4c. If it is an ADDRESS_ACK_REJECT cell which is received, the state of the VCI **6** entry in VCI table **800** is returned to IDLE, and the VCI table entry for VCI **6** is freed.

4e. PHY **0** subsequently propagates the translated cell on its upstream link.

At this point (assuming the connection request was accepted) the VCI Tables on both the upstream and downstream PHYs are filled out for the newly established connection and cells associated with the connection may flow.

Note that the downstream PHY manages any Reject—VCI conflicts which occur when the attached expander happens to choose the same VCI in the case of crossing ADDRESS_REQUEST cells. Also, it should be apparent from this description that ADDRESS_REQUEST cells and ADDRESS_ACK cells take priority over established connection cells to ensure that all VCI tables along the path are fully and expeditiously filled out prior to needing to handle traffic.

Exemplary Steps for Routing Cells in a Virtual Connection:

1. PHY **0** receives a cell from its link.

1a. The ingress VCI **6** is used to look up the egress PHY and egress VCI as well as the associated state and priority from VCI Table **800** entry **6**. It is confirmed that VCI **6** is indeed in a CONNECTED state, otherwise error recovery may be necessary.

1b. The VCI in the cell is translated from the ingress VCI of **6** to the egress VCI of **9**.

1c. The crossbar is configured to link PHY **0** to PHY **3**.

1d. The translated cell is sent to the egress PHY **3** as indicated by VCI Table **800** entry.

2. PHY **3** receives the translated cell from PHY **0**.

2a. PHY **3** recognizes the received cell is of a type associated with an established connection (i.e. not an Address Cell type used for opening or closing a connection) and no further action may be necessary.

2b. PHY **3** sends the cell to its link.

3. Similar steps occur when propagating cells associated with an established virtual connection in the opposite direction.

If cells within a connection encounter a VCI table entry in the IDLE state, it should be recognized as an error condition due to either a corrupt VCI or a corrupt VCI table entry, and the cell may be discarded.

Exemplary Steps for Closing a Virtual Connection:

1. PHY **0** receives an ADDRESS_CLOSE primitive cell for VCI **6** from its link.

1a. The egress PHY (PHY **3**), egress VCI (VCI **9**) as well as the associated state are read from VCI Table **800** entry **6**. The egress information is used to send the cell on its way while the state is used for error checking. For example, a close for a connection that is not actually connected is an error.

1b. PHY **0** sets the state for VCI **6** in its VCI table to PENDING CLOSE.

1c. The VCI in the cell is translated from the ingress VCI of **6** to the egress VCI of **9**.

1d. The crossbar is configured to link PHY **0** to PHY **3**.

1e. The cell is sent to the egress PHY, PHY **3**, as indicated by the associated VCI Table **800** entry.

2. PHY **3** receives the ADDRESS_CLOSE primitive cell from PHY **0**.

2a. PHY **3** sets the state for location **9** in the VCI Table to PENDING CLOSE

2b. PHY **3** propagates the ADDRESS_CLOSE cell on its downstream link

3. Subsequently an ADDRESS_CLOSE cell for VCI **9** is returned on the link to PHY **3**.

3a. The egress PHY (PHY **0**), Egress VCI (VCI **6**) as well as the associated state are read from VCI Table entry **9**.

3b. PHY **0** sets the state for VCI **6** in its VCI table to IDLE and frees the VCI table entry for VCI **6**.

3c. The VCI in the cell is translated from the Ingress VCI of **9** to the Egress VCI of **6**.

3d. The crossbar is configured to link PHY **3** to PHY **0**.

3e. The cell is sent to the egress PHY **0** as indicated by VCI Table **800** entry.

4. PHY **0** Receives the ADDRESS_CLOSE cell from PHY **3**.

4a. PHY **0** sets the state for location **9** in the VCI Table to IDLE frees the VCI table entry for VCI **9**.

4b. PHY **0** propagates the ADDRESS_CLOSE cell on the upstream link.

At this point the connection at both the upstream and downstream PHYs is completely torn down, and the associated resources may be reused for other connections.

While the invention has been illustrated and described in the drawings and foregoing description, such illustration and description is to be considered as exemplary and not restrictive in character. One embodiment of the invention and minor variants thereof have been shown and described. In particular, features shown and described as exemplary software or firmware embodiments may be equivalently implemented as customized logic circuits and vice versa. Protection is desired for all changes and modifications that come within the spirit of the invention. Those skilled in the art will appreciate variations of the above-described embodiments that fall within the scope of the invention. As a result, the invention is not limited to the specific examples and illustrations discussed above, but only by the following claims and their equivalents.

What is claimed is:

1. A Serial Attached SCSI (SAS) expander comprising:

a first physical Link (PHY);

a second PHY; and

a Virtual Connection Manager (VCM) adapted to exchange information over a plurality of concurrently established full-duplex virtual pathways directly between the first PHY and the second PHY.

2. The SAS expander of claim **1** wherein:

the information comprises one or more virtual SAS cells, wherein each of the one or more virtual SAS cells includes a segmented SAS communication.

3. The SAS expander of claim **2** further comprising:

an Expander Link Layer (XL) coupled with the first PHY, the XL adapted to exchange SAS communications with legacy SAS devices; and

a virtual fabric layer coupled with the XL and the VCM, the virtual fabric layer adapted to receive the SAS communication from the XL, to segment the SAS communication into the one or more virtual SAS cells, and to provide the one or more virtual SAS cells to the VCM for routing to the second PHY.

4. The SAS expander of claim **2** further comprising:

a cell flow control module adapted to manage the flow of the one or more virtual SAS cells based on a number of cell buffers that are available for exchanging the one or more virtual SAS cells with a PHY of the SAS expander and an external device.

5. The SAS expander of claim **4** wherein the cell flow control module further comprises:

a data flow control module adapted to manage a flow of virtual data cells based on a number of data cell buffers