A hub port (200) in a hub which preferably eliminates the transfer of jitter in the signal transmitted to an attached node port. In one embodiment, jitter transfer is preferably eliminated by transmitting data to an attached node port (212) using a local clock signal internal to the hub rather than a clock signal recovered from the datastream. The hub port includes smoothing circuits (220) to synchronize data received from the attached node port to the local clock.
### FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

<table>
<thead>
<tr>
<th>Code</th>
<th>Country</th>
<th>Code</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>Albania</td>
<td>ES</td>
<td>Spain</td>
</tr>
<tr>
<td>AM</td>
<td>Armenia</td>
<td>FI</td>
<td>Finland</td>
</tr>
<tr>
<td>AT</td>
<td>Austria</td>
<td>FR</td>
<td>France</td>
</tr>
<tr>
<td>AU</td>
<td>Australia</td>
<td>GA</td>
<td>Gabon</td>
</tr>
<tr>
<td>AZ</td>
<td>Azerbaijan</td>
<td>GB</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>BA</td>
<td>Bosnia and Herzegovina</td>
<td>GE</td>
<td>Georgia</td>
</tr>
<tr>
<td>BB</td>
<td>Barbados</td>
<td>GH</td>
<td>Ghana</td>
</tr>
<tr>
<td>BE</td>
<td>Belgium</td>
<td>GN</td>
<td>Guinea</td>
</tr>
<tr>
<td>BF</td>
<td>Burkina Faso</td>
<td>GR</td>
<td>Greece</td>
</tr>
<tr>
<td>BG</td>
<td>Bulgaria</td>
<td>HU</td>
<td>Hungary</td>
</tr>
<tr>
<td>BJ</td>
<td>Benin</td>
<td>IE</td>
<td>Ireland</td>
</tr>
<tr>
<td>BR</td>
<td>Brazil</td>
<td>IL</td>
<td>Israel</td>
</tr>
<tr>
<td>BY</td>
<td>Belarus</td>
<td>IS</td>
<td>Iceland</td>
</tr>
<tr>
<td>CA</td>
<td>Canada</td>
<td>IT</td>
<td>Italy</td>
</tr>
<tr>
<td>CF</td>
<td>Central African Republic</td>
<td>JP</td>
<td>Japan</td>
</tr>
<tr>
<td>CG</td>
<td>Congo</td>
<td>KE</td>
<td>Kenya</td>
</tr>
<tr>
<td>CH</td>
<td>Switzerland</td>
<td>KG</td>
<td>Kyrgyzstan</td>
</tr>
<tr>
<td>CI</td>
<td>Côte d’Ivoire</td>
<td>KP</td>
<td>Democratic People's Republic of Korea</td>
</tr>
<tr>
<td>CM</td>
<td>Cameroon</td>
<td>KR</td>
<td>Republic of Korea</td>
</tr>
<tr>
<td>CN</td>
<td>China</td>
<td>KZ</td>
<td>Kazakhstan</td>
</tr>
<tr>
<td>CU</td>
<td>Cuba</td>
<td>LC</td>
<td>Saint Lucia</td>
</tr>
<tr>
<td>CZ</td>
<td>Czech Republic</td>
<td>LI</td>
<td>Liechtenstein</td>
</tr>
<tr>
<td>DE</td>
<td>Germany</td>
<td>LK</td>
<td>Sri Lanka</td>
</tr>
<tr>
<td>DK</td>
<td>Denmark</td>
<td>LR</td>
<td>Liberia</td>
</tr>
<tr>
<td>EE</td>
<td>Estonia</td>
<td>LS</td>
<td>Lesotho</td>
</tr>
<tr>
<td>FI</td>
<td>Finland</td>
<td>LT</td>
<td>Lithuania</td>
</tr>
<tr>
<td>FR</td>
<td>France</td>
<td>LU</td>
<td>Luxembourg</td>
</tr>
<tr>
<td>GA</td>
<td>Gabon</td>
<td>LV</td>
<td>Latvia</td>
</tr>
<tr>
<td>GB</td>
<td>United Kingdom</td>
<td>MC</td>
<td>Monaco</td>
</tr>
<tr>
<td>GE</td>
<td>Georgia</td>
<td>MD</td>
<td>Republic of Moldova</td>
</tr>
<tr>
<td>GH</td>
<td>Ghana</td>
<td>MG</td>
<td>Madagascar</td>
</tr>
<tr>
<td>GN</td>
<td>Guinea</td>
<td>MK</td>
<td>The former Yugoslav Republic of Macedonia</td>
</tr>
<tr>
<td>GR</td>
<td>Greece</td>
<td>ML</td>
<td>Mali</td>
</tr>
<tr>
<td>HU</td>
<td>Hungary</td>
<td>MN</td>
<td>Mongolia</td>
</tr>
<tr>
<td>IE</td>
<td>Ireland</td>
<td>MR</td>
<td>Mauritania</td>
</tr>
<tr>
<td>IL</td>
<td>Israel</td>
<td>MW</td>
<td>Malawi</td>
</tr>
<tr>
<td>IS</td>
<td>Iceland</td>
<td>MX</td>
<td>Mexico</td>
</tr>
<tr>
<td>IT</td>
<td>Italy</td>
<td>NE</td>
<td>Niger</td>
</tr>
<tr>
<td>JP</td>
<td>Japan</td>
<td>NL</td>
<td>Netherlands</td>
</tr>
<tr>
<td>KE</td>
<td>Kenya</td>
<td>NO</td>
<td>Norway</td>
</tr>
<tr>
<td>KG</td>
<td>Kyrgyzstan</td>
<td>NZ</td>
<td>New Zealand</td>
</tr>
<tr>
<td>KP</td>
<td>Democratic People's Republic of Korea</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KR</td>
<td>Republic of Korea</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LC</td>
<td>Saint Lucia</td>
<td>PL</td>
<td>Poland</td>
</tr>
<tr>
<td>LI</td>
<td>Liechtenstein</td>
<td>PT</td>
<td>Portugal</td>
</tr>
<tr>
<td>LK</td>
<td>Sri Lanka</td>
<td>RO</td>
<td>Romania</td>
</tr>
<tr>
<td>LR</td>
<td>Liberia</td>
<td>RU</td>
<td>Russian Federation</td>
</tr>
<tr>
<td>SI</td>
<td>Slovenia</td>
<td>SK</td>
<td>Slovakia</td>
</tr>
<tr>
<td>SN</td>
<td>Senegal</td>
<td>SZ</td>
<td>Swaziland</td>
</tr>
<tr>
<td>TD</td>
<td>Chad</td>
<td>TG</td>
<td>Togo</td>
</tr>
<tr>
<td>TJ</td>
<td>Tajikistan</td>
<td>TM</td>
<td>Turkmenistan</td>
</tr>
<tr>
<td>TR</td>
<td>Turkey</td>
<td>TT</td>
<td>Trinidad and Tobago</td>
</tr>
<tr>
<td>UA</td>
<td>Ukraine</td>
<td>VG</td>
<td>Uganda</td>
</tr>
<tr>
<td>US</td>
<td>United States of America</td>
<td>UZ</td>
<td>Uzbekistan</td>
</tr>
<tr>
<td>VN</td>
<td>Viet Nam</td>
<td>YU</td>
<td>Yugoslavia</td>
</tr>
<tr>
<td>ZW</td>
<td>Zimbabwe</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HUB PORT WITHOUT JITTER TRANSFER

TECHNICAL FIELD

The present invention relates to an electronic network switching device, and more specifically to a network hub port which synchronizes data received from an attached node port to an internal clock in order to eliminate jitter transfer.

BACKGROUND INFORMATION

Electronic data systems are frequently interconnected using network communication systems. Area-wide networks and channels are two approaches that have been developed for computer network architectures. Traditional networks (e.g., LAN's and WAN's) offer a great deal of flexibility and relatively large distance capabilities. Channels, such as the Enterprise System Connection (ESCON) and the Small Computer System Interface (SCSI), have been developed for high performance and reliability. Channels typically use dedicated short-distance connections between computers or between computers and peripherals.

Features of both channels and networks have been incorporated into a new network standard known as “Fibre Channel”. Fibre Channel systems combine the speed and reliability of channels with the flexibility and connectivity of networks. Fibre Channel products currently can run at very high data rates, such as 266 Mbps or 1062 Mbps. These speeds are sufficient to handle quite demanding applications, such as uncompressed, full motion, high-quality video. ANSI specifications, such as X3.230-1994, define the Fibre Channel network. This specification distributes Fibre Channel functions among five layers. The five functional layers of the Fibre Channel are: FC-0 - the physical media layer; FC-1 - the coding and encoding layer; FC-2 - the actual transport mechanism, including the framing protocol and flow control between nodes; FC-3 - the common services layer; and FC-4 - the upper layer protocol.

There are generally three ways to deploy a Fibre Channel network: simple point-to-point connections; arbitrated loops; and switched fabrics. The simplest
topology is the point-to-point configuration, which simply connects any two Fibre
Channel systems directly. Arbitrated loops are Fibre Channel ring connections that
provide shared access to bandwidth via arbitration. Switched Fibre Channel networks,
called "fabrics", are a form of cross-point switching.

Conventional Fibre Channel Arbitrated Loop ("FC-AL") protocols provide
for loop functionality in the interconnection of devices or loop segments through node
ports. However, direct interconnection of node ports is problematic in that a failure at
one node port in a loop typically causes the failure of the entire loop. This difficulty
is overcome in conventional Fibre Channel technology through the use of hubs. Hubs
include a number of hub ports interconnected in a loop topology. Node ports are
connected to hub ports, forming a star topology with the hub at the center. Hub ports
which are not connected to node ports or which are connected to failed node ports are
bypassed. In this way, the loop is maintained despite removal or failure of node ports.

For example, FIG. 1A illustrates a hub 100 with six hub ports 102, 104, 106, 108, 110, 112 (the number of hub ports and node ports illustrated in FIGS. 1A
and 1B is for illustrative purposes only and does not limit the application of the
technology provided by the preferred embodiment). Hub ports 102 - 112 are inter-
connected by internal hub links 114, 116, 118, 120, 122, 124. Four node ports 126,
128, 130, 132 are attached to hub ports 102, 106, 110, 112, respectively. A physical
star topology is created by node ports 126 - 132 around hub 100. The internal
connection of hub ports 102 - 112 by internal hub links 114 - 124 forms an internal
loop within hub 100 and creates a loop topology among node ports 126 - 132.

Each node port is connected to a hub port by a pair of data channels. For
example, node port 126 is connected to hub port 102 by data channels 134 and 136.
Data channel 134 transmits data from hub port 102 to node port 126. Data channel
136 transmits data from node port 126 to hub port 102. In this way, a datapath is
created around the loop including each of the hub ports and the node ports.

A synchronization problem occurs from the accumulation of time based
noise (i.e., loss of precision in a signal), commonly referred to as "jitter", in signals.

As a signal passes through logic circuits, a finite amount of jitter is typically intro-
duced to the signal. Jitter accumulates as a signal passes through progressively more logic circuits and accumulates more noise. Because the datastream in a conventional hub typically passes through multiple hub ports before reaching the destination node port, the accumulated jitter may be enough to prevent the destination node port from being able to recover a suitable clock signal from the datastream. If this recovery fails, the node port typically sees many invalid 10B characters and so has a high bit-error rate.

If two node ports are communicating across hubs, the number of hub ports and logic circuits through which the datastream passes typically increases, resulting in yet more jitter. This increase in jitter from connecting hub ports in hub to hub connections limits the cascadability of hubs in conventional implementations.

For example, as shown in FIG. 1B, two hubs 150, 170 are connected through a pair of hub ports 158, 172. Each hub 150, 170 has four hub ports 152, 154, 156, 158, and 172, 174, 176, 178, respectively. Two node ports 160, 162 are connected to hub ports 154, 156 of hub 150, and two node ports 180, 182 are connected to hub ports 174, 176 of hub 170. Two data channels 184, 186 interconnect hub port 158 and hub port 172. Data channel 184 carries data from hub port 172 to hub port 158. Data channel 186 carries data from hub port 158 to hub port 172. Due to the interconnection of hubs 150 and 170 the number of hub ports in the loop increases from four to eight. Thus, the number of hub ports through which the datastream passes may substantially increase. Before the interconnection, when node port 160 sends data to node port 162, the data passes through two hub ports: 154 and 156. After the interconnection, node port 160 may send data to node port 162, node port 180, or node port 182. When node port 160 sends data to node port 182, the data passes through eight hub ports and two node ports: 154, 156, 162 (where the data is re-timed), 156, 158, 172, 174, 180 (re-timed), 174, and 176. Thus, as hubs are cascaded to increase the number of node ports in the loop, the number of hub ports through which data may pass increases and so the potential accumulation of jitter increases.
Under conventional FC-AL protocols, when synchronization is lost, such as by accumulated jitter as described above, loop initialization primitive ("LIP") primitive sequences may be generated by one or more node ports to reinitialize the loop. LIP primitive sequences are formed from three consecutive LIP ordered sets of the same type. LIP ordered sets are known and defined in conventional FC-AL protocols. When node ports receive a LIP primitive sequence, node ports typically end ordinary processing and may generate LIP primitive sequences of their own. Under conventional FC-AL protocols, the loop may not return to ordinary processing for as long as approximately 500 to 600 microseconds. This delay in and of itself is an undesirable result of synchronization errors. In addition, some computer software applications being used across the loop may not be tolerant of the LIP process, and, at the very least, all applications experience a delay. An application may "crash" (i.e., abruptly and completely stop execution) and need to be restarted to continue processing. Such a crash increases the amount of delay experienced by users or other processes connected to the loop. In addition, data loss or data corruption may result as data storage devices may not be properly accessed upon the crash generated by the LIP process.

Accordingly, the inventors have determined that it would be desirable to provide a hub port which internally synchronizes signals passed around a loop network having characteristics similar to Fibre Channel Arbitrated Loop networks.

**SUMMARY**

A hub port of the preferred embodiment provides a clock local to the hub. The local clock reduces and preferably eliminates the transfer of jitter in the signal transmitted to a node port.

In one implementation, the hub port includes smoothing circuits to synchronize data received from the attached node port to a local clock that is internal to the hub and common to all hub ports in the hub. Jitter transfer is preferably eliminated by transmitting data to an attached node port using an internal clock of the
hub rather than a clock signal recovered from the datastream. In addition, hub
cascadability is not limited by jitter accumulation.

An additional advantage provided by the synchronization of data from a
node port to a local clock is that a constant phase is maintained in the datastream of
the hub.

BRIEF DESCRIPTION OF DRAWINGS

FIG. 1A shows a prior art configuration of a loop including a hub.
FIG. 1B shows a prior art configuration of two hubs interconnected
through hub ports.
FIG. 2 is a block diagram of a first preferred implementation of a hub port.
FIG. 3 is a block diagram of a second preferred implementation of a hub
port.

DETAILED DESCRIPTION

As described above, the invention will be explained below in the context
of a Fibre Channel Arbitrated Loop ("FC-AL") as an illustration of preferred embodi-
ments. However, the invention may have applicability to networks with similar
characteristics as FC-AL networks.

The preferred embodiment avoids the problem of jitter transfer described
above by using a local clock signal and including a smoothing circuit within each hub
port. Each hub port is connected to a clock which is local to the hub and transmits
data to an attached node port using the local clock signal. The smoothing circuit
compensates for any discrepancy between the clock signal of the node port and the
local clock signal.

Data is transmitted from the hub port to the attached node port using the
local clock signal. The use of the local clock preferably eliminates transfer of jitter
from the node port to the hub port. As a result, preferably no jitter is passed through
by hub ports and the cascadability of hubs is not limited by jitter transfer.
Each hub port may receive data from potentially two sources: an attached node port and an upstream node port. While a hub port is not connected to a node port, the hub port is in bypass mode and passes data from the upstream hub port to a downstream hub port. When a node port is connected to a hub port and is transmitting valid data to the hub port, the hub port switches out of bypass mode and switches from outputting the datastream from the upstream hub port to outputting the datastream from the node port. The hub port monitors data received from the attached node port and synchronizes that data to the local clock signal using a smoothing circuit. The node port transmits data to the hub port using a clock internal to the node port. The hub port recovers the node port’s clock signal from the data from the node port and uses that recovered clock signal to receive the data from the node port. The frequencies of the recovered clock signal and the local clock signal are typically similar but not necessarily the same and so the smoothing circuit inserts and deletes fill words from the datastream to compensate for lack of synchronization.

In addition, the hub port ensures that data from the node port enters the datastream of the loop only upon bit and word boundaries. Because each hub port in the hub inserts data into the hub loop using a local clock, the datastream on the hub loop is synchronized to the local clock. Because the datastream from the upstream hub port and the smoothed datastream from the node port are synchronized to a local clock, rather than using a clock signal recovered from the datastream from the node port, the switch from one datastream to another does not affect the phase of the datastream on the hub loop. Similarly, when an attached node port is disconnected or fails and the hub port switches into bypass mode, switching from the datastream from the node port to the datastream from the upstream hub port does not affect the phase of the hub loop datastream.

More particularly, FIG. 2 is a block diagram of a first preferred embodiment of a hub port. As described above, the jitter transfer problem of conventional hub ports is typically caused by transmitting a data signal with accumulated noise to a node port using a clock signal recovered from the data signal. This implementation
transmits data to a node port using a local clock and so transmits a clean data signal to the node port. Thus, the hub port preferably eliminates jitter transfer.

As shown in FIG. 2, a hub port 200 includes an incoming internal hub link 202 connected to a data input of a switching device such as a multiplexer 204 and to a transmit circuit 206. Transmit circuit 206 is also connected to a hub clock line 208. Hub clock line 208 is connected to a clock (not shown) which is internal to the hub containing hub port 200. Hub clock line 208 is also connected to each hub port in the hub containing hub port 200. Transmit circuit 206 transmits data along data channel 210 to node port 212 using the local clock signal on hub clock line 208. Thus, data is transmitted to node port 212 using a clean clock signal (i.e., substantially free of noise). Accordingly, the data signal to node port 212 is clean and preferably no jitter is transferred.

Node port 212 transmits data to hub port 200 along data channel 214. Data channel 214 is connected to a receive circuit 216 and a clock recovery circuit 218. Clock recovery circuit 218 recovers a clock signal from data received from node port 212 and supplies that clock signal to receive circuit 216. Receive circuit 216 outputs data using the recovered clock signal to a smoothing circuit 220. Smoothing circuit 220 is connected to hub clock line 208 and outputs data to a data input of multiplexer 204 using the local clock signal on hub clock line 208. Thus, data from node port 212 is received using the recovered clock signal and output onto the loop using the local clock signal. As noted above and described in more detail below with reference to FIG. 3, the recovered clock signal and the local clock signal are not necessarily the same exact frequency and so smoothing circuit 220 avoids underflow and overflow in the datastream from any discrepancy in the clock rates. Smoothing circuit 220 also supplies control signals to a control input of multiplexer 204 to control the selection of data inputs for multiplexer 204. Multiplexer 204 outputs data onto outgoing internal hub link 222 into the hub loop of the hub containing hub port 200. Thus, hub port 200 synchronizes data received from node port 212 using smoothing circuit 220 and the local clock signal on hub clock line 208.
Hub port 200 transmits data to node port 212 using the local clock signal on hub clock line 208. Because the data signal is output from transmit circuit 206 using a clean clock signal, the data signal transmitted to node port 212 is clean and substantially free of jitter. Thus, hub port 200 avoids jitter transfer and improves hub cascadability.

In an alternative implementation, the smoothing circuit is contained in the transmit circuit of the hub port. This implementation may not provide the additional benefit of constant phase, however.

FIG. 3 is a block diagram of a second preferred implementation of a hub port 300 showing the components in more detail. Hub port 300 includes components for transmitting data to a node port 314, for receiving data from node port 314, and for bypassing or inserting hub port 300 in and out of the hub loop.

An incoming internal hub link 302 connects hub port 300 to an upstream hub port (not shown) in the loop of the hub containing hub port 300. Incoming internal hub link 302 is connected to a data input of a multiplexer 304 and to an encoder 306. Encoder 306 is preferably a conventional 8B/10B encoder. Encoder 306 is connected to a transmit circuit 308. Transmit circuit 308 is connected to a serializer/deserializer ("SERDES") circuit 310. SERDES circuit 310 serializes data and sends the data through a data channel 312 to node port 314. Each of encoder 306, transmit circuit 308, and SERDES circuit 310 is connected to hub clock line 336 and outputs data using that clock signal. Thus, data is transmitted to node port 314 using a clean clock signal. Accordingly, the data signal to node port 314 is clean and preferably no jitter is transferred in the data signal to node port 314.

A data channel 316 carries data from node port 314 to SERDES circuit 310. SERDES circuit 310 deserializes serial data received from node port 314 and passes the data to a receive circuit 318. SERDES circuit 310 also recovers a clock signal from the deserialized data. SERDES circuit 310 sends the recovered clock signal via a recovered clock line 317 to receive circuit 318, a word synchronization detect circuit 322 and a "smoothing" first-in-first-out ("FIFO") buffer 324. Alternatively, smoothing buffer 324 need not be a FIFO buffer.
Receive circuit 318 passes the data to a decoder 320. Decoder 320 is preferably a conventional 10B/8B decoder. Decoder 320 passes the decoded data to word synchronization detect circuit 322 and smoothing FIFO buffer 324. Word synchronization detect circuit 322 is connected to a smoothing FIFO control circuit 326, which is also coupled to smoothing FIFO buffer 324. Smoothing FIFO buffer 324 is connected to a data input of multiplexer 304, a fill word detect circuit 330, and a current fill word generator 332. Current fill word generator 332 is connected to fill word detect circuit 330 and to a data input of multiplexer 304. Smoothing FIFO control circuit 326 is connected to a hub port output control circuit 328, which is connected to a control input of multiplexer 304. Multiplexer 304 is connected to an output register 334.

A hub clock line 336 is connected to a local clock (not shown). The local clock supplies a common local clock signal to each hub port in the hub. Hub clock line 336 supplies a local clock signal to encoder 306, transmit circuit 308, SERDES circuit 310, smoothing FIFO buffer 324, current fill word generator 332, and output register 334.

Output register 334 outputs data from multiplexer 304 onto outgoing internal hub link 338 using the local clock. Output register 334 functions as a pipeline stage between connecting hub ports.

In transmitting data from hub port 300 to node port 314, encoder 306 encodes the un-encoded Fibre Channel data, preferably using 8B/10B encoding, received along incoming internal hub link 302 from the upstream hub port. Encoder 306 does not incorporate a clock signal recovered by the SERDES circuit of an upstream hub port because that recovered clock signal is not sent by that upstream hub port. As described above, data is output onto outgoing internal hub link 338 of a hub port using the local clock signal on hub clock line 336, not the recovered clock signal. Encoder 306 preferably incorporates the local clock signal on hub clock line 336. The encoded data is forwarded to transmit circuit 308. Transmit circuit 308 ensures that only complete words are transmitted to node port 314. In addition, transmit circuit 308 adjusts the running disparity of the data (i.e., the difference
between the number of 1's and 0's in the data signal) as necessary, such as by changing the disparity to positive or negative. The data is passed to SERDES circuit 310 to be serialized and transmitted to node port 314 using the local clock on hub clock line 336. As described above, by transmitting data to node port 314 using the local clock signal, preferably no jitter which may have accumulated in the datastream is transferred to node port 314.

In receiving data from an attached node port 314, an important part of the preferred hub port design is smoothing FIFO buffer 324 and smoothing FIFO control circuit 326. Smoothing FIFO buffer 324 in conjunction with smoothing FIFO control circuit 326 synchronizes data received from node port 314 to the internal clock of the hub port. Smoothing FIFO buffer 324 provides a smoothing function between the two potentially asynchronous clock domains of the clock signal recovered from the data and internal clock of hub port 300.

In operation, when hub port 300 is connected to and receiving data from node port 314, SERDES circuit 310 deserializes an incoming serial data stream from node port 314, generating deserialized 10B-encoded data. SERDES circuit 310 also recovers a clock signal from the serial data. SERDES circuit 310 outputs the recovered clock signal to recovered clock line 317 to be received by receive circuit 318, word synchronization detect circuit 322 and smoothing FIFO buffer 324. Receive circuit 318 receives the 10B-encoded data and outputs the data to decoder 320, synchronized to the recovered clock signal. Decoder 320 then performs a 10B/8B decoding of the data. The decoded data is forwarded to word synchronization detect circuit 322 and smoothing FIFO buffer 324.

Word synchronization detect circuit 322 preferably follows procedures defined in clause 12.1 of ANSI publication X3.230-1994 (FC-PH, Rev 4.3, June 1, 1994). In particular, when word synchronization detect circuit 322 detects three consecutive valid Fibre Channel ordered sets emerging from decoder 320, hub port 300 has acquired word synchronization and word synchronization detect circuit 322 enters a synchronization-acquired state. Word synchronization detect circuit 322 enables smoothing FIFO control circuit 326, which in turn enables subsequent Fibre
Channel words to be written into smoothing FIFO buffer 324. However, whenever a number of invalid Fibre Channel words specified by Fibre Channel protocols are decoded by decoder 320 and detected by word synchronization detect circuit 322, word synchronization detect circuit 322 enters a synchronization-lost state. Smoothing FIFO control circuit 326 disables words from being written into smoothing FIFO buffer 324 if word synchronization is lost.

Smoothing FIFO buffer 324 includes a FIFO buffer and, in conjunction with smoothing FIFO control circuit 326, synchronizes the received Fibre Channel data to the local clock frequency on hub clock line 336. Data received from node port 314 is written into smoothing FIFO buffer 324 at a data rate determined by the recovered clock on recovered clock line 317. The data is synchronized to the local clock signal determined by a local oscillator (not shown in FIG. 3) within the hub. Data is output from smoothing FIFO buffer 324 into multiplexer 304 at a data rate determined by the local clock signal on hub clock line 336.

Multiplexer 304 outputs data into output register 334. Switching of multiplexer 304 is controlled by hub output control circuit 328, which is connected to the control input of multiplexer 304. When data received from node port 314 is to be inserted into the loop, the data input corresponding to smoothing FIFO buffer 324 is selected. When fill words are to be inserted into the loop (described below), the data input corresponding to current fill word generator 330 is selected. When hub port 300 is in a bypass mode, the data input corresponding to incoming internal hub link 302 is selected. Each of the data streams have a constant phase and switching among the data inputs occurs on bit boundaries to preserve the constant phase. Output data register 334 outputs data onto outgoing internal hub link 338 using the local clock signal received on hub clock line 336.

The recovered clock frequency and the local clock frequency may vary by a small amount, such as ±100ppm. The difference in the recovered and local clock frequencies results in a difference between the data rates for writing data into (using the recovered clock signal) and removing data from (using the local clock signal) smoothing FIFO buffer 324. Smoothing FIFO control circuit 326 “smooths” out the
differences in incoming and outgoing data rates for smoothing FIFO buffer 324 using fill words. Fill words are typically inserted into inter-frame gaps between data frames in Fibre Channel data streams. Essentially every data stream includes some number of fill words. A current fill word is typically selected from a list formed according to FC-AL protocols. By scheduling either insertions or deletions of current fill words when the data in smoothing FIFO buffer 324 reaches various thresholds, smoothing FIFO control circuit 326 prevents smoothing FIFO buffer 324 from running dry or from overflowing. When smoothing FIFO control circuit 326 determines that deleting a fill word is appropriate, smoothing FIFO control circuit 326 waits until a suitable fill word is found in an inter-frame gap before a delete is performed. When an insert is appropriate, smoothing FIFO control circuit 326 waits for an inter-frame gap and then causes multiplexer 304 to select the data input corresponding to current fill word generator 332 to be output to output register 334. At the same time, smoothing FIFO control circuit 326 temporarily stalls data reads from smoothing FIFO buffer 324. Connected to current fill word generator 332 is fill word detect circuit 330. Fill word detect circuit 330 keeps track of the latest suitable fill word by updating current fill word generator 332 with the fill word to be used if a fill word insert is appropriate.

As data is removed from smoothing FIFO buffer 324 and output to multiplexer 304, if the local clock frequency is faster than the recovered clock frequency, the amount of data in smoothing FIFO buffer 324 diminishes. The amount of data in smoothing FIFO buffer 324 may also diminish when writing into smoothing FIFO buffer 324 is disabled due to loss of word synchronization. If the amount of data in smoothing FIFO buffer 324 falls below a minimum threshold, smoothing FIFO control circuit 326 causes hub port output control circuit 328 to bypass hub port 300. Hub port output control circuit 328 causes multiplexer 304 to select the data input corresponding to incoming internal hub link 302 to output data directly from the upstream hub port rather than the output of smoothing FIFO buffer 324. Once the amount of data in smoothing FIFO buffer 324 again exceeds the minimum threshold, smoothing FIFO control circuit 326 causes hub port output control circuit 328 to end
the bypass. Hub port output control circuit 328 causes multiplexer 304 to select data from smoothing FIFO buffer 324, returning node port 314 to the loop. Alternatively, different thresholds may be used for removing and inserting node ports into the loop.

If the local clock frequency is slower than the recovered clock frequency, the amount of data in smoothing FIFO buffer 324 increases. If the amount of data in smoothing FIFO buffer 324 exceeds a maximum threshold, smoothing FIFO control circuit 326 deletes fill words as described above.

In operation, when hub port 300 is not connected to a node port 314, hub port 300 is in bypass mode. Multiplexer 304 connects the data input corresponding to incoming internal hub link 302 to output register 334. When node port 314 is initially connected to hub port 300 and begins transmitting a datastream, a process similar to that described above occurs. The data is deserialized by SERDES circuit 310, decoded by decoder 320, and supplied to word synchronization detect circuit 322. If valid ordered sets are being received, word synchronization detect circuit 322 enters a synchronization-acquired state and the received data starts to be written into smoothing FIFO buffer 324. As data continues to be written into smoothing FIFO buffer 324, smoothing FIFO buffer 324 exceeds a minimum threshold. At this point, smoothing FIFO control circuit 326 causes hub port output control circuit 328 to end bypass mode for hub port 300. Hub port output control circuit 328 causes multiplexer 304 to select the data input corresponding to smoothing FIFO buffer 324 rather than data from the upstream hub port on incoming internal hub link 302. Because data received from node port 314 is inserted into the loop using a local clock signal to read data out of smoothing FIFO buffer 324, node port 314 is inserted into the loop without disrupting the synchronization of the datastream.

In operation, when the datastream being received by hub port 300 from node port 314 is disrupted, such as by loss of word synchronization in the datastream, failure of node port 314, or disconnection of node port 314, word synchronization detect circuit 322 detects the loss of valid synchronized data and enters a synchronization-lost state. Subsequent data is not written into smoothing FIFO buffer 324. Data continues to be removed from smoothing FIFO buffer 324 and output by multiplexer
304 to output register 334 causing smoothing FIFO buffer 324 to approach empty. As described above, when smoothing FIFO control circuit 326 detects that smoothing FIFO buffer 324 has fallen below a minimum threshold, smoothing FIFO control circuit 326 causes hub port output control circuit 328 to bypass hub port 300. Hub port output control circuit 328 causes multiplexer 304 to select the data input corresponding to incoming internal hub link 302 and data from the upstream hub port. Thus, while valid data is not received from node port 314, hub port 300 enters a bypass mode. Because the bypass occurs in synchronization with the local clock signal, the failure of node port 314 does not disrupt the datastream of the loop.

In this manner, jitter transfer between hub ports is substantially eliminated. In addition a node port or a loop segment of multiple node ports may be seamlessly inserted into or removed from the loop by a hub port without causing a disruption in the phase of data transmitted to a downstream node port.

A number of embodiments of the invention have been described. However, these embodiments are illustrative and not limiting. Alternative embodiments and variations will be apparent to one of ordinary skill in the art. For example, an internal clock may be included in each hub port or a single internal clock may be provided in the hub and connected to each hub port. The scope of the invention is limited only by the scope of the following claims.
WHAT IS CLAIMED IS:

1. A hub port for connecting a hub to a node port, comprising:
   (a) a hub clock line connected to a clock internal to the hub for supplying a
        local clock signal; and
   (b) a transmit circuit which transmits data to the node port synchronized to the
        local clock signal.

2. The hub port of claim 1 further comprising:
   (a) a receive circuit which receives data from the node port; and
   (b) a smoothing circuit which receives data from the receive circuit and
        synchronizes such data to the local clock signal.

3. The hub port of claim 1 where the transmit circuit ensures that only full words of
   data are transmitted to the node port.

4. A hub port for connecting a hub to a node port, comprising:
   (a) a hub clock line connected to a clock internal to the hub for supplying a
        local clock signal; and
   (b) a transmit circuit which transmits data to the node port synchronized to the
        local clock signal.

5. The hub port of claim 4 further comprising a smoothing circuit which synchronizes data received from the node port to the local clock signal.
6. A hub port for connecting a hub to a node port, comprising:
   (a) an incoming internal hub link;
   (b) an outgoing internal hub link;
   (c) a hub clock line connected to a clock internal to the hub for supplying a local clock signal;
   (d) a transmit circuit connected to the incoming internal hub link and to the node port, where the transmit circuit receives data from the incoming internal hub link and sends such data to the node port using the local clock signal;
   (e) a receive circuit connected to the node port, where the receive circuit receives data from the node port; and
   (f) a smoothing circuit connected to the receive circuit and to the outgoing internal hub link, where the smoothing circuit synchronizes data from the receive circuit to the local clock signal and sends such data to the outgoing internal hub link.
7. A hub port for connecting a hub to a node port, where the hub includes at least two hub ports interconnected in a loop with internal hub links, the hub port comprising:

(a) a hub clock line connected to a clock internal to the hub for supplying a local clock signal;

(b) a multiplexer which includes a first data input, a second data input, a third data input, and a control input;

(c) a hub output register connected to the multiplexer and to the hub clock line;

(d) an incoming internal hub link connected to the third data input of the multiplexer;

(e) an outgoing internal hub link connected to the hub output register;

(f) a transmit circuit connected to the hub loop and to the hub clock line;

(g) an encoder connected to the transmit circuit and to the hub clock line;

(h) a current fill word generator connected to the second data input of the multiplexer and to the hub clock line;

(i) a fill word detect circuit connected to the current fill word generator;

(j) a smoothing buffer connected to the first data input of the multiplexer, the current fill word generator, the fill word detect circuit, and to the hub clock line;

(k) a smoothing control circuit connected to the smoothing buffer;

(l) a hub port output control circuit connected to the smoothing control circuit and the control input of the multiplexer;

(m) a word synchronization detect circuit connected to the smoothing control circuit;

(n) a receive circuit connected to the word synchronization detect circuit and the smoothing buffer;

(o) a decoder connected to the receive circuit; and
(p) a serializer/deserializer circuit connected to the encoder, the decoder, the hub clock line, and the node port, where the serializer/deserializer circuit includes a recovered clock output connected to the decoder, the word synchronization detect circuit, and the smoothing buffer, and further where the serializer/deserializer circuit transmits data to the node port using the local clock signal.

8. The hub port of claim 7 where the smoothing buffer is a FIFO buffer.

9. The hub port of claim 7 where the encoder is a 8B/10B encoder.

10. The hub port of claim 7 where the decoder is a 10B/8B decoder.

11. The hub port of claim 7 where the local clock is contained in the hub port.

12. The hub port of claim 7 where the transmit circuit ensures that only full words of data are transmitted to the node port.

13. A method of reducing jitter transfer in a hub port for connecting a hub to a node port, where the hub includes at least two hub ports arranged in a logical order, the method comprising:
(a) synchronizing data from an upstream hub port to a clock signal local to the hub port; and
(b) transmitting the synchronized data from the upstream hub port to the node port.
14. The method of claim 13 further comprising:
   (a) synchronizing data from the node port to the local clock signal; and
   (b) outputting the synchronized data from the node port to a downstream hub
       port.

15. A method of reducing jitter transfer in a hub port for connecting a hub to a node
    port, where the hub includes at least two hub ports arranged in a logical order,
    the method comprising:
   (a) sending data from the node port to the hub port;
   (b) deserializing the data from the node port;
   (c) recovering a recovered clock signal from the data from the node port;
   (d) decoding the deserialized data;
   (e) checking the decoded data for word synchronization;
   (f) writing the decoded data to a buffer using the recovered clock signal;
   (g) reading the decoded data from the buffer using a local clock signal;
   (h) outputting the decoded data from the hub port to a downstream hub port;
   (i) receiving data from an upstream hub port; and
   (j) transmitting the data received from an upstream hub port to the node port
       using the local clock signal.
16. A method of reducing jitter transfer in a hub port for connecting a hub to a node port, where the hub includes at least two hub ports arranged in a logical order, the method comprising:
   (a) sending data from the node port to the hub port;
   (b) deserializing the data from the node port;
   (c) recovering a recovered clock signal from the data from the node port;
   (d) decoding the deserialized data;
   (e) checking the decoded data for word synchronization;
   (f) writing the decoded data to a buffer using the recovered clock signal;
   (g) reading the decoded data from the buffer using a local clock signal;
   (h) outputting the decoded data from the hub port to a downstream hub port;
   (i) repeating steps (e) to (h) so long as the data received from the node port is word synchronized;
   (j) while word synchronization is lost, outputting data from the hub port which is received from another hub port in the hub;
   (k) when an amount of data in the buffer falls below a minimum threshold, outputting at least one current fill word from the hub port;
   (l) when the amount of data in the buffer exceeds a maximum threshold, deleting at least one fill word from the data stored in the buffer;
   (m) receiving data from an upstream hub port; and
   (n) transmitting the data received from an upstream hub port to the node port using the local clock signal.

17. A system for reducing jitter transfer in a hub port for connecting a hub to a node port, comprising:
   (a) means for synchronizing data from an upstream hub port to a clock signal local to the hub port; and
   (b) means for transmitting the synchronized data from an upstream hub port to the node port.
18. The system of claim 17 further comprising:

(a) means for synchronizing data from the node port to the local clock signal; and

(b) means for outputting the synchronized data from the node port to a downstream hub port.
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : H04J 3/06
US CL. : 370/ 516, 468
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)


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

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

APS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category*</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>U.S 5,506,846 A (EDEM ET AL) 09 April 1996, Figs. 3 and 10; column 2 lines 25-64, column 4 lines 20-57, column 6 lines 15-67, column 7 lines 1-37, column 9 lines 34-67, column 10 lines 1-12.</td>
<td>1-6, 13-16</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C. See patent family annex.

Date of the actual completion of the international search 01 JULY 1999

Date of mailing of the international search report 27 AUG 1999

Name and mailing address of the ISA/US Commissioner of Patents and Trademarks

Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

Ricardo Pizarro

Telephone No. (703) 305-1121

Form PCT/ISA/210 (second sheet) (July 1992)