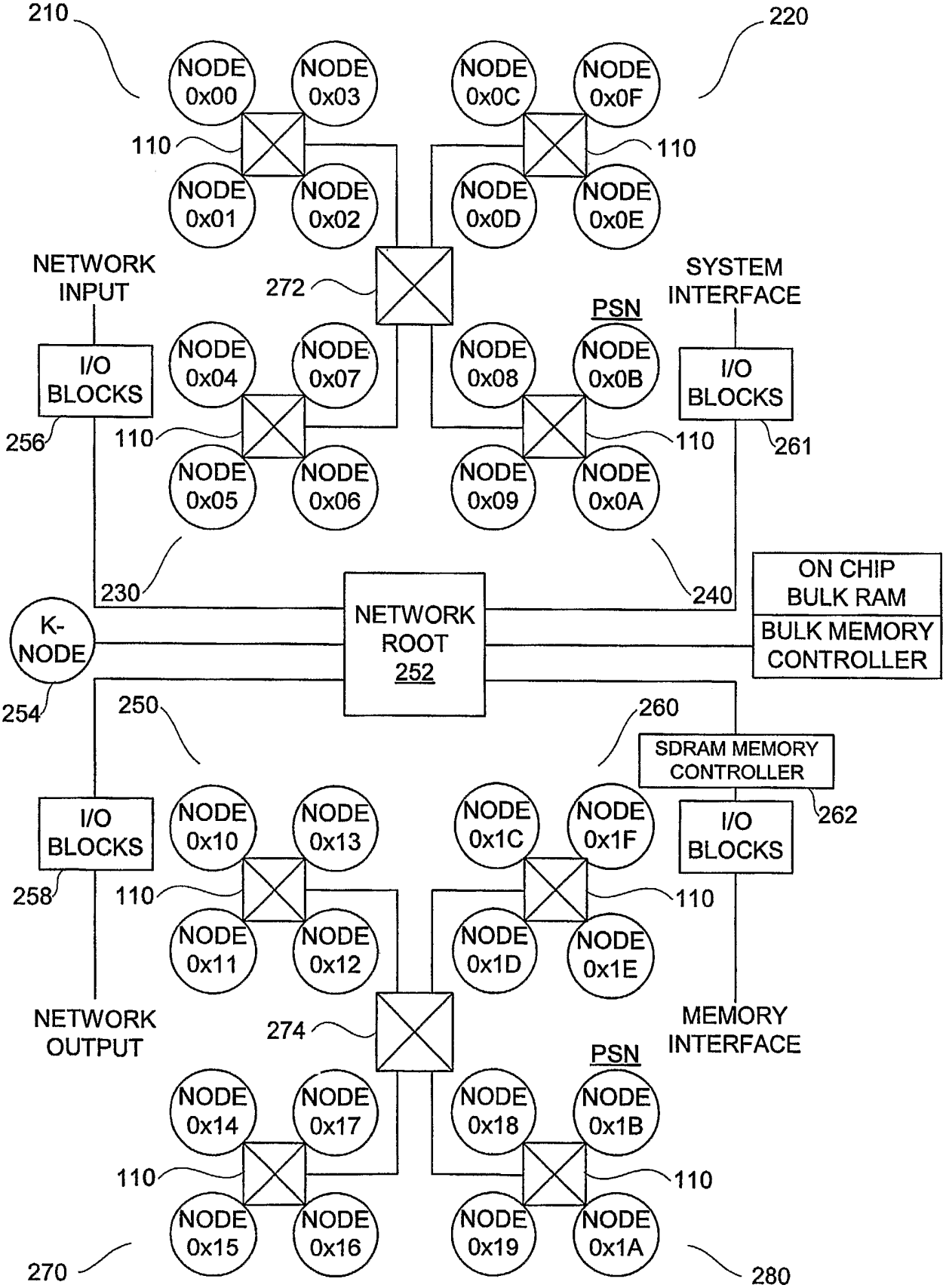

210
220
NODE 0x00
NODE 0x03
NODE 0x0C
NODE 0x0F
110
NODE 0x01
NODE 0x02
NODE 0x0D
NODE 0x0E
NETWORK INPUT
272
SYSTEM INTERFACE
PSN
I/O BLOCKS
NODE 0x04
NODE 0x07
NODE 0x08
NODE 0x0B
I/O BLOCKS
256
110
261
NODE 0x05
NODE 0x06
NODE 0x09
NODE 0x0A
230
240
ON CHIP BULK RAM
K-NODE
NETWORK ROOT 252
BULK MEMORY CONTROLLER
254
250
260
SDRAM MEMORY CONTROLLER
I/O BLOCKS
NODE 0x10
NODE 0x13
NODE 0x1C
NODE 0x1F
I/O BLOCKS
262
258
110
NODE 0x11
NODE 0x12
NODE 0x1D
NODE 0x1E
NETWORK OUTPUT
274
MEMORY INTERFACE
PSN
NODE 0x14
NODE 0x17
NODE 0x18
NODE 0x1B
110
NODE 0x15
NODE 0x16
NODE 0x19
NODE 0x1A
270
280

FIG. 2

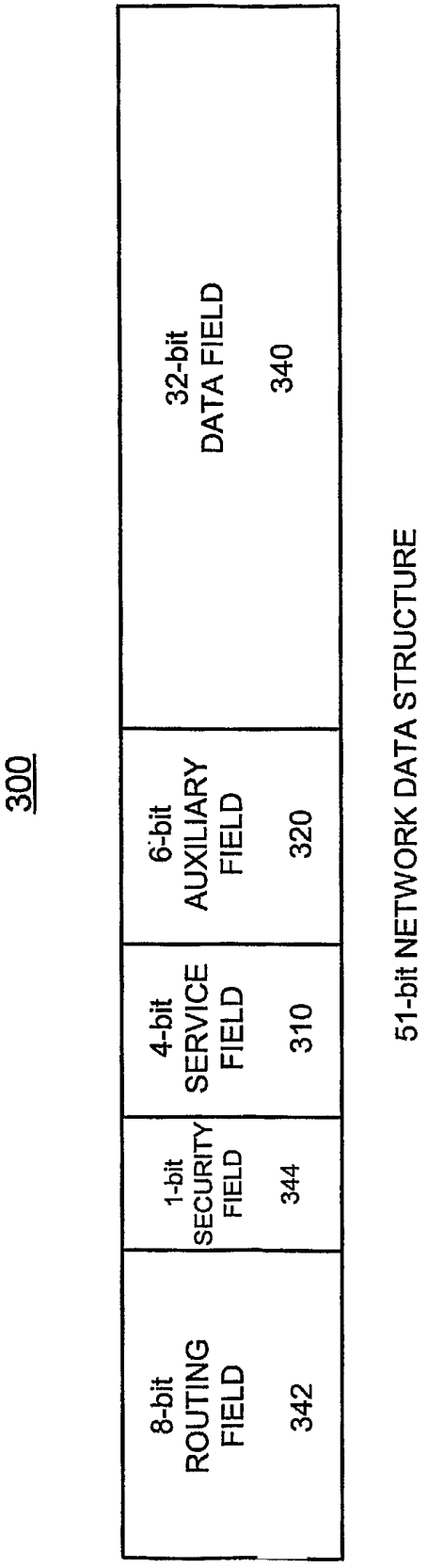

300
8-bit ROUTING FIELD
342
1-bit SECURITY FIELD
344
4-bit SERVICE FIELD
310
6-bit AUXILIARY FIELD
320
32-bit DATA FIELD
340
51-bit NETWORK DATA STRUCTURE

FIG. 3

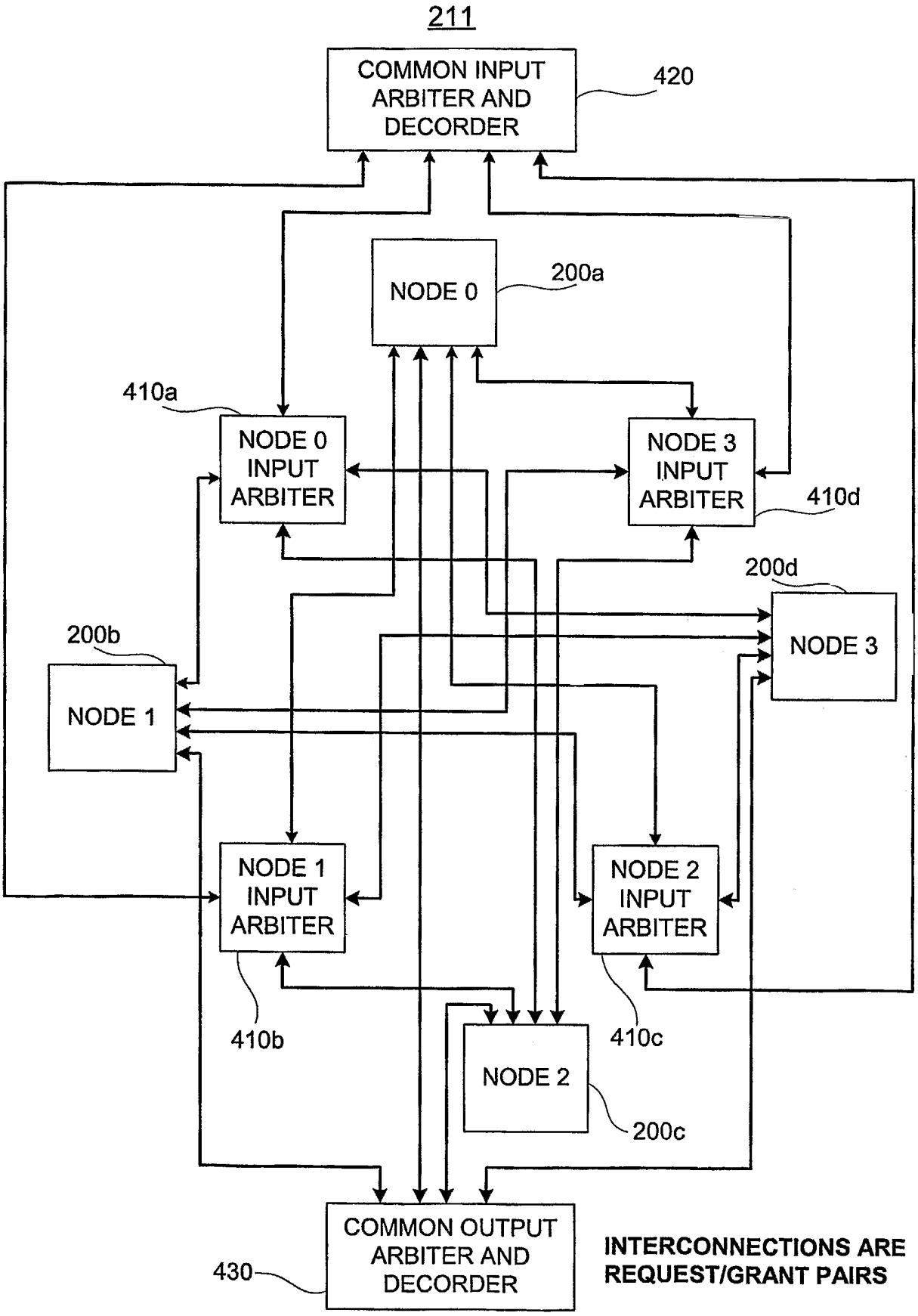
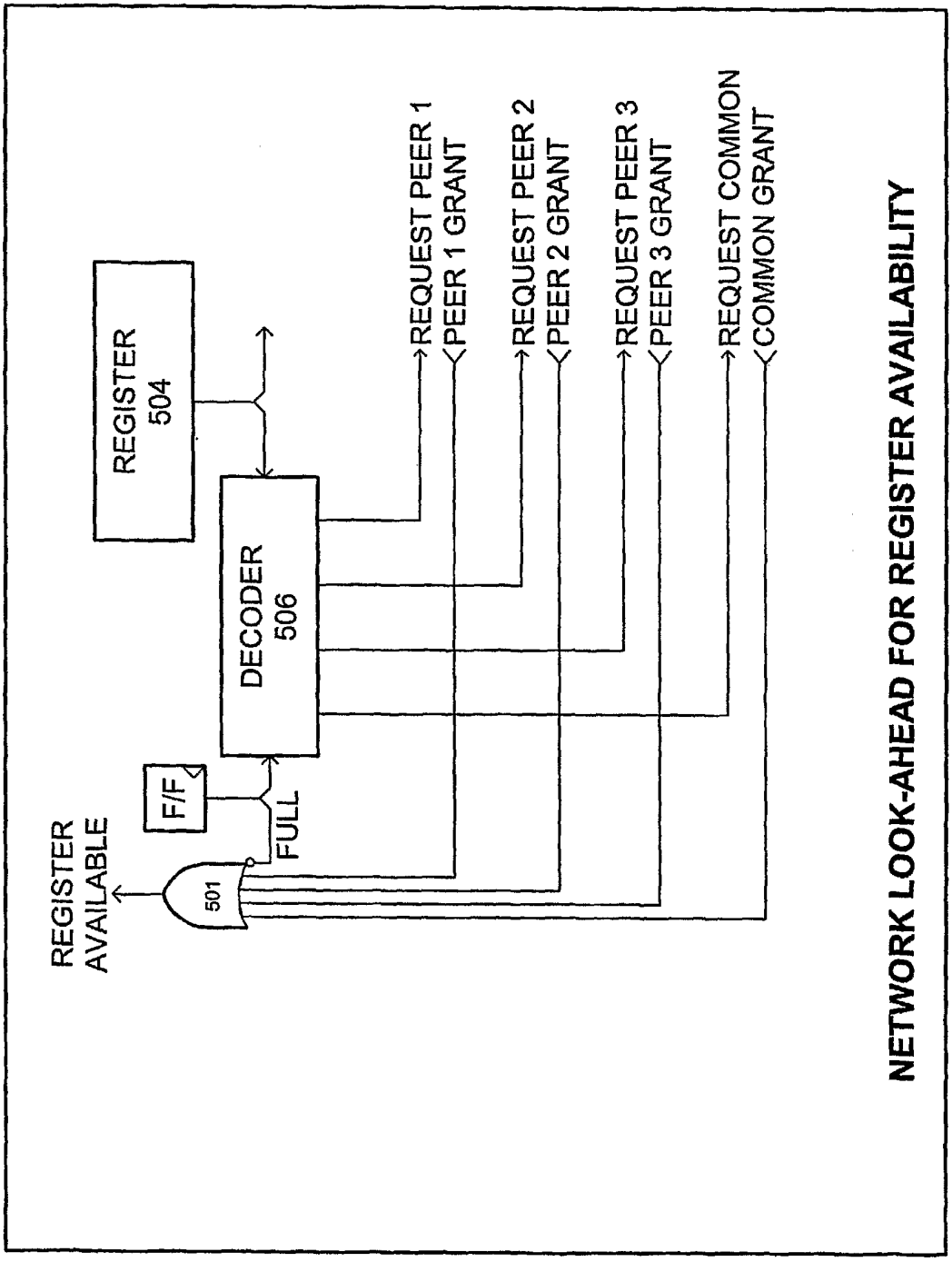

211
COMMON INPUT ARBITER AND DECORDER
420
NODE 0
200a
410a
NODE 0 INPUT ARBITER
NODE 3 INPUT ARBITER
410d
200d
NODE 3
200b
NODE 1
NODE 1 INPUT ARBITER
NODE 2 INPUT ARBITER
410b
410c
NODE 2
200c
COMMON OUTPUT ARBITER AND DECORDER
430
INTERCONNECTIONS ARE REQUEST/GRANT PAIRS

FIG. 4


NETWORK LOOK-AHEAD FOR REGISTER AVAILABILITY
REGISTER 504
DECODER 506
F/F
501
REGISTER AVAILABLE
FULL
REQUEST PEER 1
PEER 1 GRANT
REQUEST PEER 2
PEER 2 GRANT
REQUEST PEER 3
PEER 3 GRANT
REQUEST COMMON
COMMON GRANT

500

FIG. 5

INTERCONNECTION DIAGRAM FOR ACM CORE

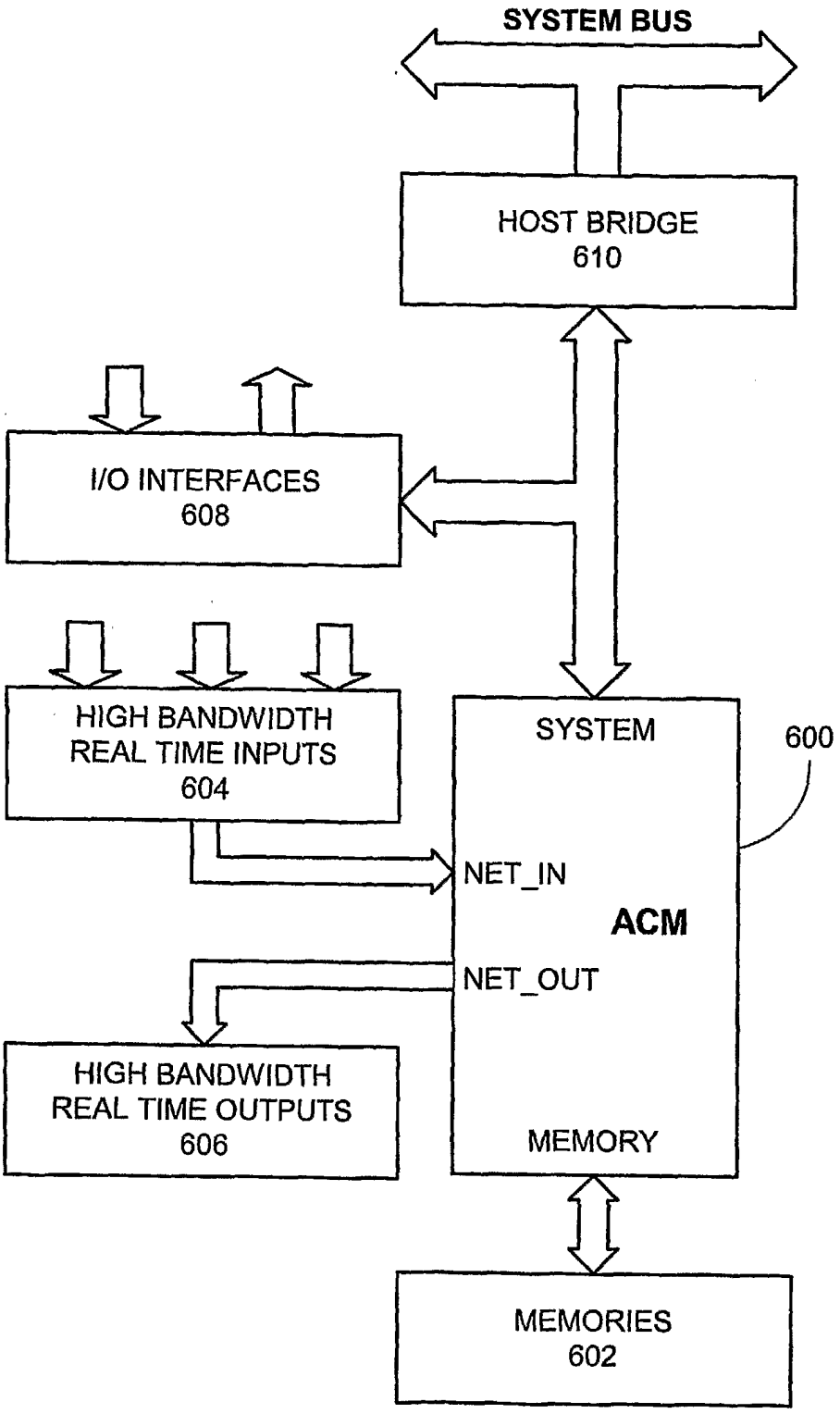


FIG. 6

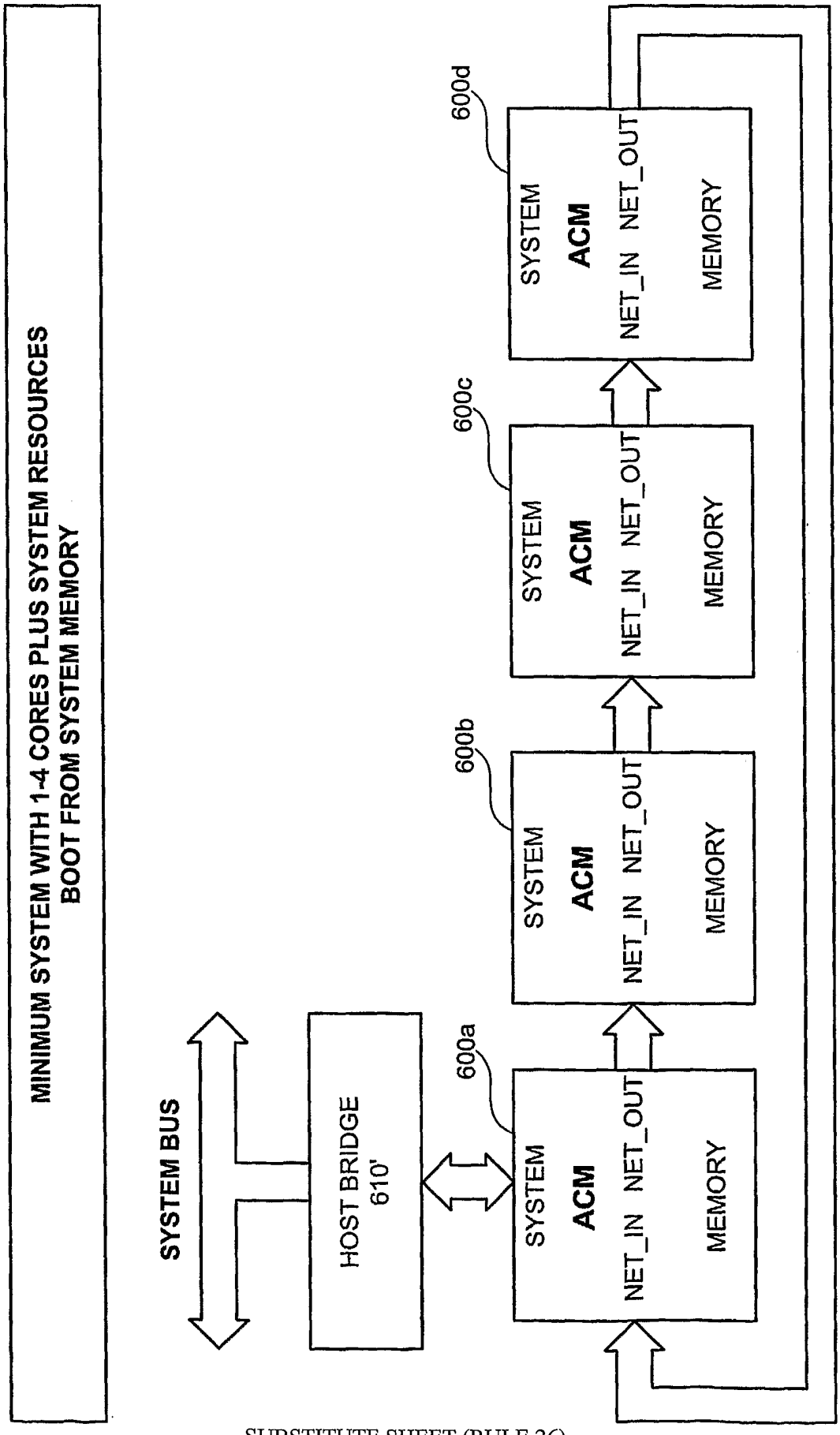

MINIMUM SYSTEM WITH 1-4 CORES PLUS SYSTEM RESOURCES
BOOT FROM SYSTEM MEMORY
SYSTEM BUS
HOST BRIDGE
610'
600a
600b
600c
600d
SYSTEM
ACM
NET_IN NET_OUT
MEMORY

FIG. 7

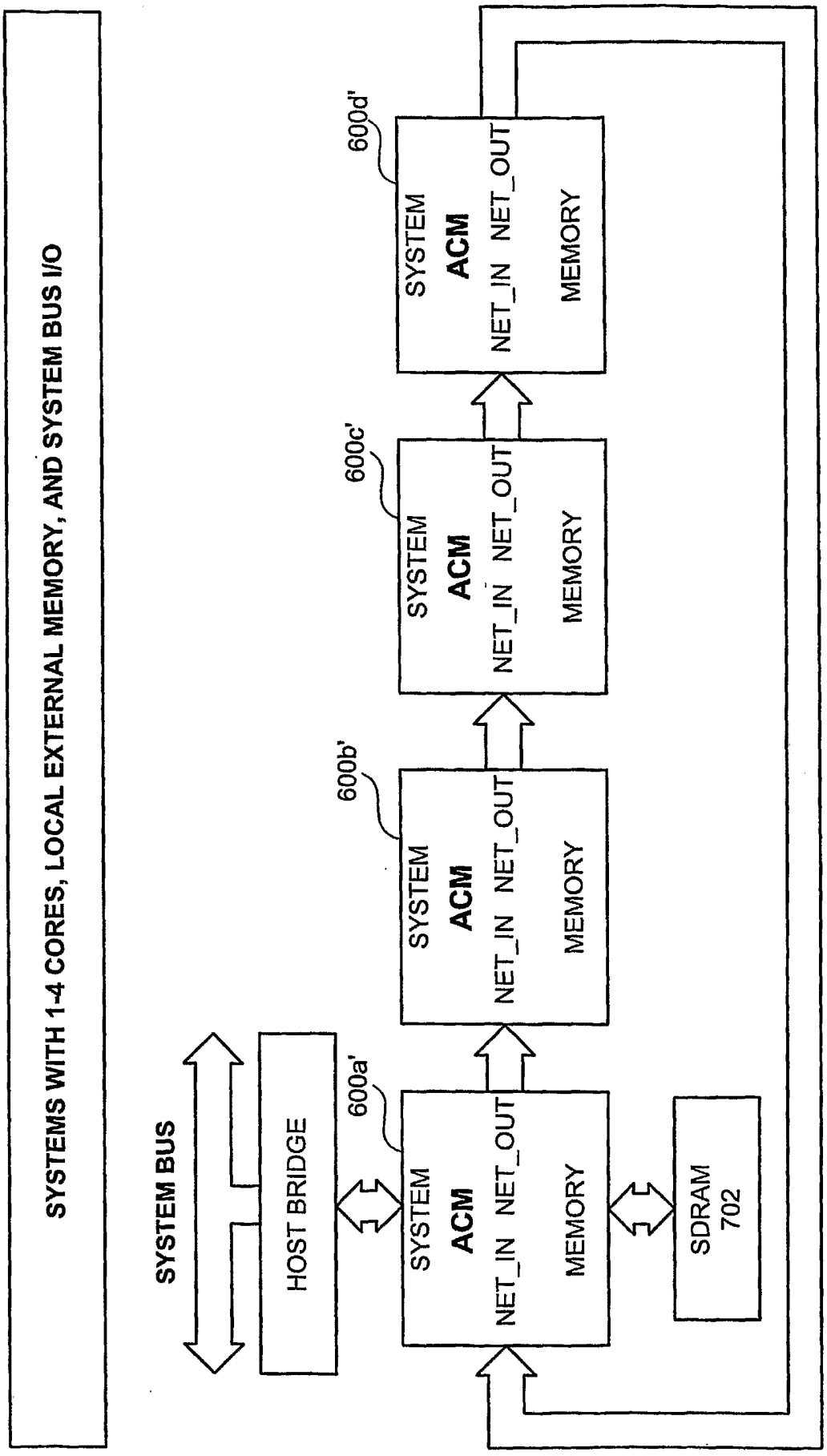

SYSTEMS WITH 1-4 CORES, LOCAL EXTERNAL MEMORY, AND SYSTEM BUS I/O
SYSTEM BUS
HOST BRIDGE
600a'
SYSTEM
ACM
NET_IN NET_OUT
MEMORY
SDRAM
702
600b'
SYSTEM
ACM
NET_IN NET_OUT
MEMORY
600c'
SYSTEM
ACM
NET_IN NET_OUT
MEMORY
600d'
SYSTEM
ACM
NET_IN NET_OUT
MEMORY

FIG. 8

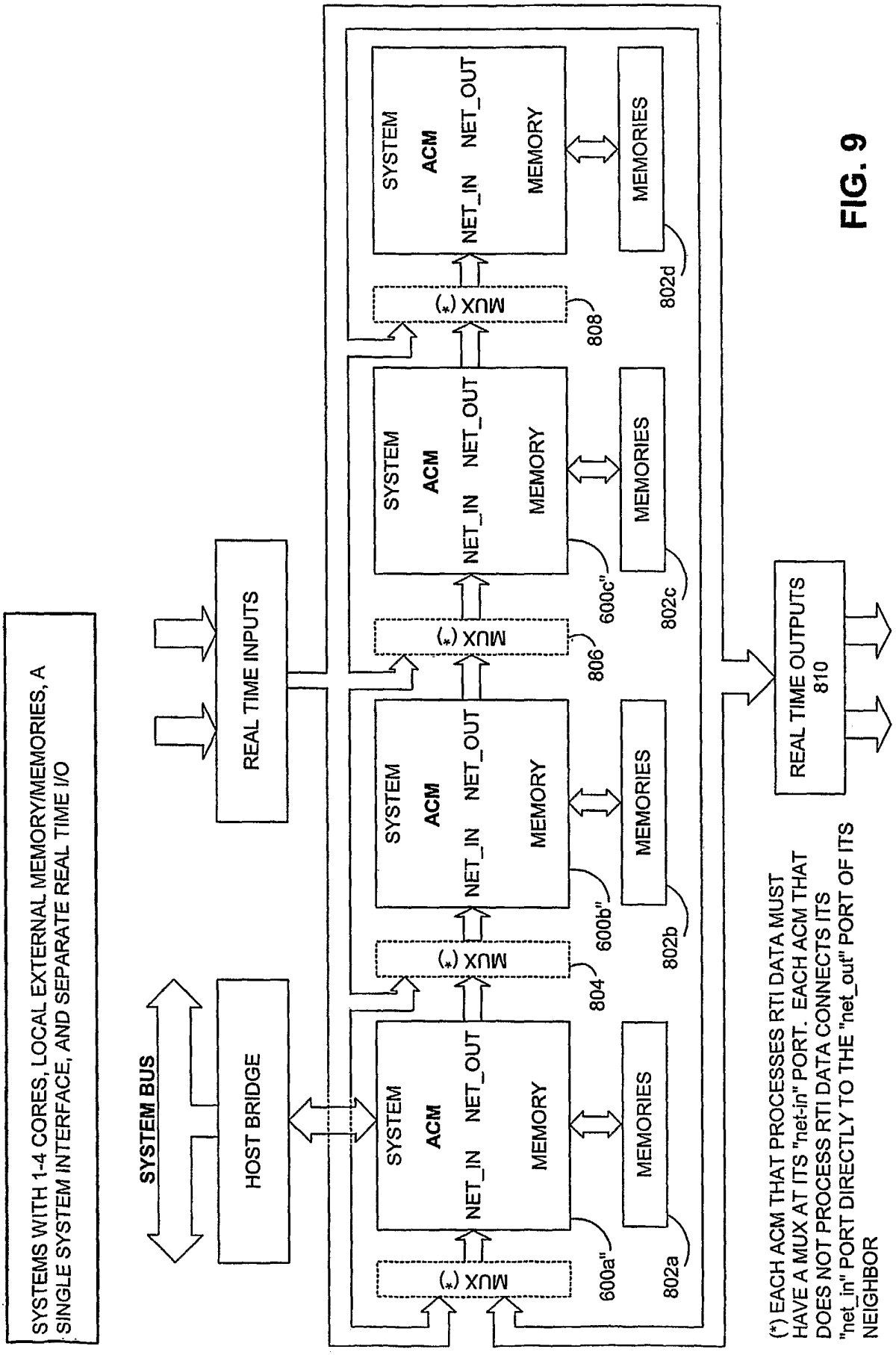

SYSTEMS WITH 1-4 CORES, LOCAL EXTERNAL MEMORY/MEMORIES, A SINGLE SYSTEM INTERFACE, AND SEPARATE REAL TIME I/O
SYSTEM BUS
HOST BRIDGE
REAL TIME INPUTS
SYSTEM ACM NET_IN NET_OUT MEMORY
MEMORIES
MUX (*)
600a"
600b"
600c"
802a
802b
802c
802d
804
806
808
REAL TIME OUTPUTS 810
(*) EACH ACM THAT PROCESSES RTI DATA MUST HAVE A MUX AT ITS "net-in" PORT. EACH ACM THAT DOES NOT PROCESS RTI DATA CONNECTS ITS "net_in" PORT DIRECTLY TO THE "net_out" PORT OF ITS NEIGHBOR

FIG. 9