US 20080201588A1

(19) **United States**

(12) **Patent Application Publication**

PYEON et al.

(10) **Pub. No.: US 2008/0201588 A1**

(43) **Pub. Date: Aug. 21, 2008**

(54) **SEMICONDUCTOR DEVICE AND METHOD FOR REDUCING POWER CONSUMPTION IN A SYSTEM HAVING INTERCONNECTED DEVICES**

(75) Inventors: **Hong Beom PYEON**, Kanata (CA); **Jin-Ki Kim**, Kanata (CA); **HakJune Oh**, Kanata (CA)

Correspondence Address:
MOSAID TECHNOLOGIES INCORPORATED
11 HINES ROAD
KANATA, ON K2K-2X1

(73) Assignee: **MOSAID Technologies Incorporated**, Kanata (CA)

(21) Appl. No.: **12/018,272**

(22) Filed: **Jan. 23, 2008**

Related U.S. Application Data

(60) Provisional application No. 60/902,003, filed on Feb. 16, 2007, provisional application No. 60/891,108, filed on Feb. 22, 2007, provisional application No. 60/943,442, filed on Jun. 12, 2007.

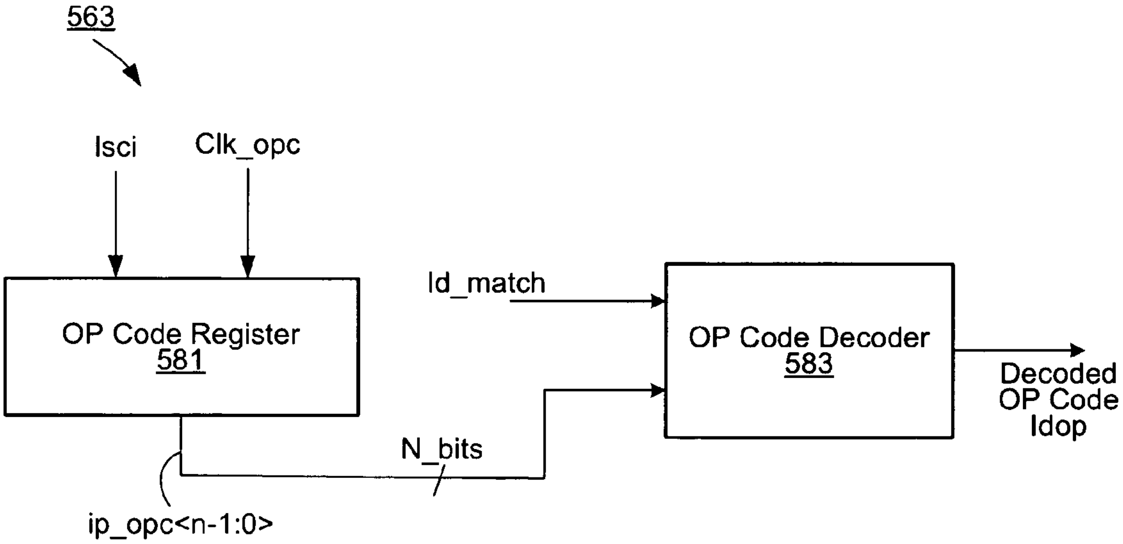
Publication Classification

(51) **Int. Cl.**
G06F 1/32 (2006.01)

(52) **U.S. Cl.** **713/320**

(57) **ABSTRACT**

A system includes a plurality of memory devices connected in-series that communicate with a memory controller. A memory device designated by an ID number performs operations at a normal power consumption level. The other devices not designated perform signal forwarding operations at a reduced power consumption level. The designated memory device enables its internal clock generator to generate all clocks necessary for operations. The non-designated memory devices generate clocks to perform partial operations for forwarding commands to next memory devices. In another example, memory devices do not forward the input command to the next memory device when there is no ID match. In another example, a memory device transmits the command replacing the content thereof with a static output when there is an ID match. Such partial clock generation, non-forwarding of commands and replacing the command contents will cause the system to operate at the reduced power consumption level.



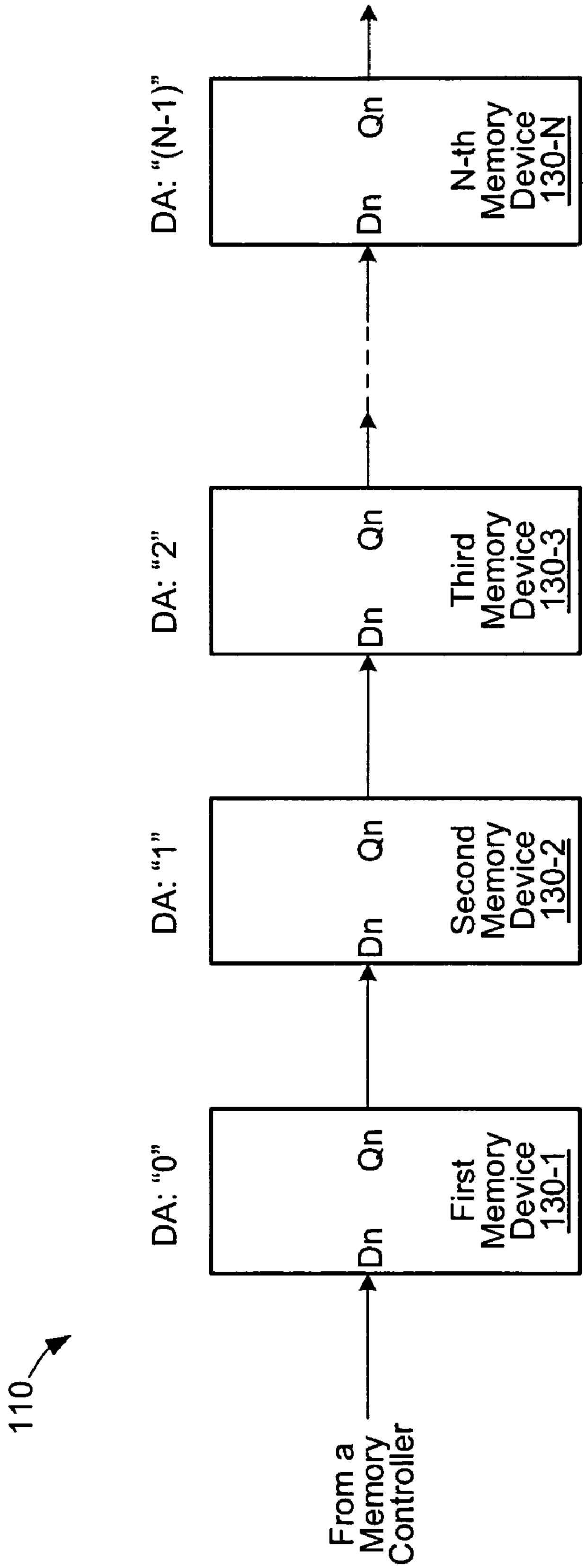

110
From a Memory Controller
DA: "0"
Dn
Qn
First Memory Device 130-1
DA: "1"
Dn
Qn
Second Memory Device 130-2
DA: "2"
Dn
Qn
Third Memory Device 130-3
DA: "(N-1)"
Dn
Qn
N-th Memory Device 130-N

FIG. 1

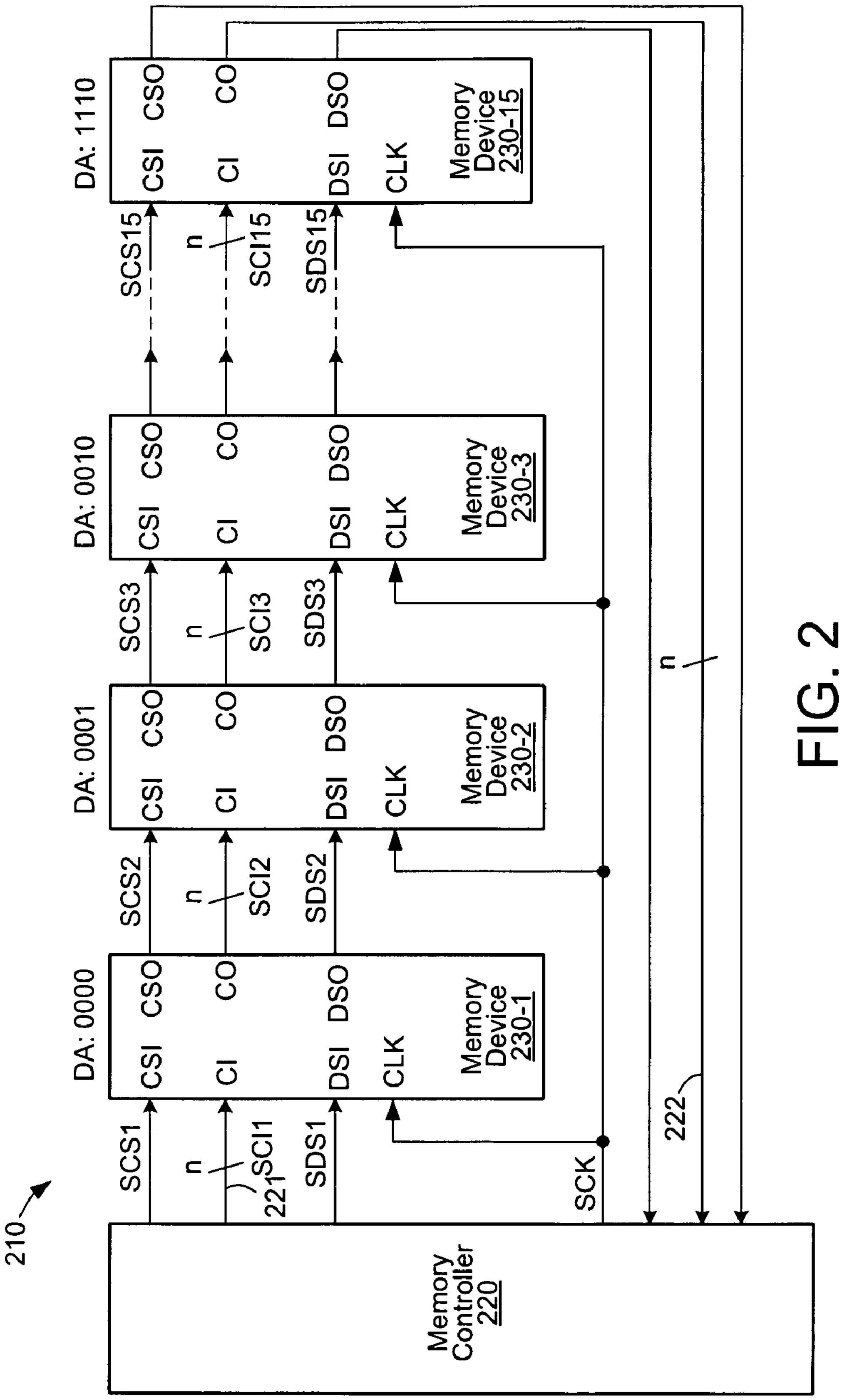

210
Memory Controller 220
SCS1
n
SCI1
221
SDS1
SCK
222
n
DA: 0000
CSI CSO
CI CO
DSI DSO
CLK
Memory Device 230-1
SCS2
n
SCI2
SDS2
DA: 0001
CSI CSO
CI CO
DSI DSO
CLK
Memory Device 230-2
SCS3
n
SCI3
SDS3
DA: 0010
CSI CSO
CI CO
DSI DSO
CLK
Memory Device 230-3
SCS15
n
SCI15
SDS15
DA: 1110
CSI CSO
CI CO
DSI DSO
CLK
Memory Device 230-15

FIG. 2

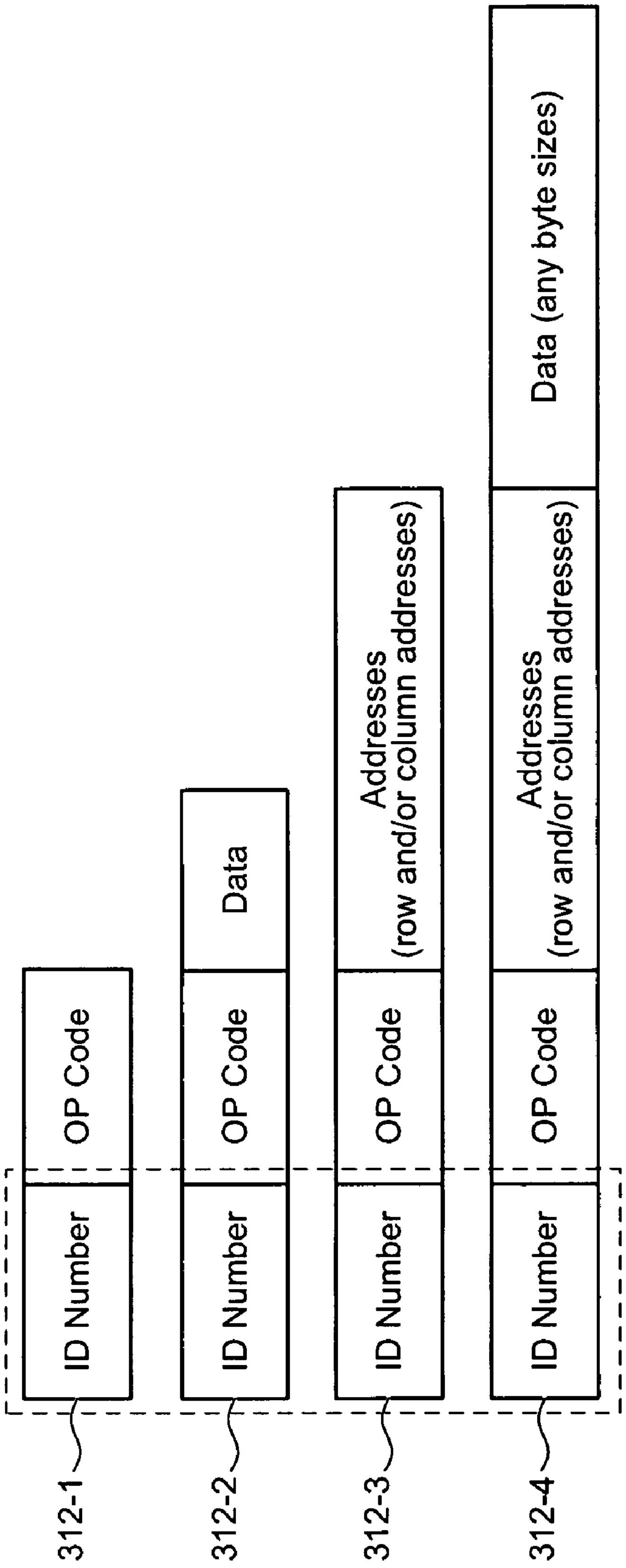

312-1
ID Number
OP Code
312-2
ID Number
OP Code
Data
312-3
ID Number
OP Code
Addresses (row and/or column addresses)
312-4
ID Number
OP Code
Addresses (row and/or column addresses)
Data (any byte sizes)

FIG. 3

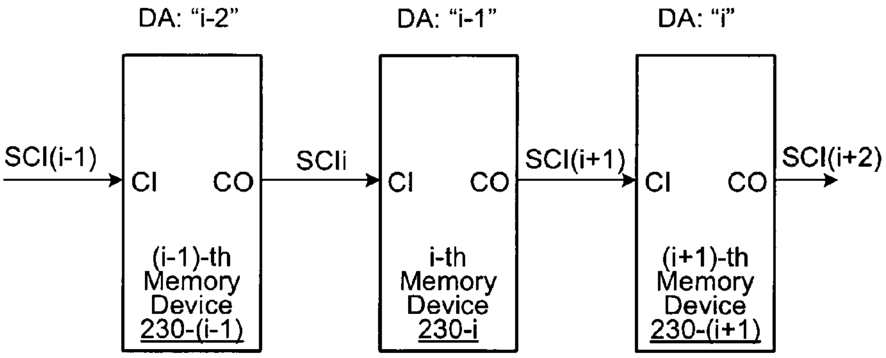

DA: "i-2"
DA: "i-1"
DA: "i"
SCI(i-1)
CI
CO
SCIi
CI
CO
SCI(i+1)
CI
CO
SCI(i+2)
(i-1)-th Memory Device 230-(i-1)
i-th Memory Device 230-i
(i+1)-th Memory Device 230-(i+1)

FIG. 4

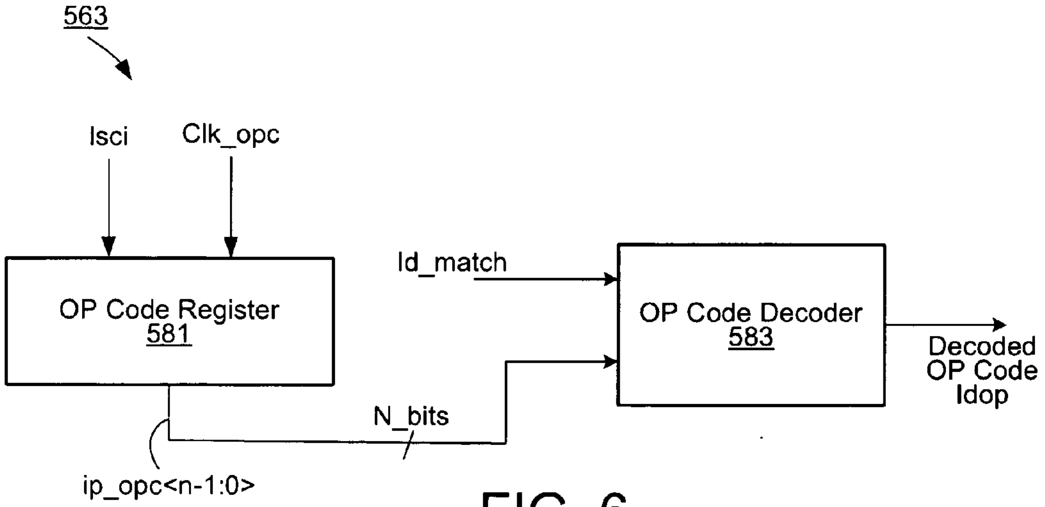

563
Isci
Clk_opc
OP Code Register 581
Id_match
OP Code Decoder 583
Decoded OP Code Idop
N_bits
ip_opc<n-1:0>

FIG. 6

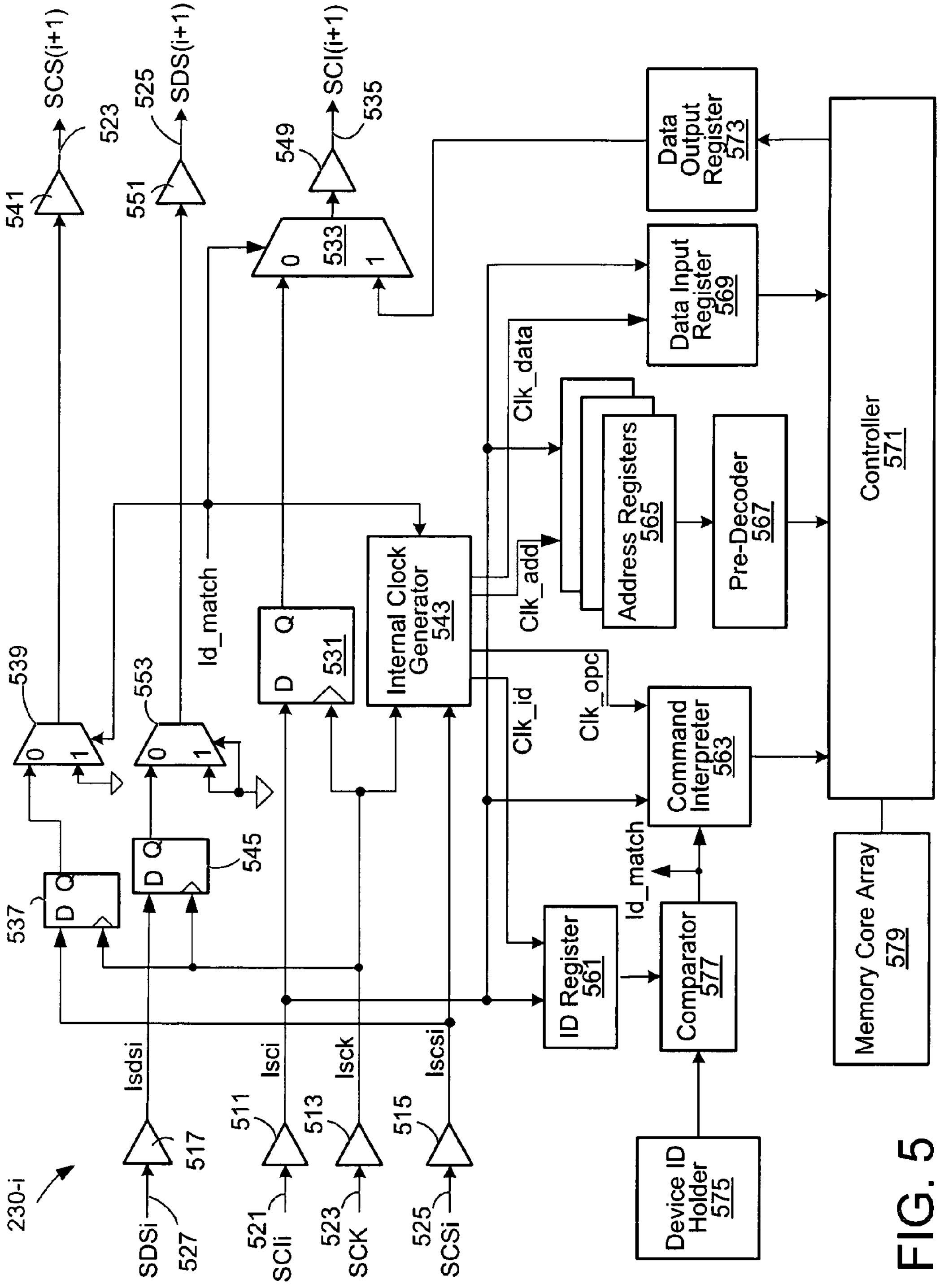

230-i
SDSi
527
517
Isdsi
521
SCIi
511
Isci
523
SCK
513
Isck
525
SCSi
515
Iscsi
537
D Q
545
D Q
539
0
1
553
0
1
541
523
SCS(i+1)
551
525
SDS(i+1)
Id_match
D Q
531
549
535
SCI(i+1)
0
533
1
Internal Clock Generator 543
Clk_id
Clk_opc
Clk_add
Clk_data
ID Register 561
Comparator 577
Id_match
Device ID Holder 575
Command Interpreter 563
Address Registers 565
Pre-Decoder 567
Data Input Register 569
Data Output Register 573
Controller 571
Memory Core Array 579

FIG. 5

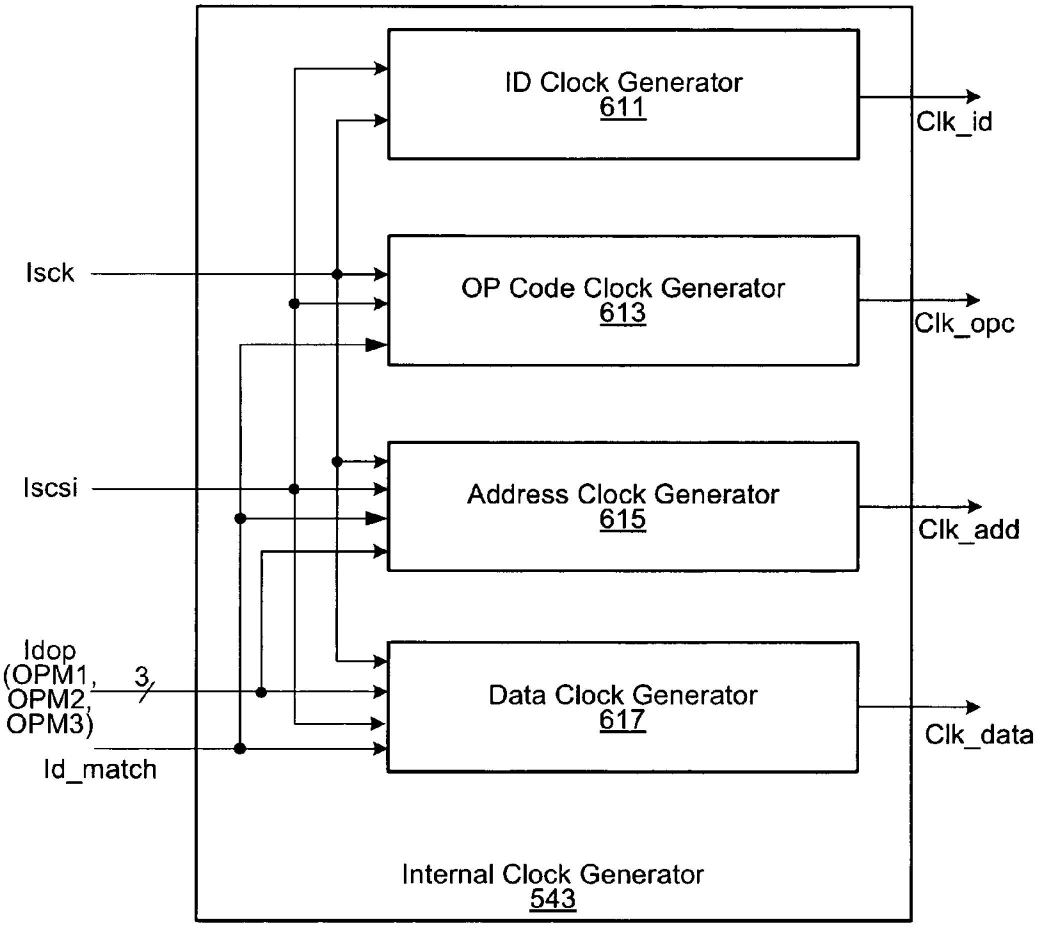

ID Clock Generator
611
Clk_id
Isck
OP Code Clock Generator
613
Clk_opc
Iscsi
Address Clock Generator
615
Clk_add
Idop
(OPM1,
OPM2,
OPM3)
3
Data Clock Generator
617
Clk_data
Id_match
Internal Clock Generator
543

FIG. 7

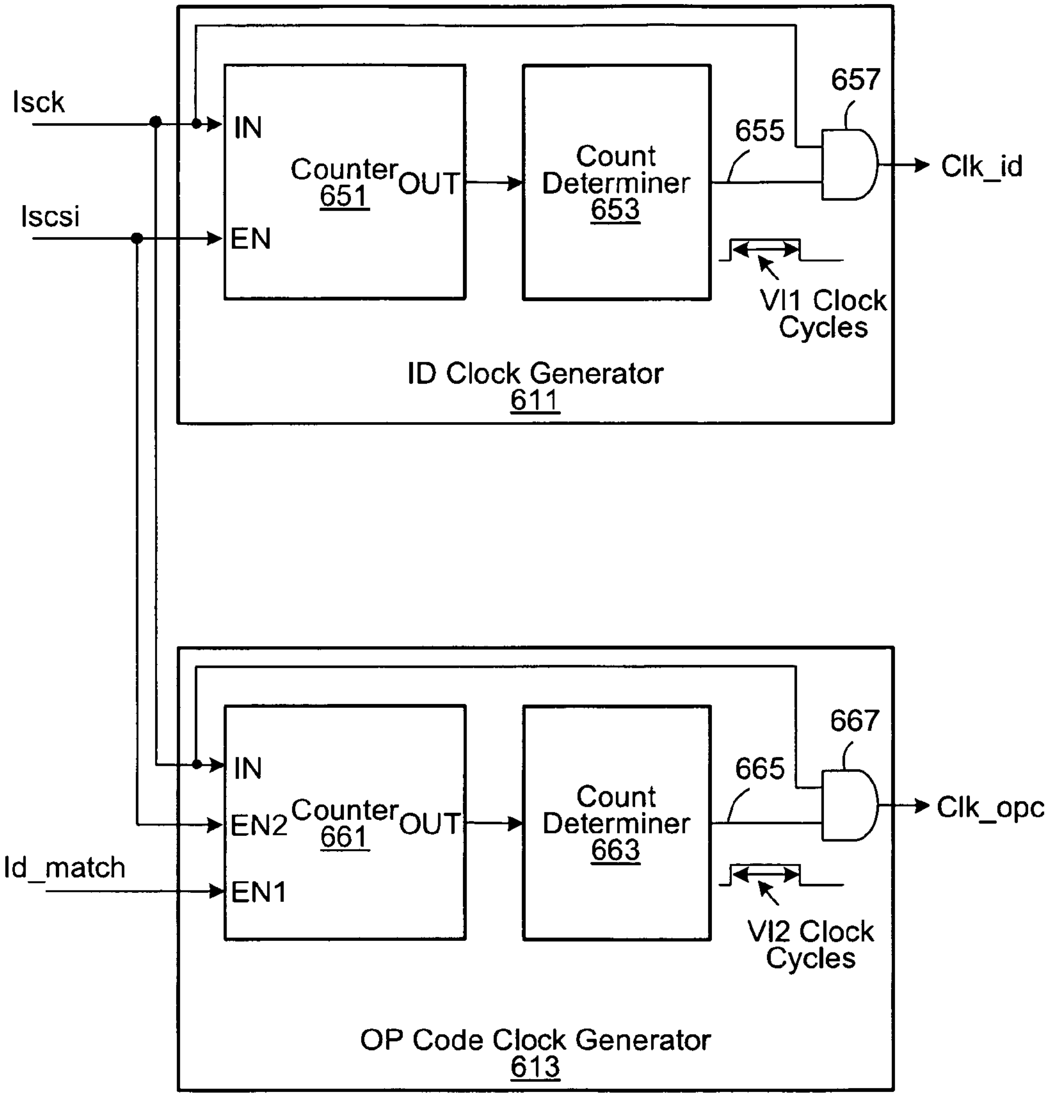

Isck
Iscsi
IN
EN
Counter
651
OUT
Count
Determiner
653
655
657
Clk_id
VI1 Clock
Cycles
ID Clock Generator
611
IN
EN2
Counter
661
OUT
Id_match
EN1
Count
Determiner
663
665
667
Clk_opc
VI2 Clock
Cycles
OP Code Clock Generator
613

FIG. 8A

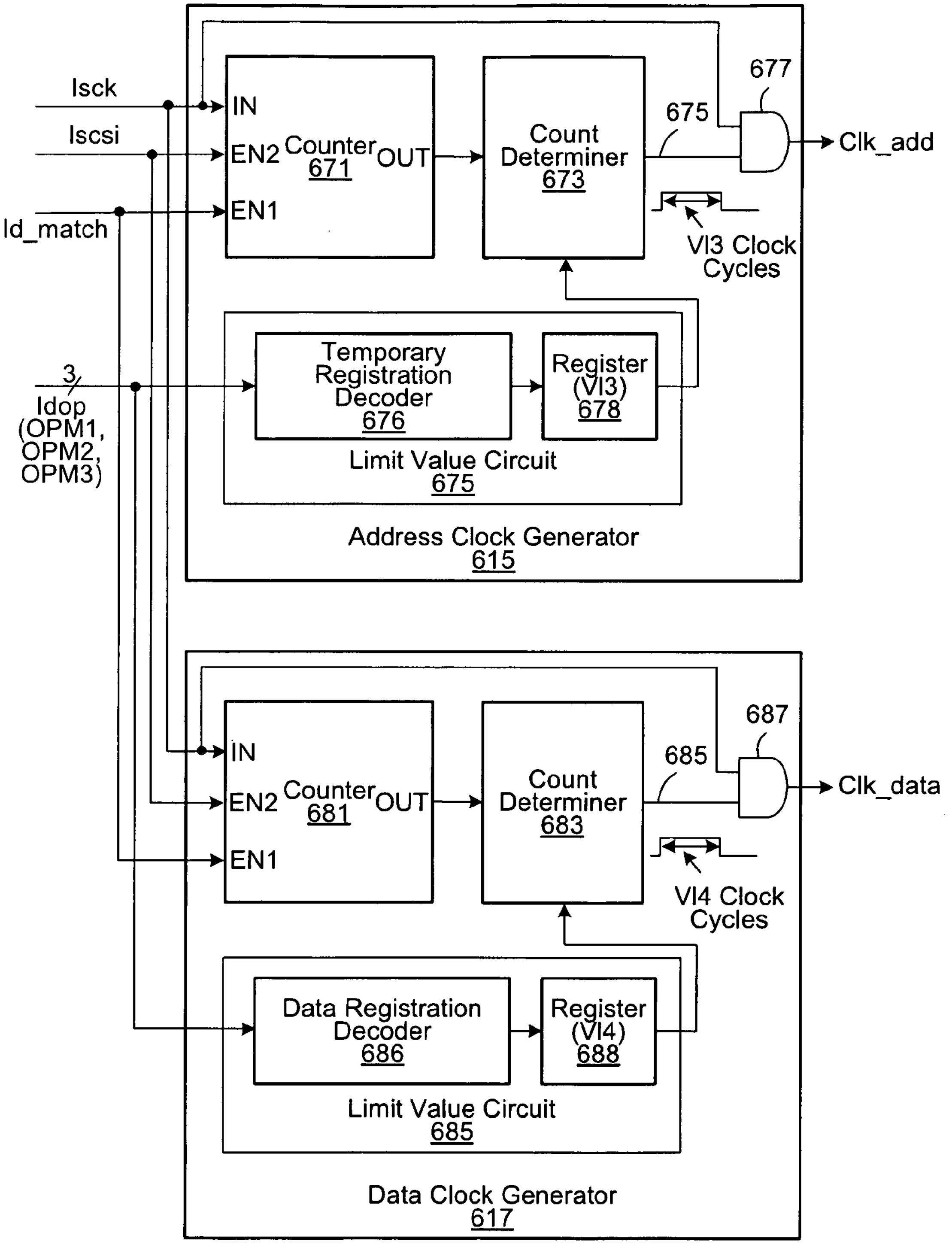

Isck
Iscsi
Id_match
3
Idop
(OPM1,
OPM2,
OPM3)
IN
EN2
EN1
Counter
671
OUT
Count
Determiner
673
675
677
Clk_add
VI3 Clock
Cycles
Temporary
Registration
Decoder
676
Register
(VI3)
678
Limit Value Circuit
675
Address Clock Generator
615
IN
EN2
EN1
Counter
681
OUT
Count
Determiner
683
685
687
Clk_data
VI4 Clock
Cycles
Data Registration
Decoder
686
Register
(VI4)
688
Limit Value Circuit
685
Data Clock Generator
617

FIG. 8B

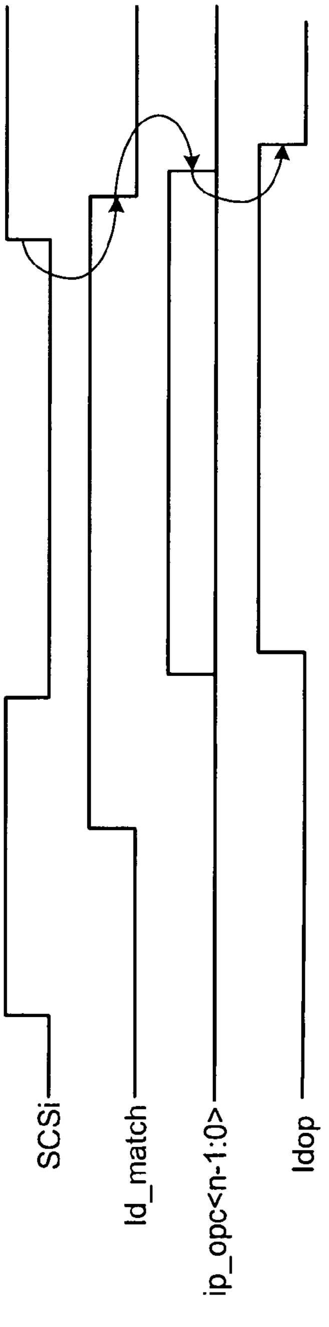

SCSi
ld_match
ip_opc<n-1:0>
ldop

FIG. 9A

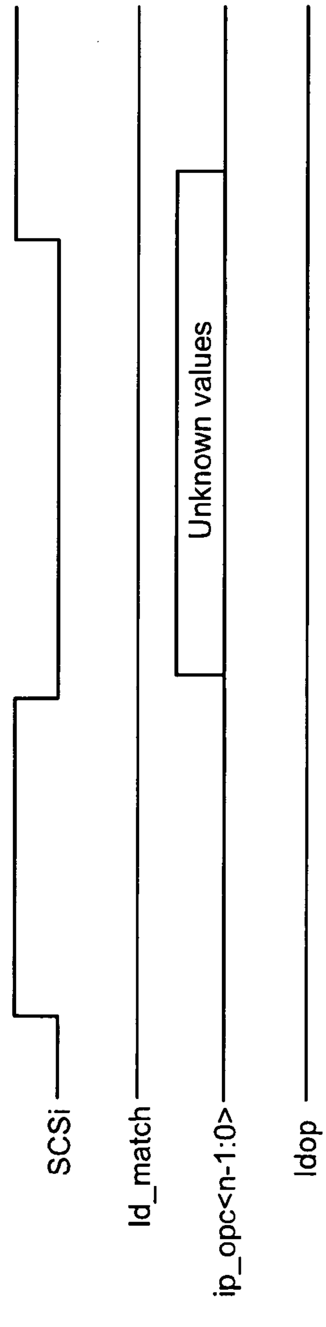

SCSi
ld_match
ip_opc<n-1:0>
ldop
Unknown values

FIG. 9B

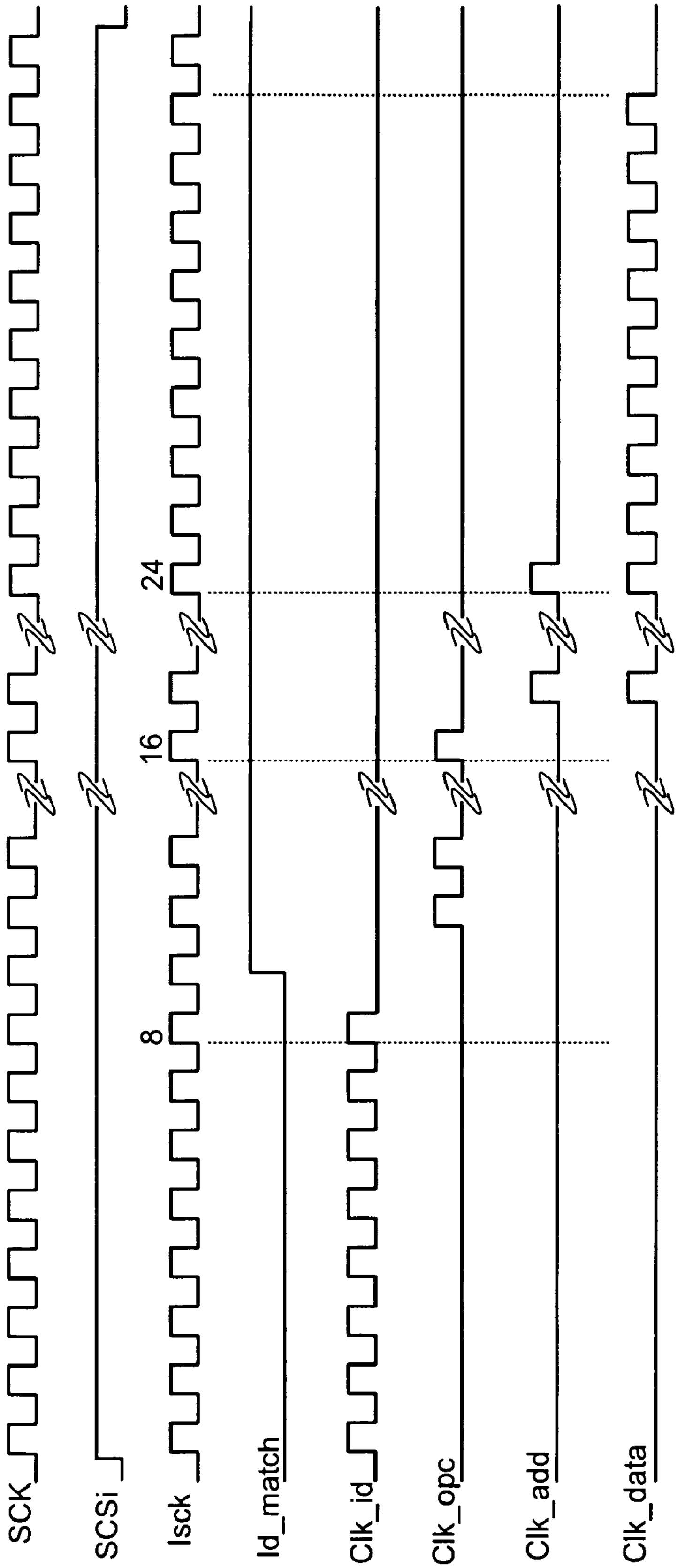
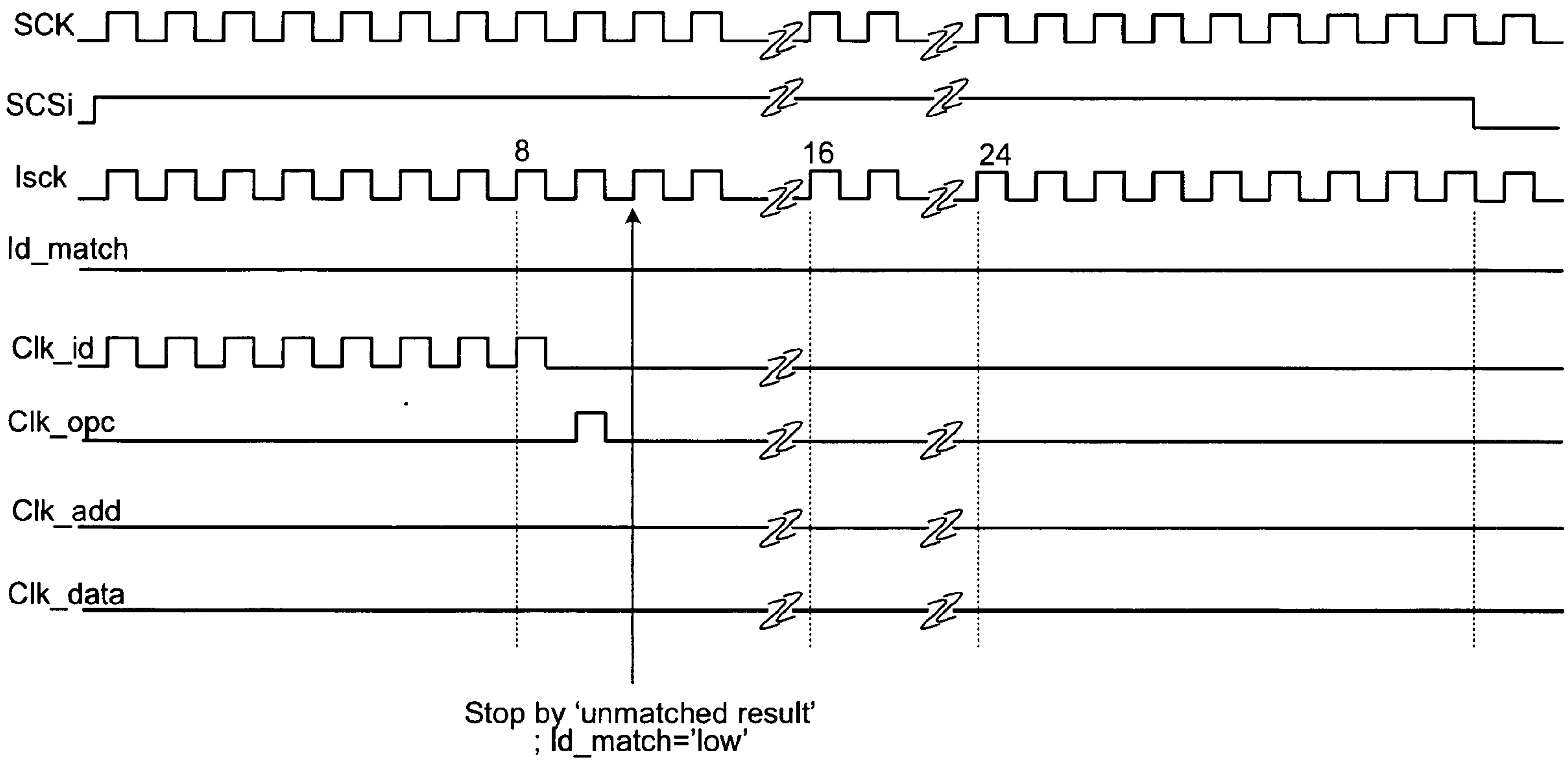


FIG. 10A



Stop by 'unmatched result'
; Id_match='low'

FIG. 10B

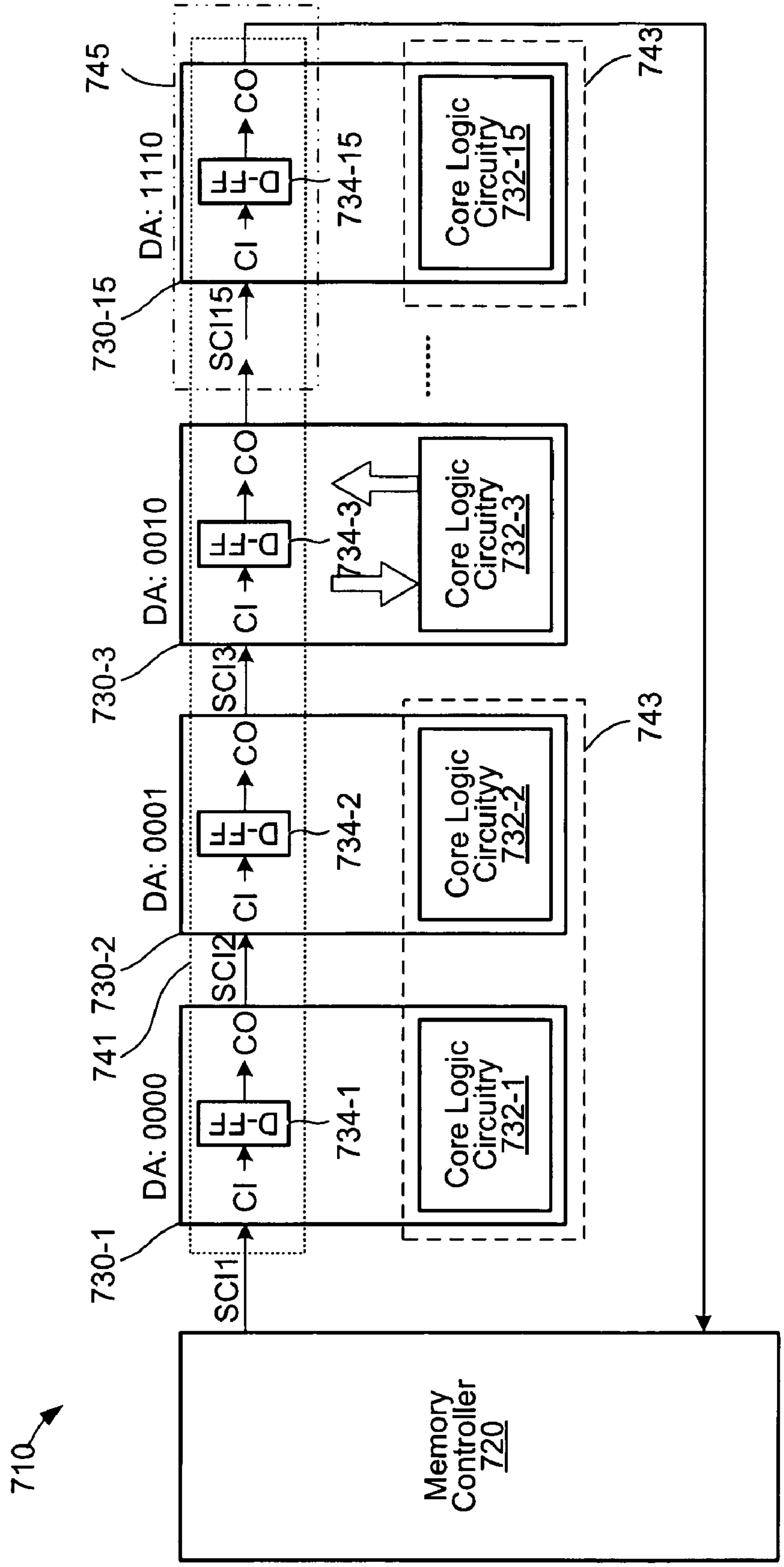

710
Memory Controller 720
SCI1
730-1
DA: 0000
CI
D-FF
CO
734-1
Core Logic Circuitry 732-1
741
730-2
SCI2
DA: 0001
CI
D-FF
CO
734-2
Core Logic Circuityy 732-2
743
730-3
SCI3
DA: 0010
CI
D-FF
CO
734-3
Core Logic Circuitry 732-3
......
SCI15
730-15
DA: 1110
CI
D-FF
CO
734-15
Core Logic Circuitry 732-15
745
743

FIG. 11

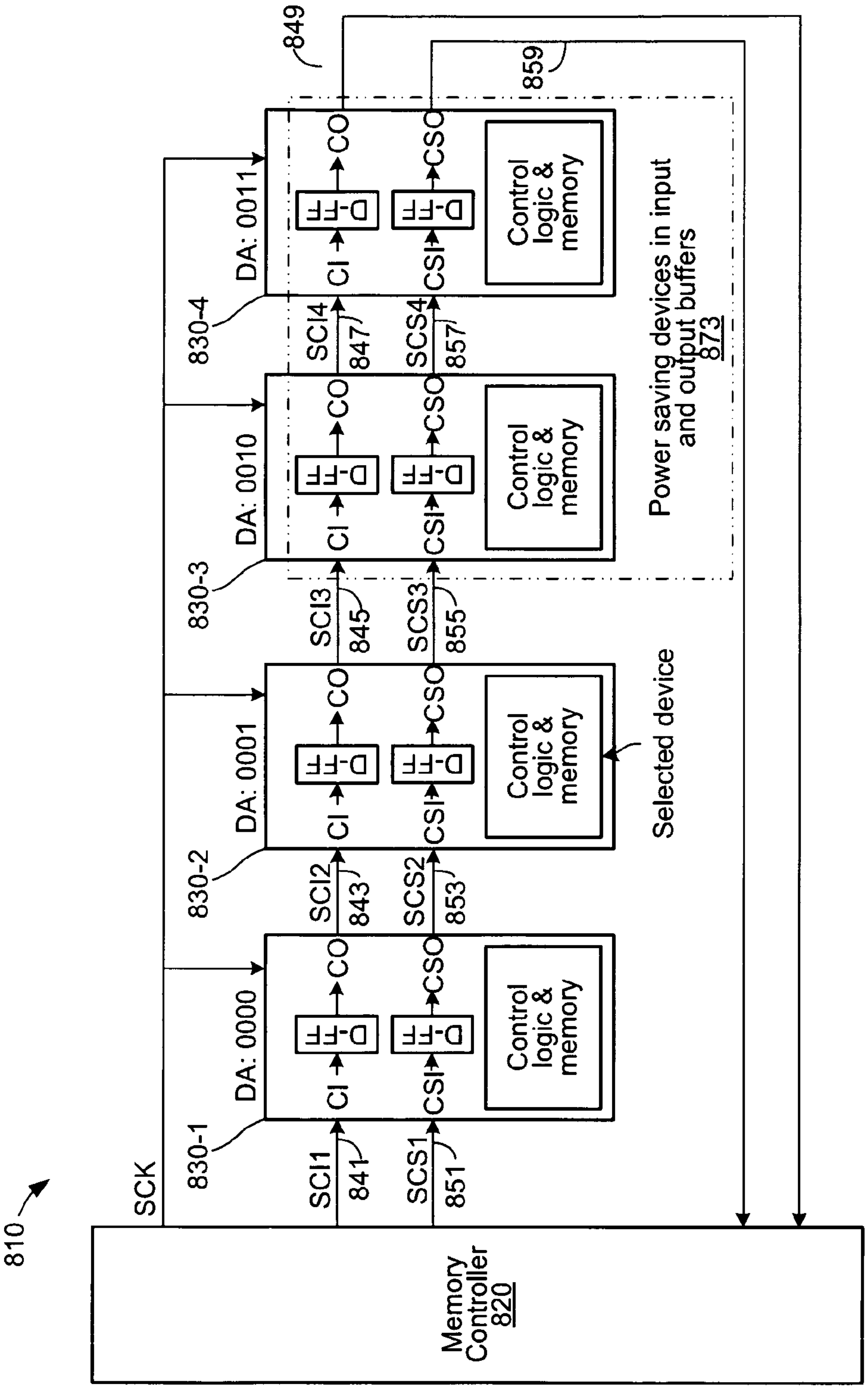

810
Memory Controller 820
SCK
SCI1
841
SCS1
851
830-1
DA: 0000
CI
D-FF
CO
CSI
D-FF
CSO
Control logic & memory
SCI2
843
SCS2
853
830-2
DA: 0001
CI
D-FF
CO
CSI
D-FF
CSO
Control logic & memory
Selected device
SCI3
845
SCS3
855
830-3
DA: 0010
CI
D-FF
CO
CSI
D-FF
CSO
Control logic & memory
SCI4
847
SCS4
857
830-4
DA: 0011
CI
D-FF
CO
CSI
D-FF
CSO
Control logic & memory
Power saving devices in input and output buffers 873
849
859

FIG. 12

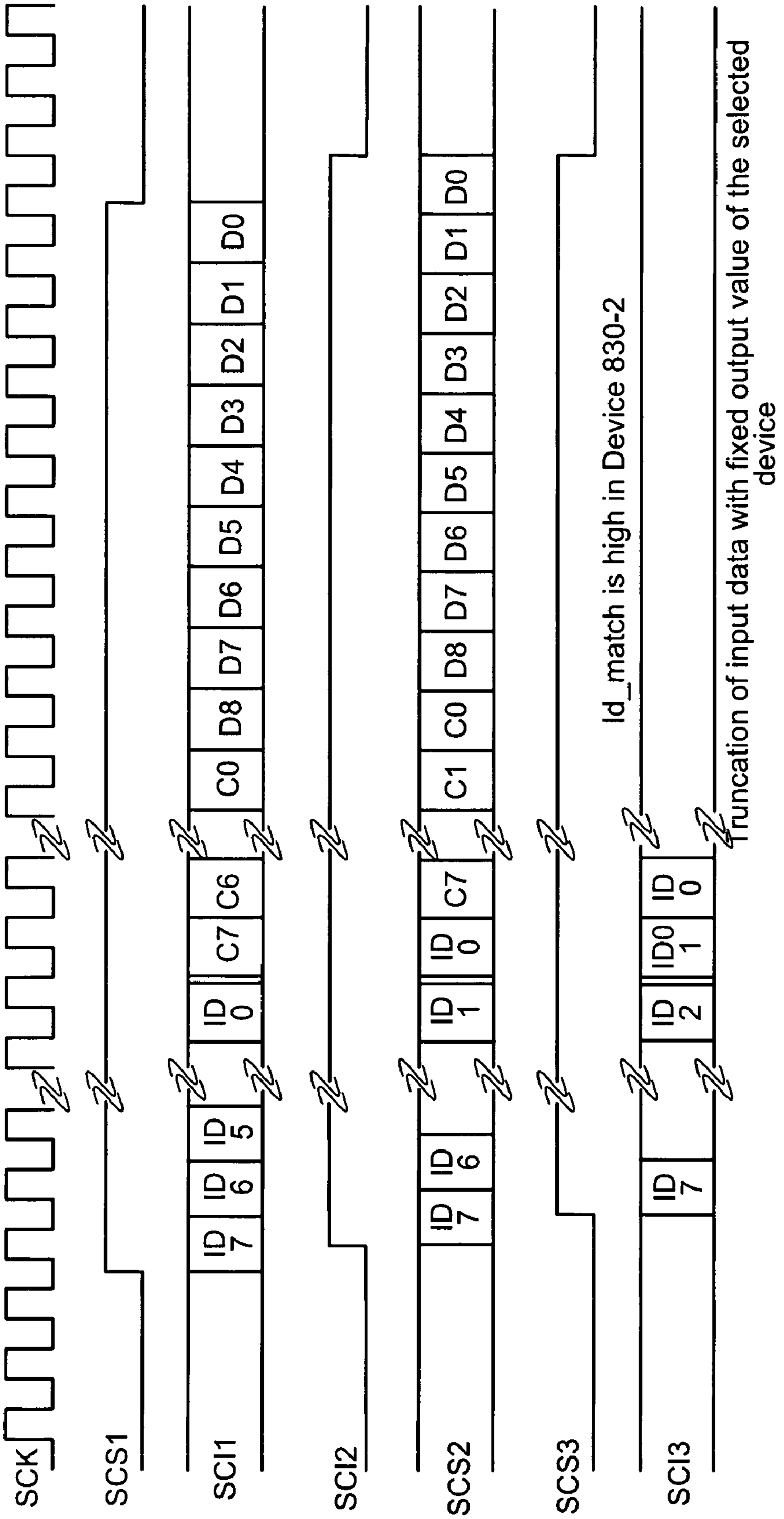

SCK
SCS1
SCI1
ID 7
ID 6
ID 5
ID 0
C7
C6
C0
D8
D7
D6
D5
D4
D3
D2
D1
D0
SCI2
SCS2
ID 7
ID 6
ID 1
ID 0
C7
C1
C0
D8
D7
D6
D5
D4
D3
D2
D1
D0
SCS3
SCI3
ID 7
ID 2
ID0 1
ID 0
Id_match is high in Device 830-2
Truncation of input data with fixed output value of the selected device

FIG. 13

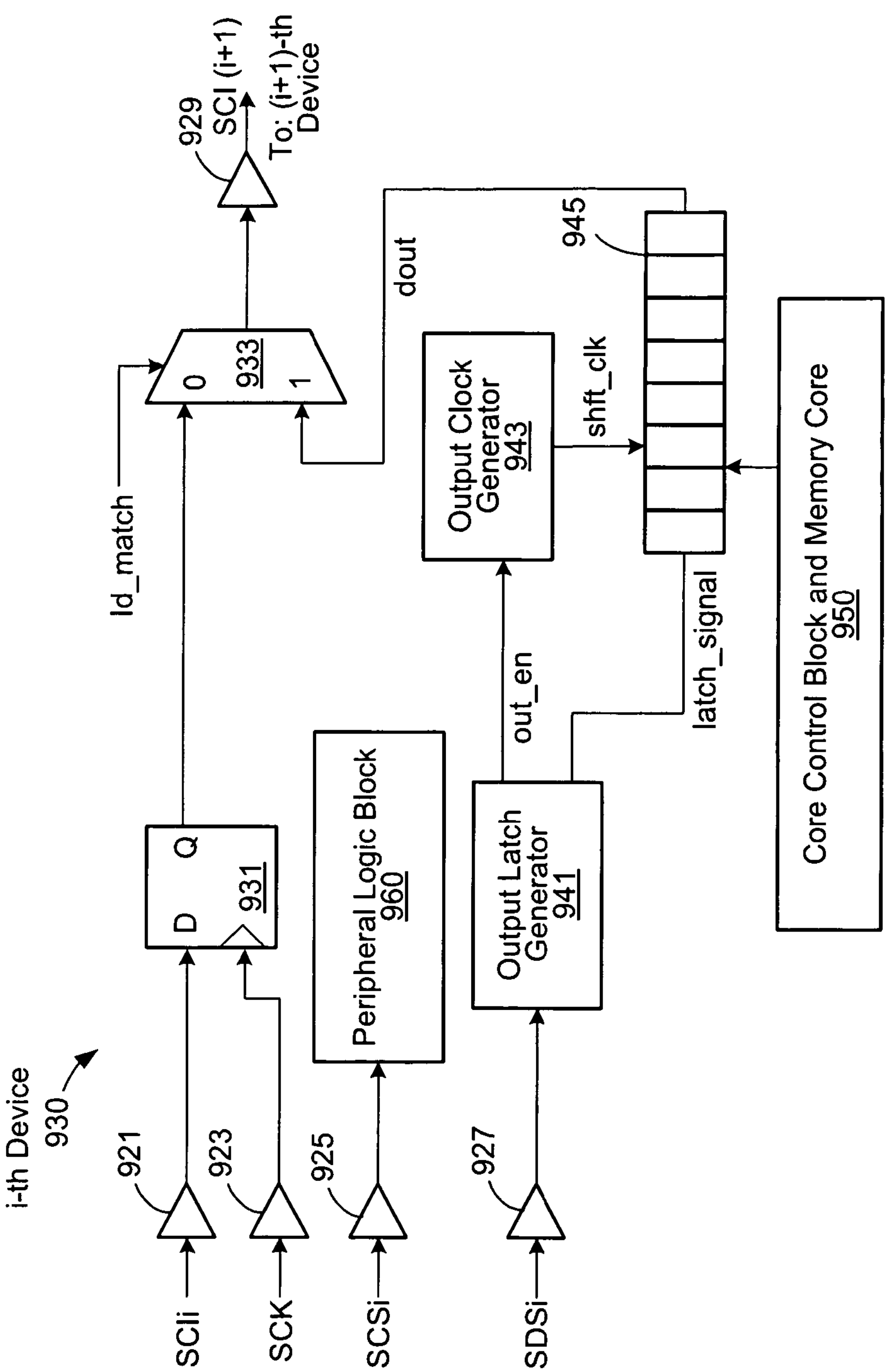

i-th Device
930
SCIi
SCK
SCSi
SDSi
921
923
925
927
D
Q
931
Peripheral Logic Block
960
Output Latch Generator
941
out_en
latch_signal
Output Clock Generator
943
shft_clk
Id_match
0
933
1
dout
945
929
SCI (i+1)
To: (i+1)-th Device
Core Control Block and Memory Core
950

FIG. 14

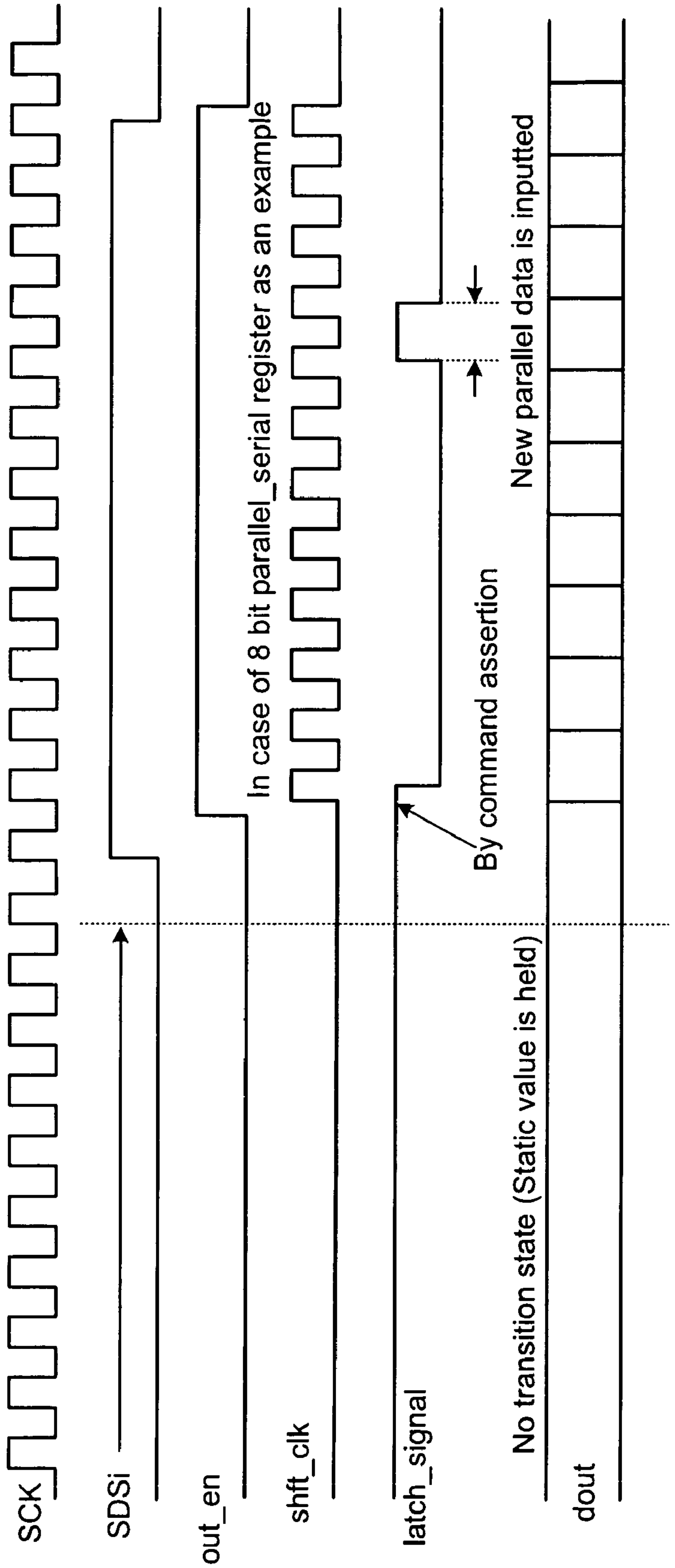

SCK
SDSi
out_en
shft_clk
latch_signal
dout
In case of 8 bit parallel_serial register as an example
By command assertion
New parallel data is inputted
No transition state (Static value is held)

FIG. 15

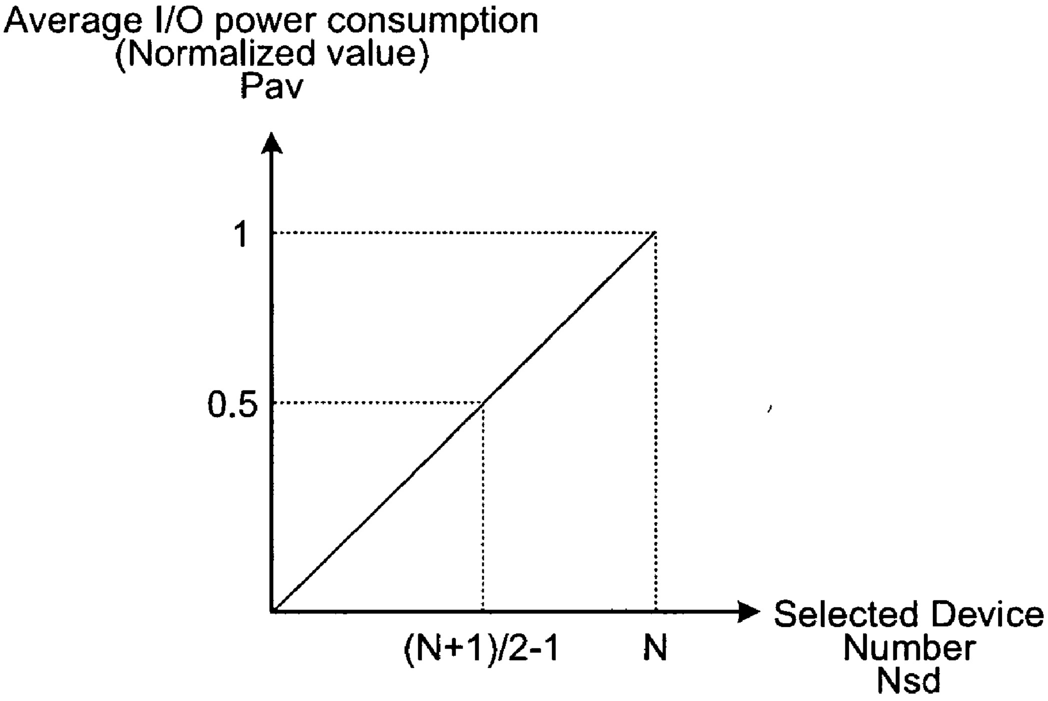

Average I/O power consumption
(Normalized value)
Pav
1
0.5
(N+1)/2-1
N
Selected Device
Number
Nsd

FIG. 16

SEMICONDUCTOR DEVICE AND METHOD FOR REDUCING POWER CONSUMPTION IN A SYSTEM HAVING INTERCONNECTED DEVICES

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of priority from U.S. Provisional Patent Application No. 60/902,003 filed Feb. 16, 2007; U.S. Provisional Patent Application No. 60/891,108 filed Feb. 22, 2007; and U.S. Provisional Patent Application No. 60/943,442 filed Jun. 12, 2007, the he disclosures of which are expressly incorporated herein by reference in their entirety.

FIELD OF THE INVENTION

[0002] The present invention relates generally to semiconductor devices. More particularly, the present invention relates to a system having an arrangement having a plurality of semiconductor devices.

BACKGROUND OF THE INVENTION

[0003] Power consumption control is important in memory and system design, and there are potential benefits in reducing power consumption. For instance, a memory system that is battery-powered benefits from a longer battery life if power consumption by the memory system is reduced. There also are benefits in reducing power consumption for a memory system that is not battery-powered. For instance, reducing power consumption in a main logic system reduces electric currents and voltages thereby reducing heat generation during active operation.

[0004] Most memory systems utilize a multi-drop connection between the memory controller and multiple memory devices to increase the memory density on the system board and at the package level. However, this approach does not guarantee good signal integrity and enough timing margin in high speed applications, for example, over 100 MHz frequency. Therefore, there is a need for an alternative memory system architecture that achieves acceptable signal integrity for high speed applications. Such an alternative has been found in architectures that feature multiple devices that are serially interconnected. In such an architecture, a memory controller is connected to a first memory device with a link, and that memory device is connected to a next memory device with another link and so on. The use of links between devices overcomes some of the problems with signal integrity for high speed applications. However, unlike a multi-drop architecture, where specific devices can be activated while remaining devices are passive, all of the devices in a serially interconnected architecture are active because they must be available to pass signals on to the next device in the serial interconnection.

SUMMARY OF THE INVENTION

[0005] In accordance with one aspect of the present invention, there is provided a semiconductor device for use in a serial interconnection arrangement of semiconductor devices. The semiconductor device comprises: command circuitry for receiving commands and forwarding at least some of the commands; and core circuitry configured to operate with normal power consumption for those commands that are addressed to the semiconductor device, and operate with reduced power consumption for those commands that are not addressed to the semiconductor device.

[0006] For example, each command comprises an ID number. The semiconductor device may further comprise a determiner for determining for each command if the command is addressed to the semiconductor device based on the ID number of the command and the device address of the semiconductor device. The core circuitry may comprise an internal clock producer for producing at least one clock for processing commands. The internal clock producer may be enabled upon determining that the ID number of the command matches the device address of the semiconductor device, the core circuitry thereby operating with the normal power consumption. The internal clock producer may be disabled upon determining that the ID number of the command does not match the device address of the semiconductor device, the core circuitry thereby operating with the reduced power consumption.

[0007] The command may further include an OP code. The internal clock producer may include an OP code clock producer for producing an OP code clock if the ID number of the command matches the device address of the semiconductor device. The OP code clock facilitates processing of an OP code of the command.

[0008] The core circuitry may provide a read output that is a substantially static output comprising an output from the memory that is generated while a read operation is not in progress.

[0009] According to another broad aspect of the invention, there is provided a method in a semiconductor device of a serial interconnection arrangement of semiconductor devices. The method comprises: receiving commands and forwarding at least some of the commands at the semiconductor device; operating the semiconductor device with normal power consumption for those commands that are addressed to the semiconductor device; and operating the semiconductor device with reduced power consumption for those commands that are not addressed to that semiconductor device.

[0010] According to another broad aspect of the invention, there is provided a system comprising: a controller; and a plurality of semiconductor devices connected in-series, one of the semiconductor devices being connected to the controller. Each of the semiconductor devices includes: command circuitry for receiving commands and forwarding at least some of the commands; and core circuitry configured to operate with normal power consumption for those commands that are addressed to the semiconductor device, and operates with reduced power consumption for those commands that are not addressed to the semiconductor device.

[0011] According to an embodiment of the present invention, there is provided a system having a memory controller and a plurality of memory devices connected in-series that communicate with the memory controller. A memory device designated by an ID number performs operations at a normal power consumption level. The other devices that are not designated perform signal forwarding operations at a reduced power consumption level. The designated memory device enables its internal clock generator to generate all clocks necessary for operations. However, in the non-designated memory devices, their internal clock generators generate clocks to perform partial operations for forwarding commands to next memory devices. Such partial operations are performed at the reduced power consumption level.

[0012] A memory device according to another embodiment of the present invention does not forward the input command to the next memory device when there is an ID match. Therefore, the memory devices that do not receive the command will not perform operations at the normal power consumption level, with the result that the power consumption by the system will be reduced.

[0013] A memory device according to another embodiment of the present invention transmits the command replacing the content thereof with a static output. Therefore, the memory devices that receive such commands with replacements perform operations at the reduced power consumption level, with the result that the power consumption by the system will be reduced.

[0014] Examples of the semiconductor devices are processors and memory devices that may operate with different power consumption levels. The memory devices may be volatile memory devices (e.g., random accesses memories) or non-volatile memory devices (e.g., flash devices).

[0015] Other aspects and features of the present invention will become apparent, to those ordinarily skilled in the art, upon review of the following description of specific embodiments of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] Embodiments will now be described with reference to the attached drawings in which:

[0017] FIG. **1** is a block diagram of a serial interconnection arrangement of a plurality of memory devices, to which embodiments of the present invention are applicable;

[0018] FIG. **2** is a block diagram of a system having series-connected memory devices, to which embodiments of the present invention are applicable;

[0019] FIG. **3** is a schematic of example command formats for the system as shown in FIG. **2**;

[0020] FIG. **4** is a block diagram showing part of the series-connected memory devices shown in FIG. **2**;

[0021] FIG. **5** is a block diagram showing details of one of the memory devices shown in FIG. **2**;

[0022] FIG. **6** is a schematic showing a command interpreter shown in FIG. **5**;

[0023] FIG. **7** is a block diagram showing an internal clock generator shown in FIG. **5**;

[0024] FIGS. **8**A, **8**B are block diagrams showing details of the internal clock generator shown in FIG. **7**;

[0025] FIGS. **9**A, **9**B are signaling diagrams for example signals in the circuitry shown in FIG. **6**;

[0026] FIGS. **10**A, **10**B are signaling diagrams of example signals in the circuitry shown in FIG. **5**;

[0027] FIG. **11** is a block diagram of a system having memory devices that are connected in-series;

[0028] FIG. **12** is a block diagram of the system in which details of the common operation path are provided;

[0029] FIG. **13** is a signaling diagram of example signals in the system shown in FIG. **12**;

[0030] FIG. **14** is a schematic of example circuitry of a memory device for implementing a command truncation feature;

[0031] FIG. **15** is a signaling diagram of example signals in the circuitry shown in FIG. **14**; and

[0032] FIG. **16** is a graph illustrating an example correlation between a selected device number and power consumption with the truncation feature.

DETAILED DESCRIPTION

[0033] In the following detailed description of sample embodiments of the invention, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific sample embodiments in which the present invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the present invention, and it is to be understood that other embodiments may be utilized and that logical, mechanical, electrical, and other changes may be made without departing from the scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined by the appended claims.

[0034] The present invention relates generally to a system having an arrangement including a plurality of semiconductor devices. The devices may operate with different power consumption levels. Examples of the semiconductor devices are processors and memory devices that may operate with different power consumption levels. The memory devices may be volatile memory devices (e.g., random accesses memories) or non-volatile memory devices (e.g., flash devices).

[0035] Examples of detailed system architectures that employ multiple devices connected in-series are provided in commonly assigned and co-pending U.S. patent application Ser. No. 11/594,564 entitled "Daisy Chain Cascading Devices" filed Jul. 31, 2006, the disclosure of which is hereby incorporated by reference in its entirety. Other example details of series-connected memory devices are provided in United States Provisional Patent Application No. 60/868,773 entitled "System and Method of Operating Memory Devices of Varying Type" filed Dec. 6, 2006, the disclosure of which is hereby incorporated by reference in its entirety.

[0036] Examples described herein can find application in serial interconnections having a plurality of devices connected in-series. More generally, the embodiments described herein can find application in system architectures employing serial interconnection configurations having a plurality of semiconductor devices connected in-series by serial links or parallel links. Serial interconnection arrangements form a subset of architectures in which devices are connected together in-series with serial or parallel links. An overview of the system architecture having series-connected memory devices is provided below with reference to FIG. **1**.

[0037] FIG. **1** shows a serial interconnection arrangement of a plurality of memory devices that are connected in-series. Referring to FIG. **1**, an arrangement **110** includes N memory devices **130-1**, **130-2**, **130-3**, - - - , **130**-N that are connected in-series, N being an integer greater than one. A memory controller (not shown) sends a group of signals of data and information to the memory devices of the arrangement **110**. Data or information to be processed is sent to the first device **130-1** and propagated through the devices of the series-connected devices. In one implementation, the output of the last device **130**-N is open. In another implementation, the output of the last (N-th) device **130**-N is connected to the memory controller, so that the memory controller can use the feedback data from the last device **130**-N.

[0038] Each of the memory devices **130-1**, **130-2**, **130-3**, - - - , **130**-N of the arrangement **110** is associated with a unique device address (DA) or device identification number (ID) that is assigned thereto. In the illustrated example, the device addresses of the device **130-1**, **130-2**, **130-3**, - - - , **130**-N are "0", "1", "2", - - - , "N–1", respectively. In a case of N being 15, the device addresses (DAs) are represented by, for example, a binary code or number of four bits: "0000", "0001", "0010", - - - , "1110", respectively. Each device has its MSB (most significant bit) first and has its LSB (least significant bit) last. In another implementation, the DA can be changed to the LSB first and the MSB last. Also, the DAs can be successive numbers from another value (e.g., "1"). Furthermore, the DAs can be decremented numbers from a maximum value.

[0039] The memory controller issues control and data information including a specific device address (for a target or designated device) and an operation command. Such information is sent to the first device **130-1** and it is propagated through the devices. During the propagation, an input Dn of each device receives incoming information and outgoing information is transmitted from its output Qn to a next device. A specific memory device identified as a target or designated device by the specific device address executes the operation command, in accordance with device address matching. Examples of the operation command are memory accesses and data processes.

[0040] FIG. **2** shows a system having a serial interconnection arrangement of a plurality of memory devices connected in-series, to which embodiments of the present invention are applicable. In the illustrated example, 15 devices are connected in-series. Referring to FIG. **2**, a system **210** having memory devices **230-1**, **230-2**, **230-3**, - - - , **230-15** connected in-series. The system **210** has a memory controller **220** connected to a first memory device **230-1** via a link **221**. Each pair of adjacent memory devices is interconnected via a link. In the particular example shown in FIG. **2**, the last memory device **230-15** is connected to the memory controller **220** via a link **222** thereby creating a specific type of series architecture, namely a ring architecture. Alternatively, if the last memory device **230-15** is not connected to the memory controller **220**, the series-connection architecture would be a linear architecture. In a linear architecture, the last memory device **230-15** may be connected to some other component instead of the memory controller **220**. Ring architectures and linear architectures are both examples of architectures featuring devices connected together in-series. The links interconnecting the memory devices **230-1**, **230-2**, **230-3**, - - - , **230-15** are used to transmit commands over a path that traverses the memory devices **230-1**, **230-2**, **230-3**, - - - , **230-15**.

[0041] Each command issued by the memory controller **220** includes an ID number and a command OP code (hereinafter simply 'OP code'), and may also include address information and/or data. The ID number included in each command matches the device address of a specifically selected (i.e., a target or designated) memory device. A command is contained in a command input (CI) signal sent to the memory devices by the memory controller **220**. Whenever a memory device receives a command contained in a command input signal, it determines whether the ID number of the command matches the device address associated with the device. If there is a match, then the memory device executes the operation (OP) code identified by the OP code field of the command. Otherwise, the memory device does not execute the OP code identified by the OP code field of the command and merely forwards the command on to the next memory device.

[0042] FIG. **3** shows example command formats for the memory devices connected in-series as shown in FIG. **2**. Referring to FIG. **3**, a first command format **312-1** includes an ID number and an OP code. The ID number is used to uniquely identify a selected memory device, while the OP code field contains the OP code to be executed by the selected or designated device. Commands with the first command format **312-1** may, for example, be used for commands containing OP codes for reading a register value. A second command format **312-2** includes an ID number, an OP code and data. Commands with the second command format **312-2** may, for example, be used for commands containing OP codes for writing data to a register. A third command format **312-3** includes an ID number, an OP code, and additional addresses. The additional addresses may, for example, include row and/or column addresses for addressing a location in memory cells. Commands with the third command format **312-3** may, for example, be used for commands containing OP codes for reading data from memory cells of a selected memory device. A fourth command format **312-4** includes an ID number, an OP code, additional addresses, and data. Commands with the fourth command format **312-4** may, for example, be used for commands containing OP codes for writing data to the memory cells of a selected memory device. Note that all of four example command formats **312-1**, **312-2**, **312-3**, **312-4** start with an ID number for addressing purposes. For example, each of the fields of the ID number and the OP code is one byte (eight bits). The byte sizes of data and addresses are variable and information on the bytes (or bits) thereof is indicated by the OP code.

[0043] Referring back to FIG. **2**, as part of the path, the first memory device **230-1** has a command input connection CI and a corresponding command output connection CO. Similarly, each of the other memory devices **230-2**, **230-3**, - - - , **230-15** has a command input connection CI and a corresponding command output connection CO. In the illustrated example, the link **221** of the path form an n-bit wide path. The links are serial links when the path is a single bit wide path (i.e., when n is equal to 1). Otherwise, the links are parallel links (i.e., when n is greater than 1).

[0044] In the illustrated example, there is also a control path that traverses the memory devices **230-1**, **230-2**, **230-3**, - - - , **230-15**. As part of this control path, the first memory device **230-1** has a command strobe input (CSI) connection and a corresponding command strobe output (CSO) connection. Similarly, each of the other memory devices **230-2**, **230-3**, - - - , **230-15** has a CSI connection and a corresponding CSO connection. The CSI connection of the first device **230-1** is for receiving a command strobe signal SCS**1** issued by the memory controller **220** for enabling the command of the command input signal SCI**1** input to the memory devices **230-1**. The CSO connection of the device **230-1** is for forwarding the command strobe signal to the next device **230-2**. Therefore, the forwarded command strobe signal is transmitted as a command strobe signal SCS**2** to the CSI connection of the second memory device **230-2**. Similarly, each of the other devices **230-2**, **230-3**, - - - , **230-15** performs the forwarding of the incoming command strobe signal to the next memory device. As such, the command strobe signal is forwarded from one device to the next device.

[0045] There is another control signal which is a data strobe signal for enabling output from the memory device during read operation. The first memory device **230-1** has a data strobe input (DSI) connection and a corresponding data strobe output (DSO) connection. Similarly, each of the other memory devices **230-2**, **230-3**, - - - , **230-15** has a DSI connection and a corresponding DSO connection. The DSI connection of the first device **230-1** is for receiving a data strobe signal DCS**1** issued by the memory controller **220** for enabling the output from the memory devices **230-1** to the next memory device **230-2**. The DSO connection of the device **230-1** is for forwarding the data strobe signal to the next device **230-2**. Therefore, the forwarded data strobe signal is transmitted as a data strobe signal DCS**2** to the DSI connection of the memory device **230-2**. Similarly, each of the other devices **230-2**, **230-3**, - - - , **230-15** performs the forwarding of the incoming data strobe signal to the next memory device. As such, the command strobe signal is forwarded from one device to the next device.

[0046] Furthermore, a clock signal SCK is provided by the memory controller **220** to all of the devices commonly (i.e., a common synchronous clocking fashion). Additional signals are provided for operation, for example, a chip select and a reset (not shown).

[0047] Although only 15 memory devices **230-1**, **230-2**, **230-3**, - - - , **230-15** are shown in the illustrated example, more generally, there may be any appropriate number of memory devices connected in-series that communicate with the memory controller **220**. A serial interconnection arrangement can include more devices.

[0048] The system **210** shown in FIG. **2** implements a ring architecture. In operation of the system **210**, the memory controller **220** sends commands over an output communication path including the link **221** and receives responses over an input communication path including the link **222** for those commands that requisition a response. For each command, the memory controller **220** issues the command strobe signal SCS1 for enabling the first device **230-1** to receive the command. The command strobe signal is propagated through the devices. Also, the memory controller **220** issues the data strobe signal SDS1 for enabling the first device **230-1** after the data read operation is instructed to output the data from the selected device. The data strobe signal is propagated through the devices. Each of the memory devices **230-1**, **230-2**, **230-3**, - - - , **230-15** is associated with a unique device address (DA). In the illustrated example, the memory devices **230-1**, **230-2**, **230-3**, - - - , **230-15** have device addresses represented by binary codes of 0000, 0001, 0010, - - - , 1110, respectively, but more generally the addresses are implementation specific.

[0049] During operation, the system **210** consumes power. Each of the memory devices **230-1**, **230-2**, **230-3**, - - - , **230-15** consumes power to process each command regardless of whether the command is addressed to the particular device. Signal and data transmission through the series-connected memory devices **230-1**, **230-2**, **230-3**, - - - , **230-15** causes each of the devices to consume power due to signal flow through input and output pins or connections, such as command input connection CI, corresponding command output connection CO.

[0050] Each of the memory devices **230-1**, **230-2**, **230-3**, - - - , **230-15** transmits command input streams on the command input signal to the next adjacent memory device. The power consumption due to signal flow through a set of input and output pins or connections may be relatively low. However, each of the memory devices **230-1**, **230-2**, **230-3**, - - - , **230-15** has a respective set of input and output connections and therefore, the total power consumption increases with the number of memory devices. The number of series-connected memory device can be implementation specific and depends on the power consumption. Further explanation is provided below with reference to Table 1 that outlines example power consumption of memory devices.

TABLE 1

Device	Vdd (V)	Cload (pF)	Frequency (MHz)	Data Rate (Mbps/pin)	Power (mW)	Current per Output (mA)
Multi-drop NAND	3.3	50	33	33	18	5.4
Serial Interconnection NAND	1.8	5	100	200	3.2	1.8
Serial Interconnection NAND	1.8	5	133	266	4.3	2.4

[0051] Table 1 illustrates power consumption and current for memory devices under various operating conditions. A NAND flash device used in a multi-drop arrangement may use an operating voltage of Vdd=3.3V, a capacitive load of Cload=50 pF, an operating frequency of 33 MHz, and a data rate of 33 Mbps/pin. Such a device will consume, for example, 18 mW of power and 5.4 mA of current per output. By contrast, a NAND flash device used in a serial interconnection arrangement with an operating voltage of, for example, Vdd=1.8V, a capacitive load of Cload=5 pF, an operating frequency of 100 MHz, and a data rate of 200 Mbps/pin will consume only 3.2 mW of power and only 1.8 mA of current per output. Increasing the frequency results in greater power consumption. For example, increasing the operating frequency from 100 MHz to 133 MHz for the series-connected NAND flash devices results in the power consumption increasing from 3.2 mW to 4.3 mW while the current per output increases from 1.8 mA to 2.4 mA.

[0052] For example, a NAND flash device having 8 I/Os each at 5.4 mA has a total current of 43.2 mA. Therefore, the number of memory devices operating at 100 MHz in the serial interconnection arrangement with similar I/O current as the multi-drop NAND flash with 8 I/Os would be 43.2 mA/1.8 mA, which is equal to approximately 24. Also, the number of memory devices operating at 133 MHz in the serial interconnection arrangement with similar I/O current as the multi-drop NAND flash with 8 I/Os would be 43.2 mA/2.4 mA, which is equal to approximately 18. If individual memory devices in the serial interconnection arrangement can consume less power, then additional memory devices may be feasible.

[0053] The last two rows of Table 1 indicate power consumption of memory devices connected in-series with serial links, wherein each device has one pair of I/O. Note that power consumption for memory devices connected in-series with parallel links may differ.

[0054] In other system arrangements such as a multi-drop arrangement, power saving can be accomplished by enabling or disabling memory devices. Devices that are not enabled consume less power. However, in the serial interconnection arrangement, all memory devices are enabled during operation so that commands can be propagated through the series-connected devices. Therefore, the approach to the multi-drop arrangement cannot be applied to the serial interconnection arrangement. An approach for saving power consumption in memory devices of a serial interconnection arrangement featuring devices that are connected in-series with links will be described below. Note that this example and the subsequent examples described below relate to systems having memory devices connected in-series with serial links. However, it is to be understood that embodiments of the present invention are applicable to systems having series-connected memory devices with parallel links.

[0055] FIG. **4** shows part of the series-connected memory devices shown in FIG. **2**. As shown, command input signal SCIi input to a device **230**-*i* can be transmitted to the next device **230**-(*i*+1).

[0056] FIG. **5** shows details of a representing i-th memory device shown in FIGS. **2** and **4**. It is to be understood that the circuitry of FIG. **5** is very specific and is provided for example purposes only. Referring to FIG. **5**, a memory device **230**-*i* includes ID match and command forwarding circuitry and control logic circuitry for its operations. The ID match and command forwarding circuitry performs forwarding on the input to the output and for determining whether or not to operate the control logic circuitry with reduced power consumption. However, it is to be understood that the division between the ID match and command forwarding circuitry and the control logic circuitry are rather arbitrary.

[0057] The device **230**-*i* includes four input connections: (i) a command input (CI) connection **521** for receiving a command input signal SCIi including the incoming command; (ii) a clock input connection **523** for receiving a clock signal SCK; (iii) a command strobe input (CSI) connection **525** for receiving a command strobe signal SCSi for enabling command input; and (iv) a data strobe input (DSO) connection **527** for receiving a data strobe signal SDSi. All four signals SCIi, SCK, SCSi, SDSi are buffered by respective input buffers **511**, **513**, **515**, **517**. A buffered version, Isci, of the command input signal SCIi is passed through a D-type flip-flop (D-FF) **531** onto a multiplexer **533**, which provides through an output buffer **549** and a command output connection **535** a command input signal SCI(i+1) to a next device (i+1). The buffered version, Iscsi, of the command strobe signal SCSi is fed to an internal clock generator **543** and a D-FF **537**. An output of the D-FF **537** is fed to a multiplexer **539**, the output of which is through an output buffer **541** to the next device as a command strobe signal SCS(i+1). Similarly, a buffered version, Isdsi, of the data strobe signal SDSi is fed to a D-FF **545**, the output of which is fed to a multiplexer **553**. An output of the multiplexer **553** is through an output buffer **551** to the next device as a data strobe signal SDS(i+1). A buffered version, Isck, of the clock signal SCK is fed to clock inputs of the D-FFs **531**, **537** and **545** and the internal clock generator **543**. The buffered command input signal Isci is also fed to an ID register **561**, a command interpreter **563**, address registers **565**, and a data input register **569**. The internal clock generator **543** has outputs connected to the ID register **561**, the command interpreter **563**, the address registers **565**, and the data input register **569**.

[0058] A comparator **577** has a first input from the ID register **561** and a second input from a storage element **575** holding a device ID, which represents the actual device ID (or the device address DA) of the memory device. The comparator **577** compares the ID number contained in the command to the ID held in the device ID holder **575** to determine whether there is a match between the two IDs. In the event that there is a match, an ID match signal Id_match having the high state is provided from the comparator **577** to the command interpreter **563**, the internal clock generator **543**, and the multiplexers **533**, **539**. The address registers **565** are connected to a pre-decoder **567**. The command interpreter **563**, the pre-decoder **567**, and the data input register **569** are connected to a controller **571**. The controller **571** is connected to a data output register **573**, which is connected to "1" input of the multiplexer **533**. A memory core array **579** for storing data is associated with the controller **571**.

[0059] FIG. **6** shows example details of the command interpreter **563** shown in FIG. **5**. Referring to FIG. **6**, the command interpreter **563** includes an OP code register **581** and an OP code decoder **583**. The OP code register **581** receives the buffered command input signal Isci and the OP code clock signal Clk_opc.

[0060] Referring to FIGS. **5** and **6**, in operation, a command is received over the command input signal SCIi with the command strobe signal SCSi being asserted. The internal clock generator **543** generates an ID clock signal Clk_id for the ID register **561**. As described in FIG. **3**, the command starts with an ID number, which is loaded into the ID register **561**. The comparator **577** compares the contents of the ID register **561** with the device ID held in the device ID holder **575** of the memory device. If there is a match, then the comparator **577** asserts the ID match signal Id_match in order to indicate an ID match. Otherwise, the comparator **577** does not assert the ID match signal Id_match.

[0061] In the illustrated example, the ID match signal Id_match is active high. Alternatively, the ID match signal Id_match can be active low. The ID match signal Id_match has a role in activating the internal clock generator **543** and the OP code decoder **583**. The internal clock generator **543** generates an OP code clock signal Clk_opc for the OP code register **581** of the command interpreter **563**, an address clock for the address registers **565**, and a data clock for the data input register **569** if the ID match signal Id_match is high. Therefore, the OP code, additional addresses, and data of the command are conditionally loaded into the OP code register **581**, the address registers **565** and the data input register **569** for further processing by the OP code decoder **583**, the pre-decoder **567**, the controller **571**. In the event that there is no ID match, then the internal clock generator **543** does not generate OP code clock signal Clk_opc, address clock signal Clk_add, and data clock signal Clk_data because there is no need for the memory device to load the OP code, additional addresses, and the data from the command. When the ID match signal Id_match is low, internal operations are halted (by not generating the clocks) so that unnecessary power consumption can be avoided. For instance, the OP code decoder **583** does not operate and the controller **571** does not activate a memory bank operation. Bank control and other logic blocks placed after the OP code decoder **583** are controlled by results of the OP code decoder **583**. This can result in power savings. Holding a high or low state in CMOS logic results in low power consumption, as leak current is very low.

[0062] In the illustrated example, the multiplexer **533** conditionally passes the command input signal SCIi as the command input signal SCI(i+1) based on whether the ID match signal Id_match is high. Although this relates to the truncation, its operation is briefly described here because it is provided in combination with the clock generation. If the ID match signal Id_match is low, then the output of the multiplexer **533** passes the command input signal SCIi, so that subsequent memory devices (e.g., the memory device (i+1)) can receive the command. However, if there is a match, then the command input signal SCI(i+1) is provided by the output of the data output register **573**, which contains static data so long as no data is being read out from the data output register **573**. This results in the command input signal SCIi(i+1) having no transitions unlike the command input signal SCIi. This results in power savings for subsequent memory devices. Although examples presented herein use a multiplexer for the truncation, more generally, any data path selector can be implemented. More generally, the command input signal SCI (i+1) can be connected to something that produces no transitions than forwarding the command input. For example, the output may be connected to ground. Further details of the truncation are provided later as truncation embodiment.

[0063] The internal clock generator **543** shown in FIG. **5** contains appropriate circuitry for implementing the functionality described above. Example circuitry will be later described with reference to FIG. **7**.

[0064] In operation, the OP code register **581** contains an OP code from a received command when the OP code clock has been enabled for the particular device, i.e., when the command contained an ID that matched that of the particular device. The contents of the OP code register **581** are provided to the OP code decoder **583** as ip_opc<n−1:0>. If the ID match signal Id_match is high, then the OP code decoder **583** decodes the OP code ip_opc<n−1:0>. The OP code decoder **583** outputs a decoded OP code Idop, which is a decoded version of the OP code ip_opc<n−1:0> from the OP code register **581**. However, if the ID match signal Id_match is low, then the OP code decoder **583** does not decode the OP code ip_opc<n−1:0>. The decoded OP code Idop is implementation specific and may for example include any one or more appropriate signals for execution of the OP code.

[0065] FIG. **7** shows the internal clock generator **543** shown in FIG. **5**. Referring to FIG. **7**, the internal clock generator **543** includes an ID clock generator **611**, an OP code clock generator **613**, an address clock generator **615** and a data clock generator **617**. Details of the ID clock generator **611** and the OP code clock generator **613** are provided below with reference to FIG. **8**A, while details of the address clock generator **615** and the data clock generator **617** are provided below with reference to FIG. **8**B.

[0066] Each of the clock generators **611**, **613**, **615**, **617** is connected to receive the buffered clock signal Isck and the buffered command strobe signal Iscsi. The ID clock generator **611** starts when the buffered command strobe signal Iscsi is enabled, and generates the ID clock signal Clk_id with enough transitions to process the ID number of the incoming command. The internal clock generator **543** receives the ID match signal Id_match, which determines whether to generate the OP code clock signal Clk_opc, address clock signal Clk_add, data clock signal Clk_data used for processing the command. The address clock generator **615** and the data clock generator **617** are connected to receive the decoded OP code signal Idop containing OPM**1**, OPM**2**, OPM**3**, which indicate whether there is address and data in the command.

[0067] FIG. **8**A shows the ID clock generator **611** and the OP code clock generator **613** shown in FIG. **7**. Referring to FIG. **8**A, the ID clock generator **611** includes a counter **651** having a clock input IN for receiving the buffered clock signal Isck and an enable input EN for receiving the buffered command strobe signal Iscsi and the count output signal from its output OUT is fed to a count determiner **653**. An output signal **655** of the count determiner **653** is fed to an AND gate **657**. When the count starts, the signal **655** goes high. The count reaches a pre-determined count number (e.g., 8 corresponding to the bit number of the ID), the signal **655** goes low. During the high state of the signal **655** (VI**1** clock cycle), the clock pulses of the clock signal Isck pass the AND gate **657**, so that the ID clock signal Clk_id is provided.

[0068] Similarly, the OP code clock generator **613** includes a counter **661**, a count determiner **663** and an AND gate **667**. An input IN of the counter **661** receives the clock signal Isck, an enable input EN**1** receives the ID match signal Id_match and another enable input EN**2** receives the buffered command strobe signal Iscsi. When the counter **661** is enabled by Id_match and the command strobe signal Isci, it counts the clocks of Isck. When the count starts, a determination signal **665** goes high and the count reaches a pre-determined count number (e.g., 8 corresponding to the bit number of the OP code), the signal **665** goes low. During the high state of the signal **665** (VI**2** clock cycle), the clock pulses of the clock signal Isck pass the AND gate **667**, so that the OP code clock signal Clk_opc is provided.

[0069] FIG. **8**B shows the address clock generator **615** and the data clock generator **617** shown in FIG. **7**. Referring to FIG. **8**B, the address clock generator **615** and the data clock generator **617** operate only if the ID match signal Id_match is asserted. The address clock generator **615** includes a counter **671**, a count determiner **673**, a limit value circuit **675** and an AND gate **677**. The limit value circuit **675** includes a temporary registration decoder **676** and a register **678**. The counter **671** is activated by the ID match signal Id_match and the buffered command strobe signal Iscsi and thereafter, counts pulses of the buffered clock signal Isck continuously that is fed to its count input IN.

[0070] Referring to FIG. **8**B, the three bit operation mode signals OPM**1**, OPM**2** and OPM**3** contained in the decoded OP code signal Idop are fed to the temporary registration decoder **676** of the limit value circuit **675**. The temporary registration decoder **676** decodes OPM**1**, OPM**2** and OPM**3** and its decoded value V**13** is registered in the register **678**. The count determiner **673** determines whether the count by the counter **671** reaches the limit value V**13** defined by OPM**1**, OPM**2** and OPM**3** held in the register **678**. The count determiner **673** provides a count determination output signal **675** to the AND gate **677** that receives the buffered clock signal Isck. The count determination output signal **675** becomes “high” when the counter **671** starts counting and “low” when the count reaches the limit value V**13**. By gating in response to the signal **675**, the AND gate **677** outputs V**13** clocks that are to be contained in the address clock signal Clk_add. In this example, V**13** defined by OPM**1**, OMP**2** and OPM**3** represents the total number of bits of the column and row addresses. The address clock generator **615** allows for the generation of an address clock with varying numbers of pulses, the number of pulses being determined based on the OPM**1**, OPM**2**, OPM**3**. Of course, if the number of pulses is fixed, then a circuit similar to that used for OP code clock generation can be employed. Other mechanisms for determining the length of the address clock can be employed.

[0071] Similarly, the data clock generator **617** includes a counter **681**, a count determiner **683**, a limit value circuit **685** and an AND gate **687**. The limit value circuit **685** includes a data registration decoder **686** and a register **688**. The decoding function of the data registration decoder **686** is different from that of the temporary registration decoder **676**. The counter **681** is activated by the ID match signal Id_match and the buffered command strobe signal Iscsi and thereafter, counts pulses of the buffered clock signal Isck continuously, the Isck being input to an input IN. The three bit operation mode signals OPM**1**, OPM**2** and OPM**3** of the decoded OP code signal Idop are decoded by the data registration decoder **686** and its decoded value VI**4** is registered in the register **688**.

[0072] The count determiner **683** determines whether the count by the counter **681** reaches the limit value VI**4** defined by OPM**1**, OPM**2** and OPM**3** held in the register **688**. The count determiner **683** provides a count determination output signal **685** to the AND gate **687**. The count determination output signal **685** becomes "high" when the counter **681** starts counting and "low" when the count reaches the limit value VI**4**. By gating in response to the signal **685**, the AND gate **687** outputs VI**4** clocks that are to be contained in the data clock signal Clk_data. In this example, VI**4** defined by OPM**1**, OPM**2** and OPM**3** represents the total number of bits of the data. Therefore, the data length is determined based on OPM**1**, OPM**2** and OPM**3**.

[0073] In other implementations, the length of the data clock is determined based on how long the command strobe signal SCSi is asserted by the memory controller. In such implementations, generation of the data clock signal Clk_data is stopped by a transition of the OP code strobe signal SCSi from high to low, and not by the count determination circuit **683** determining when the counter **681** has reached VI**4** as defined by OPM**1**, OPM**2** and OPM**3**. Accordingly, such implementations may not have the counter **681**, the count determination circuit **683**, or the limit value circuit **685**. Rather, such implementations would be provided with any appropriate circuitry so that generation of the data clock signal Clk_data starts and finishes upon a transition of the command strobe signal SCSi from high to low.

[0074] FIG. **9**A shows a signaling diagram of example signals that are generated by the circuitry shown in FIG. **6** where it is assumed a command has been received that contains an ID that matches that of the particular device. The signaling diagram includes the command strobe signal SCSi, the ID match signal Id_match, the OP code ip_opc<n−1:0>, and the decoded OP code Idop. In this example, it is assumed that there is an ID match and therefore the ID match signal Id_match is asserted. This causes the OP code ip_opc<n−1:0> to be decoded so as to generate the decoded OP code Idop.

[0075] FIG. **9**B is another signaling diagram of example signals that are generated by the circuitry shown in FIG. **6** where it is assumed a command has been received that contains an ID that does not match that of the particular device. The signaling diagram includes the buffered OP code strobe input signal Iscsi, the ID match signal Id_match, the OP code ip_opc<n−1:0>, and the decoded OP code Idop. In this example, it is assumed that there is no ID match. Therefore, the ID match signal Id_match is not asserted. Accordingly, the input OP code is an unknown value because the OP code of the command was not loaded into the OP code register **581**. With reference to FIG. **5**, this is due to the OP code clock signal Clk_opc not being generated by the internal clock generator **543**. The decoded OP code is low because the OP code decoder **583** does not decode the input OP code.

[0076] The ID match signal Id_match is used to initiate OP code decoding. The ID match signal Id_match is not always high. Rather, the ID match signal Id_match is high only at the target memory device in the series-connected devices. Invalid glitch logic generation from decoding logic is avoided since the OP code decoder **583** takes the ID match signal Id_match. The ID match signal Id_match can control the result of the OP code decoder **583**. Using this logic, when the ID match signal Id_match is low, current generation after the OP code decoder **583** can be reduced and therefore reduced power consumption can be realized.

[0077] FIG. **10**A is a signaling diagram for example signals wherein the device ID matches the ID contained in an incoming command of the command input signal SCIi in the circuitry shown in FIG. **5**.

[0078] Referring to FIG. **10**A, the signaling diagram includes the clock signal SCK, the command strobe signal SCSi, the buffered internal clock signal Isck, the ID match signal Id_match, the ID clock signal Clk_id, the OP code clock signal Clk_opc, the address clock signal Clk_add, and the data clock signal Clk_data. Note that the ID clock signal Clk_id is active for the first 8 clock cycles in order to load 8 bits of the ID number of the command. In this example, it is assumed that the ID number matches the device address. Therefore, the ID match signal Id_match is asserted. This causes the OP code clock signal Clk_opc, the address clock signal Clk_add, and the data clock signal Clk_data to be subsequently generated in turn in order to load and process the OP code, additional addresses, and data of the command.

[0079] FIG. **10**B is a signaling diagram for example signals wherein the device ID does not match the ID contained in the incoming command signal SCIi in the circuitry shown in FIG. **5**. The signals correspond to and are identically labeled as those of FIG. **9**A. The ID clock signal Clk_id is active for the first 8 clock cycles in order to load 8 bits of the ID number of the command. In this example, it is assumed that the ID number does not match the device address. Therefore, the ID match signal Id_match remains low. This causes clock generation to be stopped for the OP code clock signal Clk_opc, the address clock signal Clk_add, and the data clock signal Clk_data. In the illustrated example, this occurs at around the tenth clock cycle. It is, thus, that no OP code clock signal Clk_ope is generated in the event that the ID match signal Id_match remains low. It may, however, one pulse may be generated for the OP code clock signal Clk_opc before its generation is halted, depending on the circuit performance and the clock frequency, as shown in FIG. **10**B.

[0080] FIG. **11** shows a system having a serial interconnection arrangement including a plurality of devices. The system of FIG. **11** corresponds to the system shown in FIG. **2** and thus, devices shown in FIG. **11** has similar structure to that of the devices of FIG. **2**. Note that control signals such as the command and data strobe signals are not shown in FIG. **11**, for the purpose of simplicity.

[0081] Referring to FIG. **11**, a system **710** includes a plurality of memory devices that are connected in-series. The system **710** includes a memory controller **720** together with a plurality of memory devices **730-1**, **730-2**, **730-3**, - - - , **730-15** that are interconnected via serial links. The serial links form a common operation path **741**. Each of the memory devices **730-1**, **730-2**, **730-3**, - - - , **730-15** has ID match and command forwarding circuitry **734-1**, **734-2**, **734-3**, - - - , **734-15**, respectively, and control logic circuitry **732-1**, **732-2**, **732-3**, - - - , **732-15**, respectively. The ID match and command forwarding circuitry is responsible for taking the input command, and forwarding it on to the next device. In the illustrated example, the ID match and command forwarding circuitry in each memory device is shown to constitute a D-FF. A more detailed example of the ID match and command forwarding circuitry and control logic circuitry is similar to that of FIG. **8**. In the illustrated example, 15 memory devices **730-1**, **730-2**, **730-3**, - - - , **730-15** are shown. However, more generally, there may be any appropriate number of memory devices.

[0082] In operation, the memory controller **720** sends commands over the common operation path **741**. For each of the memory devices **730-1**, **730-2**, **730-3**, - - - , **730-15**, the respective control logic circuitry **732-1**, **732-2**, **732-3**, - - - , **732-15** operates with normal power consumption for commands that are addressed to the memory device, and is operated with reduced power consumption for commands that are not addressed to the memory device. In the illustrated example, it is assumed that the memory controller **720** sends a command addressed to the third memory device **730-3** that is a target or designated device. The command includes an ID number (a binary code of "0010") as described in FIG. **3**, so as to uniquely identify the designated memory device **730-3**.

[0083] The first memory device **730-1** receives the command contained in the command input signal SCI**1** and determines whether it is addressed to another memory device and therefore its control logic circuitry **732-1** is operated with reduced power consumption. The first memory device **730-1** forwards the command to the second memory device **730-2**, the control logic circuitry **732-2** of which similarly operates with reduced power consumption because the command is addressed to another memory device. The second memory device **730-2** forwards the command to the third memory device **730-3**, the control logic circuitry **732-3** of which operates with normal power consumption for processing the command because the command is addressed to the third memory device **730-3**. For each of subsequent memory devices (i.e., the fourth device—the 15th device **730-15**), the control logic circuitry (e.g., the control logic circuitry **732-15**) operates with reduced power consumption, because no command is received that is addressed to the memory device **730-15**.

[0084] Note that the memory devices other than the designated memory device **730-3** reduce their power consumption in a first area as indicated at **743**. The first area **743** includes the control logic circuitry **732-1**, **732-2**, - - - , **732-15** of the memory devices **730-1**, **730-2**, - - - , **730-15**. Power consumption is reduced in the first area **743** because there is no need for the other memory devices to employ their control logic circuitry to process the command. Only the designated memory device **730-3** processes the command. In some implementations, power consumption is reduced in each control logic circuitry **732-1**, **732-2**, - - - , **732-15** by holding off from generating one or more clock signals used for command processing. Further details of how power consumption can be reduced by holding off from generating one or more clock signals is provided later.

[0085] Power consumption can be reduced in a second area because **745** there is no need for the subsequent memory device to receive the command transmitted over the common operation path **741**. However, note that if the command contains an OP code that requisitions an output by the third memory device **730-3** (for example a core memory output, or a status register output), then the third memory device **730-3** provides the output to the subsequent memory devices. In this case, the output from the device may be enabled by a data strobe signal from the memory controller **720**. This would result in logic transitions in the second area **745** thereby causing power consumption. There is still a reduction in power consumption in the second area **745** because the command per se is not forwarded. Further details of how power consumption can be reduced by truncating commands are provided below.

[0086] Power consumption is reduced by designing an appropriate set of OP codes for use in commands in order to achieve fewer logic transitions. Further details of how this may be accomplished are provided below.

[0087] Note that in a given implementation, mechanisms to power down the control logic circuitry of memory devices that have not been designated, and/or mechanisms to reduce logic transitions in devices subsequent to a designated device, and/or reduced transition OP code sets may be implemented.

[0088] In another implementation, power consumption can be reduced in devices subsequent to a memory device to which a command is addressed. This is achieved by the memory device to which the command is addressed not forwarding at least part of the command on to the next memory device in the set of memory devices interconnected. For the specific example of FIG. **11**, for each of the memory devices **730-1**, **730-2**, **730-3**, - - - , **730-15**, the ID match and command forwarding circuitry **734-1**, **734-2**, **734-3**, - - - , **734-15** forwards the command only if the command is not addressed to the memory device. The ID match and command forwarding circuitry **734-1**, **734-2** of the first two memory devices **730-1**, **730-2** forward the command because the command is addressed to another memory device. However, the ID match and command forwarding circuitry **734-3** of the third memory device **730-3** does not forward the command to the subsequent memory device. Rather, the ID match and command forwarding circuitry **734-3** of the third memory device **730-3** provides a substantially static output for the subsequent memory device. Due to the latency within a device determining a given command is addressed to itself, part of the command may be forwarded to the subsequent device. The remainder is truncated upon the device determining the command is addressed to itself. In this manner, the ID match and command forwarding circuitry **734-3** of the third memory device **730-3** truncates the command so that subsequent memory devices receive a substantially static signal with no transitions. The substantially static signal results in no transitions in a second area indicated at **745**, which results in reduced power consumption.

[0089] In embodiments featuring truncation, after a given memory device determines that a received command is for that device, the memory device does not forward the command input on to the command input of the next device. An example of a system featuring such truncation will now be described with reference to FIG. **12**.

[0090] FIG. **12** shows a system according to another embodiment of the present invention. A system arrangement shown in FIG. **12** is similar to that of FIG. **11**. Referring to

FIG. **12**, a system **810** has a memory controller **820** and a plurality of devices. In the particular example, four devices are connected in-series. Another arrangement including different number of devices is applicable.

[0091] The system **810** has memory devices **830-1**, **830-2**, **830-3**, **830-4** that are connected in-series. The devices have respective link forwarding logic circuitry that is comprised of a respective D-FF for the command input, and a respective D-FF for a command strobe signal. A common connection for the clock SCK is also shown. The command output from the memory controller **820** follows a path including links **841**, **843**, **845**, **847**, **849** and a command strobe signal from the memory controller **820** follows a path including links **851**, **853**, **855**, **857**, **859**. In the absence of truncation, the command output from the memory controller **820** will be passed along all the links **841**, **843**, **845**, **847**, and the command strobe signal will be passed along all the links **851**, **853**, **855**, **857**, **859**. In contrast, where truncation is to be performed, the command input will not be passed along the links that follow the device to which a given command is addressed. In the event that the read data is passed through the links **845**, **847**, **849** with the data strobe signal so that the memory controller knows the start point of valid read data, when read operation is instructed, the data strobe signal is not truncated even though the strobe signal arrives at the designated device.

[0092] For the purpose of illustration, FIG. **12** shows the serial interconnection arrangement of series-connected devices implementing truncation. In the particular example, a command is addressed to the second memory device **830-2**. The memory controller **820** outputs a command from its command output, and this is forwarded along links **841**, **843** to device **830-2**. The command includes an ID number ID=0001 that matches the device address DA=0001 of the second memory device **830-2**. The second memory device **830-2**, upon determining there is a match in the ID number, operates to truncate the command input by not forwarding it along to the following device **830-3** along path **845**. The result is that subsequent devices **830-3**, **830-4** consume less power. The devices generally indicated at **873** can operate with reduced power consumption. The link forwarding logic of the following devices does not need to process the command.

[0093] FIG. **13** is a signaling diagram of example signals generated by the system shown in FIG. **12**. Referring to FIG. **13**, the signaling diagram includes the clock signal SCK, the command strobe signal SCS**1** and the command input signal SCI**1** to the first memory device **830-1**, the command strobe signal SCS**2** and the command input signal SCI**2** to the second memory device **830-2**, and the command strobe signal SCS**3** and the command input signal SCI**3** to the third memory device **830-3**. Note that the command input signals SCI**1**, SCI**2** to the first and second memory devices **830-1**, **830-2** include all bits of the command, which in this example includes 8 bits for the ID number, 8 bits for the OP code, and 8 bits for the data. However, the command input signal SCI**3** to the third memory device **830-3** has been truncated by the second memory device **830-2** after the 8 bits of the ID number. This is because the second memory device **830-2** has determined that it is the target device and therefore subsequent devices **830-3**, **830-4** do not require the command.

[0094] FIG. **14** shows example circuitry of a memory device for implementing a command truncation feature. Referring to FIG. **14**, circuitry **930** receives a command input signal SCIi, a clock signal SCK, a command strobe signal SCSi, and a data strobe signal SDSi. All signals are buffered by respective input buffers **921**, **923**, **925**, **927**. The command input SCIi and the clock signal SCK are fed to a D-FF **931**, which in turn is connected to a multiplexer **933**. The multiplexer **933** provides a command input signal SCI(i+1) to the next device (i+1) through an output buffer **929**. The command strobe signal SCSi is provided to a peripheral logic block **960**. The data strobe signal SDSi is fed to an output latch generator **941**, which in turn is connected to an output clock generator **943** and a register **945**. A core control block and memory core **950** is connected to the register **945**. The register **945** has an output to the multiplexer **933**.

[0095] In operation, the circuitry **930** functions to, on the one hand, forward on the command input signal SCIi as SCI(i+1) to the next (i+1)-th device when a command is not addressed to the i-th device. The circuitry **930** functions, on the other hand, to connect the output to the output register **945**, both upon receipt of a command that is addressed to that device so as to prevent the command from being forwarded to the next device, and to allow outputs from that device to be produced as the command input signal SCI(i+1) for the next (i+1)-th device.

[0096] The command input signal SCIi is fed through the D-FF **931** to the multiplexer **933**. The multiplexer **933** determines whether the command input signal SCI(i+1) is to be provided from the i-th command input signal SCIi or from the output of the register **945**. The multiplexer **933** selects between the command input signal SCIi and the output of the register **945** based on whether the ID match signal Id_match is high, which is high when the memory device has been selected and low when the i-th memory device is unselected. In the illustrated example, the ID match signal Id_match is determined by the peripheral logic block **960** that corresponds to those of the circuitry of FIG. **5**. When the memory device receives the command including the ID number that matches its device address, then the ID match signal Id_match goes to high and the multiplexer **933** is switched to select the output dout of the register **945**, so that the dout is provided as SCI(i+1). If the ID match signal Id_match is low, then the command input signal SCIi is provided as the SCI (i+1). The command input signal SCIi is forwarded on as the command input signal SCI(i+1) for a brief period of time that is determined by the internal logic combination, (in this case, a time period equal to a number of cycles necessary to clock in the device ID) and after that the input is truncated, and a fixed output value from the register **945** is transmitted as the (i+1)-th command input signal SCI(i+1).

[0097] The command input signal SCI(i+1) derived from dout is static because no output of register **945** is produced (i.e., clocked out) until the output clock is generated by output clock generator **943**, and this is not done until the register **945** is asserted to output. The switching operation of the multiplexer **933** allows subsequent (downstream) devices to receive the static level in the event that the i-th memory device has been selected. Therefore, subsequent devices take the static value and no transitions occur at their input and output buffers. However, note that if the command contains an OP code that is a read OP code requisitioning a response, then read data is clocked out of the register **945** and provided to the (i+1) command input signal SCI(i+1) while the data strobe signal SDSi is asserted.

[0098] Generation of the output dout from the register **945** will be described with reference to FIG. **15**. FIG. **15** shows a signaling diagram of example signals that are generated by the circuitry **930** shown in FIG. **14**.

[0099] Referring to FIGS. **14** and **15**, the signaling diagram includes the clock signal SCK, the data strobe signal SDSi, the output enable signal out_en, the shift clock signal shft_clk, the latch_signal latch_signal, and the data out dout. Operation starts when the data strobe signal SDSi is asserted after the read command is issued. The output latch generator **941** generates the latch_signal latch_signal to instruct the register **945** to latch the contents read from the memory core of the core control block and memory core **950**. The output latch generator **941** also generates the output enable signal out_en to enable the output clock generator **943**. The output clock generator **943** generates the shift clock signal shft_clock when the output enable signal out_en is high. The shift clock signal shft_clk is used to clock data out of the register **945**. Accordingly, the data output dout from the register **945** is provided to the multiplexer **933**.

[0100] For write operations with long command input, this works with the same effect. No matter what OP code is input with the ID number, if the ID number is matched with that of the device, the ID match signal Id_match is asserted and then the fixed level of data output dout is transmitted to the next device as the command input signal SCI(i+1). After the selected device, unselected devices can save power consumption even at the input and output buffers.

[0101] In the illustrated example, truncation of data is performed between the command input signal SCIi and the command input signal SCI(i+1). Truncation of the command strobe can also be performed, but it does not affect the total power saving because the command strobe has two transitions when compared to data so that early truncation of the command input signal SCIi has same power consumption as no truncation case of the command input signal SCIi. Note that a static value can be taken from the register **945** for truncating the command strobe.

[0102] In the illustrated example, a static value is taken from the register **945**. During a read operation the contents from the register **945** are clocked out and sent as the next command input. Therefore, taking the static value from the register **945** during a write operation makes use of an existing data path. However, it is to be understood that a static value can be taken from any source that can provide the static value. The static value may, for example, be taken directly from a supply ground. Other implementations are possible.

[0103] FIG. **16** shows an example correlation between a selected device number and power consumption with the truncation feature. Referring now to FIG. **16**, the graph plots average I/O power consumption Pav (normalized value) verses the selected device number Nvd. An approximately linear relationship can be seen. This means that power consumption is greater when the selected device number is greater. With reference to FIG. **12**, this is due to the area **873** being smaller, which results in power consumption savings being smaller. This power consumption depends on which device is selected with the ID number. If the last memory device in the serial interconnection arrangement is selected, then power saving for I/O part cannot be expected with the truncation embodiment. However, in other cases where a memory device that is not the last memory device is selected, power saving for I/O part is accomplished.

[0104] Power consumption in a specified period of time is affected by the number of transitions of logic value in the specified period. Therefore, if one can assign OP codes resulting in fewer transitions of logic, then power savings can be attained. Although the total power saving from this technique may be minor when compared to other methodologies such as those described above, this technique should nonetheless be considered for reducing input/output power consumption when data flows through unselected memory devices before reaching the target memory device.

[0105] Designing OP codes may not significantly affect the total power saving in the case of transitions due to OP codes being much smaller in number than transitions due to the data. Also, the data can include random bit combinations. However, the OP code consideration to reduce power may be helpful for OP code based operations like multiple register access.

[0106] Table 2 shows example OP codes and corresponding numbers of bit transitions.

TABLE 2

OP Code	Translations
00h (0000 0000)	No transition
01h (0000 0001)	Two transitions (low_to_high, high_to_low)
02h (0000 0010)	Two transitions (low_to_high, high_to_low)
03h (0000 0011)	Two transitions (low_to_high, high_to_low)
04h (0000 0100)	Two transitions (low_to_high, high_to_low)
05h (0000 0101)	Four transitions
06h (0000 0110)	Two transitions (low_to_high, high_to_low)
07h (0000 0111)	Two transitions (low_to_high, high_to_low)
08h (0000 1000)	Two transitions (low_to_high, high_to_low)
09h (0000 1001)	Four transitions
0Ah (0000 1010)	Four transitions
0Bh (0000 1011)	Four transitions
0Ch (0000 1100)	Two transitions (low_to_high, high_to_low)
0Dh (0000 1101)	Four transitions
0Eh (0000 1110)	Two transitions (low_to_high, high_to_low)
0Fh (0000 1111)	Two transitions (low_to_high, high_to_low)

[0107] Table 2 includes 16 possible OP codes from 00h up to 0Fh. Other OP codes are possible. A bit transition is counted when two adjacent bits have different values. For example, OP code 05h (0000 0101) has four transitions: 0→1, 1→0, 0→1, and 1→0. Note that the number of bit transitions counted does not include possible bit transitions that may occur with adjacent bits that are not shown. For example, with reference to FIG. **3**, the OP code is preceded by an ID number, and may be followed by data or an address. Bit transitions between the least significant bit of the ID number and the most significant but of the OP code are not considered. Similarly, bit transitions between the most significant bit of the OP code and the least significant but of the address or data is not considered. However, if such bit transitions were considered, then values indicated in the table would be different. For instance, the first OP code 00h may have up to two bit transitions depending on whether the least significant bit of the ID number and most significant bit of the following data/address are '1'. It is assumed that such preceding/following transitions are random.

[0108] In Table 2, some of the OP codes have four bit transitions, while others have only two bit transitions. The first OP code 00h has zero transitions. The OP codes that involve fewer transitions are preferred because power savings can be realized with fewer bit transitions. Therefore an OP code set can be designed to include OP codes having fewer transitions. An example OP code set may for example consist of 00h, 01h, 02h, 03h, 04h, 06h, 07h, 08h, 0Ch, 0Eh, and 0Fh. Other OP code sets are possible. In addition, OP codes can be assigned on the basis of the statistical likelihood of their occurring. The most frequently used operations can be associated with OP codes having fewer transitions, while the least frequently used operations can be associated with OP codes having more transitions.

[0109] In the embodiments described above, the device elements and circuits are connected to each other as shown in the Figures, for the sake of simplicity. In practical applications of the present invention, elements, circuits, etc. may be connected directly to each other. As well, elements, circuits etc. may be connected indirectly to each other through other elements, circuits, etc., necessary for operation of the devices or apparatus. Thus, in actual arrangement of devices and apparatus, the elements and circuits are directly or indirectly coupled with or connected to each other.

[0110] The above-described embodiments of the present invention are intended to be examples only. Alterations, modifications and variations may be effected to the particular embodiments by those of skill in the art without departing from the scope of the invention, which is defined solely by the claims appended hereto.

What is claimed is:

1. A semiconductor device for use in a serial interconnection arrangement of semiconductor devices, the semiconductor device comprising:

command circuitry for receiving commands and forwarding at least some of the commands; and

core circuitry configured to operate with normal power consumption for those commands that are addressed to the semiconductor device, and operate with reduced power consumption for those commands that are not addressed to the semiconductor device.

2. The semiconductor device of claim **1** wherein each command comprises an ID number, the semiconductor device further comprising:

a storage element for storing a device address of the semiconductor device; and

a determiner for determining for each command if the command is addressed to the semiconductor device based on the ID number of the command and the device address of the semiconductor device.

3. The semiconductor device of claim **2** wherein:

the core circuitry comprises an internal clock producer for producing at least one clock for processing commands;

for each command,

the internal clock producer is enabled upon determining that the ID number of the command matches the device address of the semiconductor device, the core circuitry thereby operating with the normal power consumption;

the internal clock producer is disabled upon determining that the ID number of the command does not match the device address of the semiconductor device, the core circuitry thereby operating with the reduced power consumption.

4. The semiconductor device of claim **3** wherein:

for each command received, the command circuitry forwards the command if the ID number of the command does not match the device address of the semiconductor device.

5. The semiconductor device of claim **4** wherein each command further comprises an OP code, the internal clock producer comprising an OP code clock producer for producing an OP code clock if the ID number of the command matches the device address of the semiconductor device, the OP code clock facilitating processing of an OP code of the command by the core circuitry.

6. The semiconductor device of claim **5** wherein each command further comprises data, the internal clock prouder comprising a data clock producer for producing a data clock if the ID number of the command matches the device address of the semiconductor device and the OP code of the command indicates that there is data included in the command, the data clock facilitating processing of the data of the command by the core circuitry.

7. The semiconductor device of claim **5** wherein each command further comprises address information, the internal clock producer comprising an address clock producer for producing an address clock if the ID number of the command matches the device address of the semiconductor device and the OP code of the command indicates that there is memory address information included in the command, the address clock facilitating processing of the address information of the command by the core circuitry.

8. The semiconductor device of claim **5** wherein each command further comprises address information and data, the internal clock producer comprising an address and data clock producer for producing address and data clocks if the ID number of the command matches the device address of the semiconductor device and the OP code of the command indicates that there are address information and data included in the command, the address and data clocks facilitating processing of the address and data of the command by the core circuitry.

9. The semiconductor device of claim **1** wherein the core circuitry comprises a memory for storing data.

10. The semiconductor device of claim **1** wherein each command comprises an ID number, the command circuitry comprises:

a storage element for storing a device address of the semiconductor device; and

an ID match determiner for determining whether the command is addressed to the semiconductor device based on the ID number, the ID match determining for each command if the command is addressed to the semiconductor device by comparing the ID number of the command with the device address of the semiconductor device;

(a) if the ID number of the command does not match the device address of the semiconductor device, forwards the command; and

(b) if the ID number of the command matches the device address of the semiconductor device, provides a substantially static output.

11. The semiconductor device of claim **9** wherein:

the core circuitry provides providing a read output, the substantially static output comprising an output from the memory that is generated while a read operation is not in progress.

12. The semiconductor device of claim **1** wherein the core circuitry is associated with a register that contains a most recently read output from the memory, wherein the output from the memory comprises an output of the register.

13. The semiconductor device of claim **1**, wherein the core circuitry comprises:

a processor for executing a received command to perform the normal power operation in response to the match.

14. The semiconductor device of claim **13**, wherein the core circuitry further comprises:

enabling circuitry for enabling the processor to perform the normal power operations in response to the match.

15. The semiconductor device of claim **14**, wherein the enabling circuitry comprises:

a driving signal provider for providing a driving signal to the processor in response to the match, so that the processor performs the normal power operation in response to the driving signal.

16. The semiconductor device of claim **1** wherein comprises:

a memory element containing an identification of a device address;

an input connection for receiving a command comprising an ID number;

an output connection;

an internal clock producer for:

(a) if the ID number matches the device address, producing internal clock signals used for processing the command; and

(b) if the ID number does not match the device address, holding off from generating the internal clock signals.

17. The semiconductor device of claim **16** further comprising:

a data path selector for:

(a) if the ID number does not match the device address, forwarding the command over the output interface; and

(b) if the ID number matches the device address, providing a substantially static output over the output interface.

18. The semiconductor device of claim **1** wherein the core circuitry comprises a processor for processing incoming commands addressed to the device.

19. The semiconductor device of claim **18** wherein the core circuitry comprises a memory for storing data, the memory being configured to be accessed by the processor.

20. A method in a semiconductor device of a serial interconnection arrangement of semiconductor devices, the method comprising:

receiving commands and forwarding at least some of the commands at the semiconductor device;

operating the semiconductor device with normal power consumption for those commands that are addressed to the semiconductor device; and

operating the semiconductor device with reduced power consumption for those commands that are not addressed to that semiconductor device.

21. The method of claim **20** wherein each command comprises an ID number, the method further comprising:

maintaining a device address of the semiconductor device;

determining whether the command is addressed to the semiconductor device based on the ID number of the command and a device address associated with the semiconductor device, in response to a determination result, the semiconductor device being operated with normal power consumption or reduced power consumption.

22. The method of claim **21** wherein:

receiving commands and forwarding at least some of the commands;

if the command is not addressed to the semiconductor device, the semiconductor device forwarding the command; and

if the command is addressed to the semiconductor device, the semiconductor device providing a substantially static output.

23. The method of claim **22** wherein each command comprises an ID number, the method further comprising:

maintaining a device address of the semiconductor device;

determining if the command is addressed to the semiconductor device by comparing the ID number of the command to a device address of the semiconductor device.

24. The method of claim **23** further comprising:

maintaining an identification of a device address;

receiving a command comprising an ID number over an input connection;

if the ID number matches the device address, producing internal clock signals used for processing the command and providing a substantially static output over the output interface; and

if the ID number does not match the device address, holding off from generating the internal clock signals and forwarding the command over the output interface.

25. A system comprising:

a controller; and

a plurality of semiconductor devices connected in-series, one of the semiconductor devices being connected to the controller, each of the semiconductor devices including:

command circuitry for receiving commands and forwarding at least some of the commands; and

core circuitry configured to operate with normal power consumption for those commands that are addressed to the semiconductor device, and operates with reduced power consumption for those commands that are not addressed to the semiconductor device.

* * * * *