

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
28 February 2008 (28.02.2008)

PCT

(10) International Publication Number  
**WO 2008/023218 A1**

(51) International Patent Classification:  
**G06F 13/362** (2006.01)

(21) International Application Number:  
PCT/IB2006/052909

(22) International Filing Date: 23 August 2006 (23.08.2006)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant (for all designated States except US):  
**FREESCALE SEMICONDUCTOR, INC.** [US/US];  
6501 William Cannon Drive West, Austin, TX 78735  
(US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **GOREN, Ori**  
[IL/IL]; Kibutz Ruchama, 79180 Kibutz Ruchama (IL).  
**NETANEL, Yaron** [IL/IL]; Ichilov 13/4, 42445 Natanya  
(IL). **LIVAY, Aviel** [IL/IL]; Rabinovich 47, 58671 Holon  
(IL). **MORAN, Gil** [IL/IL]; Ofarim 16, P.O.B. 1507,  
85025 Meitar (IL). **NEEMAN, Yossy** [IL/IL]; Ha-Ilanot  
20/14, 70700 Gedera (IL).

(81) Designated States (unless otherwise indicated, for every  
kind of national protection available): AE, AG, AL, AM,  
AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN,  
CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI,  
GB, GD, GE, GH, GM, HN, HR, HU, ID, IL, IN, IS, JP,  
KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT,  
LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ,  
NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU,  
SC, SD, SE, SG, SK, SL, SM, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,  
ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),  
European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,  
FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT,  
RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA,  
GN, GQ, GW, ML, MR, NE, SN, TD, TG).

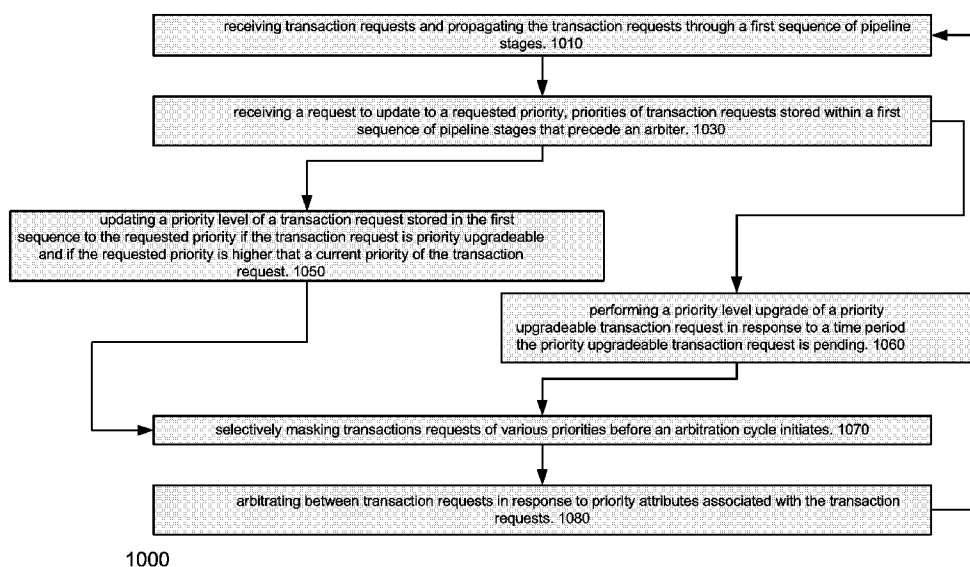
**Declaration under Rule 4.17:**

— of inventorship (Rule 4.17(iv))

**Published:**

— with international search report

(54) Title: DEVICE HAVING PRIORITY UPGRADE MECHANISM CAPABILITIES AND A METHOD FOR UPDATING PRI-  
ORITIES



(57) Abstract: A device (10, 11, 12, 13, 14) having priority update capabilities and a method (1000) for updating priorities, the method (1000) includes: receiving (1030) a request to update to a requested priority, priorities of transaction requests stored within a first sequence of pipeline stages that precede an arbiter; updating (1050) a priority level of a transaction request stored in the first sequence to the requested priority if the transaction request is priority upgradeable and if the requested priority is higher than a current priority of the transaction request; and arbitrating (1080) between transaction requests in response to priority attributes associated with the transaction requests.

## DEVICE HAVING PRIORITY UPGRADE MECHANISM CAPABILITIES AND A METHOD FOR UPDATING PRIORITIES

### FIELD OF THE INVENTION

5           The present invention relates to devices having priority upgrade capabilities and methods for upgrading capabilities.

### BACKGROUND OF THE INVENTION

Deep pipelined devices such as but not limited to on-chip interconnects,  
10       can interface with many components. Requests to receive a service or gain access to a certain bus are usually sent to an arbitrator that can apply various arbitration schemes in order to determine which service shall be granted or which component can gain access to a shared medium such as a shared bus.

The arbitration schemes can be responsive to the priority of the requesting  
15       component. Accordingly, more important components such as processors, digital signals processor and the like are associated with higher request priority. In deep pipelines devices' requests to receive a service can propagate through many pipeline stages before getting to the arbiter. These pipelines stages actually form a request queue in which lower priority request can be located before high priority  
20       requests.

There is a need to provide an efficient device having priority upgrade capabilities and methods for upgrading capabilities

### SUMMARY OF THE PRESENT INVENTION

25           A device having priority upgrade capabilities and a method for priority upgrading, as described in the accompanying claims.

### BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be understood and appreciated more fully from  
30       the following detailed description taken in conjunction with the drawings in which:

FIG 1A illustrates a device having priority upgrade capabilities according to an embodiment of the invention.

FIG. 1B illustrates modular components of a device having priority upgrade capabilities, according to an embodiment of the invention;

FIG. 2A illustrates an interconnect, according to an embodiment of the invention;

5        FIG. 2B illustrates an interconnect having pipeline limiting capabilities, according to an embodiment of the invention;

FIG. 3A illustrates a device having priority upgrade capabilities, according to an embodiment of the invention;

10       FIG. 3B illustrates a device having a pipeline limiting capabilities, according to an embodiment of the invention;

FIG. 4 illustrates an expander, according to an embodiment of the invention;

FIG. 5 illustrates a splitter, according to an embodiment of the invention;

15       FIG. 6 illustrates multiplexer and arbiter, according to an embodiment of the invention;

FIG. 7 illustrates a clock separator, according to an embodiment of the invention;

FIG. 8 illustrates a bus width adaptor, according to an embodiment of the invention;

20       FIG. 9 illustrates an arbitration method, according to an embodiment of the invention;

FIG. 10 illustrates method for priority updating, according to an embodiment of the invention; and

25       FIG. 11 illustrates method for pipeline limiting, according to an embodiment of the invention.

#### DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The following figures illustrate exemplary embodiments of the invention. They are not intended to limit the scope of the invention but rather assist in  
30       understanding some of the embodiments of the invention. It is further noted that all the figures are out of scale.

Conveniently, a device having priority update capabilities is provided. The device includes: (i) an arbiter adapted to arbitrate between transaction requests in response to priority attributes associated with the transaction requests; (ii) a first

sequence of pipeline stages that precede the arbiter, wherein at least one pipeline stage out of the first sequence is adapted to receive a request to update a priority of transaction requests stored within the first sequence to a requested priority; and wherein the device is adapted to update a priority of a transaction request stored in  
5 the first sequence if the transaction requests is priority upgradeable and if the requested priority is higher than a current priority of the transaction request.

Conveniently, a method for updating priorities is provided. The method includes: (i) receiving a request to update to a requested priority, priorities of transaction requests stored within a first sequence of pipeline stages that precede  
10 an arbiter; (ii) updating a priority level of a transaction request stored in the first sequence to the requested priority if the transaction request is priority upgradeable and if the requested priority is higher than a current priority of the transaction request; and (iii) arbitrating between transaction requests in response to priority attributes associated with the transaction requests.

15 According to an embodiment of the invention the device that has priority upgrade capabilities includes an interconnect. Device 10 can be a mobile device such as a mobile phone, music player, audio-visual device, personal data accessory, laptop computer, and the like. Device 10 can also be a stationary device such as a desktop computer, server, network node, and the like.

20 Device 10 can include multiple interconnect building blocks such as but not limited to expanders, arbiter and multiplexers, splitters, samplers, clock separators and bus width adaptors. Some of these building blocks can perform time based priority upgrade, can update priority in response to a request to upgrade priority and the like. An interconnect can include one arbiter that is  
25 followed (usually after other stages) another arbiter.

FIG 1A illustrates device 10 according to an embodiment of the invention.

Device 10 includes arbiter 800 that arbitrates between transaction requests in response to priority attributes associated with the transaction requests.

Device 10 includes a first sequence 18 of pipeline stages 18,1 – 14,4 that  
30 precede arbiter 800 and also includes a second sequence 19 of pipeline stages 19,1 – 19,5 that precede arbiter 800. Arbiter 800 arbitrates between a transaction request that is stored at the head of first sequence 18 and at the head of second sequence 19.

At least one pipeline stage out of first sequence is adapted to receive a request to update a priority of transaction requests stored within first sequence 18 to a requested priority. The request can originate from a requesting unit or from a time based priority upgrade mechanism. This pipeline stage can send the request  
 5 to the following stages of the pipeline. Such a pipeline stage can be an expander such as expander 600 of FIG. 1B.

For each transaction request stored in first sequence 18, device is adapted to update its priority if the transaction request is priority upgradeable and if the requested priority is higher than the current priority of the transaction request. The  
 10 same applies for transaction requests stored in second sequence 19.

In many cases high priority requests are stuck under lower priority requests. The arbiter is not award to the high priority requests that can eventually be served only after relatively a long time period. By upgrading the priority of all the priority upgradeable transaction requests the pending period of the originally  
 15 high priority request will be shortened.

TABLE 1 illustrates various priority upgrade scenarios. It is assumed that  $P1 < P2 < P3 < P4$ .

Pipeline stage	Current priority	Upgradeable ?	Requested priority	Result of priority upgrade
18,4	P1	Y	P3	P3
18,3	P4	N	P3	P4
18,2	P3	Y	P3	P3
18,1	P1	Y	P3	P3
19,5	P3	Y	P4	P4
19,4	P1	Y	P4	P4
19,3	P4	Y	P4	P4
19,2	P2	N	P4	P2
19,1	P3	N	P4	P3

TABLE 1

In order to add some fairness to the arbitration process the priority of an  
 20 upgradeable transaction request can be increases as its pending period increases. This mechanism is referred to as time base priority upgrade mechanism.

Those of skill in the art will appreciate that other time based priority level upgrading schemes can be applied without departing from the scope of the invention.

According to an embodiment of the invention a predefined timing threshold T1 is defined. When half of T1 passes the priority level is upgraded. When another fourth of T1 passes the priority level is further upgraded. When another eighth of T1 passes the priority level is further upgraded. This mechanism can be applied by using a multiplexer that has multiple inputs and one output. The output is connected to a priority upgrade period counter. When the counter is reset (or rolls over or reaches a predefined value) a priority upgrade occurs. The different inputs of the multiplexer are connected to different portions (offset by one bit) of a register that stores T1. The first input receives the whole T1. The second input receives T1 without its least significant bit (which equals T1/2). The third input receives T1 without its two least significant bits (T1/4), and so on. The multiplexer is controlled by a selection unit that alters its selection each time the priority upgrade period counter is reset.

FIG. 1B illustrates modular components 300-800 of device 10 according to an embodiment of the invention.

Conveniently, the modular components include: (i) expander 600, (ii) arbiter and multiplexer 800, (iii) splitter 500, (iv) sampler 700, (v) clock separator 300, and (vi) bus width adaptor 400.

It is noted that an interconnect does not necessarily include all these components. It is further noted that these components can also be used as stand-alone components in the integrated circuit. Those of skill in the art will appreciate that an inter connect can include multiple stages of these modular components.

According to an embodiment of the invention each of these modular components building blocks is using the same standard interface, such as to facilitate a glue-less connection between each of these components.

According to another embodiment of the invention each modular components can alter various attributes of various pending transaction requests. For example, various transaction requests can be associated with an arbitration priority that can be upgraded. Each modular component can upgrade the priority of the transaction request it stores, either in response to a request from another component or even apply a time based priority upgrade scheme.

Conveniently, at least one modular component can receive and generate signals that represent the beginning and/or end of the following phases: request and address phase, a data phase and an end of transaction phase.

Conveniently, at least one modular component can store one or more  
5 transaction request and also support multiple pending transaction requests that are stored in other components. For example, the expander 600 can receive up to sixteen transaction requests that were not followed by data phases and/or end of transaction phases, although it can store a more limited amount of requests.

Expander 600 allows a single master with a point-to-point interface to  
10 access a plurality of slaves, each with a point-to-point interface. The slave selection is based upon address decoding. Arbiter and multiplexer 800 allows a plurality of masters with a point-to-point interface to access a single slave with a point-to-point interface.

Splitter 500 allows a single master with a point-to-point interface to access  
15 a single slave with a point-to-point interface. The splitter 500 optimizes transactions according to the capabilities of the slave.

Sampler 700 allows a single master with a point-to-point interface to access a single slave with a point-to-point interface. It samples the transactions generated towards the slave. It is noted that the sampler 700 as well as other  
20 components can include one or more sampling circuits and optionally one or more bypassing circuit.

Clock separator 300 allows a single master with a point-to-point interface to access a single slave with a point-to-point interface. The master may operate in one clock domain while the slave operates in another clock domain. Bus width  
25 adaptor 400 allows a single master with a point-to-point interface to access a single slave with a point-to-point interface. The master's data bus width is different than the slave's data bus width.

Conveniently, each modular component out of components 200-800 includes an input interface and an output interface. For convenience of  
30 explanation these interfaces were illustrated only in FIG. 7 (input interface 305 and output interface 315) and in FIG. 8 (input interface 205 and output interface 215).

According to an embodiment of the invention multiple modular components out of components 200-800 includes a sampling circuit that can be

selectively bypassed by a bypass circuit. For convenience of explanation only FIG. 4 illustrates a sampling circuit 610 and a bypass circuit 612.

FIG. 2A illustrates device 11 having priority upgrade capabilities, according to an embodiment of the invention.

5           Device 10 includes interconnect 100. Interconnect 100 connects between M masters and S slaves. M and S are positive integers. The M masters are connected to M input ports 102(1) - 102(M) while the S slaves are connected to output ports 101(1) - 101(S). These input and output ports can support bi-directional traffic between masters and slaves. They are referred to input and  
10           output ports for convenience only. Conveniently, the input ports 102(1)- 102(M) are the input interfaces of the expanders 600(1) – 600(M) and the output ports are the output interfaces of splitters 500(1) – 500(S).

            Interconnect 100 includes M expanders 600(1) – 600(M), S arbiters and multiplexers 800(1) – 800(S) and S splitters 500(1) – 500(S). Each expander  
15           includes a single input port and S outputs, whereas different outputs are connected to different arbiter and multiplexers.

            Each arbiter and multiplexer 800 has a single output (that is connected to a single splitter) and M inputs, whereas different inputs are connected to different expanders 600. Each splitter 500 is connected to a slave.

20           It is noted that interconnect 100 can have different configuration than the configuration illustrated in FIG. 2. For example, it may include multiple samplers 700, clock separators 300 and bus width adaptors 400. These components can be required in order to support interconnects to slaves and masters that have different bus widths and operate in different frequencies.

25           Each splitter 500 is dedicated to a single slave. This splitter 500 can be programmable to optimized the transactions with that slave. Conveniently, each splitter 500 is programmed according to the slave maximal burst size, alignment and critical-word-first (wrap) capabilities.

            Interconnect 100 can operate as a low latency interconnect by utilizing the  
30           minimal amount of sampling circuits and bypassing other sampling circuits. It can also operate as latency insensitive interconnect.

            Conveniently, 100 is a non-blocking full fabric switch that supports per-slave arbitration, thus it enables maximal data bus utilization towards each of the slaves.



Each modular components of the interconnect 100 has a standard, point to point, high performance interface. Each master and slave is interfaced via that interface. These interface uses a three phase protocol. The protocol includes a request and address phase, a data phase and an end of transaction phase. Each of these phases is granted independently. The protocol defines parking grant for the request and address phase. The data phase and the end of transaction phase are conveniently granted according to the fullness of the buffers within the interconnect 100. The request is also referred to as transaction request. The end of transaction phase conveniently includes sending an end of transaction (EOT) indication.

For example, a master can send a write transaction request to an expander 600(1). The expander 600(1) can store up to three write transaction requests, but can receive up till sixteen write transaction requests, as multiple transaction requests are stored in other components of the interconnect. Thus, if it received the sixteenth write transaction request (without receiving any EOT or EOD signal from the master) it sends a busy signal to the master that should be aware that it can not send the seventeenth transaction request.

On the other hand, when the expander 600(1) stores the transaction request it sends an acknowledge to the master that can enter the data phase by sending data to the expander 600(1). Once the expander 600(1) ends to receive the whole data it sends a EOD signal to the master that can then end the transaction.

The expander 600(1) sends the transaction request to the appropriate arbiter and multiplexer. When the transaction request wins the arbitration and when the multiplexer and arbiter receives a request acknowledge signal then expander 600(1) sends the data it received to the splitter. Once the transmission ends the expander 600(1) enters the end of transaction phase. The splitter then executes the three-staged protocol with the target slave.

Interconnect 100 can use multiple sampling circuits, in order to interconnect between high frequency masters and remote slaves. The amount of sampling units affects the depth of the pipeline although the depth of the pipeline can be also responsive to other parameters such as but not limited to the buffering capabilities of the interconnect 100, and the like. The amount of sampling circuits can be increased by adding samplers, such as sampler 700 to interconnect 100, and/or by bypassing or not-bypassing the sampling circuitries within the

expanders, arbiters and multiplexers and the splitters. For example, the expander 600 includes a main sampler 640 as well as an address and attribute sampler 610 that can be bypassed.

Conveniently, interconnect 100 can terminate write transaction locally or  
5 let it be terminated by the slave. The write termination capability is enabled by an attribute that is associated with the transaction. In order to provide data coherency the slave should terminate the write transaction, otherwise the interconnect 100 can terminate the transaction locally.

Conveniently, the interconnect 100, and especially each arbiter and  
10 multiplexer 800 implements an arbitration scheme that can be characterized by the following characteristics: multiple (such as four) quality-of-service (or priority) levels, a priority upgrade mechanism, priority mapping, pseudo round robin arbitration, time based priority level upgrade, priority masking, weighted arbitration, and late decision arbitration.

15 The priority level is an attribute of each transaction. The arbiter includes a dedicated arbiter circuit per priority level. The priority upgrade mechanism allows a master (or another component) to upgrade a priority level of a pending transaction, based upon information that is acquired after the generation of that transaction request. The upgrade involves altering the priority attribute associated  
20 with the transaction request. The update can be implemented by the various components of the interconnect.

According to an embodiment of the invention some transaction requests can be labeled as non-upgradeable, while other transaction requests can be labeled as upgradeable. Non-upgradeable transaction requests are not upgraded  
25 during priority upgrade sessions.

Priority mapping allows to map master priority levels to slave priority levels or to a common priority level mapping. Pseudo round-robin arbitration involves storing the last arbitration winner and scanning a transaction request vector from the last arbitration winner until a current transaction request is  
30 detected.

Time based priority level upgrading includes updating the priority level of pending transaction requests in response to the time they are pending.

Conveniently, this feature reduces the probability of starvation. According to an embodiment of the invention a predefined timing threshold T1 is defined. When

half of T1 passes the priority level is upgraded. When another fourth of T1 passes the priority level is further upgraded. When another eighth of T1 passes the priority level is further upgraded. Those of skill in the art will appreciate that other time based priority level upgrading schemes can be applied without departing from the scope of the invention.

Priority masking includes selectively masking various request of predefined priorities, during predefined time slots. Conveniently, during one time slot the highest priority transaction requests are masked, during another timeslot the highest and the second highest priority transactions requests are blocked, and so on. Conveniently, some transaction requests can not be blocked, and during various time slots all the transaction requests are allowed. This guarantees a minimal arbitration winning slots for transactions with lower priorities, thus resolves potential starvation problems.

Weighted arbitration includes allowing an arbitration winner to participate in multiple consecutive transactions (transaction sequence) after winning an arbitration session. The weight can represent the amount of transactions that can be executed by an arbitration winner. Conveniently, if during the transactions sequence a higher priority transaction request wins the arbitration scheme then the transaction sequence stops.

Late decision arbitration includes determining a new arbitration winner substantially at the end of a currently executed transaction or substantially after a delay corresponding to the length of the current transaction ends.

Interconnect 100 is an ordered interconnect thus it does not require area-consuming re-order buffers. Conveniently, interconnect 100 is synthesized within a bounded centralized area generating star topology. This synthesis may require to add a small amount of buffers between interconnect 100 and the master and slaver that are connected to it. Nevertheless, this synthesis dramatically reduces the complexity of routing and further shortens the design and verification period.

Interconnect 100 has a relatively small area resulting in relatively low static power consumption. In addition, by applying power gating techniques the power consumption of interconnect 100 is further reduced.

Interconnect 100 includes multiple point-to-point interfaces (also referred to as ports) that inherently implement sampling. In addition interconnect 100

includes multiple sampling circuits that can be selectively bypassed, thus preventing low frequency filtering problems arising from long paths.

Interconnect 100 supports an ordered transaction protocol. In addition, to simplify implementation and eliminate reorder buffers, interconnect 100 does not  
5 generate transaction towards a new slave till all pending transaction towards that slave are completed. This behavior ensures that the order of transaction completion is the same of the order of transaction initiated. As a result the actual latency towards a certain slave may increase due to additional stall cycles.

According to another embodiment of the invention interconnect 100  
10 includes a relatively limited reorder mechanism that does not require to stall a transaction towards one slave until a previous transaction towards that slave is completed.

FIG. 2B illustrates device 12 having pipeline limiting capabilities, according to an embodiment of the invention.

15 Device 12 of FIG. 2B differs from device 11 of FIG. 2A by having multiple limiters 906(1)-906(M) and 905(1)- 905(S).

A limiter is places between a pair of modular components, such as between an expander and an arbiter and multiplexer or between an arbiter and multiplexer and between a splitter. The limiter participates in the handshake  
20 process between a pair of surrounding modular components. It receives a request to perform a transaction from one modular component. If the number of pending transaction request does not exceed a limiter threshold than the transaction request is sent to the second modular component. If, on the other hand, the number of pending transaction requests exceeds to the limiter threshold then the limiter does  
25 not pass the transaction request to the second modular component.

The limiter thresholds can be determined in view of various parameters including- the master or slave component that sends or receives the transaction requests, the priority of transaction requests, required throughput of the interconnect, latency parameters and the like.

30 For example, limiters that are connected to lower priority masters that are characterized by relatively long transactions can be set to lower limiter threshold.

Yet for another example, the limiter threshold can be set in view of required device throughput or latency. In interconnects that service many masters and slaves the optimal set of limiter thresholds can be hard to predict. In order to

assist in the limiter threshold definition the behavior of the device can be monitored, while different limiter thresholds are set.

FIG. 2B illustrates a set of monitors 956(1) – 956(M) and 955(1)- 955(S) that are connected to controller 98. Monitors 956(1) – 956(M) are connected to the inputs of expanders 600(1) – 600(M) while monitors 955(1)- 955(S) are connected to the outputs of splitters 500(1)-500(S). It is noted that limiters can be placed near the monitors and vice verse.

These monitors can monitor the traffic that passes by various junctions/buses and report their findings to a controller 98 that can determine the limiter thresholds.

It is noted that if the behavior of the device can depend upon the application that is being executed by the device. The limiter thresholds can also be set in response to the application that is being executed by the device.

Referring back to FIG. 2A, limiter 906(m,s) is positioned between the m'th expander 600(m) and the s'th arbiter and multiplexer 800(s), wherein index m ranges between 1 and M and index s ranges between 1 and S. Limiter 905(s) is positioned between the s'th arbiter and multiplexer 800(s) and the s'th splitter 500(s).

It is noted that limiters can be places between some modular components, while various other connections between modular components can be left without a limiter. For example, a limiter can be placed after expanders that are connected to low priority master components.

It is noted that at least some limiters can includes a fixed limiter threshold and are not connected to a controller such as controller 98.

FIG. 3A illustrates device 13 having priority upgrade capabilities, according to an embodiment of the invention.

Device 13 includes integrated circuit 11 that in turn includes a group of interconnects that includes interconnects 101, 102 and 103. The usage of multiple interconnects can be required when certain components should be connected by a low latency interconnect. If a single interconnect cannot provide such low latency then the components of the integrated circuit can be grouped to multiple groups. At least one group includes multiple components that are physically close to each other that are interconnected by a low latency interconnect. Interconnects 101 and 102 are low latency interconnects while interconnect 103 is a latency

insensitive interconnect. Conveniently, more sampling circuits are bypassed at interconnects 101 and 102 in comparison to interconnect 103.

Interconnect 101 interconnects a first group of components that includes processors 110 and 112 and shared on-chip memory 120. Interconnect 101 is also  
5 connected to interconnect 102 and interconnect 103. The two processors 110 and 112 are the masters of this interconnect.

Interconnect 102 interconnects a second group of components that includes processors 114 and 118 and shared on-chip memory 124. The two processors 114 and 118 are the masters of this interconnect.

10 Interconnect 103 interconnects a third group of components that includes DMA 122, external host interface (I/F) 116, peripherals 130, 132 and 134 and a memory controller 136. The memory controller 136 is connected to an off chip memory 190. Peripherals 130, 132 and 134 and memory controller 136 are the slaves of interconnect 103.

15 FIG. 3B illustrates device 14 having pipeline limiting capabilities, according to an embodiment of the invention.

Device 14 differs from device 13 of FIG. 3A by including monitors 951(1)- 951(6) and limiters 901(1)-901(6).

Monitor 951(1) and limiter 901(1) are connected between processor 110  
20 and interconnect 101. Monitor 951(2) and limiter 901(2) are connected between processor 112 and interconnect 101. Monitor 951(3) and limiter 901(3) are connected between processor 114 and interconnect 102. Monitor 951(4) and limiter 901(4) are connected between processor 118 and interconnect 102. Monitor 951(5) and limiter 901(5) are connected between interconnect 101 and  
25 DMA controller 122. Monitor 951(6) and limiter 901(6) are connected between interconnect 101 and memory controller 136.

These monitors and limiters can be connected to controller 99 that is adapted to evaluate the performance of device 14 and in response determine limiter thresholds.

30 FIG. 4 illustrates an expander 600, according to an embodiment of the invention.

Expander 600 includes input port 102, multiple (such as S) output ports 601-603, an address and attribute sampler 610, an address and priority translation unit

620, slave decoder 630, main sampler 640, de-multiplexer 650 and control unit 660.

The address and attribute sampler 610 can be bypassed. If it is not bypassed it samples the address and attributes lines.

5           Expander 600 supports priority upgrades of transaction requests that are stored in it. Thus, a priority attribute of a stored transaction request can be updated. The updated priority is taken into account by arbiters and multiplexers 800(1)-800(S). The upgrade can usually take place before the slave that is the target of the transaction acknowledges the transaction request.

10           The main sampler 640 includes a double buffer for all lines from the master to the slave (including address, write data and attribute lines). The double buffer allows to sample address, write data and attribute lines of a certain transaction before another transaction ends. The main sampler 640 provides a single buffer for the lines from the slave to the master (including, for example,  
15   read data).

          The main sampler 640 facilitates transaction priority upgrading and also time based priority upgrading. Time based priority upgrade involves increasing a priority of a pending transport request that is pending for more than a certain time threshold. Conveniently, multiple transaction priority upgrades can occur if the  
20   pending period exceed multiple time thresholds.

          The priority upgrading is conveniently initiated by a master and includes upgrading the priority of a certain pending transaction request (by altering the priority attribute). Conveniently, the priority attribute of other transaction requests that precede that certain transaction requests are also upgraded. This feature  
25   allows to maintain the order of requests while increasing the probability that a certain pipelines transaction request will be serviced before lower priority transaction requests. Conveniently, the controller 660 can control this priority upgrade, but this is not necessarily so.

          The address and priority translation unit 620 translates the upper bits of  
30   the address according to a predefined values. The priority translation involves translating master transaction priority levels to a slave transaction priority levels to common priorities levels. The translation can involve using a predefined transaction priority lookup table.

The slave decoder 630 receives an address over address lines and determines whether the transaction is aimed to a slave out of the S slaves that are connected to the interconnect or if the address is erroneous, based upon a predefined address range that is associated with each slave.

5           According to one embodiment of the invention the address ranges that are allocated to each slave are unique so that only one slave can be selected.

According to another embodiment of the invention the address ranges overlap but additional information such as slave priority are provided in order to resolve multiple matches between an input address and different address ranges.

10           Conveniently, the address ranges are stored in address registers located within the expander 600. Typically one address register stores the start address of the address range while the other address register stores the end address of the address range or an offset from the start address.

The de-multiplexer 650 sends data, address and attribute signals to the  
15   arbiter and multiplexer 800 that is connected, via a splitter 500, to the target slave.

The control unit 660 control the operation of the address and attribute sampler 610, address and priority translation unit 620, slave decoder 630, main sampler 640 and the de-multiplexer 650. The control unit 660 can control power gating techniques, and block transaction requests aimed to a certain target slave  
20   until a current transaction that is aimed to that certain target slave is completed. The transaction completion can be indicated by an end of transaction signal that is sent from the target slave.

Conveniently, the control unit 660 includes an access tracker, request generator, end of data indication generator and a transaction type tracking  
25   circuitry. The access tracker tracks transactions that did not end. The request generator sends transaction request signals towards target slaves. The end of data indication generator sends EOD indication towards the master. The transaction type tracking circuitry stores information that indicates the type (read, write, error, idle) of transactions that are currently during their data phase.

30           FIG. 5 illustrates a splitter 500, according to an embodiment of the invention.

Splitter 500 is adapted to receive data transactions from the master and convert them to one or more transactions towards the slave, and vice versa. The splitter 500 stores various slave transaction characteristics (also referred to as



attributes), such as maximal burst size, data burst alignment, wrap size, and the like. It then defines the translations towards the slave in response to these attributes. The splitter 500 also applies the three stage protocol towards the slave and towards the master. For example, if a master sends a data burst of 128 bits  
5 and the slave can receive data bursts of 32 bits then the splitter 500 converts this data burst to four slave data bursts.

The splitter 500 can be configured to be responsive to the slave transaction attributes (optimize mode) or as a sampling stage (sampler mode). In the sampler mode the splitter 500 only samples signals and sends them towards the slave. It is  
10 noted that the bus width of the input port and output port of the splitter 500 are the same, thus sampling mode can be easily executed.

The splitter 500 includes a data unit 510, a respond unit 520, a request unit 530 and a control/debug unit 540. The control/debug unit 540 controls the splitter and is also used during debug mode.

15 It is noted that other modular component of interconnect 100 includes a debug unit and/or a combined debug and control unit but for simplicity of explanation only FIG. 5 illustrates a debug unit.

The data unit 510 includes buffers that enable to exchange data between the master and slave. The respond unit 520 manages the end of transmission  
20 signal and the end of data signals. The request unit 530 performs the access optimization and manages other control signals.

The splitter 500 can store multiple transaction requests, and includes one sampling circuit as well as an optional sampling circuit that can be bypassed. The second sampling circuit is located within the request unit 530. Conveniently, two  
25 sampling circuits are activated when the splitter 500 wrap is enabled, or when the splitter 500 operates in an optimize mode.

Conveniently, when a write transaction occurs, the master sends a data burst to the splitter 500. The master also sends information reflecting the size of the burst, so that the splitter 500 can send an EOD signal towards the master once  
30 it received the whole data burst and the master-splitter data phase ends. It can also send an EOT signal once the master- splitter end of transaction phase ends. The EOD and EOT can be sent even if the data was not sent (or was not completely sent) to the slave. The splitter 500 sends data to the slave in one or more data beats, and used the three stage protocol. The slave sends to the splitter 500 EOD

and EOT signals once the splitter-slave data phase and the splitter-slave transaction end phase are completed.

According to an embodiment of the invention the splitter 500 can also support transaction priority upgrading and also time based priority upgrading.

- 5 These features can be required if the splitter 500 is followed by an arbiter.

FIG. 6 illustrates multiplexer and arbiter 800, according to an embodiment of the invention.

- Multiplexer and arbiter 800 includes multiple (such as M) input ports 801-803, output port output ports 812, an atomic stall unit 810, multiplexer 820,  
10 arbiter 830 and sampler 840. The atomic stall unit 810 receives transaction requests from various masters that are aimed to the same slave. Sampler 640 samples the arbitration result. It is connected between the multiplexer 820 and the arbiter 830.

- The arbiter 830 receives the transaction requests from the atomic stall unit  
15 810, master arbitration priority and master weights, a late arbitration control signal, and provides to the multiplexer 820 the arbitration winner and an indication that a transaction starts. The transaction start indication is responsive to a transaction acknowledgement signal sent from the splitter. The multiplexer 820 also receives the transaction requests and in response to the control signal from the  
20 arbiter 830 selects one of the pending transaction requests to be outputted to the splitter 500.

The arbiter 830 includes an arbiter engine 832, a request organizer 834 and a request generator 836.

- The request organizer 834 receives the transaction requests and their  
25 priority level and generates multiple request vectors, each vector represents the transaction requests that belong to a certain priority level. Each vector indicates the masters that sent pending transaction requests.

- The request generator 836 includes a masking unit 837 that selectively masks various transaction request of predefined priorities, during predefined time  
30 slots. For example, assuming that four priority levels exist, and that sixteen timeslots are defined. During two time slots the highest priority transaction requests are masked and the corresponding request vector is null. During two other time slots the two highest priority transaction requests are masked and the two corresponding request vectors are null. During one time slot only the lowest

priority level transaction requests are enabled and during the other time slots all the transaction requests are unmasked.

The request generator 836 also applies the weighted arbitration and the late decision arbitration, by sending to the arbiter engine 832 timing signals that indicate when to perform an arbitration cycle. For example, the request generator can receive an indication about the size of a data burst and the size of the data beat and determine when to trigger the next arbitration cycle. The request generator 836 is aware of the priorities of the pending transaction requests and can request an arbitration cycle if a higher priority request has arrived during a long transaction of a lower priority transaction request.

The request generator 826 also sends control signals such as master request signal and slave acknowledge signal in order to implement the three phase protocol.

The arbiter engine 832 includes multiple arbitration circuits, each associated with transaction requests that belong to the same priority level. The arbitration winner is the highest unmasked transaction request that won an arbitration cycle within the arbitration circuit.

The arbiter engine 832 receives multiple request vectors, each vector represents the transaction requests that belong to a certain priority level. Each vector indicates the masters that sent pending transaction requests. The arbiter engine 832 applies a pseudo round robin arbitration scheme, that takes into account only the winner of the last arbitration cycle.

Those of skill in the art will appreciate that other arbitration schemes, including well know arbitration schemes can be applied.

FIG. 7 illustrates a clock separator 300, according to an embodiment of the invention.

Clock separator 300 supports priority upgrading and also the three stage protocol. It includes an input and output interfaces as well as control path 310, data path 320 and a controller 330. The controller 330 controls the operation of the clock separator while the control path 30 is used to propagate transaction requests, control signals and attributes. These signals can include EOT signal, EOD signal, acknowledgement signals, transaction request signals and the like.

The controller 330 can receive indications about the mode of operation of the clock separator and control the clock separator 300 accordingly. For example,

the clock separator can operate in a bypass mode during which the input clock frequency and the output clock frequency are the same, in various modes in which there is a predefined relationship between the input and output clocks and the like.

5 The data path 320 includes two sampling circuits for write operations and one sampling circuit for read operations. The data path 320 usually includes a buffer for write operations and a buffer for read operations. The buffering allows to compensate for differences between the input and output clock frequencies.

10 The dashed vertical line 301 illustrates that the clock separator 300 components operate at an input frequency domain and an output frequency domain. It is noted that the frequencies can differ from each other but this is not necessarily so. The clock separator 300 can be used to synchronize between input and output clocks, reduce skew and/or jitter and the like.

FIG. 8 illustrates a bus width adaptor 400, according to an embodiment of the invention.

15 The bus width adaptor 400 supports priority upgrading and also the three stage protocol. It includes an input and output interfaces as well as control path 410, data path 420 and a controller 430. The controller 430 controls the operation of the bus width adaptor 400 while the control path 410 is used to propagate transaction requests, control signals and attributes. These signals can include EOT  
20 signal, EOD signal, acknowledgement signals, transaction request signals and the like.

The controller 430 can receive indications about the width of the different buses, alignment of data and timing parameters and control the bus width adaptor 400 accordingly. The data path 420 includes two sampling circuits for write  
25 operations and one sampling circuit for read operations. The data path 420 usually includes a buffer for write operations and a buffer for read operations. The buffering allows to compensate for differences between the input and output bus widths.

30 FIG. 9 illustrates an arbitration method 1100, according to an embodiment of the invention.

The arbitration method 1100 starts by stage 1110 of receiving at least one transaction request associated with at least one master, whereas all the transaction requests are associated with the same slave. Each transaction request is associated with a transaction request priority and a transaction request weight.

Stage 1110 is followed by stage 1120 of selectively masking the transaction requests. The selective masking can be applied in various time slots and can mask transaction requests of one or more priority, especially the higher priorities.

5           Stage 1120 is followed by stage 1130 of determining when to perform one or more arbitration cycles. The determination can be responsive to the length of a current transaction. Conveniently, the arbitration cycle is executed near the end of a current transaction. According to an embodiment of the invention there can be a time gap between the selection of an arbitration winner and the beginning of the  
10   data phase. This time gap usually occurs in read transaction, although this is not necessarily so. In write transactions the data to be transferred during the data phase is usually stored within interconnect when the arbitration takes place. In read transactions the data is usually stored within the slave when the arbitration cycle occurs. Thus, instead of waiting to the end of the data transfer in order to  
15   initiate the next arbitration cycle, the arbiter calculates the length of the currently approved data transfer and starts the next arbitration cycle after a delay that corresponds to that length.

          Stage 1130 is followed by stage 1140 of performing, for each priority level, an arbitration sequence between unmasked transaction requests.  
20   Conveniently, each arbitration cycle involves applying a pseudo round robin arbitration scheme. Stage 1140 provides an arbitration winner and also include calculating the amount of data beats that can be transferred by the winner.

          Stage 1140 is followed by stage 1150 of providing an indication about the arbitration winner. Stage 1150 can include determining the number of transactions  
25   that can be consecutively conducted by the arbitration winner. Said determination is usually responsive to the weight of the transaction request.

          Stage 1150 is followed by stage 1160 of determining when to perform the next arbitration cycle and jumping to stage 1110. It is noted that if stage 1110 is preceded by stage 1160 then stage 1130 can be skipped. It is noted that the even  
30   of a certain master won an arbitration cycle and is in the middle of a sequence of transactions then the sequence can be stopped if a higher priority transaction request won an arbitration cycle.

          Method 1100 also includes stage 1115 of updating the priority level of pending transaction requests. Stage 1115 can be executed during the execution of

other stages of method 1100. Conveniently, a priority update of a certain transaction request is blocked once the transaction request wins the arbitration, but this is not necessarily so. Stage 1115 can be time based and/or can be initiated by a master. The priority upgrade can include upgrading the priorities of

5 transaction requests that precede the certain transaction requests, especially those transaction requests that are stored at the same queue as the certain transaction request.

According to an embodiment of the invention the arbitration scheme is applied by a multiplexer and arbiter that participates in a three stage

10 communication protocol. Conveniently, the arbiter and multiplexer is a modular component that can be connected to other modular components such as to form an interconnect.

FIG. 10 illustrates method for priority updating 1000, according to an embodiment of the invention.

15 Stage 1115 of method 1100 can include at least some of the stages of method 1000.

Method 1000 starts by stage 1010 of receiving transaction requests and propagating the transaction requests through a first sequence of pipeline stages.

Stage 1010 is followed by stage 1030 of receiving a request to update to a

20 requested priority, priorities of transaction requests stored within a first sequence of pipeline stages that precede an arbiter.

Stage 1030 is followed by stages 1050 and 1060. Stage 1050 includes updating a priority level of a transaction request stored in the first sequence to the requested priority if the transaction request is priority

25 upgradeable and if the requested priority is higher than a current priority of the transaction request.

Conveniently, stage 1050 includes checking a priority level attribute and a priority upgradeability attribute associated with the transaction request.

30 Conveniently, stage 1050 includes updating priorities of transaction requests stored within multiple modular components (such as modular components 500, 600, 700, 800 of FIG. 1A, 2A, 2B, 3-8) that are adapted to support a certain point-to-point protocol.

Stage 1060 includes performing a priority level upgrade of a priority upgradeable transaction request in response to a time period the priority upgradeable transaction request is pending.

Conveniently, during repetitions of stages 1030-1080 stage 1060  
5 repetitively increments the priority level of upgradeable transaction requests wherein a time difference between consecutive priority upgrades is inversely proportional to a number of priority increases. For example, if the first time base priority upgrade occurs after a first pending period then the second time base priority upgrade occurs after a second pending period  
10 that is shorter than the first pending period.

Stage 1050 and 1060 are followed by optional stage 1070 of selectively masking transactions requests of various priorities before an arbitration cycle initiates.

Stage 1070 is followed by stage 1080 of arbitrating between transaction  
15 requests in response to priority attributes associated with the transaction requests. Stage 1080 is followed by stage 1010.

FIG. 11 illustrates method for pipeline limiting 1200, according to an embodiment of the invention.

Method 1200 starts by stage 1210 of setting limiter thresholds that define a  
20 maximal amount of pending transaction requests to be provided from one pipeline stage to another pipeline stage. Conveniently, the pipeline stages include modular components such as but not limited to expanders, arbiters and multiplexers, splitters, samplers, clock separators, but width adaptors and the like.

A limiter threshold is associated with one limiter, although this is not  
25 necessarily so. A limiter threshold can be set in response to the components connected to a limiter, to an expected program or application executed by a device, to previous performances of a device and to previous limiter thresholds, to priority of transaction requests, required throughput of the interconnect, latency parameters and the like.

30 Stage 1210 is followed by stage 1220 of executing an application that involves selectively transferring transaction requests from one stage of the pipeline to another in response to the limiter thresholds, arbitrating between transferred transaction requests, executing selected transaction requests (selected

by the arbitration), while monitoring the performance of a device that comprises pipeline limiters.

The application can be executed by a device such as interconnects 100-104 of previous figures.

5           Stage 1220 can start by stage 1221 of receiving, by a first limiter, a request to perform a transaction from one pipeline stage to another pipeline stages.

          Stage 1221 is followed by stage 1222 of selectively transferring transaction requests from one stage of the pipeline to another in response to the limiter thresholds, wherein stage 1222 may include determining if the number of  
10       pending transaction requests does not exceed a limiter threshold than stage 1220 includes sending the transaction request to the other pipeline stage, else delaying the provision of the transaction request to the other pipeline stage.

          Stage 1222 is followed by stage 1226 of arbitrating between transaction requests at a certain pipeline stage.

15           Stage 1226 is followed by stage 1228 of executing selected transaction requests provided by the arbitrating.

          Stage 1220 can also include stage 1227 of monitoring the traffic that passes by various junctions/ buses and stage 1229 of reporting these findings to a controller that can determine the limiter thresholds.

20           Conveniently, stage 1220 is followed by stage 1230 of comparing the performance of the device to required performances targets.

          Stage 1230 is followed by stage 1240 of determining whether to alter the limiter thresholds and jump to stage 1250 of altering the limiter thresholds in response to the comparison. Stage 1250 is followed by stage 1220.

25           Variations, modifications, and other implementations of what is described herein will occur to those of ordinary skill in the art without departing from the spirit and the scope of the invention as claimed. Accordingly, the invention is to be defined not by the preceding illustrative description but instead by the spirit and scope of the following claims.



## WE CLAIM

1. A device (10, 11, 12, 13, 14) having priority update capabilities, the  
5 device (10, 11, 12, 13, 14) comprises an arbiter (800) adapted to arbitrate between transaction requests in response to priority attributes associated with the transaction requests;  
wherein the device (10, 11, 12, 13, 14) is characterized by comprising a first sequence (18) of pipeline stages that precede the arbiter  
10 (800);  
wherein at least one pipeline stage (18,1) out of the first sequence (18) is adapted to receive a request to update a priority of transaction requests stored within the first sequence (18) to a requested priority; and  
wherein the device (10, 11, 12, 13, 14) is adapted to update a  
15 priority of a transaction request stored in the first sequence (18) if the transaction requests is priority upgradeable and if the requested priority is higher than a current priority of the transaction request.
2. The device (10, 11, 12, 13, 14) according to claim 1 wherein each  
transaction request is associated with a priority level attribute and a priority  
20 upgradeability attribute.
3. The device (10, 11, 12, 13, 14) according to any claim of claims 1-2 wherein the first sequence (12) of pipeline stages comprises multiple modular components (500, 600, 700, 800); whereas each modular component is adapted to support a certain point-to-point protocol.
- 25 4. The device (10, 11, 12, 13, 14) according to any claim of claims 1-3 further comprising a time based priority level upgrading module (16) adapted to upgrade a priority of an priority upgradeable transaction request in response to a time period the priority upgradeable transaction request is pending in the first sequence (12).
- 30 5. The device (10, 11, 12, 13, 14) according to claim 4 wherein the time based priority level upgrading module is adapted to perform multiple increments of the priority of the priority upgradeable transaction request, wherein a time difference between consecutive priority upgrades is inversely proportional to a number of priority increases.

6. The device (10, 11, 12, 13, 14) according to any claim of claims 1-5 wherein further adapted to selectively mask transactions requests of various priorities before an arbitration cycle initiates.
7. The device (10, 11, 12, 13, 14) according to any claim of claims 1-6  
5 wherein the first sequence (10, 11, 12, 13, 14) comprises components of multiple interconnects.
8. The device (10, 11, 12, 13, 14) according to any claim of claims 1-7 wherein the device (10, 11, 12, 13, 14) comprises multiple input ports (102(1) – 102(M)) and multiple output ports (101(1) – 101(S)), multiple modular  
10 components (500, 600, 700, 800); whereas each modular component is adapted to support a certain point-to-point protocol; whereas at least one modular component (600) comprises a sampling circuit (610) and a bypass circuit (612), whereas the sampling circuit is selectively bypassed by the bypass circuit.
9. The device (10, 11, 12, 13, 14) according to claim 10 wherein the multiple  
15 modular components comprise M expanders (600(1) – 600(M)), whereas different expanders are coupled to different masters, and wherein each expander is coupled in parallel to S arbiters and multiplexers (800(1) – 800(S)).
10. The device (10, 11, 12, 13, 14) according to any claim of claims 8-9 wherein the modular components further comprise S splitters (500(1) – 500(S)),  
20 wherein each splitter is adapted to optimize transactions towards a slave associated with the splitter.
11. A method (1000) for updating priorities, the method (1000) comprises: arbitrating (1080) between transaction requests in response to priority attributes associated with the transaction requests;  
25 the method (1000) is characterized by comprising: receiving (1030) a request to update to a requested priority, priorities of transaction requests stored within a first sequence of pipeline stages that precede an arbiter;  
updating (1050) a priority level of a transaction request stored in the first sequence to the requested priority if the transaction request is  
30 priority upgradeable and if the requested priority is higher than a current priority of the transaction request.
12. The method (1000) according to claim 11 wherein the updating (1050) comprises checking a priority level attribute and a priority upgradeability attribute associated with the transaction request.

13. The method (1000) according to any claim of claims 11-12 wherein the updating (1050) comprises updating priorities of transaction requests stored within multiple modular components (500, 600, 700, 800) that are adapted to support a certain point-to-point protocol.
- 5 14. The method (1000) according to any claim of claims 11-13 further comprising performing (1060) a priority level upgrade of a priority upgradeable transaction request in response to a time period the priority upgradeable transaction request is pending.
15. The method (1000) according to claim 14 wherein performing (1060)  
10 comprises repetitively incrementing the priority level of upgradeable transaction requests wherein a time difference between consecutive priority upgrades is inversely proportional to a number of priority increases.
16. The method (1000) according to any claim of claims 1-5 wherein further comprising selectively masking (1070) transactions requests of various priorities  
15 before an arbitration cycle initiates.

1/14

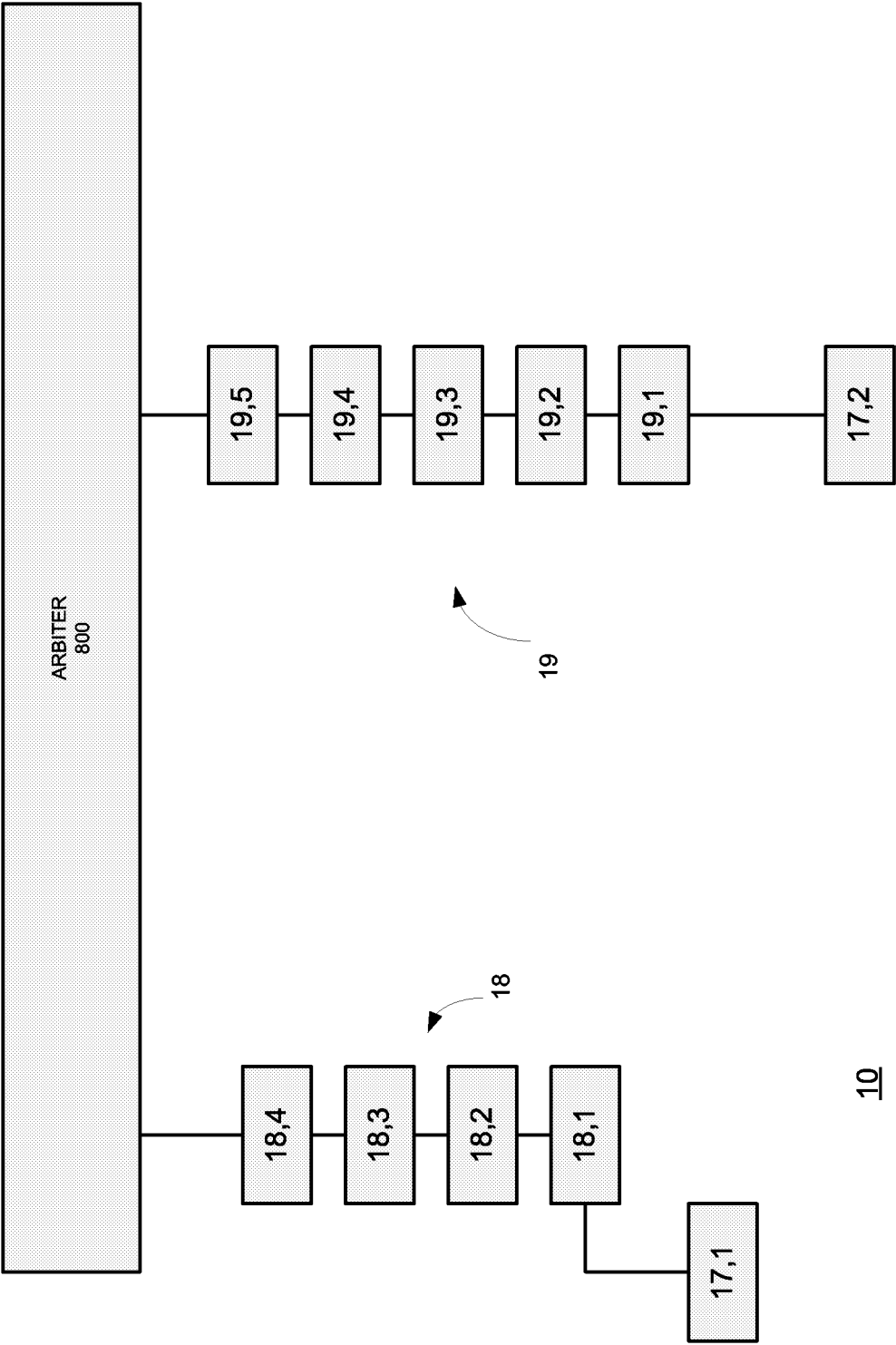


FIG. 1A

2/14

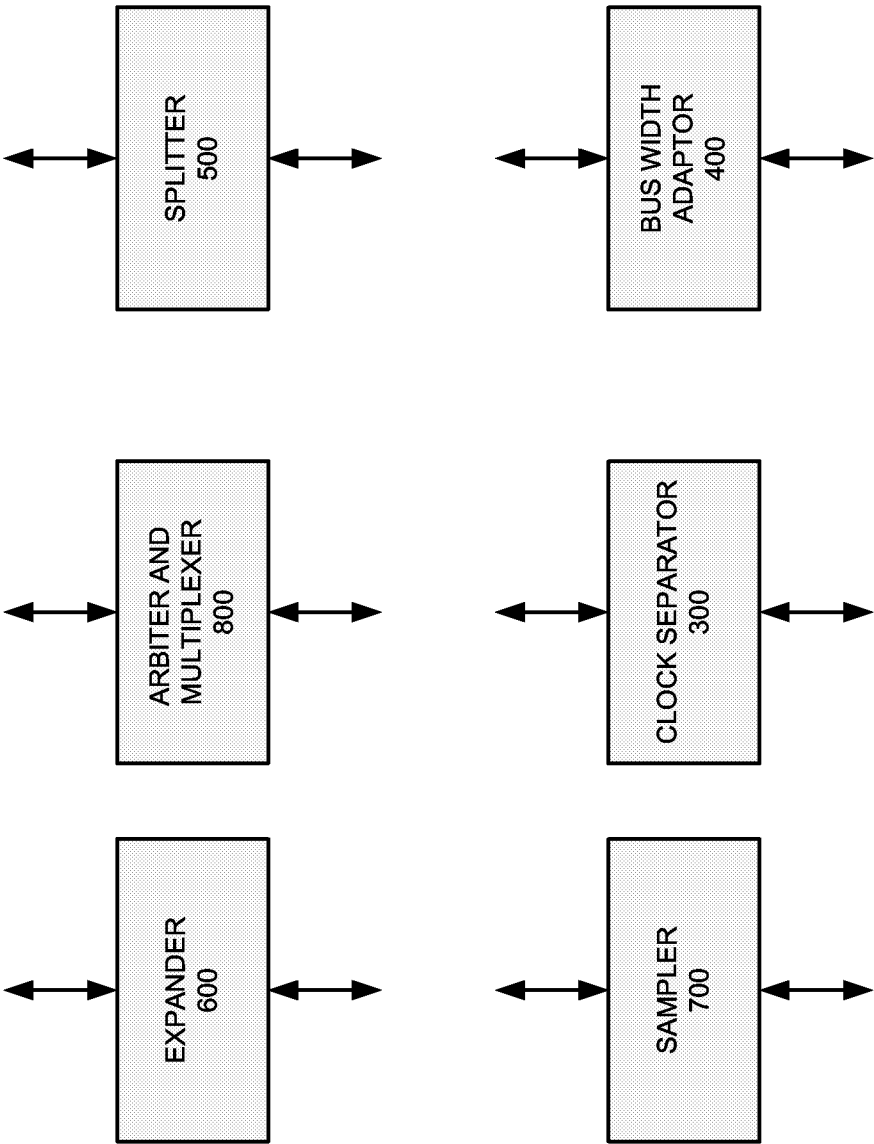


FIG. 1B

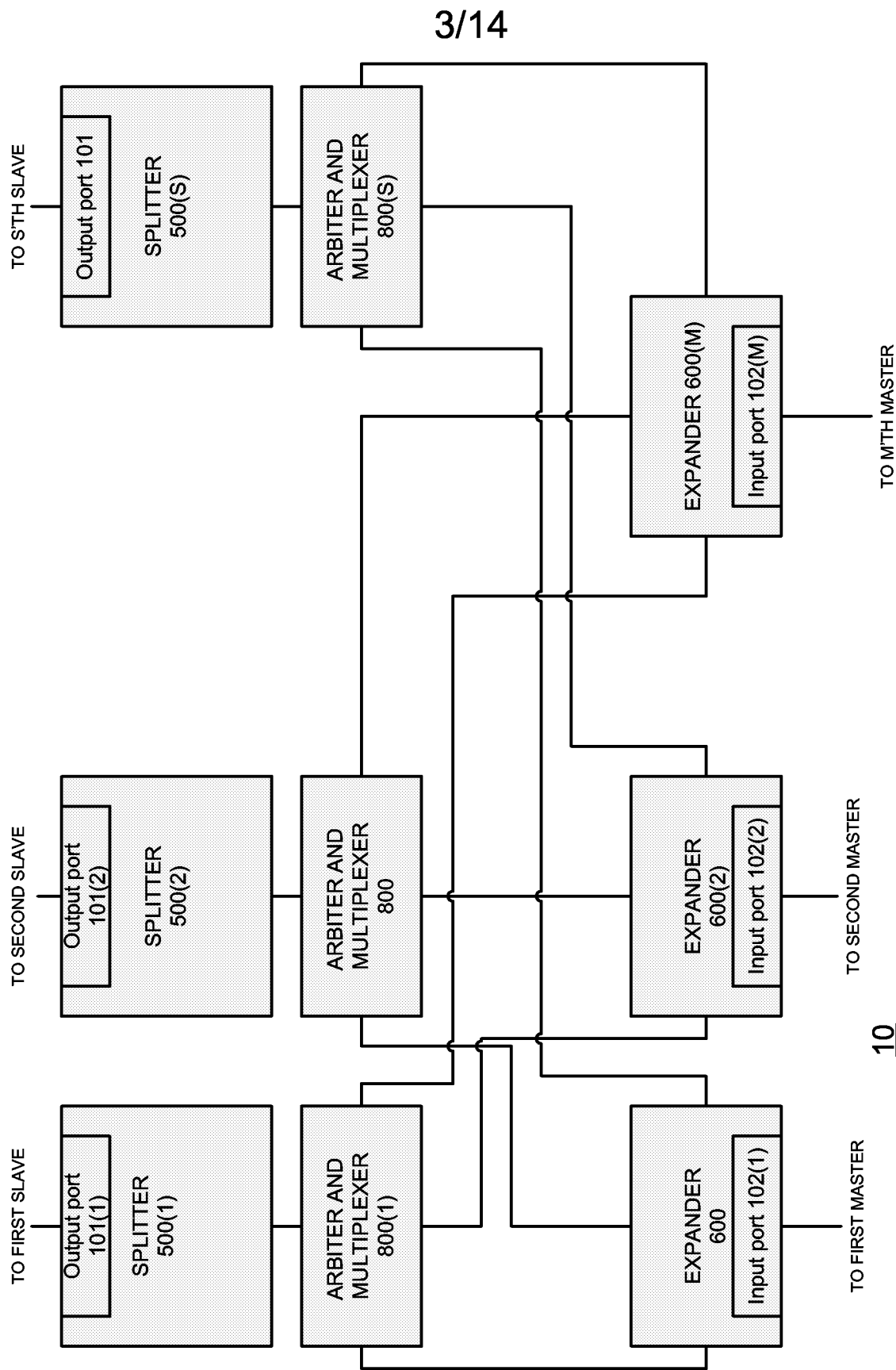


FIG . 2A

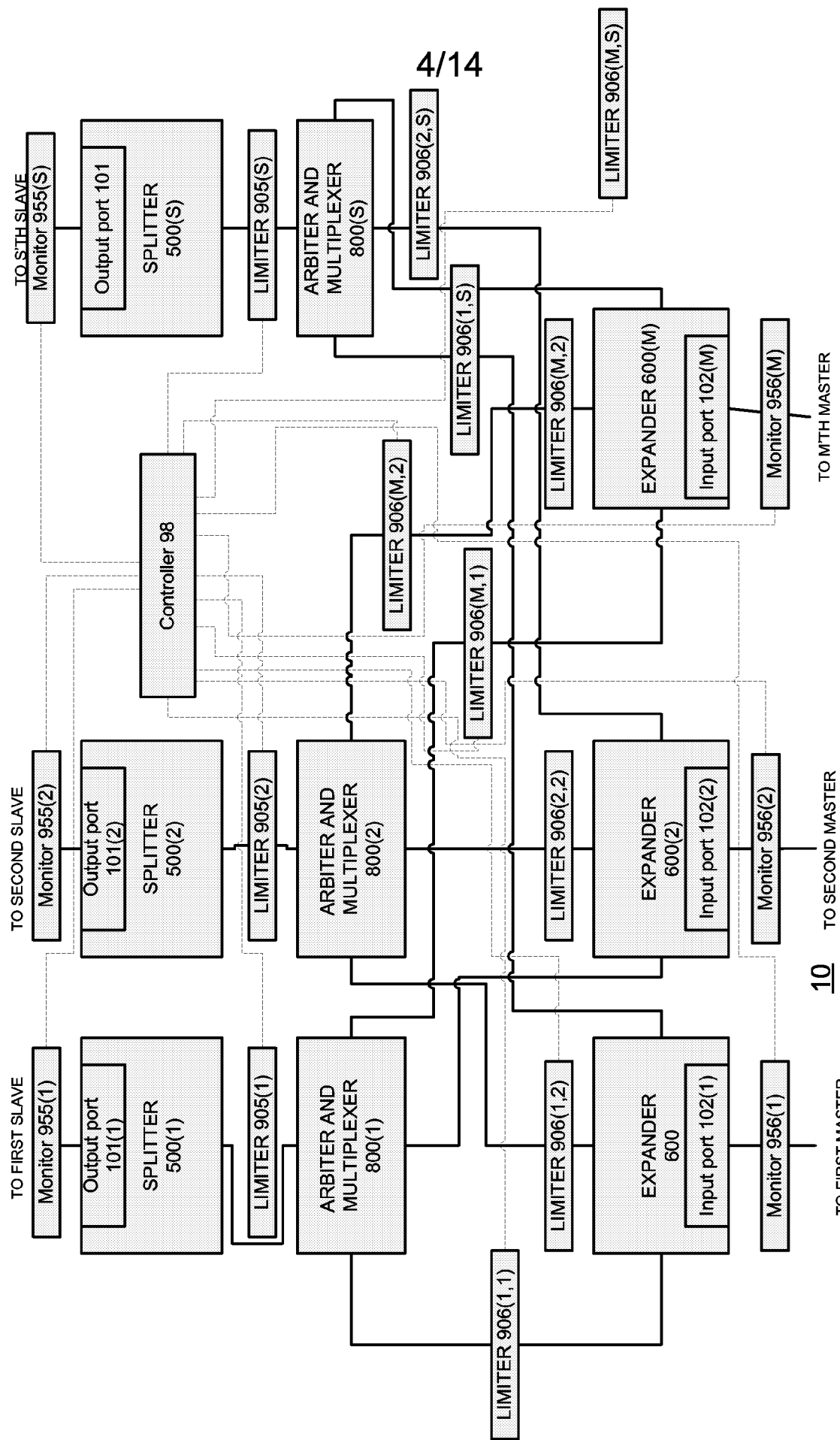


FIG. 2B

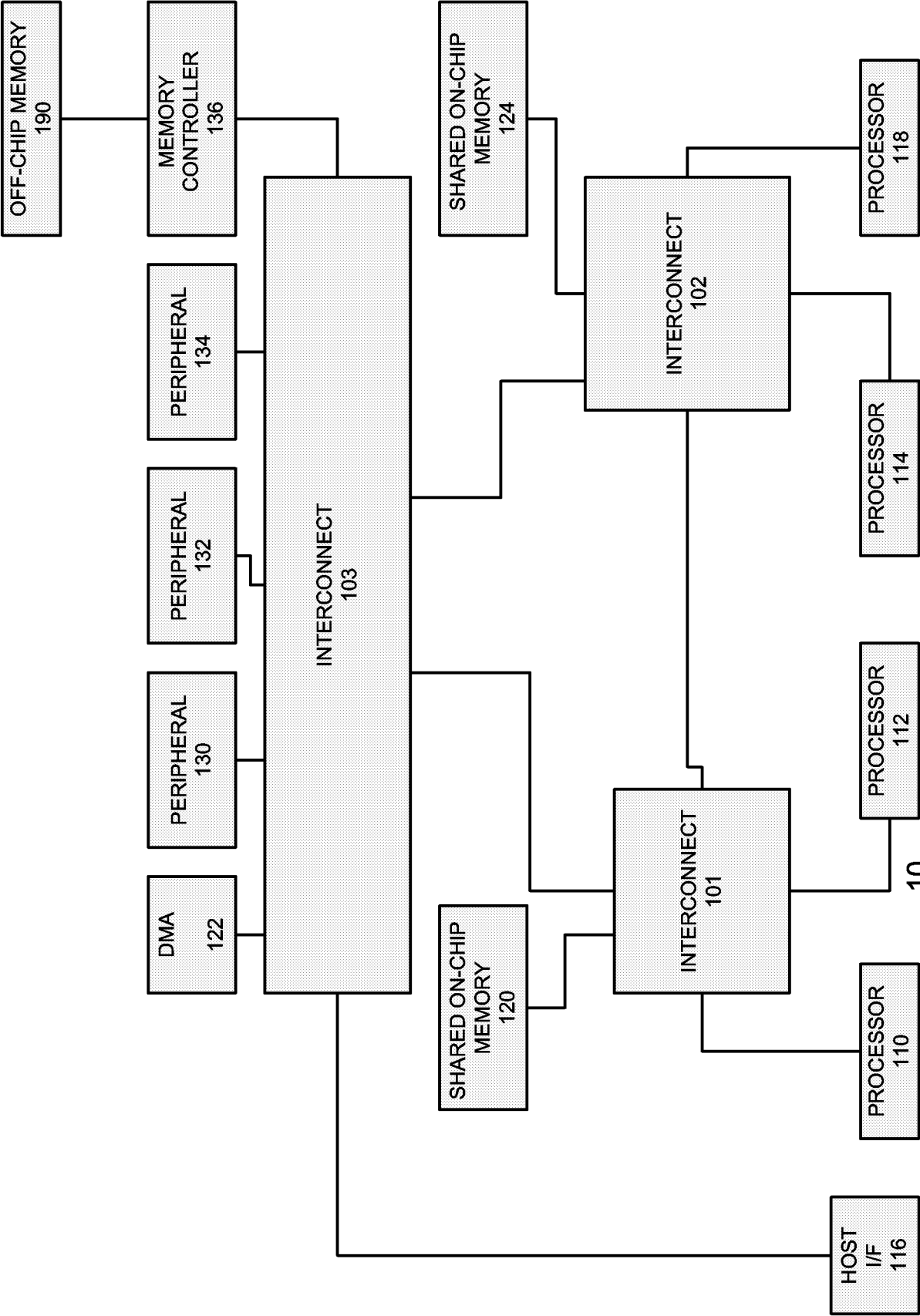


FIG. 3A



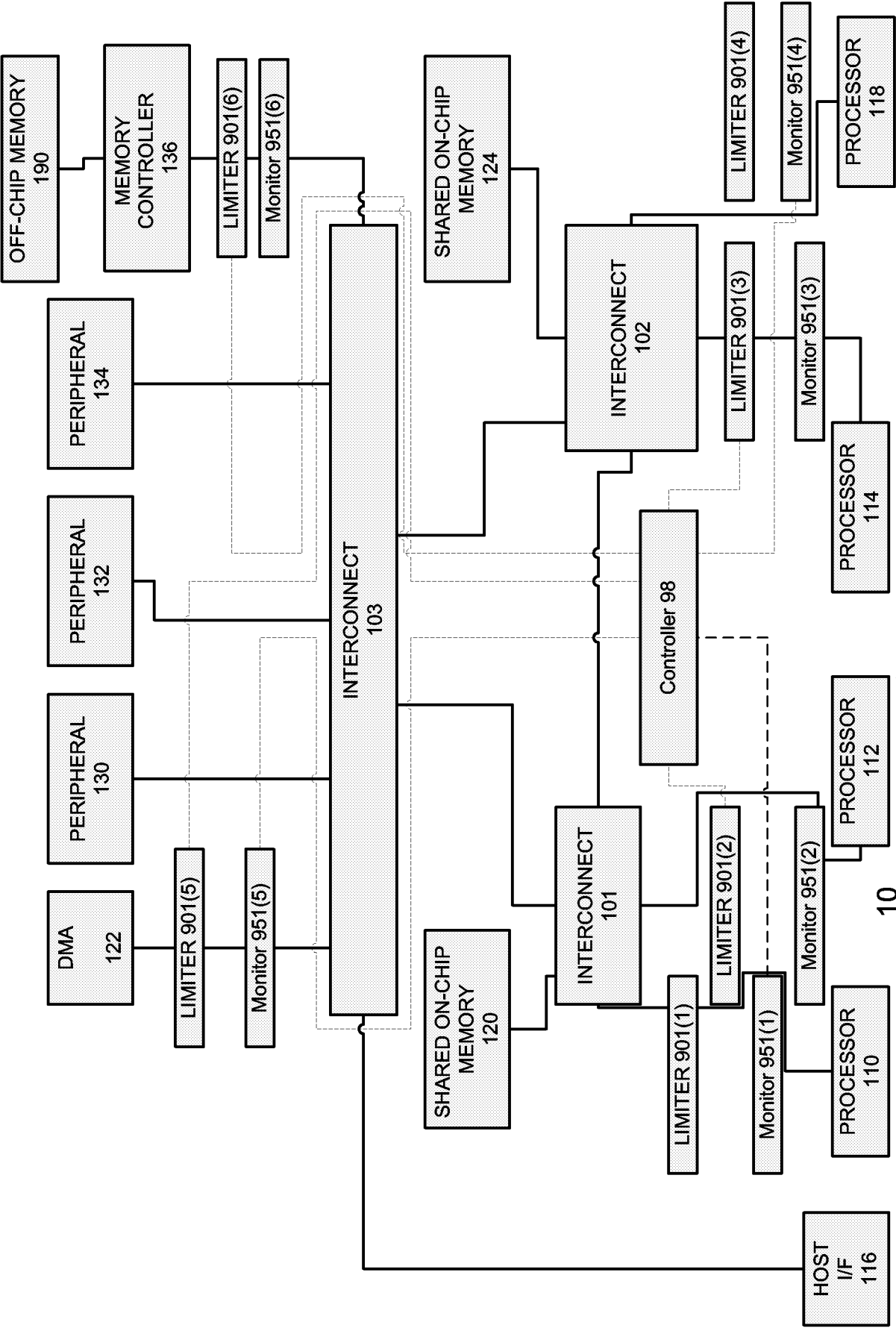


FIG. 3B

10

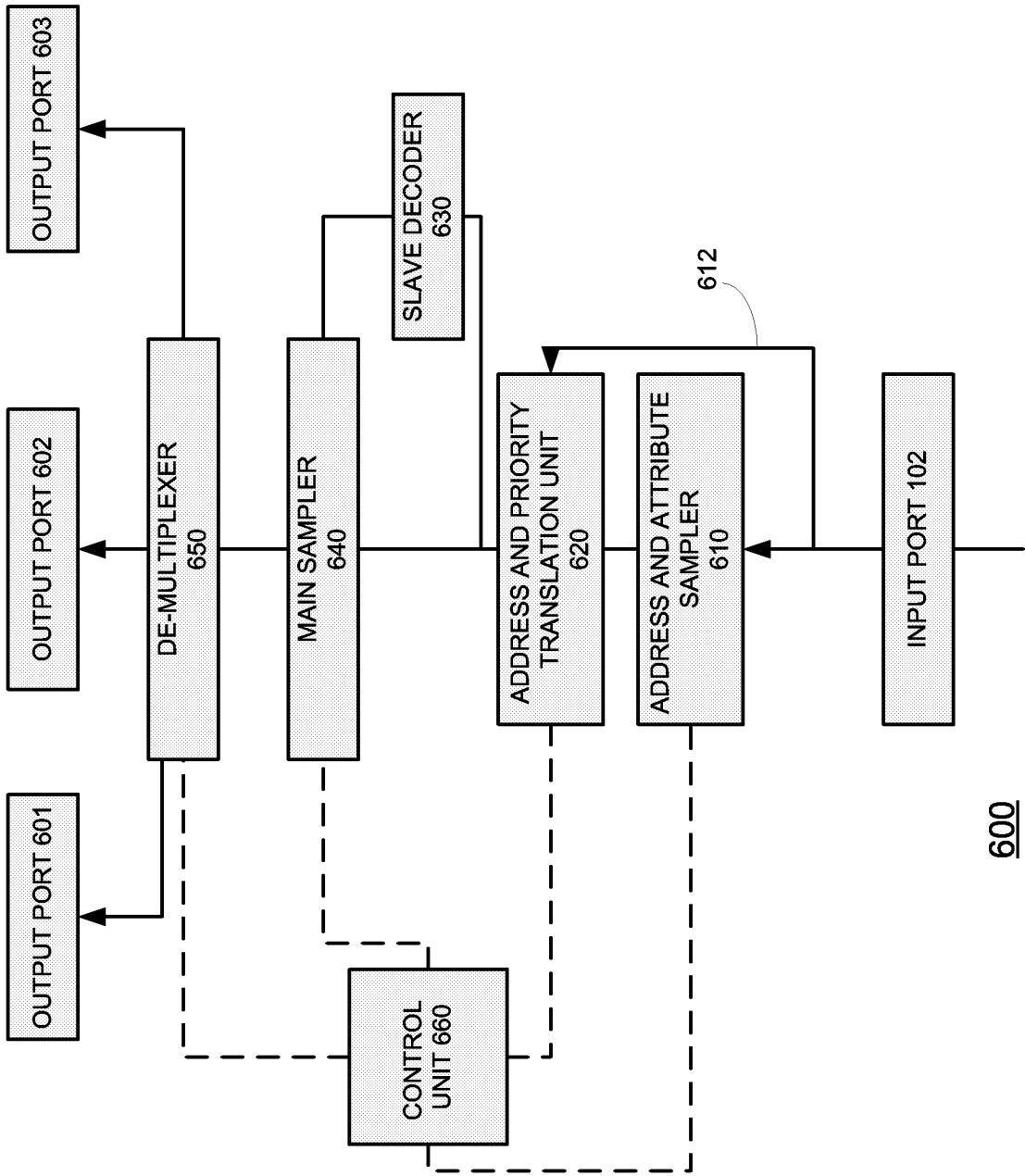
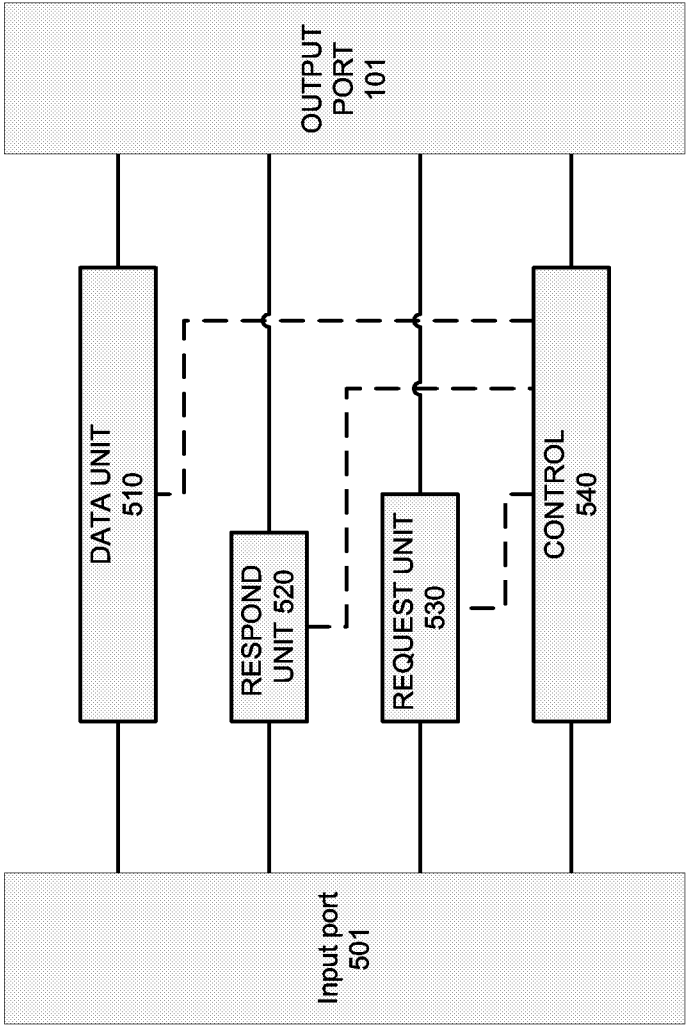


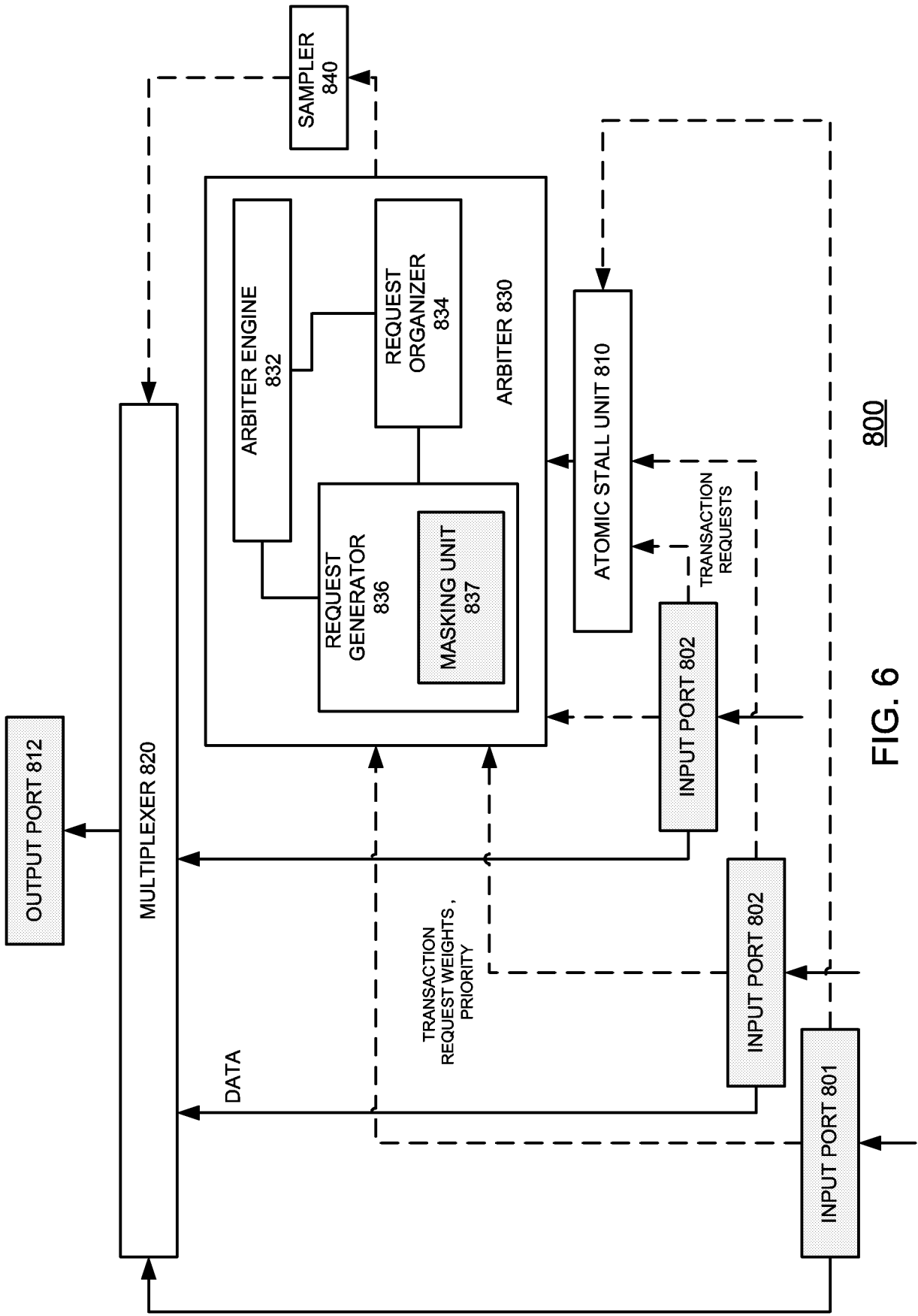
FIG. 4



500

FIG. 5

9/14



800

FIG. 6

10/14

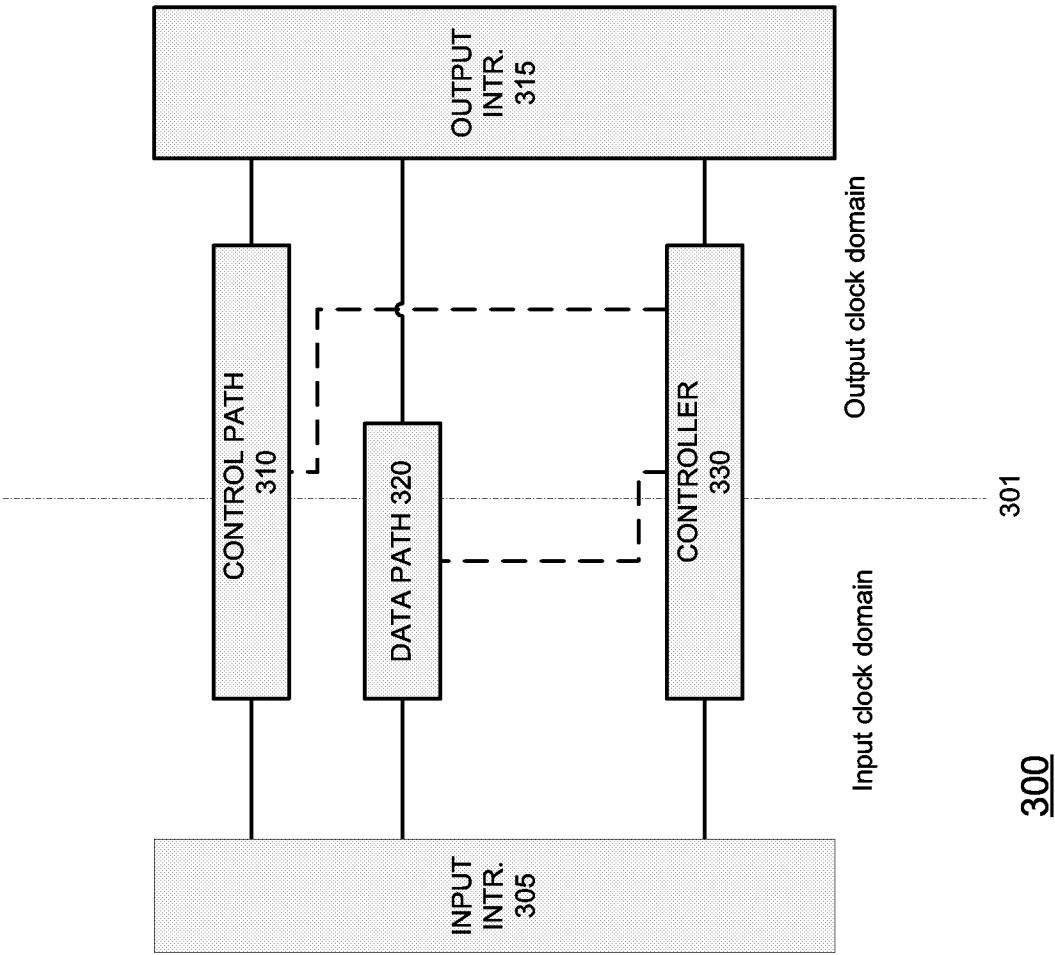
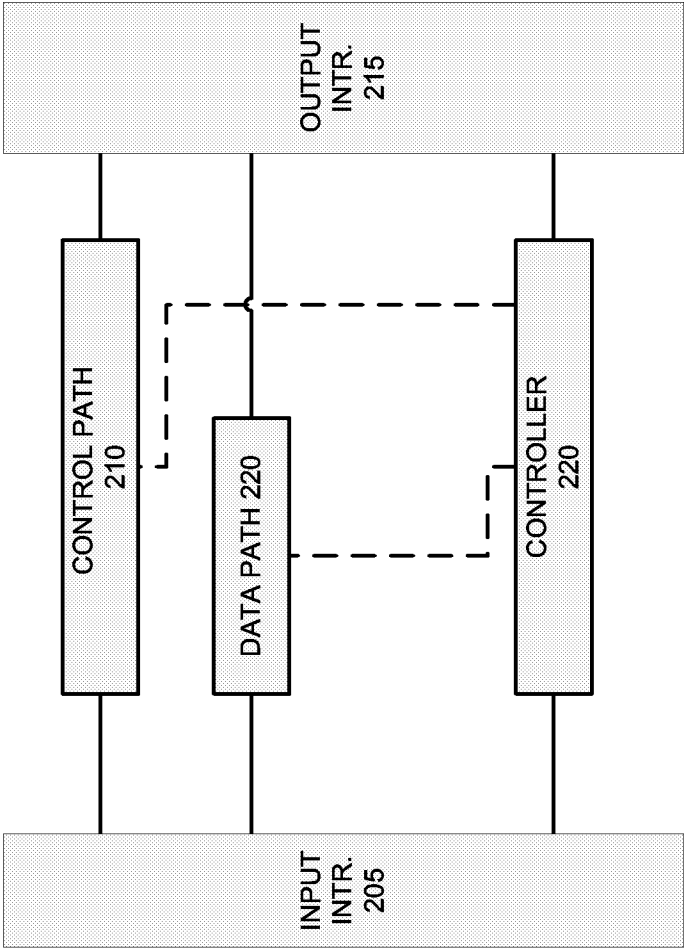
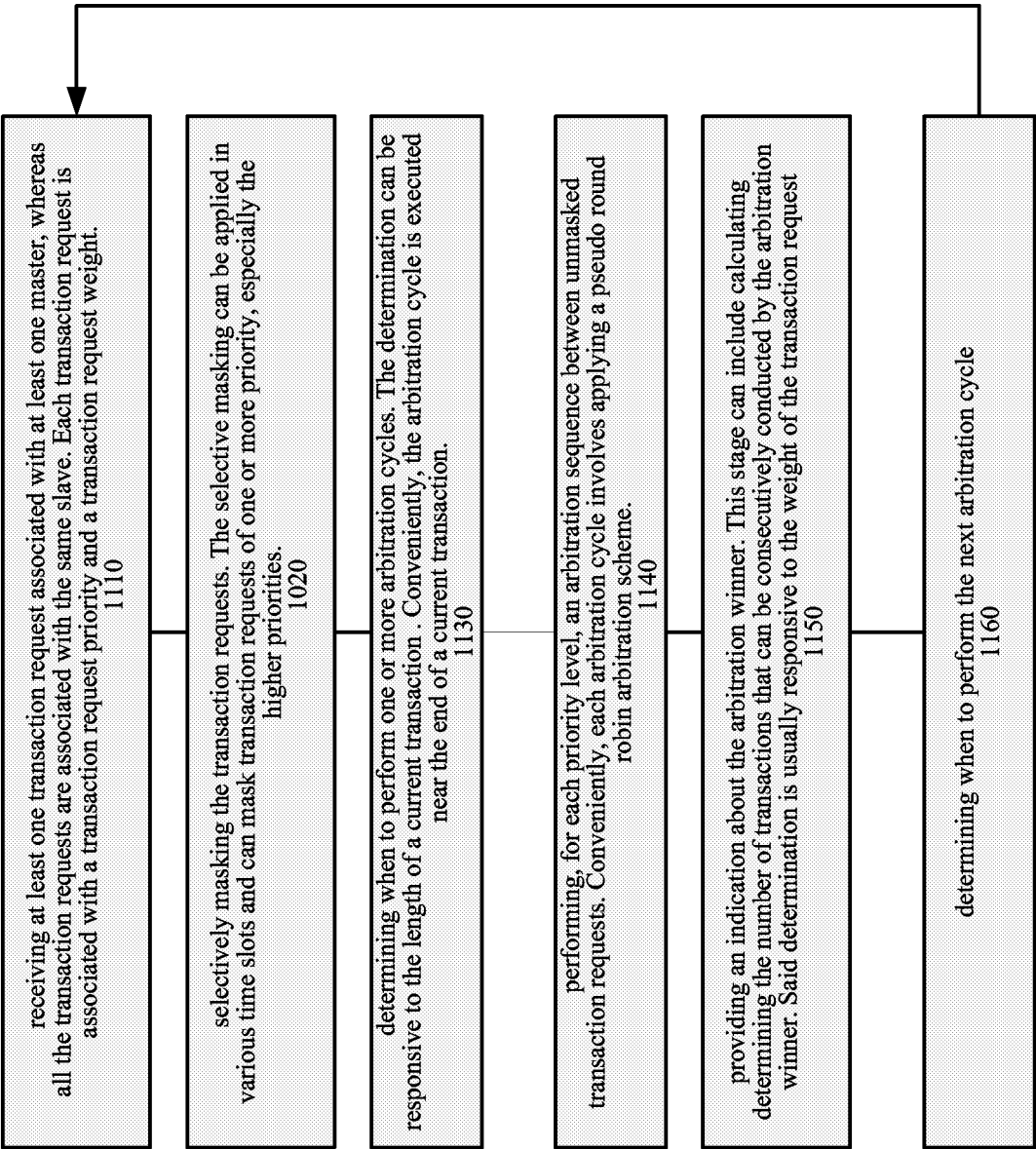


FIG . 7



200

FIG. 8



1000

FIG. 9

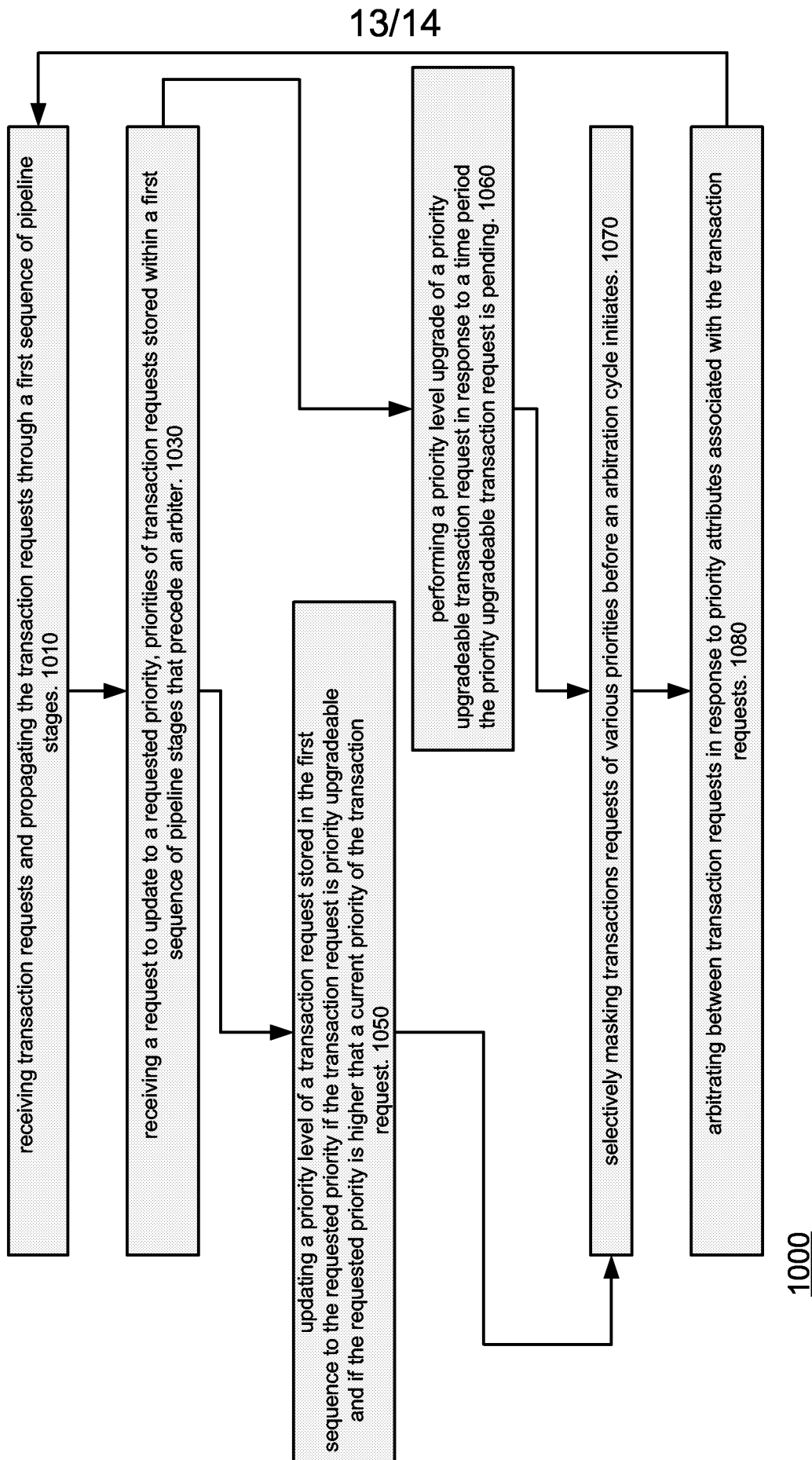


FIG. 10



14/14

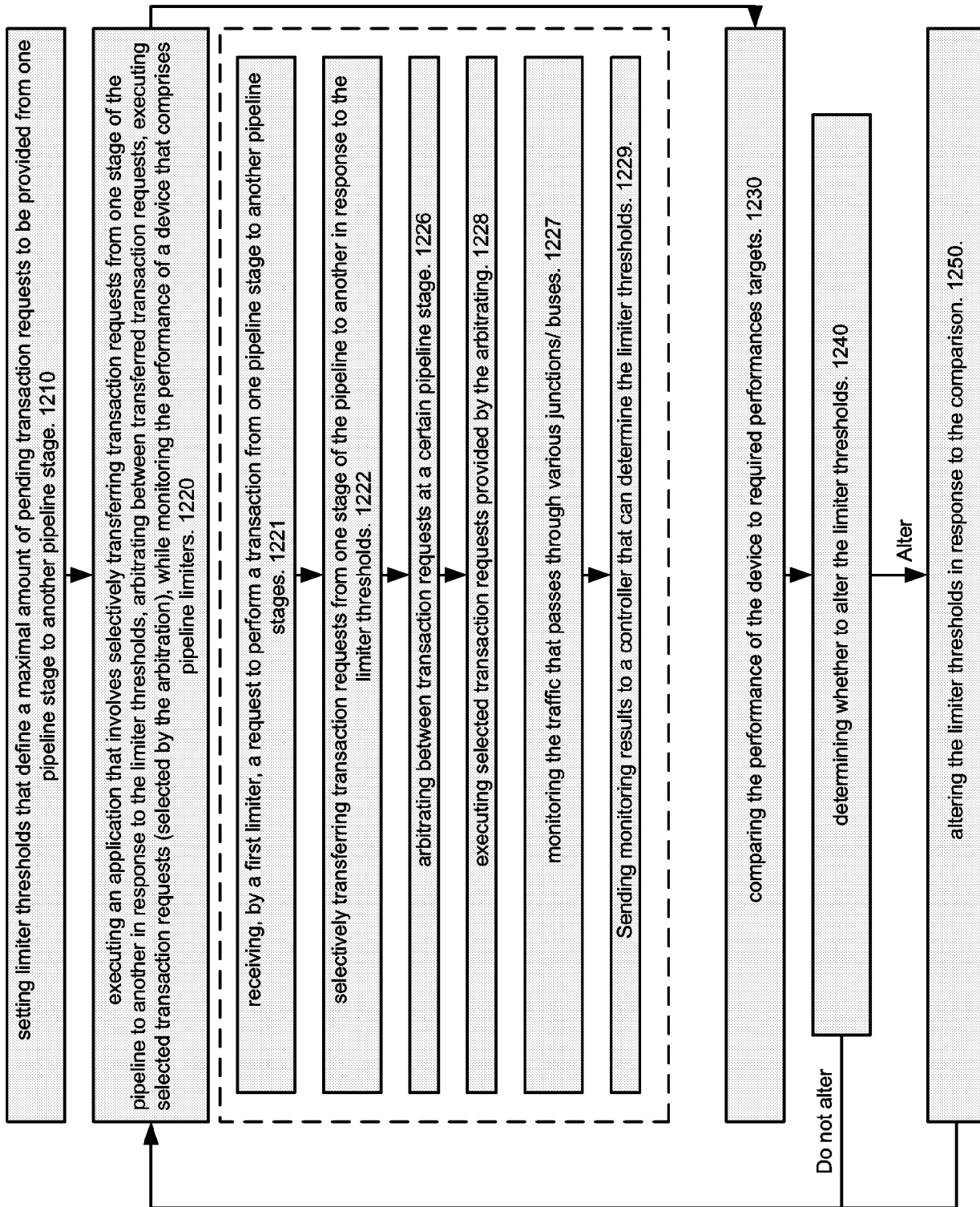


FIG. 11

# INTERNATIONAL SEARCH REPORT

International application No  
PCT/IB2006/052909

**A. CLASSIFICATION OF SUBJECT MATTER**  
INV. G06F13/362

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

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 596 749 A (CANTRELL JAY T [US] ET AL) 21 January 1997 (1997-01-21) abstract column 1, line 48 - line 59 column 6, line 4 - line 25 -----	1-16
A	US 5 796 961 A (O'BRIEN RITA M [US]) 18 August 1998 (1998-08-18) abstract -----	1-16
A	US 6 292 705 B1 (WANG YUANLONG [US] ET AL) 18 September 2001 (2001-09-18) abstract -----	1-16
A	US 6 658 485 B1 (BABER STEPHEN C [US] ET AL) 2 December 2003 (2003-12-02) abstract -----	1-16

☐ 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 document 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

18 April 2007

Date of mailing of the international search report

03/05/2007

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Sentier, Ludovic

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/IB2006/052909

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5596749	A	21-01-1997	NONE
US 5796961	A	18-08-1998	NONE
US 6292705	B1	18-09-2001	US 6466825 B1 15-10-2002
US 6658485	B1	02-12-2003	NONE