also use this interface, for example, to install special software on the microprocessor **51** in support of testing or related diagnostic functions.

The above description provides one example of an exemplary memory section. Other methods and systems may be used for implementing a memory section without departing from the scope of the invention. For example, the discussion below presents a different exemplary embodiment of a memory section using PCI bus technology.

Switches

FIG. **6** illustrates a functional diagram of a switch **22**, in accordance with methods and system consistent with the invention. As illustrated, the switch **22** includes a switch/server communications interface **204** for interfacing with a server **12**, a switch/memory section communications interface **208**, a switch fabric **206**, and a switch controller **202**. The switch/server communications interface **204** and switch/memory section communications interface **208** may be standard switch interfaces found in commercially available switches and the terms memory section and server are used to indicate the devices to which the connections leaving the switch **22** preferably connect. The switch fabric **22** may be any type of switch fabric, such as an IP switch fabric, an FDDI switch fabric, an ATM switch fabric, an Ethernet switch fabric, an OC-x type switch fabric, or a Fibre channel switch fabric. Thus, the switch **22** may be any type of commercially available switch.

In this embodiment, the management complex **26** of the storage hub **10** may exercise control over the switch **22** through the switch controller **202**, and may exercise control over the communications channel interface **46** of the memory section **30** through the section controller. For example, as discussed above, the management complex **26** may provide the switch controller **202** with an algorithm for switching traffic through the switch fabric **206**. Further, as discussed above, the management complex **26** may provide other information including, for example, providing the switch with new copies of the software it executes, a regular period to send a heartbeat (i.e., a signal that verifies the switch still can communicate), a list of valid communications network addresses, alarm acknowledgements, and command sets. Further, as discussed above, the management complex **26** may provide other information including, for example, instructions to copy a communications message, modify its contents, and then process the new message. The management complex **26** may provide other information including, for example, instructions to broadcast information to multiple addresses.

FIG. **7** illustrates an alternative functional diagram of the management of the switch **22** and the communications channel interface **46** of the memory section **30**, in accordance with methods and systems provided. In this embodiment, the switch controller **202** and memory section interfaces **208** need not be included in the switch **22**, and the management complex **26** of the storage hub **10** exercises direct control over the switch fabric **206** and server interfaces **204**. Thus, in this embodiment the communications channel interface **46** of the memory section **30** directly connects to the switch fabric **206**.

In an alternative embodiment to that of FIG. **6** and **7**, the selector **44** need not be included and all memory interface devices **64** may be connected to the switch fabric **206**.

FIG. **8** illustrates a diagram of an alternative exemplary switch **22** that may be used in the storage hub **10**, in accordance with methods and systems provided. More particularly FIG. **8** illustrates a switch **22** for connecting one or more memory sections **30** to one or more servers **12**. This example illustrates M servers **12-1**, **12-2**, . . . **12-M** connected to a single memory section **30**. In this example, the server interfaces **204** of the switch **22** include M switch/server communications interfaces (SSCI) **204-1** thru **204-M**, and the memory section interfaces **208** of the switch include N switch/memory section communications interfaces (SMCI) **208**. Additionally, the switch fabric **206** of the switch **22** includes one or more switching planes **808**.

In this example, the servers **12** each includes a device driver **28**, and the memory section **30** includes one or more communications channel interfaces (CCI) **46-1** thru **46-N**. In this example, P parallel lines connect each device driver **28** to the switch **22** and each CCI **46** to the switch **22**. Although in this example, the number of lines in each connection is equal, in other examples they may be different. The device driver **28** may be, for example, the above-discussed device driver **28**, or may be included in the device driver **28**.

Any of the M servers may generate and transfer a data request from its device driver **28** to a memory section **30** via the switch **22**. A server **12** may include in the data request a data block identifier that identifies a particular data block it wishes to write or a data block in the storage hub **10** that it wishes to read. The corresponding SSCI **204** of the switch **22** then receives the data request and forwards it to the switch controller **202**. The switch controller **202**, in this example, determines the memory section **30** to which the information request is destined from the data block identifier included in the data request.

To determine the memory section **30**, the switch controller **202** may, for example, consult a table that defines the relationship between data block identifiers and memory sections, use an algorithm to compute the address, or use some other technique.

Once the memory section is determined, the switch controller **202** then may establish a transmission path through each switching plane **808** for each of the parallel lines P from the device driver **28** to the SMCI **208** corresponding to the determined memory section **30**. The data request may also be modified by the switch controller **202** to contain a new address that may be used by the switch **22** in directing the data request to the correct memory section **30**. The modified data request is then transmitted across the switching planes **808** to the SMCI **208**. This transmission across the P lines may be synchronous.

While the path through the switch is established, the data may reside in a separate storage queue (not shown) in the switch or in a memory (not shown) for the switch controller **202**. The data request may also be copied and further modified by the switch controller **202** in accordance with any particular requirements of the storage hub **10**. For example, as previously discussed, the management complex **26** may instruct the storage hub **10** to back up all data that is written to the storage hub **10** or to one or more particular memory sections **30**. In such an example, the switch controller **202** may copy the write data request including the data to be stored and modify the request in accordance with any particular requirements of the CDA **16**. Then, the switch controller **202** may then establish a path for sending the write data request to the CDA **16** and then send the modified copy of the request to the CDA **16**, so that the write data is backed up. Likewise, subsequent data blocks that comprise the write request may also be sent to the memory device **30** are copied and sent to the CDA **16**. The management complex **26** may, for example, provide the switch controller **202** with any required information and software needed by the switch to determine how to

modify data requests, provide multiple destinations with copies of modified data requests, and provide multiple destinations with copies of data.

When a memory section **30** sends information such as data blocks to a server **12**, the data blocks from the memory section **30** arrive at the switch **22** through the SMCI **208** corresponding to the CCI **46** for the memory section **30** sending the data block. The data blocks may include an identifier that is inserted into the data by the memory section **30**. The memory interface devices of a memory section **30**, for example, may insert this address, as described below. Further, this address may be for example a data block identifier identifying the particular data block that was read from the memory section **30**, or a port or device to which the data is to be sent. In this example, P parallel lines connect each CCI **46** to the switch **22**, although the number of lines in each connection may be different. Further, P may be any number greater than or equal to 1.

The SMCI **208** then forwards the data block to the switch controller **202**, which determines the server **12** to which the data block is destined from an identifier (e.g., data block identifier, destination address, etc.) within the transmitted data. The switch controller **202** then establishes with this destination address or data block identifier, for each of the P lines from the CCI **46**, a path though the switch **22** to the SSCI **204** to which the data is to be sent. The switch **22** then transfers the data block across the switching planes **808** to the SSCI **204**. The transmission of a data block across the P lines may be, for example, synchronous.

FIG. **9** illustrates an alternative switch **22** connected to one or more memory sections **30**, in accordance with methods and systems provided. In this example, muxing (the combining of several data streams into fewer data streams, each on its own communications path) and demuxing (the separation of a set of data streams into more data streams, each on its own communications path) are used in both the memory section **30** and the switch **22**. In this example, P parallel lines connect each memory section's CCI **46** to the switch **22**, although the number of lines in each connection may be different.

In this example, in memory section **30-1**, Q lines emanate from memory interface device **64-1** and R lines emanate from memory interface device **64-2**. A corresponding mux (**902-1** and **902-2**) then multiplex the lines from each of these memory interface devices (**64-1** and **64-2**) into P streams, where, Q and R are positive integers greater than the positive integer P.

In memory section **30-2**, J lines emanate from memory interface device **64-4**, where J is a positive integer less than P. A demux **904** then demuxes these J lines to P lines.

The P parallel lines (streams), however, may also be muxed or demuxed anywhere along the switching path. For example, as illustrated, the P lines muxed into T-line by mux **906** after the SMCI **208-1**. The T-lines are then passed through the switching planes **808** to demux **908**, which demuxes the T-lines into P lines and passes the P-lines to an SSCI **204**.

Additionally, in embodiments employing a memory interface device including a shift array, one or more pipeline shift registers (not shown) may be inserted at points in the transmission and switching path to maintain the clock frequency of those transmissions at the appropriate multiple (muxing function) or sub-multiple (demuxing function) of the clock frequency of the memory interface device shift register array. For example, a shift register pipeline may be included in the CCI **46**.

FIG. **10** illustrates an exemplary pipeline shift register, in accordance with methods and systems provided. For this example, this pipeline shift register is inserted at the outputs of the CCI **46**, such that each of the P lines exiting a CCI **46** are attached to a latch shift register **1002-1**, **1002-2** . . . **1002**-P. As illustrated, each of the P lines is attached to the S input of the latch shift register, and its inverse is connected to the R input of the latch shift register. The latch shift registers, further receive a master clock signal that may be generated by a master clock circuit for the storage hub **10**. This master clock signal may be used by other components in the storage hub **10**, such as, for example, the memory sections. The master clock signal may be, for example, generated by the management complex **26** or separate circuitry may be used.

The output, Q, from the latch shift register **1002** is then fed to the S input of a second latch register **1004**, and the inverse of the output, $\overline{Q}$, is fed to the R input of the latch shift register **1004**. The second latch shift registers **1004** receive a slave clock signal. This slave clock signal may be produced by the same circuitry providing the master clock signal, and the slave clock signal may be, for example, the inverse of the master clock signal. The outputs, Q, from these second latch shift registers **1004-1**, **1004-2**, . . . , **1004**-P then provide the signal to the P lines exiting the memory section **30**. Although this description of a pipeline shift register was made with reference to latch shift registers, other types of shift registers may be used, such as, for example, dynamic shift registers. Further, although this description of a pipeline shift register was made with reference to attaching the pipeline shift registers to the outputs of the CCI **46**, pipeline shift registers, may be included elsewhere in the storage hub **10**, such as, for example, at any communications interface or between the switching planes **808**.

Memory Interface Device

FIG. **11** includes a more detailed block diagram of an embodiment of a memory interface device **64**, in accordance with methods and systems provided. As shown in FIG. **11**, the memory interface devices **64-1** and **64-2** may each include a write shift register array **72** and a read shift register array **74**. Both the read and write shift register arrays can include a plurality of shift registers **76** interconnected in series. Each shift register **76** of the shift register array (**72** and **74**) is connected to a connector circuitry **77** which connects the shift register **76** to a corresponding I/O pin of the memory device **66**.

As used herein, the term "shift register" refers to any register, device, stage or anything else with one or more selectable inputs that allows a signal to be received at an input and then output on the occurrence of some event, such as, for example, a control or clock signal. Although the term shift register sometimes refers to not just a single register stage, but also to a series of such registers, as used herein the term shift register refers to a single stage. A series of these shift registers is referred to herein as either a shift register chain, or a shift register string. The series set of registers is also sometime referred to as a "series array of (shift) registers" or shift register array that may be either a single chain of shift registers or parallel chains of shift registers. For example, the shift registers may be any type of shift register, whether dynamic or latching, whether single clock or master/slave clock, whether sampling or edge trigger, whether data (D), RS, or JK, or a stage of a charge coupled device (CCD), or any other type of device that shifts its input to an output on the basis of clock signal. The shift register arrays/chains may include any number of shift registers without departing rom the scope of the invention.

In this embodiment, all the write shift register arrays **72** of the memory section **30** are interconnected to form a longer

chain of write shift register arrays. As illustrated, the shift register array **72** of memory interface device **64-1** is connected via the top right I/o pin of memory interface device **64-2** to the write shift register array **72** of memory interface device **64-2** via its top left I/O pin. The write shift register array **72** of memory interface device **64-2** is then connected to the write shift register array **72** of the memory interface device **64-3**, and so on, such that all the write shift register arrays **72** of the memory section form a single chain of write shift register arrays. For ease in explanation with regard to this particular example, the shift register arrays of each memory interface device will be referred to as a shift register array, and the interconnection of these shift register arrays to form a longer chain of arrays will be referred to as a shift register chain.

Further, in this embodiment, the bottom right I/O pin of memory interface device **64-1** connects to the bottom left I/O pin of memory interface device **64-2** such that their read shift register arrays **74** form a chain. Referring to FIG. **5**, note that in this example the memory interface devices **64-3** and **64-4** are likewise connected, and so on. These pairs of read shifter register arrays **74** will be referred to as read chains.

The first memory interface device in each read chain also includes a read selector **70** that is connected to the read shift register array **74**. This read selector **70** is used for inserting an identifier (e.g., a destination address, data block identifier, etc.) and/or other header information into the read data. The identifier is an identifier that the switches **22** preferably use to switch the data to its appropriate destination. The identifier may be, for example, an identifier for the data being transmitted (e.g., a data block identifier) or a destination address identifying an address to which to send the data. For example, if the destination is a computer connected to the Internet, the destination address could be an IP address for the computer. Alternatively, the destination address could simply be an internal address for the switches **22** to use in routing the data to its destination server, in which case the server **12** will read and replace with a destination address that the network over which the data will travel uses to route the data to the appropriate destination.

The memory interface device **64** may also receive control and timing signals from the timing circuitry **61** of the section controller **54**. These control and timing pulses may be timed such that the data read from or written into a memory device **66** using the respective pulses are read or written in such a manner that the shift registers **76** of the memory interface device maintain their shifting as if only a shift was taking place. For a further description of memory interface devices incorporating shift registers, please see U.S. patent application Ser. No. 10/284,198 by William T. Lynch and David J. Herbison, entitled "Methods and Apparatus for Improved Memory Access," (issued as U.S. Pat. No. 6,879,526) which is incorporated by reference herein in its entirety. Additionally, data transmitted by a memory interface device **64** may, for example, be transmitted in common mode, differential mode, or in any other manner as deemed appropriate by the system designers.

Exemplary Writing Operation

FIG. **12** illustrates a flow chart for an exemplary writing operation for the memory section **30** of FIG. **5** with reference to FIGS. **1**, **6**, and **11**, in accordance with methods and systems provided. This flow chart illustrates but one example of a writing operation, and other methods may be used for implementing a write operation without departing from the scope of the invention. In this example, the memory devices **66** of the memory section **30** are partitioned into data blocks, where each data block is identifiable by a data block identifier. A more thorough description of data blocks and partitioning of memory devices are presented below.

A write request may originate, for example, from a user connected to a server **12** via, for example, a network. In this example, it is assumed that the user sends a block of data (i.e., a data block) that is to be stored by the storage hub **10**. A device driver **28** in the server then may determine a data block identifier for the data block to be stored and send the request, including the data block identifier (DBI), to a switch **22** within the storage hub **10** (Step **1202**). The device driver **28** may, for example, determine the data block identifier using standard methods, such as for example, the server **12** may be executing Oracle or a similar type application, which may be used to determine the data block identifier.

The switch **22** then may use the data block identifier (DBI) to direct the data request to the memory section **30** that is to store the data block by, for example, determining, based on the DBI, an address for the memory section that the

The switch **22** uses to route the data request to the memory section (Step **1204**). For example, when a data request arrives at an SSCI **204** of a switch **22**, the SSCI may, for example, direct the data request to the switch controller **202**, which may then, use a table to look up the address corresponding to the DBI, use an algorithm to compute the address from the DBI, or use some other method.

The switch controller **202** may then establish a path through the switch fabric **206** from the SSCI **204** to the SMCI **208** corresponding to the memory section where the data will be stored (Step **1206**). If there is no idle communications channel between the switch **22** and the memory section **30** or if there is a conflict, the switch may send a congestion message to the requesting server **12**, which may then queue the message until the conflict is resolved. In an embodiment of a switch **22**, such as that illustrated in FIG. **7**, the management complex **26** may perform the functions described above as being performed by the switch controller **202**.

Next, the switch **22** then forwards the data request to the CCI **46** of the memory section **30** (Step **1208**); the CCI **46** then may direct the request to the section controller **54** (Step **1210**). The section controller **54** identifies the data request as a write request, and determines if one of its memory devices has space to store the data (Step **1212**). For example, the section controller **54** may identify the request as a write request by information contained in the request itself, by the format or size of the data request, or by some other method. If there is no available memory device **66**, the section controller **54** sends a negative acknowledgement, NAK, message (Step **1214**), through the CCI **46** and switch **22** to the requesting server **12**, which, after receiving the NAK (Step **1216**), may attempt to rewrite the data to the storage hub **10** using the same or a different DBI (Step **1218**), may attempt to write to another device (not shown), or may inform the application.

If space is available, the section controller **54** sends a message to the device driver **28** that it may transmit the data to be stored (Step **1220**). In response, the device driver **28** transmits the data through the switch **22** to the memory section's **30** communications interface (CCI) **46** (Step **1222**). Additionally, the management complex **26** may also direct the switch **22** to also send write data to the CDA **16**. For example, the management complex **26** may provide an algorithm to the switch controller **206** which when executed causes all write data to be sent to both the memory section **30** where the data will be stored and to the CDA **16**. The version of the data stored by the CDA **16** will be treated as a back-up

version that in the event the memory section suffers a fault may be loaded onto a different functioning memory section.

The selector **44** then directs the data to the temporary store memory interface device **60**. The microprocessor **51** of the section controller **54** then checks the state of the memory device **66** where the data is to be stored to determine if the memory device **66** is available or is busy (Step **1224**). For example, the microprocessor **52** may store in its RAM **52** a status code for each memory device **66** in the memory section **30** that the microprocessor **51** may consult to determine the availability of the memory device **66**. If the memory device **66** is available, the microprocessor **51** sends a message to the memory device **66** through the memory device control circuitry **55** to ready itself for storing the data (Step **1232**). If, however, the memory device **66** is busy, the data temporary storage memory interface device **60** stores the data in the temporary memory storage device **58** (Step **1226**) and a write request is placed in a queue in the microprocessor's RAM **52** (Step **1228**).

When the memory device **66** becomes available, the memory device **66** signals the microprocessor **51** in the section controller **54** via the memory section control circuitry **55** (Step **1228**). This may, for example, be accomplished by the memory device **66** sending an interrupt signal to the microprocessor **51** via the memory device control circuitry **55**. Then, the microprocessor **51** sends a message to the memory device **66** through the memory device control circuitry **55** to ready itself for storing the data (Step **1232**). When the memory device is ready; the temporary storage memory interface device **60** passes the data to the T-selector **62**, which, because this is a write operation, passes the data to the memory interface device **64**. For example, if the memory device **66** were available at Step **1224**, the data need not be stored in the temporary memory storage device **58**. The data is then clocked into the shift register array **76** of the first memory interface device **64-1** where it is clocked through the write chain of shift register arrays **76** until it is loaded into the memory interface device **64** corresponding to the memory device **66** to which the data is to be written (Step **1234**). The data is then written to the memory device at an address supplied by the section controller **54** of the memory section **30** (Step **1236**).

A more detailed description of the connections between the shift register arrays and the memory devices and a method for writing the data is presented in the aforementioned U.S. patent application Ser. No. 10/284,198 by William T. Lynch and David J. Herbison entitled "Methods and Apparatus for Improved Memory Access" (issued as U.S. Pat. No. 6,879,526) filed on the same day as the present application.

When the write operation is complete, the memory device **66** informs the microprocessor **51** in the section controller **54** through the memory device interface **55** (Step **1238**). The section controller **54** may then send an acknowledgement to the device driver **28** in the requesting server **12** (Step **1240**).

The following provides a more detailed description of a technique that may be used in step **1236** for writing the data to the memory device **66**. This technique is referred to as memory latching. Memory latches may, for example be edge-triggered or flip flop circuits with common reset lines. That is, the memory device control circuitry **55** may, for example, include one or more of these memory latches along with other appropriate circuitry for performing this technique.

The section controller **54** may reserve memory locations in its internal memory **52** for the management of the memory devices **66**, which in this example are assumed to be DIMMs. These memory locations may be anywhere in the internal memory **52** of the section controller **54**.

The memory device control circuitry **55** may read the data stored in these assigned memory locations, for example, by reading information being transmitted between the microprocessor **51** and the memory **52** and looking at information addressed to these memory locations. These memory locations may store the starting address for data transfer, the number of binary digits to transfer, along with other control information for the DIMM **66**. This control information may include, for example, information regarding whether the operation is a read or a write whether the DIMM **66** should start or terminate operations, etc. Although, the above describes one technique for writing data to a memory device in a memory section, one of skill in the art will recognize that numerous other methods are possible without departing from the scope of the invention.

Exemplary Reading Operation

FIG. **13** illustrates a flow chart for an exemplary reading operation for the memory section of FIG. **5** with reference to FIGS. **1**, **6**, and **11**, in accordance with methods and systems provided. This flow chart illustrates but one example of a read operation and other methods may be used for implementing a read operation without departing from the scope of the invention. In this example, the memory devices **66** of the memory section **30** are partitioned into data blocks, where each data block is identifiable by a data block identifier. A more thorough description of data blocks and partitioning of memory devices is presented below. A read request may originate from a user connected to a server **12** via a network. A device driver **28** in the server then sends the request, including a data block identifier (DBI), to a switch **22** in the storage hub **10** (Step **1302**). The device driver **28** may, for example, determine the data block identifier using standard methods. For example, the server **12** may be executing Oracle or a similar type application, which may be used to determine the data block identifier.

The switch **22** then may use the data block identifier (DBI) to direct the data request to the memory section **30** that stores the data block, for example, by determining based on the DBI, an address for the memory section that the switch uses to route the data request to the memory section **30** (Step **1304**). The switch controller **202** may then establish a path through the switch fabric **206** from the SSCI **204** to the SMCI **208** corresponding to the memory section where the data will be stored (Step **1306**). In the event there is no idle communications channel between the switch **22** and the memory section **30** or there is a conflict, the switch may send a congestion message to the requesting server **12**, which may then queue the message until the conflict is resolved. In an embodiment of a switch **22**, such as that illustrated in FIG. **7**, the management complex **26** may perform the functions described above as being performed by the switch controller **202**. Next, the switch **22** then forwards the data request to the CCI **46** of the memory section **30** (Step **1308**). The CCI **46** then may direct the request to the section controller **54** (Step **1310**). The section controller **54** identifies the data request as a read request, and determines if one of its memory devices stores the identified data block (Step **1312**). For example, the section controller **54** may identify the request as a read request by information contained in the request itself, by the format or size of the data request, or by some other method.

If the requested data block is not stored in the memory section **30**, the section controller **54** sends a negative acknowledgement (NAK) message (Step **1314**), through the CCI **46** and switch **22** to the requesting server **12**. After receiving the NAK (Step **1316**), the requesting server **12** may

attempt to re-read (Step **1318**) the data block from the memory section, may attempt to read the data block from another device (not shown), or may inform the application. If the section controller **54** verifies that the memory section stores the requested data block, the microprocessor **51** in the section controller **54** determines which memory device **66** stores the data and checks its state to determine if the memory device **66** is busy or available (Step **1320**). For example, as discussed above, the microprocessor **52** may store in its internal memory **52** a status code for each memory device **66** in the memory section **30** that the microprocessor **51** may consult to determine the availability of the memory device **66**.

If the memory device is available, the microprocessor **51** reserves the memory device by, for example, changing its state in the section controller's internal memory **52** from available to busy. (Step **1326**). The section controller **54** then may send an acknowledgement (ACK) (Step **1328**), to the requesting server **12**. If the device driver **28** still wants to read the read the data block (Step **1330**), it transmits a read confirmation through the switch **22** to the memory section **30** (Step **1332**). Otherwise, it abandons the read request (Step **1334**).

Upon receipt of the read confirmation, the section controller **54** provides an identifier to the read selector **70** of the first memory interface device **64-1** in the read chain corresponding to the memory device(s) **66** from which the data will be read (Step **1332**). As discussed above, the identifier may be, for example, an identifier that the switches **22** use to switch the data to its appropriate destination. The identifier may be, for example, an identifier for the data being transmitted (e.g., a data block identifier) or a destination address identifying an address to which to send the data. For example, if the destination is a computer connected to the Internet, the destination address could be an IP address for the computer. Alternatively, the destination address could simply be an internal address for the switches **22** to use in routing the data to its destination server, in which case the server **12** will read and replace with a destination address that the network over which the data will travel uses to route the data to the appropriate destination.

The identifier (e.g., destination address, a data block identifier, etc) is then clocked through the chain shift register arrays **74** of the memory interface devices **64** until it reaches the memory interface device **64** corresponding to the memory device **66**. That is, the memory device **66** that contains the data block corresponding to the data block identifier in the data request (Step **1336**). Next, the section controller **54** may then determine, using the data block identifier, the addresses for the memory device **66** corresponding to the storage locations where the data block to be read is stored.

The section controller **54** then provides these addresses to the memory device **66** along with other appropriate control signals. In response, the memory device reads the data and it is loaded into the read shift register chain **74** of the memory interface device **64** such that the identifier (e.g., destination address, data block identifier, etc.) is appended to the front of the data (Step **1338**). The addresses and control signals may be provided to the memory device **66** using a technique such as the above discussed memory latching technique.

If the memory device **66** at step **1320** was busy, the read request may be queued in the internal memory **52** of the microprocessor **51** (Step **1322**). When the memory device becomes available, it may send an interrupt signal to the section controller **54** (Step **1324**), which then executes the read request as described beginning at Step **1326**.

When the data is clocked out of the shift register chain and sent to the selector **44** (Step **1340**), the selector **44**, under the control of the section controller **54**, connects the data to the appropriate channel of the communications channel interface **46**. The data is then passed to the server **12** through the communications interface **208**, the switch fabric **206**, and the communications interface **204** (Step **1342**).

Additionally, data may be, for example, simultaneously read from all memory devices in a chain and simultaneously loaded into the read chain of shift register arrays. In such a situation, the identifier (e.g., destination address, data block identifier, etc.) may, for example, only be inserted at the front of the chain of shift register buses.

Test Operation

A test operation for the embodiment of FIG. **5** will now be described. In certain instances, it may be desirable to test the system using known data. When testing the system, test data and a control signal are sent from the section controller **54** to the T-selector **62** such that T-selector sends the test data to the memory interface devices **64** and the test data may be passed through the system. The test data after being written to and read from the memory interface devices **64** may then be sent to the selector **44**, which may be, for example, instructed by the test circuitry **59** to direct the test data to the test circuitry **59**. The test circuitry **59** may then check the data using error detection and correction capabilities, such as, for example, parity checks and/or bit level comparisons. If the data are correct, no action is required. If the data are not correct, the test data may be resent. If the data are then correct, no action is required. If not, the section controller **54** may notify the management complex **26** that a fault has occurred and begin to isolate the fault through the error recovery capabilities present in the software it executes. In parallel, the management complex **26** may then execute fault management procedures, such as those discussed above in the section on the management complex **26**.

Additionally, the CPU **51** may provide through the Header/Test interface **63** test data to the memory interface devices **64**. For example, this data may be sent to one or more of the shift register arrays of **64** in the same manner as destination addresses/data block identifiers and other data. Transmission of the data can be processed through the storage hub **10** in the same manner as any formal data is processed and routed through the storage hub **10**. Such test data allows testing of the operations of all shift register arrays, all controls and interfaces, proper identification of server destinations, throughput time, checks on synchronism among the parallel data paths, etc. If desired, such a pathway may also be employed for any desired handshake routines prior to actual data delivery.

Parallelism and Scalability of Storage Hub

The storage hub **10** may exhibit hierarchical parallelism. The phrase "hierarchical parallelism" as used herein refers to parallelism in the memory section, parallelism between the memory section and the switch fabric, and parallelism among all the memory sections through the switch fabric's connections to servers.

In this example, so long as the requested data are resident in different memory devices **66**, the memory section **30** itself may support N simultaneous reads and one write, where N is the number of communications channel connections available to the memory section **30** and preferably does not exceed the number of memory devices **66**. For example, as illustrated in FIG. **5**, the communications channel interface **46** has **4** communications channels connections for transmitting and receiving information. The switch **22** preferably can handle

simultaneous read requests and write requests that it can fulfill. The section controller **54** of the memory section **30** preferably manages the reading and writing of data to the memory devices **66** and manages any conflicts. The section controller **54** of the memory section **30** manages conflicts through the capabilities present in the software it executes. For example, the section controller **54** may direct that write requests have a priority higher than read requests with the lower priority requests being queued. For example, as previously discussed, the data for write requests may be queued in the temporary storage device **58**, and that write and read requests may be queued in the internal memory **52** of the section controller **54**. The management complex **26** may direct the section controller **54** to resolve conflicts using other methods such as, for example, a first-to-arrive/a-first-to-be-processed algorithm.

In this example, parallelism of the memory sections **30** may be further augmented by parallelism among memory sections **30**. That is, at any point in time and so long as the number of memory sections **30** is at least as great as the number of server connections, S, from the storage hub **10**, then as many as S memory sections may be accessed. In the event the storage hub **10** has more server connections than memory sections, then, in this example, the number of simultaneous transactions equals the number of memory sections.

In addition, the storage hub **10**, in this example, may also be scalable. More particularly, if increased capacity is demanded from the storage hub **10**, this increased capacity may be handled by, for example, adding additional memory sections **30** and cabinets to house them, including backup power, higher capacity and/or additional switches **22**, and/or increasing the number and/or capacity of the connections to the storage hub **10**. Additionally, as the capacity of the storage hub **10** is increased, the capacity and/or number of management complex processors (**32** and **34**) may be increased, if necessary, to ensure that the management complex **26** has sufficient capacity to monitor the respective states of the storage hub's memory sections **30**

Additionally, in this example, the memory sections **30** may also be scalable. For example, increased capacity of a memory section **30** may be obtained by, for example, adding additional memory devices **66**, memory interface devices **64**, and/or communications channel interfaces to the memory section. Preferably, the section controller **54** for each memory section includes a sufficiently large resident memory for holding a "map" of the location of each data block resident in its memory section **30**. Commercially available microprocessors may be used by the section controller for storing this map.

A scalable architecture may initially be deployed wherein the storage hub includes only one or a few memory sections each including only small number of memory devices. Then, as increased capacity is demanded of the storage hub, this increased capacity may be managed, for example, by adding additional and/or higher capacity memory sections to the storage hub additional cabinets and backup power to house the additional memory sections, increasing the capacity of the existing one or more memory sections, increasing the capacity of the management complex **26** through the addition of control processors **34** or administration processors **34**, increasing the number and/or capacity of the switches **22** through the addition of more ports and/or switch controllers as needed. Thus, as the storage hub's capacity is increased, the performance seen by any given user preferably remains uniform as the system grows, and expected response times for users with applications generating random data requests preferably remains uniform as the system expands.

Partitioning

FIG. **14** illustrates a logical diagram of N memory devices **66**, in accordance with methods and systems provided. As discussed above, the memory devices **66** may be solid state memory devices that can store B bits, and the number of bits that each memory device can store may be different. In one example, the management complex **26** may send a command to the section controller **54** of a memory section **30**, instructing the section controller **54** to reconfigure a memory device **66** in the memory section **30** into a number of partitions including one or more data blocks each having a particular block size, where the block size is the number of bits in the block. Use of differently sized data blocks and data block partitions allows the storage hub to suit its storage structure to different applications whose data are stored on the same storage hub. For example, an application like an on-line catalog that always stored text and an image or images should preferably prefer a larger block size than an application like a customer reservation system that stored only text. Since the partitions and block sizes can be reconfigured at will through commands from the management complex, new data for new applications may be loaded into the storage hub without have to stop its operations or affect data in other memory sections that are not being changed.

As illustrated in FIG. **14**, memory device **66-1** has two partitions, where the first partition includes eight blocks and the second includes two blocks. In this example, the block size of the data block in the first partition is smaller than the data blocks in the second partition. Memory device **66-2**, in contrast, has four identical partitions of four blocks each, where each data block is of equal size Additionally, in this example, the management complex **26** may dynamically adjust the data block size in each partition, modify the partitions, or add or delete partitions. For example, at any time the management complex **26** may instruct the section controller **54** of a memory section **30** to reconfigure the memory devices **66**, wherein the current partition may have no effect on any subsequent partition.

Alternative Memory Interface Device

FIG. **15** illustrates an embodiment wherein the memory interface devices use a common shift register array for reading from and writing to the memory device, in accordance with methods and systems provided. As illustrated, each memory interface device **64** includes a shift register array **78** of a plurality of shift registers **76** interconnected in series. Further, like the embodiment of FIG. **11**, these memory interface devices are connected in pairs to form a chain.

As illustrated, the shift register array **78** of memory interface device **64-1** is connected to the shift register array **78** of memory interface device **64-2**, which is in turn connected to the selector **44**. Further, because there is not a separate write chain, the I/O pins in the upper right and left of the memory interface devices **64** shown in FIG. **5** are not necessary. Additionally, the memory interface devices of this embodiment use a write selector **82**. The operation of the write selector **82** will be described in more detail below.

The following provides a description of exemplary write operation for the embodiment of FIG. **15**. Data to be written to the memory devices **66**, as in the above embodiment of FIG. **11** is forwarded through the Temporary storage interface device **60** to the T-selector **62** under the control of the section controller **54**. In this embodiment, because there are separate chains for writing the data, the data is forwarded to the chain corresponding to the memory device **66** where the data is to

be stored. The T-selector **62**, therefore, passes the data to the write selector **82**, which receives a control signal from the section controller **54** such that the write selector **82** sends the data to the appropriate chain of shift register arrays.

The signal is then passed to the read selector **70** of the first memory interface device **64-1** (in this example) in the chain. Because, this is a write operation, the data is clocked into the shift register array **78** and is clocked through the shift registers until it is loaded into the shift register array **78** corresponding to the memory device **66** to which the data is to be written. The data is then written to the memory device **66**. Methods and systems for writing data from a shift register array **78** to a memory device **66** are presented in more detail below.

An exemplary reading operation for the embodiment of FIG. **15** will now be described. First, an identifier (e.g., destination address, data block identifier, etc.) for the data is supplied to the read selector **84** from the section controller **54**. The identifier is then clocked through the chain of shift register arrays.

Next, the data is loaded from the memory device **66** into the shift register array **78** such that the identifier (e.g., destination address, data block identifier, etc.) is appended to the front of the shift register array. The data is then clocked through the chain of shift registers **76** until it is passed to the selector **44**. The data is then passed to the end user through the server **12** as was described with reference to FIGS. **5** and **11**.

Additionally, as in FIG. **11**, test data may be inserted into the memory devices **66** and passed through the system using the T-selector **62**.

As will be obvious to one of skill in the art, other embodiments of the memory interface device are possible, without departing from the scope of the invention. For example, although each memory interface device is described as only having one read or write shift register array, the memory interface device may include any number of write or read chains of shift register arrays. Further, the shift registers **76** rather than being 1 bit shift registers may be of any depth. For example, the shift register arrays could be, for example, N×M arrays such as, for example, 2×8, 4×32, 8×16, etc. arrays as determined by the system designers for their particular implementation. Additionally, the shift registers arrays may be configured in a ring, such that the data once loaded into a chain circulates synchronously in the chain. A more detailed description of memory access using shift register arrays is set forth in the aforementioned U.S. patent application Ser. No. 10/284,198 by William T. Lynch and David J. Herbison entitled "Methods and Apparatus for Improved Memory Access" (issued as U.S. Pat. No. 6,879,526) filed on the same day as the present application.

Alternative Memory Section

FIG. **16** illustrates an alternative memory section **30**, in accordance with methods and systems provided. These memory sections are provided as exemplary mechanisms that could be employed to practice the invention, and are not intended to limit the scope of the claims.

As illustrated, the memory section **30** includes one or more communications channel input/output interfaces **46-1** and **46**-m, a section controller **54**, one or more bus interfaces **1602**, a bus **1604**, a bus controller **1606**, an analyzer **1608**, and one or more memory devices **66-1** and **66**-n. In this example, the bus **1604** uses the Peripheral Component Interconnect (PCI) standard. Each device is preferably connected to the bus via a bus interface **1602**, which in this example is a PCI interface. The bus controller **1606** in this example is a PCI bus controller for arbitrating which device may control the bus. Additionally, in this example, an analyzer circuit **1608** monitors the state of the PCI bus **1604** and informs the section controller **54** of the real time state of the bus **1604**.

The communications channel input/output interfaces **46-1** and **46**-m may be any type of communications channel interfaces, such as Fibre Channel, Telecommunications Control Protocol/Internetworking Protocol (TCP/IP) over Ethernet, Token Ring, etc. In this example, the communications channel input/output interfaces **46-1** and **46**-m use the Fibre Channel protocol. A commercially available Fibre Channel I/O component embodied on a chip may be used as the communications channel I/O interfaces **46**. These commercially available chips typically include a microprocessor for executing layer 2 and 4 protocol engines, a 2.125 Gbit transceiver, a data codec, a transmit FIFO queue, and a receiver FIFO queue. The Fibre Channel I/O components' microprocessor is capable of receiving commands from the section controller **54** regarding control of the PCI interfaces **1602** and transferring or receiving information from the memory devices **66**. In this example, m number communication channel I/O interfaces are illustrated, where m>=1.

The communications channel I/O interfaces **46** preferably also are capable of receiving control instructions from the section controller **54**. For example, the section controller **54** may send and receive control information to and from the communications channel interfaces **46** using the direct memory addressing (DMA) protocol. Additionally, although not shown, the section controller **54** may also send and receive interrupts and I/O commands to and from the communication channel interfaces **46** over one or more I/O control lines.

In this example, the section controller **54** is embodied on one or more chips including an internal random access memory, a read only memory for bootstrap loading, an interface to the Management Complex, and a microprocessor. The section controller **54** may directly control the memory devices **66** through memory latching, as in the above-described embodiment. Additionally, the section controller **54** may receive real time status information about the bus interfaces **1602** from the analyzer circuit **1608**. The section controller **54**, as illustrated, also includes a DMA interface for sending and receiving control information to/from the communications channel interfaces using the DMA protocol. Although, in this example, the DMA protocol is used, as discussed above any other suitable protocol may be used for sending control information between the communication channel interfaces **46** and the section controller **54**.

The read only memory (ROM) **56** may store the software associated with the bootstrap program used by the section controller to obtain its latest software image. This ROM **56** although illustrated as separate from the section controller **54** may be included in the section controller **54**.

The bus interfaces **1602** are used to connect the memory devices **66** and communications channel interfaces **46** to the bus **1604**, such that these devices may transmit and receive information over the bus **1604**. The bus interfaces **1602-3** and **1602-4** in this example connect to the section controller **54** via control lines over which the section controller **54** may send to the bus interfaces **1602-3** and **1602-4** control information. This control information may include, for example, an identifier (e.g., destination address, data block identifier, etc.) for data being read from the memory device **10**. Accordingly, the bus interfaces **1602-3** and **1602-4** may also be referred to as memory interface devices **1602-3** and **1602-4**.

The memory devices **66**, as in the above-described embodiments may be any type of memory devices, such as, for example DIMMs. The section controller **54** preferably controls reading from and writing to the memory devices **66**. For example, control lines between the memory devices **66** and the section controller **54** may carry signals from the section controller **54** to the memory devices **66** regarding the address to transfer data to/from, and the number of bits to transfer data to/from the memory device **66**. Additionally, the memory devices **66** may be capable of providing the section controller **54** via these control line with a real time interrupt signal when an operation is complete along with information regarding the operational status of the memory device **66**. In this example, there are n memory devices **66**, where n>=1

The following provides a brief overview of an example for a reading operation for the memory section of FIG. **16**. In this example, the communications channel interfaces **46** are preferably fibre channel I/O components. When a data request arrives at the communications channel interface **46**, the communication channel interface **46** detects it and sends an interrupt signal to the section controller **54**. This interrupt signal preferably includes information regarding the data block to be read from the memory devices. The section controller **54** then maps this data block information to an address in the memory devices. That is, the section controller determines from this data block information the memory devices **66** storing the requested data along with the addresses for this data on those memory devices. The section controller **54** then loads this address information into its internal memory, such that the addresses are transferred to the memory devices as in the above-describe memory latching example.

The requested data block is then read from the memory devices **66** and transferred to their corresponding bus interfaces **1602**. Additionally, the section controller **54** transmits an identifier (e.g., destination address, data block identifier, etc.) to the bus interface **1602**. The identifier may be, for example, an identifier for the data being transmitted (e.g., a data block identifier) or a destination address identifying an address to which to send the data. The destination address may be, for example, the internal address for the bus interface **1602** corresponding to the communication channel interface **46** to which the data is to be sent. Alternatively, the destination address may be, for example, an address of the server **12** to which the data is destined, an address for use by the switch **22** in switching the information, or any other identifier useful in routing the data towards its destination.

The bus interfaces **1602** then transfer the identifier (e.g., destination address, data block identifier, etc.) and the data over the bus **1604** according to the bus's protocol. In this example, the bus **1604** uses the PCI protocol and the PCI controller **1606** arbitrates any conflicts in accessing the bus **1604**.

The bus interface **1602** for the communication channel interface **46** to which the data is sent monitors the bus for data including its address. Because all information is transmitted over the bus, bus interfaces **1602** ignore data not including an address for the bus interface **1602**. When the bus interface **1602** for the communication channel interface **46** to which the data is to be transmitted recognizes an address for itself, it reads the data from the bus **1604**. The communication channel interface **46** then transmits the data block from the memory section **30**. In certain examples, the communications channel interface **46** may also replace the destination address with a new destination address prior to sending the data from the memory section **30**.

A write operation for the exemplary memory section **30** of FIG. **14** will now be described. When a write request is received by a communications channel interface **46**, the communications channel interface **46** determines that it is a write request and forwards the request to the section controller **54** via the DMA interface **1610**. The write request in this example may include a data block identifier indicative of where the memory section **30** is to write the data. The section controller **54** then maps this data block identifier to determine the memory devices **66** to which the memory section **30** is to write the data along with the addresses for the memory devices **66**. The section controller **54** provides these addresses to the appropriate memory devices **66**.

In addition to sending the write data request to the section controller **54**, the communications channel interface **46** obtains the data from the write request and sends the data to the communication channel interface's corresponding bus interface **1602**. Additionally, the section controller **54** may supply the communications channel interface **46** with a bus interface address for the bus interface **1602** corresponding to the memory device **66** that will store the write data. The bus interface **1602** then transmits the data and the bus interface onto the bus **1604**.

The bus interfaces **1602-3** and **1602-4** for the memory devices monitor the bus **1604** looking for data destined for them. When the bus interface **1602** senses its address is being transmitted, it reads the write data from the bus. Meanwhile, the section controller **54** informs the corresponding memory device **66** to accept write data from its bus interface **1602-3** or **1602-4**. The bus interfaces **1602-3** or **1602-4** then provide the data to their corresponding memory device **66** which write the data at the address provided by the section controller **54**.

The above description of the preferred embodiments has been given by way of example. From the disclosure given, those skilled in the art shall understand the invention and its advantages, but will also find apparent various changes and modifications that can be made to the methods and structures disclosed. We seek therefore to cover all such changes and modifications as fall within the spirit and scope of the invention, as defined by the appended claims and equivalents thereof. Thus, it is intended that the specification and examples be considered as exemplary only, with a true scope and spirit of the invention being indicated by the following claims.

What is claimed is:

1. A storage system comprising:

one or more memory sections, including

one or more memory devices including storage locations that store data, and

a memory section controller that provides addresses to the memory devices, the addresses identifying storage locations for a memory device,

wherein the memory devices use the provided addresses to perform a function selected from the set of reading out and writing data to/from the memory devices; and

one or more switches, comprising a configurable switch fabric, that receive a data request including a data block identifier and switch the data request to one or more of the memory sections determined by applying the data block identifier to an algorithm that selectively configures operation of the switch fabric, the data block identifier identifying a set of storage locations;

wherein the memory sections to which the data request was switched forward the received data block identifier to its memory section controller which maps the data block identifier to a set of addresses for the storage locations identified by the data block identifier, and provides the set of addresses to one or more of the memory section's memory devices.

2. The storage system of claim 1, further comprising

a management system that provides the algorithm to at least one of the one or more switches, wherein the switch executes the algorithm in switching the data request based on the data block identifier.

3. The storage system of claim 2, further comprising:

a management system that performs fault management in response to a fault being detected.

4. The storage system of claim 3, wherein the management system detects a fault with regard to a memory section in response to not receiving a message from the memory section.

5. The storage system of claim 3, wherein one or more of the memory sections include a fault detection component that detects a fault in the memory section.

6. The storage system of claim 3, wherein the management system includes:

one or more control processors for performing fault management; and

one or more administration processors that collect statistical data from the one or more switches and the one or more memory sections.

7. The storage system of claim 1, further comprising:

a management system that instructs the storage system to store a backup of data stored by one or more of the memory sections into a non-volatile storage device connected to at least one of the switches, receives a fault message from a memory section, and instructs the storage system to load the back-up of the memory section's data from the non-volatile storage device into a functioning memory section.

8. The storage system of claim 1, further comprising

an interface that connects to an external management system such that configuration management is performed through the external management system.

9. The storage system of claim 1, further comprising:

one or more memory interface devices that receives from at least one of the memory devices data stored in the storage locations identified by the addresses, combines the data with an identifier for use in forwarding the requested data, and forwards the data to one or more of the switches;

wherein the switches to which the data was forwarded switch the data using the identifier and forward the data to a destination device.

10. The storage system of claim 9, wherein the identifier for use in forwarding the requested data is an address for a device to which the data are to be forwarded.

11. The storage system of claim 9, wherein the identifier for use in forwarding the requested data is an identifier for identifying the data that are to be forwarded.

12. The storage system of claim 9, wherein the memory devices include outputs, and wherein at least one of the memory interface devices includes:

at least one set of shift registers interconnected in series, wherein at least one of the shift registers receives a clock signal having a shift frequency, and wherein the shift register shifts data loaded into the shift register to a next one of the shift registers in the set according to the shift frequency; and

wherein data from one or more of the output of a memory device is loaded into a corresponding shift register in the sets of shift registers and the loaded data is shifted from the shift register to a next one of the shift registers in the set according to the shift frequency, such that the shift register maintains its shift frequency during any loading of the data.

13. The storage system of claim 9, wherein the memory devices include outputs, and wherein at least one of the memory interface devices includes:

at least one set of shift registers interconnected in series, wherein at least one of the shift registers receives a clock signal having a shift frequency, and wherein the shift register shifts data loaded into the shift register to a next one of the shift registers in the set according to the shift frequency; and

wherein data in the set of shifted register is loaded according to the clock signal from one or more shift registers in the set into one or more of the memory devices via an input corresponding to the shift register, such that the shift register maintains its shift frequency during any loading of the data.

14. The storage system of claim 1, wherein at least one memory section further includes

a temporary storage device that stores data to be written to a memory device; and

a temporary storage interface device that stores data in and retrieves data from the temporary storage device;

wherein at least one of the switches, in response to receiving data to be stored in the memory section, forwards the data to the temporary storage interface device, and wherein the temporary storage interface stores the data in the temporary storage device if a memory device to which the data is to be written is busy, and wherein the temporary storage interface device retrieves the data from the temporary storage device when the memory device is no longer busy and forwards the data to the memory device such that the memory device stores the data.

15. The storage system of claim 1, wherein at least one memory section further includes at least one communications channel interface for receiving data to be stored by the memory section and transferring the data to be stored to one or more of the memory devices, and wherein the communications channel interface receives data read from the memory devices and transfers the data read from the memory devices to one or more of the switches, and wherein the communications channel interface receives data requests and forwards the data requests to the memory section controller.

16. A method for use in a storage system, comprising:

storing data in storage locations in a memory device;

receiving by a switch comprising a configurable switch fabric, a data request including a data block identifier;

the switch switching the data request to a memory section including the memory device determined by applying the data block identifier to an algorithm that selectively configures operation of the switch, the data block identifier identifying a set of storage locations in the memory device;

forwarding the received data block identifier to a memory section controller;

the memory section controller mapping the data block identifier to a set of addresses for the storage locations identified by the data block identifier; and

the memory section controller providing the set of addresses to the memory device; and

the memory device using the provided addresses to perform a function selected from the set of reading and writing data to/from the memory device.

17. The method of claim 16, further comprising

a management system providing the algorithm to the switch; and

the switch executing the algorithm in switching the data request based on the data block identifier.

18. The method of claim 17, further comprising:

a management system performing fault management in response to a fault being detected.

19. The method of claim 18, further comprising:

the management system detecting a fault in response to not receiving a message from the memory section controller.

20. The method of claim 18, further comprising:

the memory section controller detecting a fault.

21. The method of claim 16, further comprising:

storing a back-up of data stored by the memory device into a non-volatile storage device connected to the switch;

a management system receiving a fault message from the memory section controller; and

the management system instructing the non-volatile storage device to send the back-up of the memory device's data to a functioning memory device.

22. The method of claim 16, further comprising:

a memory interface device receiving from the memory device data stored in the storage locations identified by the addresses;

the memory interface device combining the data with an identifier for use in forwarding the requested data;

the memory interface device forwarding the data to the switch;

the switch switching the data using the identifier; and

the switch forwarding the data to a destination device.

23. The method of claim 16, further comprising:

a memory interface device in the memory section receiving, in response to the memory section receiving a data request, data from a memory device;

shifting data in one or more shift registers in a set of shift registers interconnected in series from the shift register to a next one of the shift registers in the set on the basis of a clock signal having a shift frequency, wherein the shift registers are included in the memory interface device;

loading data from the memory device into a corresponding shift register in the set; and

shifting the data loaded into one or more of the shift registers to a next one of the shift registers in the set according to the clock signal, wherein the shift register maintains its shift frequency during the loading of the data from the memory devices into the shift registers.

24. The method of claim 16, further comprising:

a memory interface device in the memory section receiving, in response to the memory section receiving a data request, data from a memory device;

shifting data in one or more shift registers in a set of shift registers interconnected in series from the shift registers to a next one of the shift registers in the set on the basis of a clock signal having a shift frequency, the shift registers included in the memory interface device;

loading data from one or more of the shift registers to a memory device; and

shifting the data loaded from the one or more shift registers to a next one of the shift registers in the set according to the shift frequency after the data is loaded into the memory device, wherein the shift register maintains its shift frequency during the loading of the data.

25. The method of claim 16, further comprising:

the memory section receiving data to be stored in the memory device;

forwarding the data to a temporary storage interface device;

the temporary storage interface device storing the data in a temporary storage device if the memory device to which the data is to be written is busy;

the temporary storage interface device retrieving the data from the temporary storage device when the memory device to which the data is to be stored is no longer busy; and

the memory device storing the data.

26. A storage system, comprising:

means for storing, including:

means for storing data in storage locations, the means for storing data in storage locations including means for reading data stored in the storage locations using an address;

means for controlling the means for storing, the means for controlling including:

means for mapping a data block identifier to a set of addresses,

means for providing the addresses to the means for storing data in storage locations, the addresses identifying storage locations;

means for switching, including

means for receiving a data request including a data block identifier;

means for switching the data request based on the data block identifier to

a means for storing determined by applying the data block identifier to an algorithm that selectively configures operation of the means for switching, the data block identifier identifying a set of storage locations in the means for storing data in storage locations; and

means for forwarding the received data block identifier to the means for storing.

27. A storage hub comprising

a memory section, including

a memory device including storage locations that store data, and

a memory section controller that provides an address to the memory device, the address identifying a storage location,

wherein the memory device uses the provided address to write data into the memory device; and

a switch, comprising a configurable switch fabric, that receives a data request including a data block identifier and transmits the data request to the memory section determined by applying the data block identifier to an algorithm that selectively configures operation of the switch fabric, and that receives write data associated with the data request and transmits the write data to the determined memory section;

wherein the memory section forwards the received data block identifier to the memory section controller, which determines from the data block identifier the address of a storage location and provides the address to the memory device, and the memory device stores the write data at the address.

* * * * *