Copyright

Copyright © 2000 KnowledgeTek, Inc.

All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from KnowledgeTek, Inc.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume 1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1394 Overview</td>
</tr>
<tr>
<td>2</td>
<td>1394 Asynchronous Operations</td>
</tr>
<tr>
<td>3</td>
<td>Control and Status Registers</td>
</tr>
<tr>
<td>4</td>
<td>Introduction To SCSI Over 1394</td>
</tr>
<tr>
<td>5</td>
<td>Serial Bus Protocol SBP-2</td>
</tr>
<tr>
<td>6</td>
<td>SCSI Over SBP-2</td>
</tr>
<tr>
<td>7</td>
<td>ATA Over SBP-2</td>
</tr>
<tr>
<td>Volume 2</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Physical</td>
</tr>
<tr>
<td>9</td>
<td>Arbitration</td>
</tr>
<tr>
<td>10</td>
<td>Isochronous Operations</td>
</tr>
<tr>
<td>11</td>
<td>Configuration</td>
</tr>
<tr>
<td>12</td>
<td>Bus Management</td>
</tr>
<tr>
<td>13</td>
<td>Implementation</td>
</tr>
<tr>
<td>14</td>
<td>Power Management</td>
</tr>
<tr>
<td>15</td>
<td>1394 Specifications</td>
</tr>
<tr>
<td>16</td>
<td>Audio/Video</td>
</tr>
<tr>
<td>17</td>
<td>1394B</td>
</tr>
</tbody>
</table>
# Table of Contents - Appendices

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>IEEE 1394-1995 Table of Contents</td>
</tr>
<tr>
<td>C</td>
<td>Isochronous Connection Management</td>
</tr>
<tr>
<td>D</td>
<td>Device Bay</td>
</tr>
<tr>
<td>R</td>
<td>Reduced Block Command Set</td>
</tr>
<tr>
<td>Z</td>
<td>Answers</td>
</tr>
</tbody>
</table>

---

© KnowledgeTek, Inc.

rev 5.41E  15.M.31
Section 8

Physical
Subjects Covered

- Physical and Logical Topology
- Cable Signals
- DS Encoding
- Common Mode Signaling
- Arbitration Signaling
- Plug Circuitry
- Connect and Bias Detect
1394 Physical Characteristics

Point-to-Point connectors
  Each connection end terminated

Hot-Plugging allowed

4 Wire Cable
  Two twisted Pairs for data
  No Power

6 Wire Cable
  Two twisted pairs for data transmission
  2 wires carry power

4.5 meters per hop limit
1394 Physical Topology

Multi-Port Devices repeat all bus traffic on other ports* 
All Devices see all packets

* provided the remote port can handle that packet speed - described later
Point-To-Point Connections

Data fed serially using DATA and STROBE signals
POWER supplies 8-30 Volts (@1.5 Amp)
The Cable

- Power wire
  - 22AWG /0.87 diameter wires
- 60% braided shield over foil shield
  (over signal pairs - 2X)
- Signal pairs: (2X)
  - 28 AWG/0.87 diameter twisted pairs
- 97% braided overall shield
- 0.70 thick PVC jacket
- Fillers for roundness (if required)
- Power wire
  - 22AWG /0.87 diameter wires
The Standard 6 Pin Connector

Plug

Socket

Conductive Outer Housing

Spring Contacts

Traces on Insertion Wafer
Note Power Makes Contact First

© KnowledgeTek, Inc.
rev 5.41E 15.H.31

1394
Sect 8: Physical
4 Pin Connector

No Power Signals
Data/Strobe Encoding

Data is sent via the DATA Signal encoded NRZ (Non-Return to Zero) STROBE toggles whenever DATA doesn’t

Data: 1 0 1 1 1 0 0 0 0 1

DATA

STROBE

DATA XOR STROBE
(Delayed 1/2 bit time)

Provides clock for recovering Data
Data Recovery

DATA Signal

Read Clock
DATA XOR STROBE
(Delayed)

Data:

1 0 1 1 0 0 0 1
## Clock Rates

<table>
<thead>
<tr>
<th>Model</th>
<th>Data Rate</th>
<th>Error</th>
<th>Bit Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>S100</td>
<td>98.304 Mbps ± 100ppm</td>
<td>10.17 nSec</td>
<td></td>
</tr>
<tr>
<td>S200</td>
<td>196.608 Mbps ± 100ppm</td>
<td>5.09 nSec</td>
<td></td>
</tr>
<tr>
<td>S400</td>
<td>393.216 Mbps ± 100ppm</td>
<td>2.54 nSec</td>
<td></td>
</tr>
</tbody>
</table>
Differential Signaling

DATA and STROBE signals transmitted differentially

DATA

DATA*

Vcm

Common Mode Voltage

1 0 0 1

© KnowledgeTek, Inc.
rev 5.41E 15.H.31

Sect 8: Physical
## Differential Signal Levels

<table>
<thead>
<tr>
<th></th>
<th>S100</th>
<th>S200</th>
<th>S400</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>Transmitting</td>
<td>265mV</td>
<td>172mV</td>
<td>265mV</td>
</tr>
<tr>
<td>End</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiving</td>
<td>260mV</td>
<td>142mV</td>
<td>260mV</td>
</tr>
<tr>
<td>End (Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmission)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiving</td>
<td>260mV</td>
<td>173mV</td>
<td>262mV</td>
</tr>
<tr>
<td>End (Arbitration)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Data Transmission)

(Arbitration)
Crossover Cable

Cable Signal Pairs called TPA and TPB (Twisted Pairs A & B)

Port Node x

TPA

TPA*

TPB

TPB*

Port Node y

TPA

TPA*

TPB

TPB*

V_P (Power)

V_G (Ground)

TPA Transmits STROBE
Receives DATA

TPB Transmits DATA
Receives STROBE

Crossover eliminates need for separate upstream/downstream connectors

Similar to Null-Modem cables
Maximum cable length - 4.5 meters

Note: signal names are the same on both ends

pins 3 and 4 on one end connect to pins 5 and 6 on the other end
## Cable Characteristics

<table>
<thead>
<tr>
<th>Description</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPA &amp; TPB Differential Impedance</td>
<td>110Ω ± 6Ω</td>
</tr>
<tr>
<td>TPA &amp; TPB Common Mode Impedance (to VG)</td>
<td>33Ω ± 6Ω</td>
</tr>
<tr>
<td>TPA &amp; TPB Cable Attenuation</td>
<td>2.3dB max @ 100MHz</td>
</tr>
<tr>
<td></td>
<td>3.2dB max @ 200MHz</td>
</tr>
<tr>
<td></td>
<td>5.8dB max @ 400MHz</td>
</tr>
<tr>
<td>VP &amp; VG DC Resistance</td>
<td>0.333Ω max</td>
</tr>
</tbody>
</table>

© KnowledgeTek, Inc. rev 5.41E 15.H.31
Port Basic Interface

TPBias sets Common Mode Voltage
Receiving End Terminated (Either Direction)
CMOS Driver - 4mAmp
Node A will transmit to Node B at:
- S400 if packet is for Node B
- S200 if packet is for Node C
- S200 if packet is for Node D
- S100 if packet is for Node E

Nodes must signal transmission speed at the beginning of each packet
Nodes do not forward packets that are faster than a receiving port’s speed
Speed Sensing

Common Mode Voltage used to sense speed capabilities

TPA Port sets Common Mode Voltage through termination resistors

TPB Port pulls current out of node

<table>
<thead>
<tr>
<th>Port</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>S100</td>
<td>0mA</td>
</tr>
<tr>
<td>S200</td>
<td>3.5mA</td>
</tr>
<tr>
<td>S400</td>
<td>9mA</td>
</tr>
</tbody>
</table>

These values approximate

TPA Port senses reduced Common Mode Voltage

<table>
<thead>
<tr>
<th>Port</th>
<th>Voltage Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>S100</td>
<td>1.665V - 2.015V</td>
</tr>
<tr>
<td>S200</td>
<td>1.438V - 1.665V</td>
</tr>
<tr>
<td>S400</td>
<td>1.092V - 1.438V</td>
</tr>
</tbody>
</table>

Bi-Directional - Each Port knows neighbors speed

Don’t need to sense for faster than your own speed
**Speed Sensing**

- **TPBias**
  - (1.675V - 2.025V)

- **S100**
  - +0.44mA to -0.81mA

- **S200**
  - -2.53mA to -4.84mA

- **S400**
  - -8.10mA to -12.40mA

- **Sense Common Mode Voltage**
  - To Determine Speed

- **TPA**
  - 55Ω

- **TPA**
  - 55Ω

- **TPB**
  - 55Ω

- **TPB**
  - 55Ω

- **5KΩ**

- **250pF**

- **V_G**

- **V_G**

- **TX**
Speed Sensing Example

TPA

1.8V Common Mode
1.2V Common Mode

S100 Node

TPA*

1.8V Common Mode
1.2V Common Mode

S400 Node

© KnowledgeTek, Inc.
rev 5.41E 15.H.31

1394
Sect 8: Physical
TP Bias and Disconnect Sensing

If TP Bias is turned off,

Connect
Detect

If Cable Disconnected,
Resistor Pulls Down Common Mode Voltage

Bi-directional Disconnect Sense
Port Interface with Speed & Disconnect Sensing

Connect Detect
TPBias (1.675V - 2.025V)
STROBE Tx
STROBE Enab
DATA Rx

S100
S200
S400
Threshold Comparer
TPBias

I_CD

0.8V
7KΩ
7KΩ

TPA
TPB

55Ω
55Ω
55Ω

V_G
V_G

Port Status
STROBE Rx
DATA Enab
DATA Tx

Speed
S100 = 0.0 mA
S200 = 3.5 mA
S400 = 9.0 mA

© KnowledgeTek, Inc.
rev 5.41E 15.H.31

1394
Sect 8: Physical
Three Level Signaling

During Arbitration, a third signal level is used:

TPA or TPB

TPA* or TPB*

Vcm

Vcm

0 1 Z Z 1
Three Level Signals

TPA - TPA*
or
TPB - TPB*
At Receiver

Drive Z signal by disabling transmitter driver
Detect Z by window comparison
# Driving Three Level Signals

<table>
<thead>
<tr>
<th>Signal</th>
<th>Tx</th>
<th>Enab</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Z</td>
<td>X</td>
<td>0</td>
</tr>
</tbody>
</table>
Receiving Three Level Signals

<table>
<thead>
<tr>
<th>POS</th>
<th>NEG</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>Z</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>0</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>1</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>Bad</td>
</tr>
</tbody>
</table>

To Rest of Transceiver
1394 Port Transceiver

TPA Transceiver – Port x

Connect Detect

TPBias (1.675V - 2.025V)

STROBE Tx

STROBE Enab

DATA Rx

Arb A POS

Arb A NEG

Window

Compare

Threshold

Compares

TPBias

I_{CD}

V_{G}

55\Omega

55\Omega

7K\Omega

7K\Omega

S100

S200

S400

TPB Transceiver – Port y

0.8V

Bias Detect

7K\Omega

7K\Omega

STROBE Rx

DATA Enab

DATA Tx

Arb B POS

Arb B NEG

Window

Compare

Threshold

Compares

Speed

S100 = 0.0 mA

S200 = 3.5 mA

S400 = 9.0 mA

© KnowledgeTek, Inc.

rev 5.41E  15.H.31

1394

Sect 8: Physical

8 - 28
Transceiver Big Picture

Port on Node x

STROBE Tx
STROBE Enab
DATA Rx
ARBA Rx
Speed
Connect Detect

TPA Transceiver

Port on Node y

STROBE Tx
STROBE Enab
DATA Rx
ARBA Rx
Speed

TPA Transceiver

STROBE Rx
DATA Tx
DATA Enab
ARBB Rx
Speed
Port Status

TPB Transceiver

STROBE Rx
DATA Tx
DATA Enab
ARBB Rx
Speed
Bias Detect

TPB Transceiver
Arbitration Signaling

Signal contained on both TPA & TPB
Both nodes drive both!
Both nodes receive both!

TPA Transceiver

ARB A Xmit
ARB A Receive

Window Compare

TPB Transceiver

ARB B Xmit
ARB B Receive

Window Compare

TPA Transceiver

ARB A Xmit
ARB A Receive

Window Compare

TPB Transceiver

ARB B Xmit
ARB B Receive

Window Compare
## Effect of Two Drivers

<table>
<thead>
<tr>
<th>Arb A Transmits</th>
<th>Arb B Transmits</th>
<th>Cable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>Z</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Z</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>Z</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Z</td>
</tr>
<tr>
<td>1</td>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Z</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ Z = \text{Not driven} \]
## Arbitration Signaling Deduction

<table>
<thead>
<tr>
<th>If I am sending</th>
<th>And I see on cable</th>
<th>He must have been sending</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>Z</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Z</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0 or Z</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>broke</td>
</tr>
<tr>
<td>1</td>
<td>Z</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>broke</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1 or Z</td>
</tr>
</tbody>
</table>
this page is intentionally blank
## Arbitration Signal Encoding - Transmit

<table>
<thead>
<tr>
<th>Signal Transmitted</th>
<th>Arb A Tx</th>
<th>Arb B Tx</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle</td>
<td>Z</td>
<td>Z</td>
<td>Sent to indicate a gap</td>
</tr>
<tr>
<td>Request</td>
<td>Z</td>
<td>0</td>
<td>Sent to parent to request the bus</td>
</tr>
<tr>
<td>Grant</td>
<td>Z</td>
<td>0</td>
<td>Sent to child when bus is granted</td>
</tr>
<tr>
<td>Parent Notify</td>
<td>0</td>
<td>Z</td>
<td>Sent to parent during Tree-ID</td>
</tr>
<tr>
<td>Data Prefix</td>
<td>0</td>
<td>1</td>
<td>Sent before data packets</td>
</tr>
<tr>
<td>Child Notify</td>
<td>1</td>
<td>Z</td>
<td>Sent to child to Ack parent notify</td>
</tr>
<tr>
<td>Ident Done</td>
<td>1</td>
<td>Z</td>
<td>Sent to parent, self-ID done</td>
</tr>
<tr>
<td>Data End</td>
<td>1</td>
<td>0</td>
<td>Sent at end of packet transmission</td>
</tr>
<tr>
<td>Bus Reset</td>
<td>1</td>
<td>1</td>
<td>Sent to force a bus reconfiguration</td>
</tr>
<tr>
<td>Tx Disable Notify</td>
<td>Z</td>
<td>1</td>
<td>Requests peer node to enter suspend state</td>
</tr>
<tr>
<td>Tx Suspend</td>
<td>0</td>
<td>0</td>
<td>Requests peer node to handshake Tp Bias and enter suspend state,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>propagate suspend to all active ports</td>
</tr>
</tbody>
</table>
### Arbitration Signal Encoding - Receive

<table>
<thead>
<tr>
<th>Arb A Rx</th>
<th>Arb B Rx</th>
<th>Signal Received</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>Z</td>
<td>Idle</td>
</tr>
<tr>
<td>Z</td>
<td>0</td>
<td>Parent Notify or Request Cancel</td>
</tr>
<tr>
<td>Z</td>
<td>1</td>
<td>Ident Done</td>
</tr>
<tr>
<td>0</td>
<td>Z</td>
<td>Self ID Grant or Request</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>Root Contention or Grant or Rx suspend</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Parent Handshake or Data End</td>
</tr>
<tr>
<td>1</td>
<td>Z</td>
<td>Child Handshake or Rx disable notify</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Data Prefix</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Bus Reset</td>
</tr>
</tbody>
</table>

**Notice Cable Twist**
- Data Prefix Transmit: $A = 0$, $B = 1$
- Data Prefix Received: $A = 1$, $B = 0$
Bi-directional Signaling

Arbitration Signal Encoder chosen carefully
When both sides transmit - cable still has correct signal

Example

<table>
<thead>
<tr>
<th>Transmit</th>
<th>Data Prefix</th>
<th>Receive</th>
<th>Data End</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Z</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

This works. Data Prefix should nullify Request
Sending Packets

Gap

Bus is in Idle state (Send A=Z B=Z) \textit{varies in length}

Arb

Arbitration - covered in next section

Data Prefix

Bus is in Data Prefix State (Send A=0 B=1) 
Signals Data Coming 
4 - 160 T_{BR} (40nSec - 1.63\mu\text{Sec})

Packet

Normal Data Encoding (Send A=STROBE B=DATA)

Data End

Bus is in Data End State (Send A=1 B=0) 
24 T_{BR} (240 - 260 nSec)

\[ T_{BR} = \text{Base Rate Bit Time} \approx 10 \text{ nSec} \]
Asynchronous Subactions

Transaction

Ack Gap
4 $T_{BR}$ (40 - 50 nSec)

SubAction Gap
Defaults to 1036 $T_{BR}$ (10μSec)
Can and should be set shorter

© KnowledgeTek, Inc.
Concatenated Asynchronous Subactions

subaction 1: request

subaction 2: response
Physical Review

1. Explain DS encoding
2. What are the defined speeds for 1394a - 2000?
3. Explain arbitration signaling
4. How is speed sensing done?
5. Explain connect detect and bias detect
6. What does “differential” signaling mean?
Physical Notes
Section 9

Arbitration
Subjects Covered

Normal arbitration
Arbitration enhancements
Fairness
Priority register
Physical Topology

Nodes are numbered automatically during Configuration.
Highest numbered node is the Root.
Leaf - Only one connection
Branch - Two or more connected ports
Root - Leaf or branch with no parent
The Need For Arbitration

Instead - Nodes go through Arbitration before transmitting

Arbitration grants permission to one Node
Loser waits and tries again at next gap
Arbitration Strategy

Node must observe Idle for Sub-Action Gap Time

Node transmits Request to Parent

If Node Receives Grant
Node won Arbitration
Send Data Prefix and then Packet

If Node Receives Data Prefix
Node lost Arbitration
Remove Request
Arbitration Strategy

Multiport Nodes

- Data Prefix from Parent: Pass it on to children
- Request from Child:
  - Echo Request to Parent
  - Send Data Prefix to other Children
- Grant from Parent: Pass it on to Requester

Root Node

First Request Wins
Arbitration Example

Node 0 and Node 2 want to send a packet

1394 Cable
Each arrow indicates Arbitration Signaling

- Idle
- Req Request
- Grant
- Data Prefix

Node 0 and Node 2 want to send a packet
Arbitration Example - Initial Requests

- Idle
- Request
- Grant
- Data Prefix

Diagram showing a tree structure with nodes 0, 1, 2, 3, and 4. Node 4 is the root, and the diagram illustrates the flow of requests and grants between nodes.
Arbitration Example - Parents Act

Parents shut down other children
Propagate Request upward

Node 0

Node 1

Node 2

Node 3

Root Node 4
Arbitration Example - Losers Remove Request

Node 3 sees Data Prefix from Parent
Removes Request
Propagate Data Prefix to Requester

Node 0

Node 1

Node 2

Node 3

Root Node 4

Req

Idle
Request
Grant
Data Prefix

1394

Sect 9: Arbitration

1394
Arbitration Example - Losers Remove Request

Idle

Request

Grant

Data Prefix

Root
Node 4

Node 0

Node 3

Node 1

Node 2
Arbitration Example - Winner Issues Data Prefix

Idle

Request

Grant

Data Prefix

Root Node 4

Node 0

Node 3

Node 1

Node 2

Grant

Grant

© KnowledgeTek, Inc.
rev 5.41E 15.H.31

1394

Sect 9: Arbitration
Arbitration Example - Root Removes Grant

Data Prefix Everywhere Ready For Packet

Idle
Request
Grant
Data Prefix

Root Node 4

Node 0
Node 3
Node 1
Node 2

© KnowledgeTek, Inc. rev 5.41E 15.H.31

Sect 9: Arbitration
Fairness

Node closest to root would always win. Why?

Add Fairness:

Nodes Arbitrate after Sub-Action Gap only if enabled
When Node wins Arbitration, disable
(Other Nodes now win Arbitration)
Extra Long Gap (Arbitration Reset Gap) resets all Nodes
Concatenated Subaction does not disable
Bus Manager’s Role in Arbitration

Function residing on some node
  May or may not be Root

Builds topology and speed map of the bus

Computes worst case propagation times

Determines optimum timings
  Data Prefix
  Sub-Action Gap
  Arbitration Reset

Sends timings throughout bus via PHY packet
PHY Configuration Packet Format

transmitted first

<table>
<thead>
<tr>
<th>PID</th>
<th>phy_ID</th>
<th>R</th>
<th>T</th>
<th>gap_cnt</th>
<th>0x0000h</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Logical inverse of first quadlet</td>
<td>transmitted last</td>
</tr>
</tbody>
</table>

PID  Phy Config Packet Identifier = 00b

T    If set - all nodes set their Gap Count to the indicated value

gap_cnt Gap Count Value to use if T=1

phy_ID Node ID that is to set its Force_Root bit if R=1
(Valid only if R=1)

R    If set causes the indicated node to set its Force_Root bit
## Types of Arbitration

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Normal</strong></td>
<td>Node Requests to use the bus</td>
</tr>
<tr>
<td><strong>Token</strong></td>
<td>Node closest to root arbitrates and passes grant to it's children. It will jump on with Fly-by arbitration.</td>
</tr>
<tr>
<td><strong>Fly-By</strong></td>
<td>Permits a transmitted packet to be concatenated to the end of a primary packet of which no ACK is permitted. <em>(PHY, ACK, BRIDCAST, ISOLCHRONOUS)</em></td>
</tr>
<tr>
<td><strong>ACK Accelerated</strong></td>
<td>PHY can arbitrate immediately following an observed ACK packet. <strong>Savings = subaction gap time.</strong></td>
</tr>
</tbody>
</table>
Arbitration Enhancements
ACK Accelerated Arbitration

Arbitrate after ACK, do not wait for subaction gap

Requires enable acceleration bit in Phy register=1
Arbitration Enhancements
Fly-by Arbitration

- Shall not use fly-by arbitration to concatenate an S100 packet after any packet of higher speed

- Fly-by arbitration permits an asynchronous packet to be concatenated to an ACK, or an isochronous packet to be concatenated to a cycle start on another isochronous packet.

- Requires enable acceleration bit in Phy registers=1 for asynch concatenation

- Disabled by Arbitration control until after
  - arb reset gap
  - cycle start packet
  - 2 subaction gaps
Arbitration Enhancements: Fairness Budget Registers

**ADDR 218**

<table>
<thead>
<tr>
<th>Pri-Max</th>
<th>r</th>
<th>Pri Req</th>
</tr>
</thead>
</table>

- **Pri Max**: Set by vendor to maximum number of requests node expects
- **Pri Req**: Written by bus manager to set number of allowed priority requests
  - $= 0$: Fairness as defined by 1394-1995
  - $\neq 0$: Priority requests, in excess of fairness defined in 1394-1995

**Transaction Codes Eligible**

- 0: Write request for data quadlet
- 1: Write request for data block
- 4: Read request for data quadlet
- 5: Read request for data block
- 9: Lock request
- A: Stream data block

- Applies to async subactions only
- Register to be written by bus manager only

---

© KnowledgeTek, Inc.
rev 5.41E 15.H.31
1394 Arbitration Review

1. What signals are used for normal arbitration?

2. How does arbitration ensure isochronous gets priority over asynchronous?

3. What are the arbitration enhancements? How does each work?
Section 10

Isochronous Operations
Subjects Covered

Need for Isochronous
How it works
Interaction between Asynchronous and Isochronous
Cycle master node
IRM
Applicable registers
Asynchronous streams
Asynchronous vs Isochronous

Asynchronous
Sender transmits data and receiver acknowledges receipt
If receipt was defective, then retry
Accuracy is critical, data must be delivered accurately
Used for financial, personnel data, etc.

Isochronous
Sender negotiates for bandwidth, is then guaranteed access to bus
Sender sends data on time regardless of errors
Time delivery is more critical than accuracy
Used for multimedia, movies, audio, etc.
Reduces size of buffer required in device
No ACK, No retry
Asynchronous Review

Translated means “not synchronized with time”
Guaranteed delivery
Acknowledged except broadcast
Used for data applications
  accuracy more critical than timing
Retries OK
Isochronous

Translated means "same/equal time"
Uniform in time, having equal duration
Just in time delivery system
Guaranteed timing
Recurring at regular intervals
Not acknowledged
timing more critical than accuracy

Never retry
How does Isochronous Work?

Every 125 μSec - A new cycle starts
Cycle Start Packet issued by Cycle Master (Root)
Isochronous Devices begin Arbitrating

Isochronous Devices win Arbitration
They don’t wait for a Sub-Action Gap

When all Isochronous Devices have transferred
Asynchronous nodes now see Sub-Action Gaps

Isochronous Cycle and Fairness Interval are independent

for async

may end/start @ same time
but don’t have to.
Isochronous Cycle

- Isochronous packets
- Asynchronous packets
- Asynchronous acknowledgment packets
- Isochronous gaps
- Subaction gaps
- Acknowledgment gaps
- Arbitration reset gap

125 usec
Cycle Start Packet Format

Indicates start of Isochronous Cycle
Places Cycle_Time_Data → Cycle_Time_Register
(For each Isochronous capable node)
# Cycle Start Packet Contents

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination_ID</td>
<td>Always set to FFFFh</td>
</tr>
<tr>
<td>Source_ID</td>
<td>Node ID of Cycle Master (Root)</td>
</tr>
<tr>
<td>tl</td>
<td>Transaction Label (set to 0)</td>
</tr>
<tr>
<td>tcode</td>
<td>Transaction Code (8 = Cycle Start)</td>
</tr>
<tr>
<td>pri</td>
<td>Priority, for backplane environment Set to Fh</td>
</tr>
<tr>
<td>Cycle Time Register Address</td>
<td>Always set to FFFF F000 0200h</td>
</tr>
<tr>
<td>Cycle Time Data</td>
<td>Time at transmission of packet</td>
</tr>
</tbody>
</table>
Example Isochronous Subactions

data prefix  data end  data prefix  data end  data prefix  data end

arb          packet         isoch gap        arb          packet         isoch gap

channel x  channel m  channel q

3 different talkers on one
Example Concatenated Isochronous Subactions

1 user - arbitrated non - each packet sent to a different channel
3 users - use fly by arbitration
2 users - combo of the 2 above
Isochronous Data-Block Packet Format

Note that packet is different from the asynchronous packet format

Look @ $tcode$ first
Isochronous Data-Block Packet Contents

Data_Length  Number of data Bytes in packet

tag  Isochronous Data Format Tag
Indicates Format of Data contents
Only 00 = unformatted defined
  01 = Defined in IEC 61883

channel  Used to logical connect transmitter & receiver

tcode = A  Transaction Code (A = Isochronous Data)

sy  Synchronization Code
(Application Specific)
Empty Isochronous Data Packets

1394 doesn't require empty iso packets to be sent, but UPL do.
### Maximum Payload Size for Asynchronous Packets with Data Block Payload

<table>
<thead>
<tr>
<th>Data Rate</th>
<th>Maximum Payload Size (bytes)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>S25</td>
<td>128</td>
<td>TTL backplane</td>
</tr>
<tr>
<td>S50</td>
<td>256</td>
<td>BTL and ECL backplane</td>
</tr>
<tr>
<td>S100</td>
<td>512</td>
<td>cable base rate</td>
</tr>
<tr>
<td>S200</td>
<td>1024</td>
<td></td>
</tr>
<tr>
<td>S400</td>
<td>2048</td>
<td></td>
</tr>
<tr>
<td>S800</td>
<td>4096</td>
<td>1394b</td>
</tr>
<tr>
<td>S1600</td>
<td>4096</td>
<td>1394b</td>
</tr>
</tbody>
</table>

Additional note: yestart packet can be delayed 83 ms, yestart packet must see gap, must be pulled back in, (longest it will take is 4 cycles) by not allowing any async devices back on.
Channels

Transmitter and Receiver assigned same channel

Transmitter sends Isochronous Packet each Cycle
Packet indicates channel

Receiver listens to Isochronous Packets of correct Channel
No retry, flow control, etc.

Broadcast - Zero or more Receivers
(Maybe no one listening)
Isochronous Nodes

Can transmit or receive Isochronous Packets

Uses Channel Number to identify data stream

Must have a free running 24.576 MHz clock

Must implement a Cycle_Time Register
  At CSR offset 0200h
  Counts 24.576 MHz clock ticks

Must synchronize Cycle_Time Register to Cycle_Start Packets
  Synchronization implementation dependent
  Can’t ever go backwards!

Must implement Configuration ROM
  Describe Isochronous Capabilities in Bus_Info_Block

<table>
<thead>
<tr>
<th>31h &quot;1&quot;</th>
<th>33h &quot;3&quot;</th>
<th>39h &quot;9&quot;</th>
<th>34h &quot;4&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>IR CI IB PMC Resv</td>
<td>cyc clk acc</td>
<td>max rec</td>
<td>reserved</td>
</tr>
<tr>
<td>Node vendor ID</td>
<td>chip ID high</td>
<td></td>
<td></td>
</tr>
<tr>
<td>chip ID low</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1394

© KnowledgeTek, Inc.
rev 5.41d 15.H.09

Sect 10: Isochronous Operations
Cycle Time Register

At Address NNNN FFFF F000 0200h:

- **second_cnt**: Counts 24.576 MHz clock ticks
  - Rolls over after 3071 (BFFh)
  - Synchronized to cycle_start offset

- **cycle_cnt**: Counts 125 µSec clock ticks (cycles)
  - Rolls over after 7999 (1F3Fh)

- **cycle_offset**: Current time in seconds
  - Rolls over after $2^7$ sec
  - but is kept track of in Bus Time Register

NNNN = Node ID
## Cycle Master

Sends Cycle Start Packets every 125 $\mu$Sec

Always the Root Node
- Must win Arbitration in order to send Cycle Starts on time

Implements the Cycle_Time Register
- Uses the Cycle_Offset value in Cycle_Start Packets

Implements the Bus_Time Register
- At CSR offset 0204h
- Keeps universal bus time in seconds
- Rolls over every 136 years
- Set by the Bus Manager

Indicates Cycle Master Capable in Config ROM

<table>
<thead>
<tr>
<th>31h &quot;1&quot;</th>
<th>33h &quot;3&quot;</th>
<th>39h &quot;9&quot;</th>
<th>34h &quot;4&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRCBPMC Resv</td>
<td>cyc clk acc</td>
<td>max rec</td>
<td>G R Link Speed</td>
</tr>
<tr>
<td>Node vendor ID</td>
<td>chip ID high</td>
<td>chip ID low</td>
<td></td>
</tr>
</tbody>
</table>

© KnowledgeTek, Inc.  
rev 5.41d 15.H.09  
Sect 10: Isochronous Operations
Bus Time Register

Address NNNN FFFF F000 0204h:

- `second_cnt_hi`:
  - write/read

- `second_cnt_lo`:
  - read only

- The `second_cnt` field of the Cycle_Time Register
- Counts overflows of Cycle_Time Register

NNNN = Node ID
Isochronous Resource Manager (IRM)

One node on bus provides the IRM
  Highest numbered node with IRM capabilities
  Not necessarily the Root

IRM provides registers to manage Isochronous Operations
  Bandwidth_Available Register (offset 220h)
  Channels_Available Registers (offset 224h)

IRM provides location of Bus Manager
  Bus_Manager_ID Register (offset 21Ch)

IRM has Bus Management Obligations (sect 11 & 12)
Bandwidth Available Register

Address NNNN FFFF F000 0220h:

bw_remaining Amount of Isochronous Bandwidth remaining
Measured in allocation units (au)
1 allocation unit = Quadlet time @ S1600

Maximum Bandwidth (100 μSec) = 4915 au

Register for information only, no direct control

NNNN = Node ID
Channels Available Register

Address NNNN FFFF F000 0224h:

0
32
63

channels_avail_hi
channels_avail_lo

Bit indicates corresponding channel available

1 = available
0 = owned, in use

Must be accessed through Lock (compare & swap)

Register for information only, no direct control

Channel 31 = default broadcast channel
(Automatically allocated by IRM)

NNNN = Node ID
Bus Manager ID Register

Address NNNN FFFF F000 021Ch:

<table>
<thead>
<tr>
<th>reserved</th>
<th>bmgr_id</th>
</tr>
</thead>
</table>

bmgr_id Bus Manager ID on reset = 3F

Node ID on this bus of Bus Manager

Must be accessed with a Lock (compare & swap)

NNNN = Node ID
Asynchronous Streams

Isochronous Cycle

Isochronous Format Packet
- No physical address - Uses channel number
- tcode = Ah
- 1 Quadlet Header
- No Acknowledge

Transmitted during Asynchronous Time
Arbitration & Fairness

Uses Isochronous Hardware
Does not compete for ISOCH Bandwidth

Example of use: store movie to hard drive
Asynchronous Stream Packet

Format the same as for Isochronous packets
Maximum Data Length the same as for Asynchronous packets
Why Asynchronous Streams?

Has Isochronous Advantages
Broadcast and Multicast
Channel Model - Easily Filtered

Does not consume Isochronous Bandwidth
Isochronous Operations Review

1. What is the benefit of Isochronous?
2. How does Isochronous get guaranteed bandwidth?
3. What defines Isochronous cycle?
4. What are the Isochronous resources?
   How does an Isochronous owner get resources?
Isochronous Operations Notes
Isochronous Operations Notes
Section 11

Configuration
Subjects Covered

Resets
Tree ID
Self ID
Physical Topology (Review)

Numbering and Root determined during Configuration
Logical Topology (Review)

Root

Parent

Child

Leaf - Only one connection
Branch - More than one connection
Root - Leaf or Branch with no Parent

Leaf

Branch

Root

Leaf

Leaf
Configuration Process

1. Reset
2. Tree Identify
   - Identifies Root
3. Self Identify
   - Numbers the nodes
   - Communicates speed info
4. Cycle Master Starts
   - Generates Cycle Start Packets
   - Prior Isochronous Transfers resume
5. IRM Identified
   - Identifies which Node contains Isochronous Resource Manager

Bus Management Initialization
Resets

Power Reset
- Resets all CSRs to initial values
- Reset Physical Layer (Phy)
- Initiate a Bus Reset

Bus Reset
- Arbitration Signaling
  - Sent on: Change in Topology
  - Reception of a Bus Reset
  - Power Reset

Command Reset
- Does not reset Physical Layer (Phy)
- Does not initiate Bus Reset
- Initiated by writing to the Reset Start CSR
Following Reset and Bus Initialization

Each Node knows which of its Ports are connected. **How?**

Two categories of Nodes:
- **Leaf**: Only one Port connected
- **Branch**: More than one Port connected

Nodes do not know their ID

Root is unknown
Example Topology

- Branch
  - 1
  - 2

- Leaf
  - 1
  - 2
  - 3
  - 4
  - 5

- Connected Port

- Unconnected Port
Tree Identify

Reset

Tree Identify

Identifies Root

Self Identify

Cycle Master Starts

IRM Identified

Bus Management Initialization

Sect 11: Configuration
Tree Identify Strategy

Leaf Nodes
Transmit Parent_Notify through only port
Wait to receive Child_Notify
Transmit Idle

Branch Nodes
Wait for Parent_Notify on ports
Return Child_Notify to those ports
Take that port off the list of possible parents
When only one port remains - That’s the Parent !
Transmit Parent_Notify to parent
Wait to receive Child_Notify
Transmit Idle
## Tree Identify Signaling

<table>
<thead>
<tr>
<th>Signal Transmitted</th>
<th>Arb A Tx</th>
<th>Arb B Tx</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle</td>
<td>Z</td>
<td>Z</td>
<td>Sent to indicate a gap</td>
</tr>
<tr>
<td>Request</td>
<td>Z</td>
<td>0</td>
<td>Sent to parent to request the bus</td>
</tr>
<tr>
<td>Grant</td>
<td>0</td>
<td>1</td>
<td>Sent to child when bus is granted</td>
</tr>
<tr>
<td>Parent Notify</td>
<td>0</td>
<td>Z</td>
<td>Sent to parent during Tree-ID</td>
</tr>
<tr>
<td>Data Prefix</td>
<td>0</td>
<td>1</td>
<td>Sent before data packets</td>
</tr>
<tr>
<td>Child Notify</td>
<td>1</td>
<td>Z</td>
<td>Sent to child to Ack parent notify</td>
</tr>
<tr>
<td>Ident Done</td>
<td>1</td>
<td>Z</td>
<td>Sent to parent, self-ID done</td>
</tr>
<tr>
<td>Data End</td>
<td>1</td>
<td>0</td>
<td>Sent at end of packet transmission</td>
</tr>
<tr>
<td>Bus Reset</td>
<td>1</td>
<td>1</td>
<td>Sent to force a bus reconfiguration</td>
</tr>
<tr>
<td>Tx Disable Notify</td>
<td>Z</td>
<td>1</td>
<td>Requests per node to enter suspend state</td>
</tr>
<tr>
<td>Tx Suspend</td>
<td>0</td>
<td>0</td>
<td>Requests per node to handshake Tp Bias and enter suspend state; propagate suspend to all active ports</td>
</tr>
</tbody>
</table>
Tree Identify: Leaf’s Send Parent Notify

Each leaf sends a Parent_Notify signal through its single connected port.
Tree Identify: Parents Acknowledge Children

Each parent acknowledges the parent_notify signal by returning a child_notify signal.
Tree Identify: Branches Notify Deduced Parents

BOTH branch nodes send parent_notify signals to the other at the same time. This is a conflict because they can not each be the parent of the other...

So they wait each a random amount of time (one short, one long)
Tree Identify: Branch/Root Identification

both branches withdraw their parent notify signal and after a random time, one will reassert the signal.
Tree Identify: Branch/Root Identification

The other branch will respond with child_notify. The branch with no parents is the Root.

What can a Node do that wants to be Root?
Tree Identify: Complete

All ports now know whether they are talking to a child or parent. Each node knows if it is a leaf, branch or root. All parent_notify and child_notify signals have been withdrawn.
Self Identify

1. Reset
2. Tree Identify
   - Numbers the nodes
   - Communicates speed information
3. Self Identify
4. Cycle Master Starts
5. IRM Identified
6. Bus Management Initialization

Sect 11: Configuration
Self Identify Strategy

All Nodes
- Echo any Data Prefix from parent to children
- Echo any Self-ID Packet to other ports
- Waits for Grant from parent
- Sends Grant to lowest numbered port (Data Prefix to others)
- Echoes any Self-ID Packets to the other ports
- When it receives Ident_Done, goes on to next port
- Counts Self-ID Packets to determine next node ID
- When there are no more ports, establishes Node ID
- Sends Self-ID Packet to parent
- Sends Ident_Done to parent

Root
- Same as above but issues first Grant
- Doesn't send Self-ID Packet to parent - Completes Self-ID
## Self Identify Signaling

<table>
<thead>
<tr>
<th>Signal Transmitted</th>
<th>Arb A Tx</th>
<th>Arb B Tx</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle</td>
<td>Z</td>
<td>Z</td>
<td>Sent to indicate a gap</td>
</tr>
<tr>
<td>Request</td>
<td>Z</td>
<td>0</td>
<td>Sent to parent to request the bus</td>
</tr>
<tr>
<td>Grant</td>
<td>Z</td>
<td>0</td>
<td>Sent to child when bus is granted</td>
</tr>
<tr>
<td>Parent Notify</td>
<td>0</td>
<td>Z</td>
<td>Sent to parent during Tree-ID</td>
</tr>
<tr>
<td>Data Prefix</td>
<td>0</td>
<td>1</td>
<td>Sent before data packets</td>
</tr>
<tr>
<td>Child Notify</td>
<td>1</td>
<td>Z</td>
<td>Sent to child to Ack parent notify</td>
</tr>
<tr>
<td>Ident Done</td>
<td>1</td>
<td>Z</td>
<td>Sent to parent, self-ID done</td>
</tr>
<tr>
<td>Data End</td>
<td>1</td>
<td>0</td>
<td>Sent at end of packet transmission</td>
</tr>
<tr>
<td>Bus Reset</td>
<td>1</td>
<td>1</td>
<td>Sent to force a bus reconfiguration</td>
</tr>
<tr>
<td>Tx Disable Notify</td>
<td>Z</td>
<td>1</td>
<td>Requests per node to enter suspend state</td>
</tr>
<tr>
<td>Tx Suspend</td>
<td>0</td>
<td>0</td>
<td>Requests per node to handshake Tp Bias and enter suspend state; propagate suspend to all active ports</td>
</tr>
</tbody>
</table>
Self-ID: Root Issues Grant To One Port

Root sends "grant" out its lowest number port and "data prefix" out all others.

EVERY NODE HAS SELF ID COUNTER

Self-ID count = 0
The node that received the Grant either passes it on to its children, if any, or IDs itself and returns Data_prefix followed by the Self-ID packet.

Each branch that received a Data_prefix passes that on to its children.

Self-ID count = 0
Self-ID Counter

Each Node has a Self-ID Counter
Counts each new Self-ID observed

Can’t just count packets! (Self-ID can be multiple packets)
Value of Counter determines Node ID when Node Identifies itself
Who increments each Node’s Self_ID Counter?
Self-ID: Node 0 Sends Ident.Done

When node and all its children are identified, it sends ident_done to its parent.

Parent responds with data_prefix. In this example Root Port 1 is completed.
Self-ID: Root Sends Grant To Second Port

The root now sends a Grant out its next lowest numbered port and a data_prefix out all other ports.
Self-ID: Branch Sends Grant To Lowest Port

The Branch will send Grant out its lowest child port and data_prefix out all other ports.

Self-ID count = 1
Self-ID: Node 1 Sends Self-ID Packet

The leaf that receives the Grant has no children so takes the Self-ID count and returns the Self-ID packet, with a data_prefix and data_end.

Self-ID count = 1
Self-ID: Node 1 Sends Ident.Done

Following the self-ID packets, the node will send ident_done.

The branch sends data_prefix to the newly identified node.

Self-ID count = 2

The self-ID count incremented on data_end.
Self-ID: Branch Sends Grant To Other Node

The root issues a Grant to its lowest, unidentified port (again port 2) to its child, who passes it to its lowest unidentified child port.

Node 0

Self-ID count = 2

Root
1-c 2-c

Grant

Branch
1-p
2-c 3-c

Data_prefix

node 1

Leaf
1-p
Self-ID: Node 2 Sends Self-ID Packet

The lowest node on the tree to receive this Grant will take the Self-ID count

Self-ID count = 2
After sending its self-ID packet, the node will send ident_done. Each node will increment its self-ID count when it sees data_end at the end of the Self_ID packet.
Self-ID: Finished

After all nodes have taken a Node number and passed their self-ID packets to all other nodes, the self-ID step is finished.

Self-ID count = 5

Nodes can also know the Node ID of the Root. How?

HIGHEST NODE OR SELF-ID COUNT - 1
Self-ID Packet Format

First Packet of Self-ID:

Packet #2 (if required) of Self-ID:

Packet #3 (if required) of Self-ID:
Self-ID Packet Fields

SID
Self-ID Packet Identifier = 10b

phy_ID
Physical Node ID of packet originator

L
Link Active (1 = Link and Transaction active)

gap_cnt
Current value of the Phy gap count

sp
Speed
00b = S100
01b = S200
10b = S400
11b = Reserved for future expansion

1394-1995 Delay (00b = 144nsec)
1394-2000 Obsolete
1394.1 Bridge
00b = Not a bridge
01b = unspecified
10b = Bridge - net topology unchanged
11b = Bridge - net topology changed

Contender for Bus Manager or IRM
Initiated Reset (it's my fault)

More Self-ID Packets (IF MORE PORTS)
Self-ID Packet Fields (continued)

p0, ..., p15  Port Status  
00b = Not Present  
01b = Not Active, Disabled or Suspended  
10b = Connected To Parent  
11b = Connected To Child

pwr  Power Class:

<table>
<thead>
<tr>
<th>Power Code</th>
<th>Node Power</th>
<th>Power Supplied</th>
<th>Phy Layer Power</th>
<th>Link Layer Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>001</td>
<td>Self</td>
<td>15W</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>010</td>
<td>Self</td>
<td>30W</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>011</td>
<td>Self</td>
<td>45W</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>100</td>
<td>Bus/Self</td>
<td>None</td>
<td>3W</td>
<td>None</td>
</tr>
<tr>
<td>101</td>
<td>Reserved</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>110</td>
<td>Bus</td>
<td>3W</td>
<td>3W</td>
<td>3W</td>
</tr>
<tr>
<td>111</td>
<td>Bus</td>
<td>3W</td>
<td>3W</td>
<td>7W</td>
</tr>
</tbody>
</table>
Cycle Master Starts

Reset

Tree Identify

Self Identify

Cycle Master Starts

Generates Cycle Start Packets
Prior Isochronous Transfers resume

IRM Identified

Bus Management Initialization

HIGHEST NODE #
CAPABLE OF IRM
Cycle Master Starts

After Self-Identify previous Cycle Master checks to see if it is Root
    If it is not - Turns off

Root turns on Cycle Master
    Starts issuing Cycle Start Packets

Cycle Time Register is NOT Reset
    Transfers pick up where they left off
    Devices should have 1 sec (reset & config time) of buffer
Isochronous Resource Manager Identified

Reset

Tree Identify

Self Identify

Cycle Master Starts

IRM Identified

Identifies which Node contains Isochronous Resource Manager

Bus Management Initialization
Isochronous Resource Manager (IRM) Identified

IRM is the highest number node with IRM Capabilities

IRM and other interested nodes monitor Self-ID Packets:

The last Self-ID Packet with the Contender Bit and link active set is the IRM
Configuration Review

1. Name and describe each of the three types of resets.
2. Define the signals and process of tree-ID.
3. Define the signals and process of self-ID.
4. Define selection of cycle master and IRM.
Configuration Notes
Section 12

Bus Management
Subjects Covered

Functions of:

Root
Cycle Master
IRM and Bus Manager
Gap Time determination
Bus Management

Logical functions that supervise and control bus operations
Usually implemented in software \( \text{EXCEPT FOR ROOT & CYCLE-MASTER IN HW} \)

Six standardized managers:
- Root
- Cycle Master
- Isochronous Resource Manager (IRM)
- Bus Manager
- Power Manager (covered in section 14)
- IP Manager (not covered in this course)

Each of these management functions implemented on a node
(Not necessarily the same node)
Possible Bus Configuration

Root
Cycle Master
Bus Manager
Power Manager
IRM
IP Manager
Another Possible Bus Configuration

Root
Cycle Master

IRM
IP Manager

Bus Manager
Power Manager

(NOT NECESSARILY SAME NODE)

(USUALLY) ALWAYS SAME NODE
(REQUIRES LOTS OF INTELLIGENCE)
Logical Functions (Software)

Bus Manager Capable

IRM Capable

Cycle Master Capable

Isochronous Capable

Transaction Capable

Repeater

Hardware Functions

Each level must be capable of all lower level functions

Node Capabilities

Sect 12: Configuration
Hardware Level Node Capabilities

Repeater
   All multiport nodes are repeaters
   Repeat Packets onto other ports

Transaction
   Active Link Layer (can be source and destination)
   Must implement the following Registers:
      State_Clear, State_Set, Node_IDs, Reset_Start, & Split_Timeout

Isochronous
   Must implement Cycle_Time Register
   Free RMnnning 24.576 MHz clock
   Configuration ROM in General ROM format
Cycle Master Capable

- Bus Manager Capable
- IRM Capable
- Cycle Master Capable
- Isochronous Capable
- Transaction Capable
- Repeater
Additional Hardware Responsibilities

Isochronous Capable - implements Cycle_Time Register
Implement Bus_Time Register
Originate Cycle_Start Packets every 125 μsec (8 KHz)
Must be the Root Node

Additional Logical Function Responsibilities

Indicates Cycle Master Capable in Config ROM
Automatically starts if Root Node
Should monitor for too much Isochronous traffic and turn off
Root

Additional Hardware Responsibilities
  Arbitration Resolution

Additional Logical Function Responsibilities
  Initiates Self-Identify after Tree-Identify
  Starts Cycle_Master

What if a Node without Cycle Master Capabilities ends up Root?
Isochronous Resource Manager (IRM) Capable

- Bus Manager Capable
- IRM Capable
- Cycle Master Capable
- Isochronous Capable
- Transaction Capable
- Repeater
Isochronous Resource Manager

Additional Hardware Responsibilities

Implement Bus_Manager_ID Register
Implement Bandwidth_Available register
Implement Channels_Available register(s)

Additional Logical Function Responsibilities

Recognize itself from Self_ID Packets
Verify received Self_ID Packets are good
  Node IDs in order and the Check-Quadlets good
Issue a Bus Reset if not
Initialize Bus_Manager_ID to indicate none
If no Bus_Manager 625mSec after reset - assumes limited Bus_Manager role
  Turns-on powered down links
  Sets default Gap Timing
  Implements No Cycle Master detection
Bus Manager Capable

1. Bus Manager Capable
2. IRM Capable
3. Cycle Master Capable
4. Isochronous Capable
5. Transaction Capable
6. Repeater
Bus Manager Selection

Occurs after Self-Identify process

Previous (before reset) Bus Manager
  Sets its ID in the Bus_Manager_ID Register (in IRM)
  Uses Lock Compare & Swap transaction
  If it receives 3Fh - It is now Bus Manager

All other Bus Manager Capable nodes
  Wait 125 mSec after reset
  Set their IDs in the Bus_Manager_ID Register
    Using Lock Compare & Swap transaction

Isochronous Resource Manager
  Waits 625 mSec after reset
  If no Bus Manager - declares itself to be a limited Bus Manager
Bus Manager

Additional Hardware Responsibilities
None

Additional Logical Function Responsibilities
Power on units whose link layer is off *
Builds Topology Map
Perform gap count optimization
Performs power management
Detects absence of Cycle Master
Sets Force Root bit in a Cycle Master Capable node
Issues a Bus Reset

* If enough power is available

PID phy_ID R T gap_cnt 0000h

logical inverse of first quadlet

wait 8.3 ms before sending parent notify
probably bus manager will be root
Bus Manager - Building The Topology Map

Used to determine Speed Map and Gap Timing

Self-ID Packets during Self Identify furnish the information
Node ID
Port Status for every port on that node

Port Status | Meaning
-------------|------------------
00           | Unimplemented port (no port)
01           | Port not active
10           | Connected to parent
11           | Connected to child
Building The Topology Map - Example Topology

P = Connected to Parent
C = Connected to Child

© KnowledgeTek, Inc.
rev 5.41d 15.H.09

Sect 12: Configuration
# Topology Map for Example Topology

A topology map is a graphical representation of a network's structure. In this example, the map shows the connections between nodes, with each row representing a node and its connections to other nodes.

## Table

<table>
<thead>
<tr>
<th>phy_ID</th>
<th>Port 0</th>
<th>Port 1</th>
<th>Port 2</th>
<th>Port 3</th>
<th>Port 4</th>
<th>Port 5:10</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>parent</td>
<td>uncon</td>
<td>uncon</td>
<td>uncon</td>
<td>uncon</td>
<td>no port ...</td>
</tr>
<tr>
<td>01</td>
<td>uncon</td>
<td>parent</td>
<td>no port</td>
<td>no port</td>
<td>no port</td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>parent</td>
<td>no port</td>
<td>no port</td>
<td>no port</td>
<td></td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>child</td>
<td>child</td>
<td>parent</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>child</td>
<td>child</td>
<td>no port</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- Port connections are shown as 'uncon' if not connected or 'parent'/'child' if connected to another node.
- Working backwards from the root, go out next highest port # (1).
Reconstructing Topology from the Topology Map

Bottom Up Approach
Detailed in the 1394-1995 Specification Annex E
Look for Leaves (no children)
Identify Branches above Leaves by their node numbering

Top Down Approach
Recursive Approach
Start with Root: Count = Root ID
Evaluate Node by:
  If this Node is a Leaf, evaluation of this node complete
  Test each connected Port on this node from highest to lowest
    Count = Count -1
    This port connected to Node ID Count
    Evaluate that Node
  When all connected Ports tested, evaluation of this node complete
Reconstructing Topology

Node 4
0C 1C

ROOT

Count = 3

Node 3
2P
0C 1C
Reconstructing Topology

ROOT

Node 4
0C 1C

Node 3
2P
0C 1C

Node 2
0P

Count = 2
Reconstructing Topology

ROOT

Count = 1

Node 4

0C 1C

Node 3

2P

0C 1C

Node 1

0 1P

Node 2

0P

1394

Sect 12: Configuration
Reconstructing Topology

ROOT

Node 4
0C 1C

Count = 0

Node 0
0P

Node 3
2P
0C 1C

Node 1
0 1P

Node 2
0P

Sect 12: Configuration
Bus Manager - Gap Timing

What is the timing on each gap type?
Bus Manager - Gap Timing

Maximum of 16 hops:

Root just sent a cycle start packet

Must wait sub-action gap time before next packet
   Time for gap to propagate to Node A (16 hops)
   Time for Node A to respond (40-50 nSec)
   Time for request to propagate back to root (16 hops)

Hop delay time = Cable delay + Phy delay ≈ 167 μs
   Cable delay = 5 nSec/m • 4.5m = 22.7 nSec
   Phy delay = Electronic repeater delay (see self-ID packet) 144 nS

Need to add time to prevent race condition

6.9 μSec before any one can arbitrate for bus
Optimizing Gap Timing

Use fewer hops (requires human planning)

Bus Manager optimizes gap timing based on cable topology
Computes maximum number of hops from Topology Map
If IRM acting as a limited Bus Manager: hops = 16
Sends PHY_Config Packet to configure all nodes
Issues a Bus_Reset to activate timing
Checks gap count in each node
## Setting Gap Timing

<table>
<thead>
<tr>
<th>Gap Type</th>
<th>Detection Time</th>
<th>Minimum Delay</th>
<th>Maximum Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK Isoch.</td>
<td>40ns</td>
<td>50ns</td>
<td></td>
</tr>
<tr>
<td>Subaction</td>
<td>(27 + gap_count * 16) / base rate</td>
<td>(29 + gap_count * 16) / base rate</td>
<td></td>
</tr>
<tr>
<td>Arb. Reset</td>
<td>(51 + gap_count * 32) / base rate</td>
<td>(53 + gap_count * 32) / base rate</td>
<td></td>
</tr>
</tbody>
</table>

Base rate = 98.304 MHz
### Setting Gap Timing

<table>
<thead>
<tr>
<th>1394a Gap Count</th>
<th>Max. Hops</th>
<th>1394 Subaction Gap</th>
<th>Arb. Delay</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0.6002</td>
<td>0.0814</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>0.9257</td>
<td>0.1628</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>6</td>
<td>1.2512</td>
<td>0.2441</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>8</td>
<td>1.5767</td>
<td>0.3255</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>10</td>
<td>1.9023</td>
<td>0.4069</td>
</tr>
<tr>
<td>15</td>
<td>6</td>
<td>12</td>
<td>2.2278</td>
<td>0.4883</td>
</tr>
<tr>
<td>43</td>
<td>16</td>
<td>33</td>
<td>5.6458</td>
<td>1.3428</td>
</tr>
<tr>
<td>45</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

All times in microseconds
For Your Reference:

7 Node Bus Configuration Example
Bus Configuration with Bus Manager and IRM

Configuration after Reset, Tree identify, Self-ID

Bus Manager is only device that is Bus Manager capable

IRM is the device with the highest Phy_ID and is IRM capable.

Phy ID 6
Link Off

Phy ID 2
Computer

SBP Disk 1
isc

SBP Disk 2
isc

Phy ID 1

Phy ID 0
SBP Disk
isc

Phy ID 5
DVCR
irmc
isc

Phy ID 4
Printer
irmc
isc

Phy ID 3
Camera
isc

Root Not Cycle Master Capable
Phy Packets

Configuration Packet

00  Root_ID  R  T  gap_cnt  00  00

logical inverse of first quadlet

Ping Packet

00  Dest ID  Type (0)  00  00

logical inverse of first quadlet

Link On

01  Dest ID  00  00  00

logical inverse of first quadlet
Bus Configuration with Bus Manager and IRM

Configuration after Phy_ID 6 Link layer is powered on

STEP#1
BUS MAN.
TURNS ON #6

© KnowledgeTek, Inc.
rev 5.41d 15.H.09

Sect 12: Configuration
Bus Configuration with Bus Manager and IRM

Configuration after Second Reset

- Phy ID 5: Printer
- Phy ID 4: Camera
- Phy ID 3: TV
- Phy ID 2: Computer
- Phy ID 1: SBP Disk 2
- Phy ID 0: SBP Disk 1
- Phy ID 6: DVCR

Diagram showing the bus configuration with connections and phy IDs.
1. Name and define each management responsibility on the bus.
Bus Management Notes
Section 13

Implementation
Subjects Covered

Phy duties and responsibilities
Link duties and responsibilities
Phy - Link communication
Phy registers
1394 Protocol Stack

- Configuration & Error Control
- Read, Write, Lock
- Isochronous Channels
- Transaction Layer
  - Packets

- Serial Bus Management
- Firmware

- Link Layer
  - Cycle Control
  - Packet Transmitter
  - Packet Receiver
  - Symbols (BITS)

- Physical Layer
  - Encode/Decode
  - Arbitration
  - Media Interface
  - Electrical Signals & Mechanical Interface

- Hardware
Phy Layer Functions

Implemented all in hardware
Serializes, deserializes data
Acts as repeater for multiple ports
Drives cables (differential and common mode)
Detects speed, port connected, arbitration
Provides control and clock to Link
Generates PHY packets, checks validity of incoming PHY packets
Tree ID, Self ID
Implement PHY registers
Link Layer Functions

Manage packets
Add headers
Generate and check CRC
Examine RX packets, ignore if not for this node or if bad
If packet is good, send ACK as directed by Transaction layer

Current communication between Phy & Link = 50 MHz
Different complexity depending on functions
Recognize channels assigned by application
Detect ARB Reset Gap and ACK missing
Generates or detects the start of an Isochronous cycle
Communicate Transaction layer request for TX to PHY so
PHY can arbitrate
Transaction Layer Function

Implement split timeout and busy timeout registers
Implement ACK and Retry protocols
Handle the following inbound errors
  Request data error
  Unsolicited response
  Response format error
  ACK missing
  Response retry timeout

Form Read, Write or Lock transactions based on input from the Bus Management or application
Set Transaction code

Does not manage Isochronous packets
Bus Management Functions

IRM
   Implement IRM registers
   Verify Self-ID packets
   Limited Bus Manager function

Bus Manager
   Power management
   Speed and Topology maps

Detect errors
   Exceed maximum occupancy
   Cycle too long (detected by cycle Master in 1394a)
   Duplicate Channel Detected (detected by talker on a given channel)
   Unknown Transaction code detected
   etc (see 1394-1995 8.2.3)

Implement CSRs and Configuration ROM
Implement Cycle Master, IRM and Bus Manager state machines
Standard Link/Phy Connection

Link Hardware

- Data [0:n]
- Control [0:1]
- LReq
- SCLK
- LPS
- Link on

Phy Chip

- TPA
- TPB

Direct

Electrical Isolation

Power Regulator

Power (8-30V)

Clk 25

Backplane

Direct

SCLK 25 MHz

SIDEWAYS ASSIGNED W/ BACKPLANE

SAME SIDE
# Standardized Link/Phy Connection Definitions

Data
- [0:1] for S100
- [0:3] for S200
- [0:7] for S400
  - Speed frozen at 50 MHz
  - Move more bits to go faster

Control
- Defines the meaning of the data lines

LReq
- Serial command to the Phy

SCLK
- 49.152 MHz clock

LPS
- Link power status, defined in 1394a

Backplane
- Set high if PHY is connected to backplane

CLK25
- Set high to notify link to use 24.565 MHz

Link On
- Commands Link to Power On

Direct
- Indicates Link and Phy are directly connected
**Control [0:1]**

Phy is driving

<table>
<thead>
<tr>
<th>Ctl[0:1]</th>
<th>Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>00b</td>
<td>Idle</td>
<td>No activity</td>
</tr>
<tr>
<td>01b</td>
<td>Status</td>
<td>The PHY is sending status information to the link</td>
</tr>
<tr>
<td>10b</td>
<td>Receive</td>
<td>An incoming packet is being transferred from PHY to link</td>
</tr>
<tr>
<td>11b</td>
<td>Grant</td>
<td>The link has granted the bus to send a packet</td>
</tr>
</tbody>
</table>

**Link is driving**

<table>
<thead>
<tr>
<th>Ctl[0:1]</th>
<th>Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>00b</td>
<td>Idle</td>
<td>Transmission complete, release the bus</td>
</tr>
<tr>
<td>01b</td>
<td>Hold</td>
<td>The link wishes to hold the bus</td>
</tr>
<tr>
<td>10b</td>
<td>Transmit</td>
<td>The link is sending a packet to the PHY</td>
</tr>
<tr>
<td>11b</td>
<td>--</td>
<td>Unused</td>
</tr>
</tbody>
</table>
Requests from Link to the Phy

Request Formats

- Bus Request for Cable Environment (8 bit)
- Bus Request for Backplane Environment (11 bit)
- Register Read Request (9 bit)
- Register Writes Request (17 bit)
LREQ - Bus Requests

Bus request for cable environment

<table>
<thead>
<tr>
<th>Bits</th>
<th>Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Start</td>
<td>Indicates start of transfer, always 1</td>
</tr>
<tr>
<td>1:3</td>
<td>Request type</td>
<td>Indicates which type of request is being performed</td>
</tr>
<tr>
<td>4:6</td>
<td>Request speed</td>
<td>Speed at which the PHY will be sending the packet</td>
</tr>
<tr>
<td>7</td>
<td>Stop</td>
<td>Indicates end of transfer, always 0</td>
</tr>
</tbody>
</table>

Bus request for backplane environment

<table>
<thead>
<tr>
<th>Bits</th>
<th>Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Start</td>
<td>Indicates start of transfer, always 1</td>
</tr>
<tr>
<td>1:3</td>
<td>Request type</td>
<td>Indicates which type of request is being performed</td>
</tr>
<tr>
<td>4:5</td>
<td>Request speed</td>
<td>Ignored, set to 0 in backplane environment</td>
</tr>
<tr>
<td>6:9</td>
<td>Request priority</td>
<td>Indicates priority or urgent requests (fair requests only)</td>
</tr>
<tr>
<td>10</td>
<td>Stop</td>
<td>Indicates end of transfer, always 0</td>
</tr>
</tbody>
</table>
LREQ - Arbitration Control Request

<table>
<thead>
<tr>
<th>Bits</th>
<th>Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Start</td>
<td>Indicates start of transfer, always 1</td>
</tr>
<tr>
<td>1:3</td>
<td>Request type</td>
<td>Indicates which type of request is being performed</td>
</tr>
<tr>
<td>4</td>
<td>Accelerate</td>
<td>0 = Phy may not use accelerated arbitration, 1 = Phy may</td>
</tr>
<tr>
<td>5</td>
<td>Stop bit</td>
<td>Indicates end of transfer, always 0</td>
</tr>
</tbody>
</table>
# LREQ - Register Requests

## Register read request

<table>
<thead>
<tr>
<th>Bits</th>
<th>Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Start</td>
<td>Indicates start of transfer, always 1</td>
</tr>
<tr>
<td>1:3</td>
<td>Request type</td>
<td>Indicates which type of request is being performed</td>
</tr>
<tr>
<td>4:7</td>
<td>Address</td>
<td>Internal PHY address to be read</td>
</tr>
<tr>
<td>8</td>
<td>Stop</td>
<td>Indicates end of transfer, always 0</td>
</tr>
</tbody>
</table>

## Register write request

<table>
<thead>
<tr>
<th>Bits</th>
<th>Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Start</td>
<td>Indicates start of transfer, always 1</td>
</tr>
<tr>
<td>1:3</td>
<td>Request type</td>
<td>Indicates which type of request is being performed</td>
</tr>
<tr>
<td>4:7</td>
<td>Address</td>
<td>Internal PHY address to be written</td>
</tr>
<tr>
<td>8:15</td>
<td>Data</td>
<td>Data to be written to the specific address</td>
</tr>
<tr>
<td>16</td>
<td>Stop</td>
<td>Indicates end of transfer, always 0</td>
</tr>
</tbody>
</table>

Note: Always follow a LReq with 2 stop bits, not just one
# Request Types

<table>
<thead>
<tr>
<th>LReq[1:3]</th>
<th>Name</th>
<th>Comment, Used for:</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Imm Req</td>
<td>Take control of bus upon detecting idle. ACK transfers</td>
</tr>
<tr>
<td>001</td>
<td>Iso Req</td>
<td>Arbitrate for bus, no gaps. Isonchronous Transfers</td>
</tr>
<tr>
<td>010</td>
<td>Pri Req</td>
<td>Arbitrate after subaction gap, ignore fair protocol. Cycle start</td>
</tr>
<tr>
<td>011</td>
<td>Fair Req</td>
<td>Arbitrate for bus using fair protocol. Fair and urgent transfers</td>
</tr>
<tr>
<td>100</td>
<td>Reg Read</td>
<td>Return specified register contents through status transfers</td>
</tr>
<tr>
<td>101</td>
<td>Reg Write</td>
<td>Write to the specified register</td>
</tr>
<tr>
<td>110</td>
<td>Acc Cntr</td>
<td>Disables or Enables Phy Arbitration Acceleration</td>
</tr>
<tr>
<td>111</td>
<td>---</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
## Request Speed

<table>
<thead>
<tr>
<th>LReq[4:6]</th>
<th>Data Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>S100</td>
</tr>
<tr>
<td>001</td>
<td>S1600</td>
</tr>
<tr>
<td>010</td>
<td>S200</td>
</tr>
<tr>
<td>011</td>
<td>S3200</td>
</tr>
<tr>
<td>100</td>
<td>S400</td>
</tr>
<tr>
<td>110</td>
<td>S800</td>
</tr>
<tr>
<td>Others</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
Status

Note: Bits 4:15 are transferred only in response to a register read request or to transfer the PHY's new physical ID after a bus reset

Note: Status is transferred across D[0:1]
## Legacy PHY Registers

<table>
<thead>
<tr>
<th>Bits</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Physical ID</td>
<td>Root</td>
</tr>
<tr>
<td>0001b</td>
<td>RHB</td>
<td>IBR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Gap Count</td>
<td></td>
</tr>
<tr>
<td>0010b</td>
<td>Speed</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Total Ports</td>
<td></td>
</tr>
<tr>
<td>0011b</td>
<td>A Status-0</td>
<td>B Status-0</td>
<td>CH-0</td>
<td>Con-0</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0100b</td>
<td>A Status-1</td>
<td>B Status-1</td>
<td>CH-1</td>
<td>Con-1</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total ports</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+0010b</td>
<td>A status-n</td>
<td>B Status-n</td>
<td>CH-n</td>
<td>Con-n</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+0011b</td>
<td>Environment</td>
<td></td>
<td></td>
<td></td>
<td>Register Count</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+0100b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Vendor dependent</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Extended PHY Registers

<table>
<thead>
<tr>
<th>Bits</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000b</td>
<td></td>
<td></td>
<td></td>
<td>Physical ID</td>
<td></td>
<td></td>
<td>Root</td>
<td>CPS</td>
</tr>
<tr>
<td>0001b</td>
<td>RHB</td>
<td>IBR</td>
<td></td>
<td>Gap Count</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0010b</td>
<td></td>
<td></td>
<td>Extended = 7</td>
<td>Total Ports</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0011b</td>
<td>Max speed</td>
<td>Rsv</td>
<td>Delay (1 \times 2^0 + kH)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0100b</td>
<td>L</td>
<td>C</td>
<td>Jitter</td>
<td>1394 Power class</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0101b</td>
<td>watch dog</td>
<td>ISBR</td>
<td>Loop</td>
<td>Pwr fail</td>
<td>Timeout</td>
<td>Bias</td>
<td>enab acc</td>
<td>enab multi</td>
</tr>
<tr>
<td>0110b</td>
<td></td>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0111b</td>
<td>Page select</td>
<td></td>
<td>Port select</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000b</td>
<td></td>
<td></td>
<td>Register 0 (page select)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1111b</td>
<td></td>
<td></td>
<td>Register 7 (page select)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** These registers are referred to as “Enhanced” in 1394-1995 and “Extended” in 1394a.

© KnowledgeTek, Inc.

rev 5.41d 15.H.09

Sect 13: Implementation
Every port has 8 pages
Every page has 8 registers

Register numbers

Extended Registers
## Extended PHY registers - Page 0

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000b</td>
<td>A Status</td>
<td>B Status</td>
<td>Ch</td>
<td>Con</td>
<td>Bias</td>
<td>Dis</td>
<td></td>
</tr>
<tr>
<td>1001b</td>
<td>Negotiated speed</td>
<td>Int Enable</td>
<td>Fault</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Reserved
Compliance Level
00h = not specified
01h = 1394a
02h = 1394b
03h - FFh = reserved

Vendor ID
24 bit OUI of PHY manufacturers
Review

1. At what speed does data receive transfer between Phy and Link? 
2. Where do I remember the speed of each peer port? 
3. Which function breaks SCSI write data into 1394 sized packets? 
4. How many registers are available per port on each Phy?
Implementation Notes
Section 14
Power Management
Subjects Covered

Why use power management?
Types of power nodes
Restrictions on each type of power node
Power classes
Operation of suspend, resume and disable
Purpose

Extend battery life

Protect the environment
   Reduce power consumption
   Reduce noise
   Reduce heat generation

Without significantly reducing usability or performance
Power Management

Devices which do not follow these guidelines:

Device Bay

Units within the Power Manager chassis or PC

Devices which could, but chose not to follow these optional guidelines

The rest of this presentation defines only those devices which follow these guidelines
Power Specs

Specifications
1394-1995 and 1394a-2000

Implementation Guidelines
Available from 1394 Trade Association www.1394TA.org

Part 1: Cable Power Distribution TA 1999001-1
Part 2: Suspend/Resume Implementation TA 1999001-2
Part 3: Power State Management (Scheduled availability August, 2000) (Do not use existing drafts)
Part 4: Power Distribution Management (Scheduled availability August, 2000)
Types of Devices

Power providers
   Power provided as defined in Self-ID packet

Alternate power providers
   Power provided as defined in CSR register

Power consumers
   Devices which are neither power providers nor consumers

Self Powered nodes
   Does not consume cable power except for (optionally) the PHY
Method of Power Management

One device is selected as Power Manager (PM)

Always the Bus Manager, if capable

PM verifies power availability and demands

PM creates power domains

PM turns on Link layer of Nodes where power is available

PM, under control of an application, continues to power devices on or off as required by the different power policies
## Power States

<table>
<thead>
<tr>
<th>Unit</th>
<th>Link</th>
<th>PHY</th>
<th>Port</th>
<th>Power Usage</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>D0</td>
<td>L0</td>
<td>H0</td>
<td>P0</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>D1</td>
<td>Reserved</td>
<td>Reserved</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D2</td>
<td>L2</td>
<td>H2</td>
<td>P2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D3</td>
<td>L3</td>
<td>H3</td>
<td>P3</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>

Link, PHY and Port must be in a equal or higher performance state than Unit
Power Distribution: General Rules

Power providers and consumers shall have no 4 pin connectors.

When a node changes its power class, it shall cause a bus reset.
# Power Classes

<table>
<thead>
<tr>
<th>Device Type</th>
<th>Class</th>
<th>Power Supplied</th>
<th>Power Consumed by PHY</th>
<th>Power Consumed by Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self</td>
<td>000b</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Primary Provider</td>
<td>001b</td>
<td>15 watts</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Primary Provider</td>
<td>010b</td>
<td>30 watts</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Primary Provider</td>
<td>011b</td>
<td>45 watts</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>*</td>
<td>100b</td>
<td>None</td>
<td>3 watts</td>
<td>None</td>
</tr>
<tr>
<td>Reserved</td>
<td>101b</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consumer</td>
<td>110b</td>
<td>None</td>
<td>3 watts</td>
<td>3 watts</td>
</tr>
<tr>
<td>Consumer</td>
<td>111b</td>
<td>None</td>
<td>3 watts</td>
<td>7 watts</td>
</tr>
</tbody>
</table>

* - Alternate Power Provider
- Multi-port self power node
- Consumer
Primary Power Providers Rules

Primary power providers shall not pass current from one port to another.

Voltage provided shall be regulated or unregulated:
Primary power providers - 20-30 VDC
Alternate Power Providers: Rules

Power providers shall limit the current provided on each port with a current limiting device.

Multi-port, alternate power providers may pass current through between ports.

Voltage provided shall be regulated or unregulated between 8 and 30 VDC:
- If above 20 VDC, requires per port isolation diodes
- If below 20 VDC, node should stop driving power if it detects higher voltage from the cable.
Power Consumers Rules

Power consumers shall power up with PHY only on - maximum 3 watts

Power consumers shall not be multi-port nodes

Power consumers shall wait for Link-On packet to power on Link and above
Self-Power Nodes Rules

Multi-port, self-powered, class 4 nodes shall maintain power to their PHY when main power is removed.

Self-powered nodes may have all 6 pin connectors or all 4 pin connectors but may not mix connectors.

If PHY power is maintained, current may pass between ports.
Power Down Behavior

Power Providers or Self-Powered nodes

Continue to power own PHY, maintain bus topology, pass power between ports as allowed (preferred), or

Power from PHY, maintain bus topology (second preferred), or

Discontinue powering PHY and discontinue passing power between ports (least preferred)

Power Consumers

Leaf nodes, single port

Behavior not defined in spec
Suspend/Resume: Vocabulary

Suspend

Place the 1394 interface into a low power state but subject to wake events.

Suspended port propagates suspend to all other ports in this PHY and to their connected ports.

During suspend, port must monitor TPBias and connection.

Resume

Place the 1394 interface into a high power, active state.

Connected

A port on both ends of the 1394 cable.

Disconnected

No cable connection between this port and a peer port.

Disabled

Single port is “turned off”; ports beyond it are suspended.
Port States

Active
  Capable of sending and receiving packets
  Fully operational

Suspended
  Capable of detecting:
    physical disconnection - go to disconnected state
    presence of bias
      Fault bit clear - resume normal operations
      Fault bit set - wait for software to clear Fault bit, then resume

Disabled
  Not capable of generating or detecting signals
  Appears to be unpowered PHY

Disconnected
  No cable connected, or no port at other end of cable
Vocabulary

Boundary node
A node with 2 or more ports and at least one in active state and another in suspended

Private node
Excludes other nodes on the cable from suspending or resuming any of its ports

Public node
Allows other nodes to suspend or resume its ports
Direct
All power policies are controlled by another node
Indirect
Maintains its own power policies but accepts requests from other nodes

Isolated node
No active ports
Vocabulary

Suspend Manager
Part of Power Manager node

Suspend Initiator
Of a pair of connected ports, the one issuing the suspend request

Suspend Target
Of a pair of connected ports, the one receiving the suspend request

Suspend Domain
A group of suspended ports connected by suspended connections
1394 Bus Topology

All ports on all nodes are active

Suspend manager determines that Nodes 0, 1, 2 and 5 are not being used and the power policy indicates they should be placed in low power state.
Conditions Causing Suspend

Port receives Suspend command Extended PHY packet
   OR
Port detects a properly framed RX_SUSPEND
   OR
Another port on this PHY received a RX_SUSPEND
   OR
Port detected a RX_DISABLE_NOTIFY
   OR
Port no longer detects TpBias
Sensing TpBias from the peer node was called: "Connected" in 1394-1995, or "Bias" in 1394a

1394: High Speed Serial Bus
Sect 14: Power Management
1394a PHY

1394: High Speed Serial Bus
Sect 14: Power Management
## PHY Register Map

### Contents

<table>
<thead>
<tr>
<th>Address</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0010h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0011h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0100h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0101h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0110h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0111h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1111h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Extended PHY register map for the cable environment.

### Contents

<table>
<thead>
<tr>
<th>Address</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1001h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1011h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1100h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1101h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1110h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1111h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PHY register page 0: Port Status page**

---

1394: High Speed Serial Bus

Sect 14: Power Management
Suspend Command Packet

1394: High Speed Serial Bus
Sect 14: Power Management
Fields

Type (Extended PHY packet)
8 = Command packet
A = Confirmation packet
F = Resume Node

Fields

Cmd
0 = NOP
1 = Transmit TX_DISABLE_NOTIFY then disable port
2 = Initiate Suspend
3 = Reserved
4 = Clear the port's Fault bit
5 = Enable port
6 = Resume port
7 = Reserved
Confirmation Packet

Sect 14: Power Management

1394: High Speed Serial Bus
Transmit Suspend

TX Suspend - ARB(A) = 0; ARB(B) = 0
RX Suspend - ARB(A) = 0; ARB(B) = 0
Bias Handshake

Suspend Initiator sends TX_SUSPEND to peer port

Suspend Target receives RX_SUSPEND

Suspend Target drops TpBias

Suspend Initiator drives TpBias low until internal Connect Detect circuitry becomes active

Suspend Initiator disables TpBias and places its output in high impedance state

Suspend Target detects TpBias low so places its output in high impedance state

If Bias handshake fails, and port detects TpBias after timeout, then set "Fault" bit
Transmit Suspend

Nodes 0, 1 and 2 are suspended

1394: High Speed Serial Bus
Sect 14: Power Management
Disable Node 6

Suspend Manager wants Node 6, port 1 to be disabled. Peer port on Node 5 will go to suspended state.

1394: High Speed Serial Bus
Sect 14: Power Management
Confirmation Packet

### Sect 14: Power Management

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>PHY_ID=6</td>
</tr>
<tr>
<td>00</td>
<td>Type=Ah</td>
</tr>
<tr>
<td>000</td>
<td>Port=1</td>
</tr>
<tr>
<td>000</td>
<td>cmd=1</td>
</tr>
</tbody>
</table>

Logical Inverse
TX Disable

TX Disable Notify - ARB(A) = Z; ARB(B) = 1
RX Disable Notify - ARB(A) = 1; ARB(B) = Z
Disable Node 6 Completed

Node 5 is now suspended
Resume

Sect 14: Power Management
Conditions Causing Resume

A peer port asserts TpBias
Resume command Extended PHY packet to a port
Resume Extended PHY packet to a node
making a request to this LReq
Bias Handshake

Resume Initiator will apply its TpBias
Peer node will detect TpBias and apply its own
Resume Initiator will detect TpBias from peer
Resume Initiator will issue Bus Reset
  Boundary nodes will wait 3 Reset_Detect times and issue Short Bus Reset on the active bus
  Other Resume initiators will wait 7 Reset_Detect times and issue regular bus reset

No node will transition to active state until after the reset
Resume Propagated

1394: High Speed Serial Bus
Sect 14: Power Management
Power Management Review

1. What arbitration signaling is used for suspend and resume?
2. What arbitration signaling is used for disable
3. What phy packets are used for suspend and resume?
4. How does resume operate?
Power Management Notes
Section 15

1394 Standards

Where to Get the Information
And How to Understand it
Subjects Covered

1394 standards families
Where to get more information
Specifications

- DISK
- VCR
- Descrambler
- Bulletin Board

AV/C
(1394TA)

- 61883
- Camera 1.20
- SCSI
- IP

ATA-5
IDE

- OHCI

1394-1995
Serial Bus

- 1394a
- Evolutionary

- 1394b
S800, S1600, ...
Long Distance

- 1394.1
Bridges

- 1212-1994
Register Def
- 1212R

COMMAND
LEVEL

MAPPING
DOCUMENTS

HOST BUS ADAPTERS

© KnowledgeTek, Inc.  
rev 5.41d 15.H.09

Sect 15: Specifications
## Specifications - Transport Level

<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE 1212-1991</td>
<td>Control and Status Registers</td>
</tr>
<tr>
<td>IEEE 1394-1995</td>
<td>High Speed Serial Bus, Approved 1995</td>
</tr>
<tr>
<td>IEEE P1394a</td>
<td>Evolutionary improvements to 1394, Working group</td>
</tr>
<tr>
<td>IEEE P1394b</td>
<td>800, 1600 Mbps and beyond, Working group working hard</td>
</tr>
<tr>
<td>IEEE 1394.1</td>
<td>Bridges, Working in progress</td>
</tr>
</tbody>
</table>
Specifications - Command Level

AV/C Audio-Visual Digital Interface Command Set
   1394TA spec for VCR
Camera  1394-based Digital Camera Specification
   1394TA spec for Cameras
ATA-5 IDE standard
SCSI-3 Small Computer System Interface
   Multiple standards, some approved
Reference - Where to get more information

IEEE Standards Board
WWW at http://stdsbbs.ieee.org
FTP and Gopher at stdsbbs.ieee.org
1-800-678-IEEE

Draft Standards and information:
www.3a.com
www.t10.org \textit{SCSI}
www.t13.org \textit{IDE}
www.1394ta.org (trade association)
www.phoenix.com - Phoenix Technologies, link to their library
www.apple.com/pub/standards
www.data-transit.com
www.ti.com/sc/1394
www.semiconductors.philips.com
www.microsoft.com
www.adaptec.com
www.ibm.com
www.ZAYANTE.com
Reference - Where to get more information

Reflectors:

ATA       Subscribe by sending a message of
          "Subscribe T13" to majordomo@dt.wdc.com

P1394     Subscribe by sending a message of
          "Subscribe STDC-1394" tomajordomo@majordomo.IEEE.org
1. Name the standards 1394 is built on
2. Name the 1394 transport standards
3. Name the 1394 mapping documents
4. Name the upper level protocol standards
5. Which standards body controls each?
Specifications Notes
Section 16

Audio/Video on 1394
Subjects Covered

- AV/C Command Mapping
- Plug Control Registers
- FCP
- CIP
- Camera 1.20
- Isochronous Data Transfer
Audio/Video Protocols

Connect the devices
Physical connection (camera) – (camera 1394 spec)
Plug Control Registers (AV/C) – assume physically connected, no need to connect logically.

Control the devices
Configure, Start, Play, Stop, etc.
Reading and writing CSRs (camera)
Function Control Program (AV/C)

Moving data
Isochronous data transfer
Subjects

Audio/Video Control \(^{(AVC)}\)
Connection Management
Function Control Protocol \(^{(FCP)}\)
VCR
Disk
Camera
Bulletin Boards
CA (Descrambler)
Descriptor Blocks
Isochronous data transfer
Camera 1.20
Standards

1394

CD media
MD media
Resource Allocation
Tuner
Descrambler
Bulletin Board
Disk
VCR
Enhancements
AV/C 3.0

1394TA
IEC
IEEE

61883
Camera 1.20
SBP-2
IDE
SCSI
IP

1212
1394-1995
1394a-2000
1394b

© KnowledgeTek
rev 5.41d 15.H.09

Sect 16: Audio/Video on 1394

16 - 5
61883 / 1394 Compliance

Nodes shall conform to 1394-1995 chapters 4, 6, 7 and 8

Nodes shall be IRM capable. STATE_CLEAR.cmstr bit required

Nodes shall implement plug control registers

Nodes shall implement the following registers:

- Cycle Time
- Bus Time
- Bus Manager ID
- Bandwidth Available
- Channels Available

Nodes shall implement General Configuration ROM

Unit Directory - See next page
61883 Unit Directory Entries

Unit Spec ID (Key type/key value = 12)

| 00h | A0h | 2Dh |

Unit SW Version (Key type/key value = 13)

| 01h | bit mapped CTS |

CTS = 1111b

CTS = 0000b

CTS (Command/Transaction Set) codes

| 0000b | AV/C |
| 0001b | Reserved for CAL |
| 0010b | Reserved for EHS |
| 0011 - 1101b Reserved |
| 1110b | Vendor Unique |
| 1111b | Extended CTS |
Reference - IEC 61883 (1883) Standards

<table>
<thead>
<tr>
<th>Project</th>
<th>Order Number</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>61883-1</td>
<td>100C/182/FDIS</td>
<td>General</td>
</tr>
<tr>
<td>61883-2</td>
<td>100C/183/FDIS</td>
<td>SD-DVCR Transmission</td>
</tr>
<tr>
<td>61883-3</td>
<td>100C/184/FDIS</td>
<td>HD--DVCR Transmission</td>
</tr>
<tr>
<td>61883-4</td>
<td>100C/185/FDIS</td>
<td>MPEG data Transmission</td>
</tr>
<tr>
<td>61883-5</td>
<td>100C/186/FDIS</td>
<td>SDL-DVCR Transmission</td>
</tr>
</tbody>
</table>

IEC Website - www.iec.ch
Connection Management
Isochronous Connection Management
Plug Control Registers

To establish an isochronous stream we need to set up:

Who is the Talker
Who is/are the Listener(s)
Which Channel Number will be used
How much Bandwidth is required

Plug Control Registers (PCR):

Every device has a PCR for each input or output
The PCR determines which channel it is connected to
Maximum of 32 input PCRs and 32 output PCRs per node

To connect two devices:

Every device has a PCR for each input or output
Program the talker PCR to the desired channel
Program the listener PCR to the same channel
Who does the programming has not been determined
Isochronous Connection Management
Master Plug Control Registers

AV Talkers must have one Output Master PCR
AV Listeners must have one Input Master PCR

These contain the attributes common to all PCRs

The Output Master PCR is located at offset 900h (FFFF F000 0900h)
The Output PCRs are located in the next 31 quadlets
Output PCR 0 is located at offset 904h
Output PCR 1 is located at offset 908h
and so on...

The Input Master PCR is located at offset 980h (FFFF F000 0980h)
The Input PCRs are located in the next 31 quadlets
Input PCR 0 is located at offset 984h
Input PCR 1 is located at offset 988h
and so on...
Isochronous Connection Management
Input Master PCR

<table>
<thead>
<tr>
<th>Spd</th>
<th>Reserved</th>
<th>XSpd</th>
<th>Input Plugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td></td>
<td>00</td>
<td>S100</td>
</tr>
<tr>
<td>01</td>
<td></td>
<td>01</td>
<td>S200</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>10</td>
<td>S400</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>11</td>
<td>XSpd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00</td>
<td></td>
<td>00</td>
<td>S800</td>
</tr>
<tr>
<td>01</td>
<td></td>
<td>01</td>
<td>S1600</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>10</td>
<td>S3200</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>11</td>
<td>reserved</td>
</tr>
</tbody>
</table>

Spd
00 = S100
01 = S200
10 = S400
11 = XSpd

XSpd
00 = S800
01 = S1600
10 = S3200
11 = reserved

Input Plugs
Number of Input PCRs implemented on this node
## Isochronous Connection Management

### Input Plug Control Register

<table>
<thead>
<tr>
<th>O</th>
<th>b</th>
<th>Point-to-point</th>
<th>r</th>
<th>Channel</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>b</td>
<td>Point-to-point</td>
<td>r</td>
<td>Channel</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

- **O** Online
- **b** Broadcast connection exists
- **Point-to-point** Number of point-to-point connections for this plug
- **Channel** Channel number for this plug
- **r** Reserved
# Isochronous Connection Management

## Output Master PCR

<table>
<thead>
<tr>
<th>Spd</th>
<th>Broadcast Base</th>
<th>Reserved</th>
<th>XSpd</th>
<th>Output Plugs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Spd**

- 00 = S100
- 01 = S200
- 10 = S400
- 11 = XSpd

**XSpd**

- 00 = S800
- 01 = S1600
- 10 = S3200
- 11 = reserved

**Broadcast Base**

Used to determine base isochronous channel number for broadcasts

**Output Plugs**

Number of Output PCRs implemented on this node
# Isochronous Connection Management

## Output Plug Control Register

<table>
<thead>
<tr>
<th>Ob</th>
<th>Point-to-point</th>
<th>XSpd</th>
<th>Channel</th>
<th>Spd</th>
<th>Overhead</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>Online</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>Broadcast connection exists</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Point-to-point</td>
<td>Number of point-to-point connections for this plug</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spd</td>
<td>Speed (0 = S100, 1 = S200, 2 = S400, 3 = XSpd)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XSpd</td>
<td>Speed (0 = S800, 1 = S1600, 2 = S3200, 3 = Reserved)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Channel</td>
<td>Channel number for this plug</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overhead</td>
<td>Allocation units of overhead</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Payload</td>
<td>Maximum data quadlets in a single isochronous packet (0 = 1024 quadlets)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Audio Video/Control
Function Control Protocol

AV/C
FCP
Command Packets

Command/Response format defined by IEC 61883
Device specific commands defined by 1394TA
A/V command packets are transmitted as the data portion of the 1394 packet

Types of commands: Control, Status, Notify and Inquiry

Control  Set a feature to a certain value
Status    Tell me the current setting of a feature
Notify    Tell me if a certain event occurs
Inquiry   Tell me if you support this feature and/or parameters
FCP Command/Response Buffer

- **FFFF F000 0000**: 1212 defined CSRs
- **FFFF F000 0400**: Configuration ROM
- **FFFF F000 0800**: PCR
- **FFFF F000 0900**: FCP CMD Buffer
- **FFFF F000 0B00**: FCP RSP Buffer
- **FFFF F000 0D00**: 
- **FFFF F000 0F00**: 

Sect 16: Audio/Video on 1394
**1394 Frame with FCP Frame**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination ID</td>
<td></td>
</tr>
<tr>
<td>tl</td>
<td></td>
</tr>
<tr>
<td>rt</td>
<td></td>
</tr>
<tr>
<td>tcode</td>
<td></td>
</tr>
<tr>
<td>pri</td>
<td></td>
</tr>
<tr>
<td>Source ID</td>
<td></td>
</tr>
<tr>
<td>Memory address</td>
<td></td>
</tr>
<tr>
<td>Data Length</td>
<td>0</td>
</tr>
<tr>
<td>Header CRC</td>
<td></td>
</tr>
<tr>
<td>FCP Frame</td>
<td></td>
</tr>
<tr>
<td>CTS=0</td>
<td></td>
</tr>
<tr>
<td>FCP Data</td>
<td></td>
</tr>
<tr>
<td>Pad with Zeros if necessary</td>
<td></td>
</tr>
<tr>
<td>Data CRC</td>
<td></td>
</tr>
</tbody>
</table>
1394 Frame with FCP Frame - Definitions

**tCode**
- Write Request for Quadlet = 0
- Write Request for a Block = 1

**Address**
- FFFF F000 0B00 for Command
- FFFF F000 0D00 for Response

**CTS**
- Command/Transaction Set
  - 0000b: AV/C
  - 0001b: Reserved for CAL
  - 0010b: Reserved for EHS
  - 0011 - 1101b: Reserved
  - 1110b: Vendor Unique
  - 1111b: Extended CTS

**FCP Data**
- See the rest of this section
AV/C Command Frame

<table>
<thead>
<tr>
<th>CTS</th>
<th>ctype</th>
<th>SU type</th>
<th>SU ID</th>
<th>OpCode</th>
<th>Operand(0)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Operand(1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Operand(2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Operand(3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Operand(4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Operands(n) Pad with Zeros if necessary

CTS Command Transaction Set
ctype Command type
SU type Sub-Unit type (VCR, Camera, Disk, Bulletin Board, etc.)
SU ID Sub-Unit ID (Sequential number on this bus)
OpCode Operation requested
Operand Parameters of OpCode
AV/C Response Frame

<table>
<thead>
<tr>
<th>Transmitted first</th>
<th>Transmitted last</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTS</td>
<td>resp</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Operand(1)</td>
<td>Operand(2)</td>
</tr>
</tbody>
</table>

... Pad with Zeros if necessary

- **CTS**: Command Transaction Set
- **resp**: Response code
- **SU type**: Sub-Unit type (VCR, Camera, Disk, Bulletin Board, etc.)
- **SU ID**: Sub-Unit ID (Sequential number on this bus)
- **OpCode**: Operation requested
- **Operand**: Parameters of OpCode

© KnowledgeTek
rev 5.41d 15.H.09

Sect 16: Audio/Video on 1394
### AV/C Command Frame - Definitions

<table>
<thead>
<tr>
<th>ctype</th>
<th>Command type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Control</td>
</tr>
<tr>
<td>1</td>
<td>Status</td>
</tr>
<tr>
<td>2</td>
<td>Inquiry</td>
</tr>
<tr>
<td>3</td>
<td>Notify</td>
</tr>
<tr>
<td>4-7</td>
<td>Reserved</td>
</tr>
<tr>
<td>8-F</td>
<td>Reserved for response code</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>resp</th>
<th>Response code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-7</td>
<td>Reserved for command type</td>
</tr>
<tr>
<td>8</td>
<td>Function not implemented</td>
</tr>
<tr>
<td>9</td>
<td>Accepted</td>
</tr>
<tr>
<td>A</td>
<td>Rejected</td>
</tr>
<tr>
<td>B</td>
<td>In transition</td>
</tr>
<tr>
<td>C</td>
<td>Implemented/Stable</td>
</tr>
<tr>
<td>D</td>
<td>Changed</td>
</tr>
<tr>
<td>E</td>
<td>Reserved</td>
</tr>
<tr>
<td>F</td>
<td>Interim</td>
</tr>
</tbody>
</table>
## AV/C Command Frame - Definitions

<table>
<thead>
<tr>
<th>SU type</th>
<th>SubUnit type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Video monitor</td>
</tr>
<tr>
<td>1-2</td>
<td>1-2</td>
<td>Reserved</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Disc recorder or player</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>Tape recorder or player</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Tuner</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>Reserved</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>Video camera</td>
</tr>
<tr>
<td>8-1B</td>
<td>8-1B</td>
<td>Reserved</td>
</tr>
<tr>
<td>1C</td>
<td>1C</td>
<td>Vendor unique</td>
</tr>
<tr>
<td>1D</td>
<td>1D</td>
<td>Reserved</td>
</tr>
<tr>
<td>1E</td>
<td>1E</td>
<td>Extended to next byte</td>
</tr>
<tr>
<td>1F</td>
<td>1F</td>
<td>Unit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SU ID</th>
<th>SubUnit ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-4</td>
<td>0-4</td>
<td>Instance number</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Extended to next byte</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>Reserved</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>Ignored</td>
</tr>
</tbody>
</table>
AV/C Command Frame - Definitions

Examples:

Second video camera

```
0 0 1 1 1 1
```
Subunit Type = 7  Subunit ID = 1

Fifth VCR

```
0 0 1 0 0 0
```
Subunit Type = 4  Subunit ID = 4

Subunit type of 1F and Subunit ID of 7 is defined to mean the entire unit, not a subunit.
## AV/C Command Frame - Definitions

### OpCode Groupings

<table>
<thead>
<tr>
<th>Range</th>
<th>Addressing Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - F</td>
<td>Units and Subunits</td>
</tr>
<tr>
<td>10 - 3F</td>
<td>Units</td>
</tr>
<tr>
<td>40 - 7F</td>
<td>Subunits</td>
</tr>
<tr>
<td>80 - 9F</td>
<td>Reserved</td>
</tr>
<tr>
<td>A0 - BF</td>
<td>Units and Subunits</td>
</tr>
<tr>
<td>C0 - DF</td>
<td>Subunits</td>
</tr>
<tr>
<td>E0 - FF</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
## Unit Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>OpCode</th>
<th>Support Level (by ctype)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHANNEL USAGE</td>
<td>12h</td>
<td>- R R</td>
<td>Report information on IEEE 1394 isochronous channel usage</td>
</tr>
<tr>
<td>CONNECT</td>
<td>24h</td>
<td>O O R</td>
<td>Establish connections for unspecified streams between plugs and subunits</td>
</tr>
<tr>
<td>CONNECT AV</td>
<td>20h</td>
<td>O O O</td>
<td>Establish AV connections between plugs and subunits</td>
</tr>
<tr>
<td>CONNECTIONS</td>
<td>22h</td>
<td>- O -</td>
<td>Report connection status</td>
</tr>
<tr>
<td>DIGITAL INPUT</td>
<td>11h</td>
<td>O O -</td>
<td>Make or break broadcast Serial Bus connections</td>
</tr>
<tr>
<td>DIGITAL OUTPUT</td>
<td>10h</td>
<td>O O -</td>
<td></td>
</tr>
<tr>
<td>DISCONNECT</td>
<td>25h</td>
<td>O - -</td>
<td>Break unspecified stream connections between plugs and subunits</td>
</tr>
<tr>
<td>DISCONNECT AV</td>
<td>21h</td>
<td>O - -</td>
<td>Break AV connections between plugs and subunits</td>
</tr>
<tr>
<td>INPUT/OUTPUT PLUG</td>
<td>19h/18h</td>
<td>O O -</td>
<td>Set or report signal formats for IEEE 1394.0 plugs</td>
</tr>
<tr>
<td>SIGNAL FORMAT</td>
<td>18h</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUBUNIT INFO</td>
<td>31h</td>
<td>- M -</td>
<td>Report subunit information</td>
</tr>
<tr>
<td>UNIT INFO</td>
<td>30h</td>
<td>- M -</td>
<td>Report unit information</td>
</tr>
</tbody>
</table>
## Common Subunit Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Opcode</th>
<th>Support Level (by ctype)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Descriptor</td>
<td>0Ch</td>
<td>O - -</td>
<td>Create a new descriptor structure</td>
</tr>
<tr>
<td>Open Info Block</td>
<td>05h</td>
<td>O O -</td>
<td>Gain access to the specified info block</td>
</tr>
<tr>
<td>Read Info Block</td>
<td>06h</td>
<td>O - -</td>
<td>Read the specified info block</td>
</tr>
<tr>
<td>Write Info Block</td>
<td>07h</td>
<td>O - -</td>
<td>Write data into a specified info block</td>
</tr>
<tr>
<td>Open Descriptor</td>
<td>08h</td>
<td>O O O</td>
<td>Gain rights to access descriptor</td>
</tr>
<tr>
<td>Read Descriptor</td>
<td>09h</td>
<td>O - -</td>
<td>Read data from the descriptor</td>
</tr>
<tr>
<td>Write Descriptor</td>
<td>0Ah</td>
<td>O O -</td>
<td>Write data to the descriptor</td>
</tr>
<tr>
<td>Search Descriptor</td>
<td>0Bh</td>
<td>O - -</td>
<td>Search descriptor for specified data pattern</td>
</tr>
<tr>
<td>Object Number Select</td>
<td>0Dh</td>
<td>O O O</td>
<td>Select one or more objects</td>
</tr>
<tr>
<td>Power</td>
<td>B2h</td>
<td>O O R</td>
<td>Control power state</td>
</tr>
<tr>
<td>Reserve</td>
<td>01h</td>
<td>O O R</td>
<td>Acquire or release exclusive control of a target</td>
</tr>
<tr>
<td>Plug Info</td>
<td>02h</td>
<td>- O -</td>
<td>Information about serial bus &amp; external plugs</td>
</tr>
<tr>
<td>Vendor Dependent</td>
<td>00h</td>
<td>V V V</td>
<td>Vendor dependent commands</td>
</tr>
</tbody>
</table>

| R | Recommended                  |
| O | Optional                     |
| M | Mandatory                    |
| V | Vendor Unique                |
| - | Not defined                  |

© KnowledgeTek
rev 5.41d  15.H.09

1394

Sect 16: Audio/Video on 1394

16 - 29
## VCR Subunit Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>OpCode</th>
<th>Support Level (by ctype)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANALOG AUDIO OUTPUT MODE</td>
<td>70h</td>
<td>O O -</td>
<td>Control analog audio signal</td>
</tr>
<tr>
<td>Area Mode</td>
<td>72h</td>
<td>O O -</td>
<td>Specify where on media to record input</td>
</tr>
<tr>
<td>Absolute Track Number</td>
<td>52h</td>
<td>* * -</td>
<td>Report tape position</td>
</tr>
<tr>
<td>AUDIO MODE</td>
<td>71h</td>
<td>O O -</td>
<td>Control audio signal recording mode</td>
</tr>
<tr>
<td>BACKWARD</td>
<td>56h</td>
<td>R -</td>
<td>Search for a tape position</td>
</tr>
<tr>
<td>Binary Mode</td>
<td>5Ah</td>
<td>O O O</td>
<td>Read/Write binary group data</td>
</tr>
<tr>
<td>EDIT MODE</td>
<td>40h</td>
<td>O O</td>
<td>Control editing operations prior to an anticipated playback or record command</td>
</tr>
<tr>
<td>FORWARD</td>
<td>55h</td>
<td>R -</td>
<td>Search for a tape position</td>
</tr>
<tr>
<td>INPUT SIGNAL MODE</td>
<td>79h</td>
<td>O M -</td>
<td>Control input signal mode</td>
</tr>
<tr>
<td>LOAD MEDIUM</td>
<td>C1h</td>
<td>O -</td>
<td>Control eject, open and close</td>
</tr>
<tr>
<td>Marker</td>
<td>CAh</td>
<td>R R O</td>
<td>Record or erase marker signal</td>
</tr>
<tr>
<td>MEDIUM INFO</td>
<td>DAh</td>
<td>- R</td>
<td>Report medium information</td>
</tr>
<tr>
<td>OPEN MIC</td>
<td>60h</td>
<td>* R -</td>
<td>Open or close MIC</td>
</tr>
<tr>
<td>OUTPUT SIGNAL MODE</td>
<td>78h</td>
<td>O M -</td>
<td>Control the output signal mode</td>
</tr>
</tbody>
</table>
## VCR Subunit Commands (continued)

<table>
<thead>
<tr>
<th>Command</th>
<th>OpCode</th>
<th>Support Level</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLAY</td>
<td>C3h</td>
<td>* - -</td>
<td>Control the playback mechanism</td>
</tr>
<tr>
<td>PRESET</td>
<td>45h</td>
<td>O O -</td>
<td>Establish operating parameters for the transport mechanism</td>
</tr>
<tr>
<td>READ MIC</td>
<td>61h</td>
<td>R - -</td>
<td>Read data from MIC</td>
</tr>
<tr>
<td>RECORD</td>
<td>C2h</td>
<td>* - -</td>
<td>Control the recording mode of the transport mechanism</td>
</tr>
<tr>
<td>RECORDING DATE</td>
<td>53h</td>
<td>O O -</td>
<td>Report recording date</td>
</tr>
<tr>
<td>RECORDING SPEED</td>
<td>DBh</td>
<td>O O -</td>
<td>Control recording speed</td>
</tr>
<tr>
<td>RECORDING TIME</td>
<td>54h</td>
<td>- O -</td>
<td>Report recording time</td>
</tr>
<tr>
<td>Relative Time Counter</td>
<td>57h</td>
<td>R R -</td>
<td>Search, Inquiry or clear the RTC</td>
</tr>
<tr>
<td>SEARCH MODE</td>
<td>50h</td>
<td>- R O</td>
<td>Report transport mechanism search mode status</td>
</tr>
<tr>
<td>SMPTE/EBU Recording Time</td>
<td>5Ch</td>
<td>O O O</td>
<td>Reads/Writes present recording time</td>
</tr>
<tr>
<td>SMPTE/EBU Time Code</td>
<td>59h</td>
<td>O O O</td>
<td>Reads/Writes present recording time code</td>
</tr>
<tr>
<td>Tape playback format</td>
<td>D3h</td>
<td>* * -</td>
<td>Specifies the digital playback format</td>
</tr>
<tr>
<td>Tape Recording format</td>
<td>D2h</td>
<td>* * -</td>
<td>Specifies the digital record format</td>
</tr>
</tbody>
</table>
# VCR Subunit Commands (continued)

<table>
<thead>
<tr>
<th>Command</th>
<th>OpCode</th>
<th>Support Level</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME CODE</td>
<td>51h</td>
<td>RM</td>
<td>Search or inquire about specified medium location</td>
</tr>
<tr>
<td>TRANSPORT STATE</td>
<td>DOh</td>
<td>M O</td>
<td>Report current state of transport mechanism</td>
</tr>
<tr>
<td>WIND</td>
<td>C4h</td>
<td>*</td>
<td>Control transport mechanism motion when not in playback or record</td>
</tr>
<tr>
<td>WRITE MIC</td>
<td>62h</td>
<td>O</td>
<td>Store data in MIC</td>
</tr>
</tbody>
</table>
## Playback Modes

<table>
<thead>
<tr>
<th>Playback Mode</th>
<th>Operand</th>
<th>Support Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEXT FRAME</td>
<td>30h</td>
<td>R</td>
<td>Playback the next sequential frame or field</td>
</tr>
<tr>
<td>SLOWEST FORWARD</td>
<td>31h</td>
<td>R</td>
<td>Playback at a special effect speed</td>
</tr>
<tr>
<td>SLOW FORWARD 6</td>
<td>32h</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>SLOW FORWARD 5</td>
<td>33h</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>SLOW FORWARD 4</td>
<td>34h</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>SLOW FORWARD 3</td>
<td>35h</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>SLOW FORWARD 2</td>
<td>36h</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>SLOW FORWARD 1</td>
<td>37h</td>
<td>O</td>
<td>Playback at normal speed</td>
</tr>
<tr>
<td>X1</td>
<td>38h</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>FAST FORWARD 1-7</td>
<td>39h-3Fh</td>
<td></td>
<td>Playback at a special effect speed</td>
</tr>
<tr>
<td>PREVIOUS FRAME</td>
<td>40h</td>
<td>R</td>
<td>Play the previous frame or field</td>
</tr>
<tr>
<td>SLOW REVERSE 1-7</td>
<td>41-47h</td>
<td>R</td>
<td>Playback at a special effect speed</td>
</tr>
<tr>
<td>X1 REVERSE</td>
<td>48h</td>
<td>O</td>
<td>Playback at normal speed in reverse</td>
</tr>
<tr>
<td>FAST REVERSE 1-7</td>
<td>49h-4Fh</td>
<td></td>
<td>Playback at a special effect speed</td>
</tr>
<tr>
<td>REVERSE</td>
<td>65h</td>
<td>O</td>
<td>Playback at normal speed in reverse</td>
</tr>
<tr>
<td>REVERSE PAUSE</td>
<td>6Dh</td>
<td>O</td>
<td>Pause in reverse playback</td>
</tr>
<tr>
<td>FORWARD</td>
<td>75h</td>
<td>O</td>
<td>Playback at normal speed</td>
</tr>
<tr>
<td>FORWARD PAUSE</td>
<td>7Dh</td>
<td>M</td>
<td>Pause in playback</td>
</tr>
</tbody>
</table>

These are the operands of the Play opcode (C3h) for the VCR and disk. Support levels are listed for the VCR, and are different for disk.
## Camera Subunit Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>OpCode</th>
<th>Support Level (by ctype)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE Mode</td>
<td>40h</td>
<td>* M</td>
<td>Control automatic exposure mode</td>
</tr>
<tr>
<td>AE shift</td>
<td>42h</td>
<td>O O</td>
<td>Control the amount of light</td>
</tr>
<tr>
<td>AF mode</td>
<td>C8h</td>
<td>M M</td>
<td>Control automatic focusing mode</td>
</tr>
<tr>
<td>AGC gain</td>
<td>45h</td>
<td>M M</td>
<td>Control AGC gain</td>
</tr>
<tr>
<td>AGC maximum gain</td>
<td>74h</td>
<td>- * -</td>
<td>Report maximum value of AGC gain</td>
</tr>
<tr>
<td>CCD scan mode</td>
<td>7Ah</td>
<td>R R</td>
<td>Control scan mode of imaging devices</td>
</tr>
<tr>
<td>Contrast</td>
<td>55h</td>
<td>O O</td>
<td>Control contrast</td>
</tr>
<tr>
<td>Digital zoom</td>
<td>60h</td>
<td>M M</td>
<td>Control digital zoom</td>
</tr>
<tr>
<td>Digital zoom max</td>
<td>61h</td>
<td>O O</td>
<td>Control to limit of max. magnification of zoom</td>
</tr>
<tr>
<td>Flash</td>
<td>48h</td>
<td>O O</td>
<td>Report status of electronic flash</td>
</tr>
<tr>
<td>Focal length</td>
<td>C3h</td>
<td>O O</td>
<td>Control or report focal length</td>
</tr>
<tr>
<td>Focus</td>
<td>C1h</td>
<td>* M</td>
<td>Control motion of focussing lens group</td>
</tr>
<tr>
<td>Focussing position</td>
<td>C2h</td>
<td>O O</td>
<td>Control position of focussing lens group</td>
</tr>
<tr>
<td>Freeze</td>
<td>62h</td>
<td>R R</td>
<td>Control to still the picture</td>
</tr>
<tr>
<td>Gamma</td>
<td>52h</td>
<td>O O</td>
<td>Control gamma correction</td>
</tr>
<tr>
<td>Hue</td>
<td>5Ch</td>
<td>O O</td>
<td>Control hue</td>
</tr>
<tr>
<td>Image stabilizer</td>
<td>DCh</td>
<td>O O</td>
<td>Control image stabilizer</td>
</tr>
<tr>
<td>Iris</td>
<td>43h</td>
<td>* M</td>
<td>Control diaphragm of the optical system</td>
</tr>
<tr>
<td>Iris range</td>
<td>75h</td>
<td>- * -</td>
<td>Report maximum/minimum F.No of diaphragm</td>
</tr>
</tbody>
</table>

© KnowledgeTek
rev 5.41d   15.H.09

Sect 16: Audio/Video on 1394
# Camera Subunit Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>OpCode</th>
<th>C</th>
<th>S</th>
<th>N</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ND filter</td>
<td>CBh</td>
<td>O</td>
<td>O</td>
<td></td>
<td>Control neutral density filter</td>
</tr>
<tr>
<td>Range</td>
<td>70h</td>
<td></td>
<td></td>
<td></td>
<td>Control maximum/minimum value</td>
</tr>
<tr>
<td>Reverse</td>
<td>64h</td>
<td>O</td>
<td>O</td>
<td></td>
<td>Control to reverse picture state between + and -</td>
</tr>
<tr>
<td>Pan</td>
<td>DAh</td>
<td>O</td>
<td>O</td>
<td></td>
<td>Control panhead in a panning direction</td>
</tr>
<tr>
<td>Saturation</td>
<td>5Bh</td>
<td>O</td>
<td>O</td>
<td></td>
<td>Control saturation of color</td>
</tr>
<tr>
<td>Setup level</td>
<td>54h</td>
<td>O</td>
<td>O</td>
<td></td>
<td>Control setup level</td>
</tr>
<tr>
<td>Sharpness</td>
<td>56h</td>
<td>O</td>
<td>O</td>
<td></td>
<td>Control sharpness</td>
</tr>
<tr>
<td>Shutter speed</td>
<td>44h</td>
<td>R</td>
<td>R</td>
<td></td>
<td>Control shutter speed</td>
</tr>
<tr>
<td>Support level profile</td>
<td>72h</td>
<td></td>
<td>M</td>
<td></td>
<td>Control support level of camera subunit</td>
</tr>
<tr>
<td>Tilt</td>
<td>DBh</td>
<td>O</td>
<td>O</td>
<td></td>
<td>Control panhead in a tilting direction</td>
</tr>
<tr>
<td>Video light</td>
<td>49h</td>
<td>O</td>
<td>O</td>
<td></td>
<td>Control video light</td>
</tr>
<tr>
<td>White balance</td>
<td>5Dh</td>
<td>R</td>
<td>R</td>
<td></td>
<td>Control white balance</td>
</tr>
<tr>
<td>Zoom</td>
<td>C4h</td>
<td>*</td>
<td>M</td>
<td></td>
<td>Control motion of zoom lens group</td>
</tr>
</tbody>
</table>

© KnowledgeTek

Sect 16: Audio/Video on 1394
## Disk Subunit Commands

<table>
<thead>
<tr>
<th>Category</th>
<th>Command</th>
<th>OpCode</th>
<th>Def ctypes</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Accept/Reject editing changes</td>
<td>D2h</td>
<td>X</td>
<td>Commit or reject in-progress editing changes</td>
</tr>
<tr>
<td>A</td>
<td>Associate list with plug</td>
<td>D3h</td>
<td>X</td>
<td>Associate a list with a source or destination plug</td>
</tr>
<tr>
<td>A</td>
<td>Auto update on/off</td>
<td>D4h</td>
<td>X</td>
<td>Enable/disable automatic editing change</td>
</tr>
<tr>
<td>B</td>
<td>Combine</td>
<td>41h</td>
<td>X</td>
<td>Concatenate two tracks into a single track</td>
</tr>
<tr>
<td>A</td>
<td>Configure</td>
<td>D1h</td>
<td>X</td>
<td>Prepare the subunit for recording or playback</td>
</tr>
<tr>
<td>C</td>
<td>Disc status</td>
<td>D0h</td>
<td>-</td>
<td>Request notification of status changes</td>
</tr>
<tr>
<td>B</td>
<td>Divide</td>
<td>42h</td>
<td>X</td>
<td>Separate a specified track into two blocks</td>
</tr>
<tr>
<td>B</td>
<td>Erase</td>
<td>40h</td>
<td>X</td>
<td>Erase the disk, specified track or specified portion</td>
</tr>
<tr>
<td>A</td>
<td>Import/Export medium</td>
<td>C1h</td>
<td>X</td>
<td>Put the disc into or remove it from the drive</td>
</tr>
<tr>
<td>A</td>
<td>Monitor</td>
<td>C6h</td>
<td>X</td>
<td>Listen to what is being recorded</td>
</tr>
<tr>
<td>B</td>
<td>Move</td>
<td>43h</td>
<td>X</td>
<td>Move a track to a different logical location</td>
</tr>
<tr>
<td>A</td>
<td>Increment object position no.</td>
<td>51h</td>
<td>X</td>
<td>Divide a track while recording</td>
</tr>
<tr>
<td>C</td>
<td>Object number select</td>
<td>0Dh</td>
<td>X</td>
<td>Select one or more objects for transmission</td>
</tr>
<tr>
<td>A</td>
<td>Play</td>
<td>C3h</td>
<td>X</td>
<td>Begin playing the disk (immediate response)</td>
</tr>
<tr>
<td>A</td>
<td>Record</td>
<td>C2h</td>
<td>X</td>
<td>Record a streaming object (audio track, etc.)</td>
</tr>
<tr>
<td>A</td>
<td>Record object</td>
<td>56h</td>
<td>X</td>
<td>Record a non-streaming object (still image, etc.)</td>
</tr>
<tr>
<td>A</td>
<td>Rehearsal</td>
<td>C7h</td>
<td>X</td>
<td>Playback a few positions continuously</td>
</tr>
<tr>
<td>A</td>
<td>Search</td>
<td>50h</td>
<td>X</td>
<td>Perform a relative or absolute search for the loc.</td>
</tr>
<tr>
<td>A</td>
<td>Stop</td>
<td>C5h</td>
<td>X</td>
<td>Stop the current operation</td>
</tr>
<tr>
<td>B</td>
<td>Undo</td>
<td>44h</td>
<td>X</td>
<td>Undo the most recent editing operation(s)</td>
</tr>
</tbody>
</table>
Disk Command Categories

A  Commands that affect subunit plugs
B  Commands that affect the subunit in general
C  Miscellaneous commands
X  Defined in media documents
   Not defined for this command type
Bulletin Boards

Subunits in a unit that allow other units/subunits to share information with the base unit.

Only type defined is '01' Resource Allocation.

User can schedule resources for future use thus allowing 1394 to avoid scheduling conflicts.
Command (Opcodes)

CA enable - tells CA subunit to begin descrambling

CA entitlement - controller queries subunit to see if user has entitlement for certain services

Security - validation between controller and CA subunit
Descriptor Blocks
Descriptor Blocks

Define configuration or status of a subunit

Created when manufactured or with ‘Create Descriptor’ command

Opened with ‘Open Descriptor’ command

Read with ‘Read Descriptor’ or ‘Read Info Block’ commands

Written with ‘Write Descriptor’ or ‘Write Info Block’ commands

Usually structured as hierarchical list of lists

Each block contains ID, length, pointers to children and data, as applicable.
Descriptor Blocks

Defined in:

AV/C Digital Interface Command Set, General Specification
Version 3.0

Enhancements to AV/C General Specification 3.0
Version 1.0

AV/C Disc Subunit General Specification
Version 1.0

AV/C Disc Media Type Specification - MD Audio
Version 1.0

Each subsequent specification is a further enhancement or subunit specific refinement of the former.
# Open Descriptor

## FCP Command

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Open Descriptor (08h)</th>
</tr>
</thead>
</table>

### Operands

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;</td>
<td>Descriptor ID (MSB)</td>
</tr>
<tr>
<td>&quot;</td>
<td>Descriptor ID (LSB)</td>
</tr>
<tr>
<td>&quot;</td>
<td>Subfunction</td>
</tr>
<tr>
<td>&quot;</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

## Which descriptor and how much

<table>
<thead>
<tr>
<th>Offset</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Descriptor Type</td>
</tr>
<tr>
<td>01</td>
<td>Descriptor Type</td>
</tr>
<tr>
<td></td>
<td>Specific info</td>
</tr>
</tbody>
</table>

## Descriptor block

<table>
<thead>
<tr>
<th>Offset</th>
<th>Descriptor Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Subunit Identifier Descriptor</td>
</tr>
<tr>
<td>10</td>
<td>Object List Descriptor - specified by list ID</td>
</tr>
<tr>
<td>11</td>
<td>Object List Descriptor - specified by list type</td>
</tr>
<tr>
<td>20</td>
<td>Object Entry Descriptor - specified by object position</td>
</tr>
<tr>
<td>21</td>
<td>Object Entry Descriptor - specified by object ID</td>
</tr>
<tr>
<td>80</td>
<td>Disk Status Descriptor</td>
</tr>
</tbody>
</table>

---

© KnowledgeTek

rev 5.41d  15.H.09

Sect 16b: Audio/Video on 1394
Response to Read Disk Status Descriptor

<table>
<thead>
<tr>
<th>Full Descriptor read</th>
<th>Information Block read</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk Status</td>
<td>DISK STATUS</td>
</tr>
<tr>
<td>Reference Method</td>
<td>DISK STATUS</td>
</tr>
<tr>
<td>80h</td>
<td>DISK STATUS</td>
</tr>
<tr>
<td>00h</td>
<td>DISK STATUS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Descriptor type</th>
<th>Specific Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESCRIPTOR TYPE</td>
<td>REFERENCE METHOD</td>
</tr>
<tr>
<td>DESCRIPTOR TYPE</td>
<td>INFO BLOCK REF PATH</td>
</tr>
<tr>
<td>DESCRIPTOR TYPE</td>
<td>INFO BLOCK REF PATH</td>
</tr>
<tr>
<td>DESCRIPTOR TYPE</td>
<td>INFO BLOCK REF PATH</td>
</tr>
<tr>
<td>DESCRIPTOR TYPE</td>
<td>INFO BLOCK REF PATH</td>
</tr>
<tr>
<td>DESCRIPTOR TYPE</td>
<td>INFO BLOCK REF PATH</td>
</tr>
<tr>
<td>DESCRIPTOR TYPE</td>
<td>INFO BLOCK REF PATH</td>
</tr>
<tr>
<td>DESCRIPTOR TYPE</td>
<td>INFO BLOCK REF PATH</td>
</tr>
</tbody>
</table>

Sect 16b: Audio/Video on 1394
Disk Subunit Status Descriptor

<table>
<thead>
<tr>
<th>Address</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000h</td>
<td>Descriptor length</td>
</tr>
<tr>
<td>0001h</td>
<td></td>
</tr>
<tr>
<td>0002h</td>
<td>General disk subunit status</td>
</tr>
<tr>
<td></td>
<td>Info block type 8800h</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Destination plug status area</td>
</tr>
<tr>
<td></td>
<td>Info block type 8801</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Source plug status</td>
</tr>
<tr>
<td></td>
<td>Info block type 8802</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Synchro plug group status</td>
</tr>
<tr>
<td></td>
<td>Info block type 8803</td>
</tr>
</tbody>
</table>

Operating mode
Info block 8806

Position info
Info block 0003

Plug Configuration
Info block 8807

Playback order config
Info block 8808

Audio level meter status
Info block 8809

Monitor status
Info block 880A

Synchro plug config
Info block 880B

© KnowledgeTek
rev 5.41d  15.H.09

1394

Sect 16b: Audio/Video on 1394
Isochronous Packets for Data Transfer
## AV/C Document Structure

<table>
<thead>
<tr>
<th>Data Length</th>
<th>Tag</th>
<th>Channel</th>
<th>tCode=A</th>
<th>sy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Header CRC</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CIP Header</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Audio/Video Data</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Pad, if necessary</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Data CRC</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### TAG

- **00b**: No CIP Header present
- **01b**: CIP header included
- **10b**: Reserved
- **11b**: Reserved

**1394**

Sect 16b: Audio/Video on 1394
Packetizing Data

1394 packet

1394 Header & CRC

Audio

Video frame

Source packet

61884 header

Audio & Video

Video Only

OR

CIP Header

Data CRC

480 bytes - SD - standard def
960 bytes - HD - high def
240 bytes - SDL - standard compressed

Serve up source packets until all
data is covered

1394

Sect 16b: Audio/Video on 1394

© KnowledgeTek
rev 5.41d 15.H.09
This page is intentionally blank
CIP Header
Common Isochronous Packet

2 FORMATS

<table>
<thead>
<tr>
<th></th>
<th>SID</th>
<th>DBS</th>
<th>FN</th>
<th>QPC</th>
<th>S</th>
<th>R</th>
<th>DBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>FMT</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Frames/cycles per sec

<table>
<thead>
<tr>
<th></th>
<th>SID</th>
<th>DBS</th>
<th>FN</th>
<th>QPC</th>
<th>S</th>
<th>R</th>
<th>DBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>FMT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FDF

1394

© KnowledgeTek rev 5.41d 15.H.09

Sect 16b: Audio/Video on 1394
CIP Header

- **eoh**: End of Header
- **SID**: Source node ID
- **DBS**: Data Block Size in quadlets \((00h = 256)\)
- **FN**: Fraction Number; number of data blocks into which source block is divided
- **QPC**: Quadlet Padding Count; to make every data block the same size
- **FDF**: Format Dependent Field
- **S**: Source packet header
- **DBC**: Data Block Continuity counter
- **SType**: See next page
- **SYT**: Synchronization Timer (low order 16 bits of 1394 timer)
## Packetizing Data

<table>
<thead>
<tr>
<th></th>
<th>SD-DVCR</th>
<th>HD-DVCR</th>
<th>SDL-DVCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan/Frame</td>
<td>525/60</td>
<td>625/50</td>
<td>1125/60</td>
</tr>
<tr>
<td></td>
<td>1250/50</td>
<td></td>
<td>525/60</td>
</tr>
<tr>
<td></td>
<td></td>
<td>625/50</td>
<td>625/50</td>
</tr>
<tr>
<td>Bytes per source frame</td>
<td>480</td>
<td>480</td>
<td>960</td>
</tr>
<tr>
<td></td>
<td>960</td>
<td>240</td>
<td>240</td>
</tr>
<tr>
<td>DIF Blocks</td>
<td>6</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>DBS</td>
<td>78h</td>
<td>78h</td>
<td>F0h</td>
</tr>
<tr>
<td></td>
<td>F0h</td>
<td>3Ch</td>
<td>3Ch</td>
</tr>
<tr>
<td>SType</td>
<td>00h</td>
<td>00h</td>
<td>02h</td>
</tr>
<tr>
<td></td>
<td>02h</td>
<td>01h</td>
<td>01h</td>
</tr>
</tbody>
</table>

1394

Sect 16b: Audio/Video on 1394
Common Isochronous Packet (CIP) Format

Video Frame
(video, audio, subcode, vaux)

Broken Into

Sequence of Source Packets
(may contain different data types)

Each Broken Into

1, 2, 4, or 8 Data Blocks

Each Placed Into

1394 Isochronous Packet
(with a CIP Header in front)

Source Packet Size
480 bytes - SD
960 bytes - HD
240 bytes - SDL

More than one Data Block may be placed in a single 1394 Isochronous Packet.

→ to resync: reconstruct the original blocks
Example Source Packet Sequence for SDL Frame (60Hz)

<table>
<thead>
<tr>
<th>Source Packet #</th>
<th>Source Packet = 240 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>H</td>
</tr>
<tr>
<td>1</td>
<td>VA₀</td>
</tr>
<tr>
<td>2</td>
<td>A₀</td>
</tr>
<tr>
<td>3</td>
<td>V₂</td>
</tr>
<tr>
<td>49</td>
<td>V₁₃₂</td>
</tr>
<tr>
<td>50</td>
<td>H</td>
</tr>
<tr>
<td>51</td>
<td>VA₀</td>
</tr>
<tr>
<td>52</td>
<td>A₀</td>
</tr>
<tr>
<td>53</td>
<td>V₂</td>
</tr>
<tr>
<td>99</td>
<td>V₁₃₂</td>
</tr>
<tr>
<td>100</td>
<td>H</td>
</tr>
</tbody>
</table>

This sequence repeats 5 times per frame.

- **H**: DVCR Header Block
- **V**: Video
- **A**: Audio
- **SC**: Subcode
- **VA**: VAUX

250 Source Packets per frame

© KnowledgeTek
rev 5.41d 15.09

Sect 16b: Audio/Video on 1394
Packaging the Data

Video Frame (Video, Audio, Subcode, VAUX)

Sequence of Source Packets

Blocks

1394 Isochronous Packets

CIP Header

CIP Header

1394

Sect 16b: Audio/Video on 1394
Camera 1.20
Camera Specification 1.20

Available from 1394 Trade Association  www.1394ta.org

- Isochronous talker only
- Not capable of Cycle Master, IRM, or listener
- Must be connected to a camera controller
- Must be able to do asynchronous transfers up to 32 quadlets
- Must be settable to channels 0-15
- Must implement the following registers
  - State clear/State set
  - Node ID
  - Reset Start
  - Split timeout
  - Cycle time
  - Busy timeout

works strictly w/ CSRs
Use CSRs to Control a Camera

Control a camera by writing to CSRs
Get status by reading CSRs

OFFSETS

<table>
<thead>
<tr>
<th>CSR Range</th>
<th>R/W</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>000h</td>
<td>W</td>
<td>Initialize the Registers</td>
</tr>
<tr>
<td>100 - 1FFh</td>
<td>R</td>
<td>Inquiry formats and modes supported</td>
</tr>
<tr>
<td>200 - 3FFh</td>
<td>R</td>
<td>Inquiry frame rates supported</td>
</tr>
<tr>
<td>400 - 4FFh</td>
<td>R</td>
<td>Inquiry features supported</td>
</tr>
<tr>
<td>500 - 5FFh</td>
<td>R</td>
<td>Inquiry range of feature adjustment</td>
</tr>
<tr>
<td>600 - 6FFh</td>
<td>R</td>
<td>Inquiry Status for camera</td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>Set control for camera</td>
</tr>
<tr>
<td>700 - 7FFh</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>800 - 8FFh</td>
<td>R</td>
<td>Inquiry Status for features</td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>Set control for feature</td>
</tr>
</tbody>
</table>
Locating Camera CSRs
Configuration ROM

<table>
<thead>
<tr>
<th>Offset</th>
<th>0-7</th>
<th>6-15</th>
<th>16-23</th>
<th>24-31</th>
</tr>
</thead>
<tbody>
<tr>
<td>400h</td>
<td>04h</td>
<td>CRC length</td>
<td>ROM CRC value</td>
<td></td>
</tr>
<tr>
<td>404h</td>
<td>31h</td>
<td>33h</td>
<td>39h</td>
<td>34h</td>
</tr>
<tr>
<td>408h</td>
<td>0010</td>
<td>rsvd</td>
<td>FFh</td>
<td>max rec</td>
</tr>
<tr>
<td>40Ch</td>
<td>node vendor id</td>
<td>chp_id_hi</td>
<td></td>
<td></td>
</tr>
<tr>
<td>410h</td>
<td>chip id lo</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>414h</td>
<td>04h</td>
<td>CRC length</td>
<td>CRC</td>
<td></td>
</tr>
<tr>
<td>418h</td>
<td>03h</td>
<td>modular vendor id</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41Ch</td>
<td>0Ch</td>
<td>rsvd</td>
<td>8380</td>
<td></td>
</tr>
<tr>
<td>420h</td>
<td>8Dh</td>
<td>indirect offset</td>
<td></td>
<td></td>
</tr>
<tr>
<td>424h</td>
<td>D1h</td>
<td>unit directory offset</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Root Directory

© KnowledgeTek
rev 5.41d 15.H.09

1394
Sect 16b: Audio/Video on 1394

16 - 59
Configuration ROM (continued)

<table>
<thead>
<tr>
<th>Offset</th>
<th>0-7</th>
<th>6-15</th>
<th>16-23</th>
<th>24-31</th>
</tr>
</thead>
<tbody>
<tr>
<td>000h</td>
<td></td>
<td>0003h</td>
<td></td>
<td>CRC</td>
</tr>
<tr>
<td>004h</td>
<td>12h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>008h</td>
<td>13h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00Ch</td>
<td>D4h</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Unit Directory

- Offset 000h: unit spec ID (00A02Dh)
- Offset 004h: unit sw version (00010xh)
- Offset 008h: unit dependent directory offset

X = 0 for Camera 1.04
1 for Camera 1.20

<table>
<thead>
<tr>
<th>Offset</th>
<th>0-7</th>
<th>6-15</th>
<th>16-23</th>
<th>24-31</th>
</tr>
</thead>
<tbody>
<tr>
<td>000h</td>
<td></td>
<td></td>
<td></td>
<td>CRC</td>
</tr>
<tr>
<td>004h</td>
<td>40h</td>
<td>command regs base</td>
<td></td>
<td></td>
</tr>
<tr>
<td>008h</td>
<td>81h</td>
<td>number of quadlets to vendor name leaf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00Ch</td>
<td>82h</td>
<td>number of quadlets to model name leaf</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Unit Dependent Directory

1394

Sect 16b: Audio/Video on 1394
1. Physically plug 1394 cable into camera and camera controller
2. 1394 will detect newly connected device, do a reset and reconfigure the bus and enumerate all devices
3. Camera controller will use 1394 asynchronous reads to discover camera capabilities and limitations
4. Camera controller will use 1394 asynchronous writes to enable, disable, or adjust camera features and settings
5. Camera controller will use 1394 asynchronous write to start the camera
6. Camera will use 1394 isochronous packets to transfer data to camera controller over isochronous channel defined in step 4 above
7. Camera controller will use 1394 asynchronous write to stop the camera
# Isochronous Data Packet

<table>
<thead>
<tr>
<th>Data Length</th>
<th>Tag</th>
<th>Channel</th>
<th>tCode=A</th>
<th>sy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Header CRC**

<table>
<thead>
<tr>
<th>Video Data</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Pad, if necessary</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Data CRC</th>
</tr>
</thead>
</table>

---

Sect 16b: Audio/Video on 1394
Digital Camera Initialize
For more information

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Field</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset</td>
<td>INITIALIZE</td>
<td>Initialize</td>
<td>[0]</td>
<td>If assert this bit, Camera will re-set to initial (factory setting value) state.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>[1..31]</td>
<td>Reserved (all zero)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bits 0-7</th>
<th>Bits 8-15</th>
<th>Bits 16-23</th>
<th>Bits 24-31</th>
<th>Reserved</th>
</tr>
</thead>
</table>

© KnowledgeTek
rev 5.41d   15.H.09
Sect 16b: Audio/Video on 1394
## Digital Camera - Format Inquiry

For more information

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Field</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100h</td>
<td>V FORMAT INQ</td>
<td>Format x</td>
<td>[0..7]</td>
<td>Defined below</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[8..31]</td>
<td>Reserved (All zero)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bits 0-7</th>
<th>Bits 8-15</th>
<th>Bits 16-23</th>
<th>Bits 24-31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Format 0  VGA non-compressed (Maximum 640 x 480)
Format 1  Super VGA non-compressed format 1
Format 2  Super VGA non-compressed format 2
Format 6  Still Image
Format 7  Scalable image size
## Digital Camera - Mode Inquiry

For more information

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Field</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>180h</td>
<td>V MODE INQ 0 (format 0)</td>
<td>Mode 0</td>
<td>[0]</td>
<td>160 X 120 YUV(4:4:4) Mode (24 bit/pixel)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mode x</td>
<td>[6..7]</td>
<td>Reserved for another Mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[8..31]</td>
<td>Reserved (All zero)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bits 0-7</th>
<th>Bits 8-15</th>
<th>Bits 16-23</th>
<th>Bits 24-31</th>
</tr>
</thead>
<tbody>
<tr>
<td>V Mode</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© KnowledgeTek
rev 5.41d 15.H.09

Sect 16b: Audio/Video on 1394
Digital Camera - Mode Inquiry
For more information

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Field</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>184h</td>
<td>V MODE INQ 1 (format 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mode 0</td>
<td>[0]</td>
<td></td>
<td>800 x 600 YUV(4:2:2) Mode (16 bit/pixel)</td>
</tr>
<tr>
<td></td>
<td>Mode 1</td>
<td>[1]</td>
<td></td>
<td>800 x 600 RGB Mode (24 bit/pixel)</td>
</tr>
<tr>
<td></td>
<td>Mode 2</td>
<td>[2]</td>
<td></td>
<td>800 x 600 Y (Mono) Mode (8 bit/pixel)</td>
</tr>
<tr>
<td></td>
<td>Mode 4</td>
<td>[4]</td>
<td></td>
<td>1024 x 768 RGB Mode (24 bit/pixel)</td>
</tr>
<tr>
<td></td>
<td>Mode 5</td>
<td>[5]</td>
<td></td>
<td>1024 x 768 Y (Mono) Mode (8 bit/pixel)</td>
</tr>
<tr>
<td></td>
<td>Mode x</td>
<td>[6..7]</td>
<td></td>
<td>Reserved for another Mode</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>[8..31]</td>
<td></td>
<td>Reserved (All zero)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bits 0-7</th>
<th>Bits 8-15</th>
<th>Bits 16-23</th>
<th>Bits 24-31</th>
</tr>
</thead>
<tbody>
<tr>
<td>V Mode</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Digital Camera - Mode Inquiry

For more information

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Field</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>188h</td>
<td>V MODE INQ 2</td>
<td></td>
<td></td>
<td>1280 x 960 YUV(4:2:2) Mode (16 bit/pixel)</td>
</tr>
<tr>
<td></td>
<td>(format 2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mode 0</td>
<td>[0] 1280 x 960 YUV(4:2:2) Mode (16 bit/pixel)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mode 1</td>
<td>[1] 1280 x 960 RGB Mode (24 bit/pixel)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mode 2</td>
<td>[2] 1280 x 960 Y (Mono) Mode (8 bit/pixel)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mode 4</td>
<td>[4] 1600 x 1200 RGB Mode (24 bit/pixel)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mode 5</td>
<td>[5] 1600 x 1200 Y (Mono) Mode (8 bit/pixel)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mode x</td>
<td>[6..7] Reserved for another Mode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[8..31] Reserved (All zero)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bits 0-7</th>
<th>Bits 8-15</th>
<th>Bits 16-23</th>
<th>Bits 24-31</th>
</tr>
</thead>
<tbody>
<tr>
<td>V Mode</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© KnowledgeTek
rev 5.41d 15.H.09

Sect 16b: Audio/Video on 1394
## Digital Camera - Mode Inquiry

For more information

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Field</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>198h</td>
<td>V MODE INQ 6 (format 6)</td>
<td>Mode 0</td>
<td>[0]</td>
<td>EXIF format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mode x</td>
<td>[6..7]</td>
<td>Reserved for another Mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[8..31]</td>
<td>Reserved (All zero)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bits 0-7</th>
<th>Bits 8-15</th>
<th>Bits 16-23</th>
<th>Bits 24-31</th>
</tr>
</thead>
<tbody>
<tr>
<td>V Mode</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© KnowledgeTek
rev 5.41d 15.H.09

Sect 16b: Audio/Video on 1394

1394
## Digital Camera - Mode Inquiry

For more information

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Field</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>19Ch</td>
<td>V MODE INQ 7 (format 7)</td>
<td>Mode 0</td>
<td>[0]</td>
<td>Format 7, Mode 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mode 1</td>
<td>[1]</td>
<td>Format 7, Mode 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mode 2</td>
<td>[2]</td>
<td>Format 7, Mode 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mode 3</td>
<td>[3]</td>
<td>Format 7, Mode 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mode 4</td>
<td>[4]</td>
<td>Format 7, Mode 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mode 6</td>
<td>[6]</td>
<td>Format 7, Mode 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mode 7</td>
<td>[7]</td>
<td>Format 7, Mode 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>[8..31]</td>
<td>Reserved (All zero)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bits 0-7</th>
<th>Bits 8-15</th>
<th>Bits 16-23</th>
<th>Bits 24-31</th>
</tr>
</thead>
<tbody>
<tr>
<td>V Mode</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© KnowledgeTek
rev 5.41d 15.H.09

Sect 16b: Audio/Video on 1394
## Digital Camera - Frame Rate Inquiry Register

For more information

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.875 frames per second</td>
</tr>
<tr>
<td>1</td>
<td>3.75 fps</td>
</tr>
<tr>
<td>2</td>
<td>7.5 fps</td>
</tr>
<tr>
<td>3</td>
<td>15 fps</td>
</tr>
<tr>
<td>4</td>
<td>30 fps</td>
</tr>
<tr>
<td>5</td>
<td>60 fps</td>
</tr>
</tbody>
</table>

Offset  | Description                          |
--------|--------------------------------------|
200     | Format 0, Mode 0                     |
204     | Format 0, Mode 1                     |
208     | Format 0, Mode 2                     |
20C     | Format 0, Mode 3                     |
210     | Format 0, Mode 4                     |
214     | Format 0, Mode 5                     |
218-21F | Other modes in Format 0              |
220-2FF | Modes in other formats               |

Caution: not all frame rates are used in every format/mode combination
# Inquiry Register for Basic Function

For more information

Offset 400

<table>
<thead>
<tr>
<th></th>
<th>0-7</th>
<th>8-15</th>
<th>16-23</th>
<th>24-31</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td>mem</td>
</tr>
</tbody>
</table>

A  Camera has vendor unique advanced feature
C  Camera Power on or off capability
O  One shot transmission capability
m  Multi-shot transmission capability

Mem Maximum memory channel number

- 0000b  User memory not available
- Factory setting memory only
Inquiry Register for Feature Presence

For more information

<table>
<thead>
<tr>
<th>Offset</th>
<th>0-7</th>
<th>8-15</th>
<th>16-23</th>
<th>24-31</th>
</tr>
</thead>
<tbody>
<tr>
<td>404h</td>
<td>b</td>
<td>e</td>
<td>s</td>
<td>w</td>
</tr>
<tr>
<td>408h</td>
<td>z</td>
<td>p</td>
<td>t</td>
<td>k</td>
</tr>
</tbody>
</table>
Inquiry Register for Feature Presence - Definitions

For more information

<table>
<thead>
<tr>
<th>Feature</th>
<th>Definition</th>
<th>Value 1</th>
<th>Value 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Brightness</td>
<td>Feature control is available</td>
<td>Feature control is not available</td>
</tr>
<tr>
<td>E</td>
<td>Exposure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>Sharpness</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W</td>
<td>Whiteness</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>Hue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>Saturation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>Gamma</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>Shutter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>Gain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>Iris</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>Focus</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Temperature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>Trigger</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>Zoom</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>Pan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>Tilt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>Optical filter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>Format 6 Size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>Format 6 Quality</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Inquiry Register for Feature Elements

For more information

<table>
<thead>
<tr>
<th>Offset</th>
<th>0-7</th>
<th>8-15</th>
<th>16-23</th>
<th>24-31</th>
</tr>
</thead>
<tbody>
<tr>
<td>See below</td>
<td>p</td>
<td>r</td>
<td>a</td>
<td>m</td>
</tr>
</tbody>
</table>

For EACH REGISTER/FEATURE listed below

~

Sect 16b: Audio/Video on 1394
### Inquiry Register for Feature Elements - Offsets

For more information

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>500h</td>
<td>BRIGHTNESS INQ</td>
</tr>
<tr>
<td>504h</td>
<td>EXPOSURE INQ</td>
</tr>
<tr>
<td>508h</td>
<td>SHARPNESS INQ</td>
</tr>
<tr>
<td>50Ch</td>
<td>WHITE BAL INQ</td>
</tr>
<tr>
<td>510h</td>
<td>HUE INQ</td>
</tr>
<tr>
<td>514h</td>
<td>SATURATION INQ</td>
</tr>
<tr>
<td>518h</td>
<td>GAMMA INQ</td>
</tr>
<tr>
<td>51Ch</td>
<td>SHUTTER INQ</td>
</tr>
<tr>
<td>520h</td>
<td>GAIN INQ</td>
</tr>
<tr>
<td>524h</td>
<td>IRIS INQ</td>
</tr>
<tr>
<td>528h</td>
<td>FOCUS INQ</td>
</tr>
<tr>
<td>52Ch</td>
<td>Reserved for other FEATURE HI INQ</td>
</tr>
<tr>
<td>57Ch</td>
<td></td>
</tr>
<tr>
<td>580h</td>
<td>ZOOM INQ</td>
</tr>
<tr>
<td>584h</td>
<td>PAN INQ</td>
</tr>
<tr>
<td>588h</td>
<td>TILT INQ</td>
</tr>
<tr>
<td>58Ch</td>
<td>Reserved for other FEATURE LO INQ</td>
</tr>
<tr>
<td>5FCh</td>
<td></td>
</tr>
</tbody>
</table>
Inquiry Register for Feature Elements - Definitions

For more information

<table>
<thead>
<tr>
<th>P</th>
<th>Feature is Present</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>Capability of Reading this feature</td>
</tr>
<tr>
<td>O</td>
<td>Capability of turning feature On or Off</td>
</tr>
<tr>
<td>A</td>
<td>Capability of Automode</td>
</tr>
<tr>
<td>M</td>
<td>Capability of Manual mode</td>
</tr>
<tr>
<td>Min value</td>
<td>Minimum value of this feature</td>
</tr>
<tr>
<td>Max value</td>
<td>Maximum value of this feature</td>
</tr>
</tbody>
</table>
### Control and Status Registers for Cameras

For more information

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>600h</td>
<td>Cur V Frm Rate</td>
<td>[0..2]</td>
<td>Read the current frame rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FrameRate 0 .. FrameRate 7</td>
</tr>
<tr>
<td>604h</td>
<td>Cur V Frm Mode</td>
<td>[0..2]</td>
<td>Read the current video mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mode 0 .. Mode 7</td>
</tr>
<tr>
<td>608h</td>
<td>Cur V Frm Channel</td>
<td>[0..2]</td>
<td>Read the current video format</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Format 0 .. Format 7</td>
</tr>
<tr>
<td>60Ch</td>
<td>ISO Channel</td>
<td>[0..2]</td>
<td>Isochronous channel number for video data transmission</td>
</tr>
<tr>
<td></td>
<td>ISO Speed</td>
<td>[4..5]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[6..7]</td>
<td>Isochronous transmit speed code</td>
</tr>
<tr>
<td>61Ch</td>
<td>Camera Power</td>
<td>[0]</td>
<td>1 = power-up camera</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = power-down camera</td>
</tr>
<tr>
<td>614h</td>
<td>ISO EN</td>
<td>[0]</td>
<td>1 = start ISO transmission of video data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = stop ISO transmission of video data</td>
</tr>
</tbody>
</table>
### CSR for Camera (continued)

For more information

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>618h</td>
<td>Memoray Save</td>
<td>[0]</td>
<td>1 = current status and modes are saved to Mem Sav Ch (Self cleared)</td>
</tr>
<tr>
<td>61Ch</td>
<td>One Shot</td>
<td>[0]</td>
<td>1 = only one frame of video data is transmitted (Self cleared after transmission Ignored if ISO EN = 1)</td>
</tr>
<tr>
<td>61Ch</td>
<td>Mem Save Ch</td>
<td>[0..3]</td>
<td>Write channel for Memory Save command Must be &gt;= 0001 (0 is factory settings, which cannot be overwritten (see BASIC FUNC INQ)</td>
</tr>
<tr>
<td>624h</td>
<td>Cur Memo Ch</td>
<td>[0..3]</td>
<td>When read from, returns Current Memory Channel number When written to, loads status, modes, and values from the specified memory channel</td>
</tr>
</tbody>
</table>
Status and Control Register for Feature
For more information

<table>
<thead>
<tr>
<th>Offset</th>
<th>0-7</th>
<th>8-15</th>
<th>16-23</th>
<th>24-31</th>
</tr>
</thead>
<tbody>
<tr>
<td>See below</td>
<td>p</td>
<td>o</td>
<td>a</td>
<td>reserved/u value</td>
</tr>
</tbody>
</table>

1394

Sect 16b: Audio/Video on 1394
### Status and Control Register for Features - Offsets

For more information

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>800h</td>
<td>BRIGHTNESS</td>
</tr>
<tr>
<td>804h</td>
<td>EXPOSURE</td>
</tr>
<tr>
<td>808h</td>
<td>SHARPNESS</td>
</tr>
<tr>
<td>80Ch</td>
<td>WHITE BAL</td>
</tr>
<tr>
<td>810h</td>
<td>HUE</td>
</tr>
<tr>
<td>814h</td>
<td>SATURATION</td>
</tr>
<tr>
<td>818h</td>
<td>GAMMA</td>
</tr>
<tr>
<td>81Ch</td>
<td>SHUTTER</td>
</tr>
<tr>
<td>820h</td>
<td>GAIN</td>
</tr>
<tr>
<td>824h</td>
<td>IRIS</td>
</tr>
<tr>
<td>828h</td>
<td>FOCUS</td>
</tr>
<tr>
<td>82Ch</td>
<td>Reserved for other FEATURE HI</td>
</tr>
<tr>
<td>87Ch</td>
<td></td>
</tr>
<tr>
<td>880h</td>
<td>ZOOM</td>
</tr>
<tr>
<td>884h</td>
<td>PAN</td>
</tr>
<tr>
<td>888h</td>
<td>TILT</td>
</tr>
<tr>
<td>88Ch</td>
<td>Reserved for other FEATURE LO</td>
</tr>
<tr>
<td>8FCh</td>
<td></td>
</tr>
</tbody>
</table>
Status and Control Register for Features - Definitions
For more information

P  Feature is Present

O  Write - turn this feature On or Off
   Read - Return On/Off status of this feature

A  Write - Set the mode; 1 = Auto, 0 = Manual
   Read - Return Auto/Manual status of this feature

Value  Write - Set the value in this feature
       Read - Return the value this feature is set to

U-Value  U-Value for White balance only

V-Value  V-Value for White balance only
Review

1. What mechanism does 61883 and AV/C use to simulate a physical connection?
2. What is the protocol used to move commands and status?
3. What is the protocol used to move data with AV/C?
4. What bus management capabilities are required of a 61883 node?
5. How is control done with camera 1.20 compliant nodes?
Notes
Section 17

1394b
High Speed
Long Distance
Why a New PHY

Faster Speeds
S800
S1600
S3200

Greater Distance
100 meters

New Connection/media
Unshielded Twisted Pair
Plastic Optic Fiber
Glass Optic Fiber

More Efficient
Eliminate gaps for fairness
Last one transmitting does arbitration
Arbitration is done during previous information transmission
Subjects Covered

Signaling
8b/10b → AC COUPLING
Disparity
Speed signaling
Payload

Arbitration
BOSS
Fairness

Connection Media
STP
Glass Optical Fiber
Plastic Optical Fiber
Hard Polymer Clad Fiber
Unshielded Twisted Pair

Loop Free Build
PHY Link Interface
PHY Registers
PIL-FOP
Characteristics

1394-1995

DS encoding
DC coupling
Maximum speed - 400 Mbps
Gaps for fairness and priority

1394b

8b/10b encoding
AC or DC coupled
Up to 3200 Mbps
May be Bilingual - compatible with 1394-1995
Media defined

- Category 5 UTP
- Hard Polymer Clad Fiber
- Plastic Optic Fiber
- Glass Optic Fiber
- 1394-1995 style cables

Full Duplex
No gaps
8b/10b Encoding
Synchronization

Start

Bits arriving

Adjust PLL

Bit sync (phase lock loop)

No

K28.5?

Character sync

TX Training Request

Rx Training Request?

Synchronize Descrambler

Transmit Operation Request

End

Port sync

1394

Sect 17: 1394b High Speed/Long Distance
Data Encoding

Data Byte bits  Request symbol  Control symbol
ABCDEFGH      ABCDExxH       PQRS

Scrambler

A'B'C'D'E'F'G'H'  A'B'C'D'E'00H'  P'Q'R'S'

8b/10b Encoder

a b c d e f g h j

5b/6b encoder  3b/4b encoder

Most significant bit is A, A', P or P'

Bits are transmitted "a" first
# Request Signaling

<table>
<thead>
<tr>
<th>Request</th>
<th>Symbol ABCDExxH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>0000 0xx0</td>
</tr>
<tr>
<td>Disable Notify</td>
<td>0010 0xx0</td>
</tr>
<tr>
<td>Child Notify, Ident done</td>
<td>0100 0xx0</td>
</tr>
<tr>
<td>Operation</td>
<td>0110 0xx0</td>
</tr>
<tr>
<td>Standby</td>
<td>1000 0xx0</td>
</tr>
<tr>
<td>Suspend</td>
<td>1010 0xx0</td>
</tr>
<tr>
<td>Parent Notify</td>
<td>1100 0xx0</td>
</tr>
<tr>
<td>Legacy request</td>
<td>1110 0xx0</td>
</tr>
</tbody>
</table>

*Can't use 1, 0, Z any may be AC coupling - can only use 0, 1. All 17-11.*
# Control Symbol Mapping

<table>
<thead>
<tr>
<th>Control Token</th>
<th>Control Symbol</th>
<th>RD&lt;0</th>
<th>RD&gt;0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Async Start</td>
<td></td>
<td>0000</td>
<td></td>
</tr>
<tr>
<td>Cycle Start even</td>
<td></td>
<td>0001</td>
<td></td>
</tr>
<tr>
<td>Cycle Start odd</td>
<td></td>
<td>0010</td>
<td></td>
</tr>
<tr>
<td>Attach request/Arb context</td>
<td></td>
<td>0011</td>
<td></td>
</tr>
<tr>
<td>Speeda</td>
<td></td>
<td>0100</td>
<td></td>
</tr>
<tr>
<td>Data end</td>
<td></td>
<td>0101</td>
<td></td>
</tr>
<tr>
<td>Data null</td>
<td></td>
<td>0110</td>
<td></td>
</tr>
<tr>
<td>Speedb</td>
<td></td>
<td>0111</td>
<td></td>
</tr>
<tr>
<td>Grant</td>
<td></td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>Data prefix</td>
<td></td>
<td>1010</td>
<td>1001</td>
</tr>
<tr>
<td>Reserved</td>
<td></td>
<td>1011</td>
<td></td>
</tr>
<tr>
<td>Speedc</td>
<td></td>
<td>1100</td>
<td></td>
</tr>
<tr>
<td>ARBRST even</td>
<td></td>
<td>1101</td>
<td></td>
</tr>
<tr>
<td>ARBRST odd</td>
<td></td>
<td>1110</td>
<td></td>
</tr>
<tr>
<td>Bus Reset</td>
<td></td>
<td>1111</td>
<td></td>
</tr>
</tbody>
</table>
DC Balance Disparity
What /Why Disparity

1394b may be AC coupled, DC level cannot then be transmitted

To maintain circuits in linear part of their operation, we must establish a bias or DC level

This can be done by having an equal number of 1’s and 0’s

Therefore, at the beginning of each character, we check the disparity (have there been more 0’s or 1’s or are they equal). We then select the character format to make the number of 1’s and 0’s nearly equal at the end of this character.
Character Disparity

From the 1024 bit combinations in the 10 bit code word select only those that have an equal number of 1’s and 0’s, or those that have only 2 more 1’s or 0’s.

Also, select only those that do not have runs of five 0 bits or runs of five 1 bits.

Assign the 256 combinations from the 8 bit data byte to the resulting 10 bit combinations.

Characters that have five 1 and five 0 bits are neutral disparity
Characters that have six 1 and four 0 bits are positive disparity
Characters that have four 1 and six 0 bits are negative disparity

Characters with positive disparity have another encode that provides negative disparity
Running Disparity

Both the transmitter and the receiver start by setting their running disparity to -1

For each character, if the running disparity is -1
   Select an encode with a neutral character disparity or
   Select an encode with a positive character disparity

For each character, if the running disparity is +1
   Select an encode with a neutral character disparity or
   Select an encode with a negative character disparity

Receiver checks character disparity and updates running disparity
   If running disparity is not -1 or +1, then an error has occurred

Running disparity is set to -1 via selection of data prefix encode
Example of Disparity

Previous character yielded -1 disparity

This character D9.6
Select either
100101 1101 (+2) or
100101 0010 (-2)

Now running disparity

Running Disparity
-1

Select +2 since C negative disparity

Selection of the other encode for D9.6 would have caused the receiver to indicate an error

© KnowledgeTek
rev 5.41E 15.H.31

Sect 17: 1394b High Speed/Long Distance
Speed Signaling
1394-1995 speed signaling not changed for DS ports

1394b:
Connected ports exchange speed signals, agree on slower

Sending port sends speed code characters indicating packet speed
1 character means packet speed = port speed
2 characters means packet speed = 1/2 port speed
4 characters means packet speed = 1/4 port speed

Receiving port counts characters to determine packet speed

Sequence is:
Special Considerations

S100 packet
   No speed characters
   Data Prefix (more than one), then data

Beta
   Use speed characters to indicate packet speed

S800 and greater
   Data immediately follows speed characters
## Speed Characters

Three speed characters
- Speeda
- Speedb
- Speedc

Only one Speed(x) per sequence, all others are Speedc

<table>
<thead>
<tr>
<th>Packet Speed as a function of port speed</th>
<th>Speedc</th>
<th>Speed(x)</th>
<th>Speedc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1/2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1/4</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1/8</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>1/16</td>
<td>4</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>1/32</td>
<td>5</td>
<td>1</td>
<td>26</td>
</tr>
</tbody>
</table>
Payload Speed Matching
## Padding

### Data and Control Characters

<table>
<thead>
<tr>
<th>Packet Speed as a function of Port Speed</th>
<th>Data</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same</td>
<td>D</td>
<td>C*1</td>
</tr>
<tr>
<td>1/2</td>
<td>D + P</td>
<td>C*2</td>
</tr>
<tr>
<td>1/4</td>
<td>D + (P*3)</td>
<td>C*4</td>
</tr>
<tr>
<td>1/8</td>
<td>D + (P*7)</td>
<td>C*8</td>
</tr>
<tr>
<td>1/16</td>
<td>D + (P*15)</td>
<td>C*16</td>
</tr>
<tr>
<td>1/32</td>
<td>D + (P*31)</td>
<td>C*32</td>
</tr>
</tbody>
</table>

- D = data character
- P = Speedc character
- C = Control character
BOSS Mode Arbitration
BOSS Notes

Bus Owner Supervisor/Selector

1394b is full duplex
  Data travels on one pair (TPB --> TPA)
  Other pair is used for arbitration (TPA --> TPB)

Each node always compares its isochronous and asynchronous arbitration needs with what it receives and sends the highest priority

BOSS is the last node to transmit

After transmitting, BOSS will select the highest priority request and issue a grant

If there are no requests, BOSS will issue an ARBRST and transfer control to its parent

When Root becomes BOSS, it will retain that until a request is honored
Fairness

1394b uses even and odd cycles for access fairness, not gaps

Cycles are begun with ARBRST_even or ARBRST_odd

Each arbitrating device will use all its allocation of accesses by sending Asynch_current requests. Then it will send one more request for the next cycle Asynch_even or Asynch_odd

If a node does not need the bus, it will send Asynch_none

When the BOSS sees no Asynch_current, it begins the next cycle by sending ARBRST_even or ARBRST_odd
# Asynchronous Priority

<table>
<thead>
<tr>
<th>Request Name</th>
<th>Priority Level</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Border Node</td>
<td>7 (highest)</td>
<td></td>
</tr>
<tr>
<td>Cycle Start request</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Next Odd</td>
<td>5 if last ARBRST was odd, else 2</td>
<td>This is a queued request from last cycle</td>
</tr>
<tr>
<td>Current</td>
<td>4</td>
<td>Normal requests by nodes that have not used up their fairness budget</td>
</tr>
<tr>
<td>None Even</td>
<td>3 if last ARBRST was odd, else 1</td>
<td></td>
</tr>
<tr>
<td>Next Even</td>
<td>2 if last ARBRST was odd, else 5</td>
<td></td>
</tr>
<tr>
<td>None odd</td>
<td>1 of last ARBRST was odd, else 3</td>
<td></td>
</tr>
</tbody>
</table>
# Isochronous Priority

<table>
<thead>
<tr>
<th>Request Name</th>
<th>Priority Level</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isochronous Odd</td>
<td>3 (highest) if last cycle start was odd, else 2</td>
<td>Used if last cycle start was odd and the packet is intended to transmit in the current cycle</td>
</tr>
<tr>
<td>Isochronous Even</td>
<td>2 if the last cycle start was odd, else 3</td>
<td></td>
</tr>
<tr>
<td>Isochronous None</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
# Arbitration Requests

## Asynchronous

<table>
<thead>
<tr>
<th>Request</th>
<th>Request symbol</th>
<th>Request symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td>000</td>
<td>Not used</td>
</tr>
<tr>
<td>Current</td>
<td>001</td>
<td>Isochronous none</td>
</tr>
<tr>
<td>Next even</td>
<td>010</td>
<td>Isochronous even</td>
</tr>
<tr>
<td>Cycle start</td>
<td>011</td>
<td>Isochronous odd</td>
</tr>
<tr>
<td>None odd</td>
<td>100</td>
<td>Reserved</td>
</tr>
<tr>
<td>Next odd</td>
<td>101</td>
<td>Reserved</td>
</tr>
<tr>
<td>None even</td>
<td>110</td>
<td>Reserved</td>
</tr>
<tr>
<td>Reserved</td>
<td>111</td>
<td>Not used</td>
</tr>
</tbody>
</table>

## Isochronous

<table>
<thead>
<tr>
<th>Request</th>
<th>Request symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td>00xx0</td>
</tr>
<tr>
<td>Isochronous none</td>
<td>01xx0</td>
</tr>
<tr>
<td>Isochronous even</td>
<td>10xx0</td>
</tr>
<tr>
<td>Isochronous odd</td>
<td>11xx0</td>
</tr>
<tr>
<td>Reserved</td>
<td>00xx1</td>
</tr>
<tr>
<td>Reserved</td>
<td>01xx1</td>
</tr>
<tr>
<td>Reserved</td>
<td>10xx1</td>
</tr>
<tr>
<td>Not used</td>
<td>11xx1</td>
</tr>
</tbody>
</table>
BOSS Arbitration

Node 4
Root

Arbitrating Isoch next even, asynch current

Ioch odd, Asynch current

Ioch odd, Asynch current

Transmitting Node and BOSS
Node 0

Information flow TPB --> TPA

Arbitration flow TPA --> TPB

Arbitrating Isoch none, Asynch next odd
Node 3

This node forwards HIGHEST PRI

Arbitrating Isoch odd, Asynch current
Node 1

Arbitrating Isoch none, Asynch next odd
Node 2

Last ARBRST was even
Last Cycle Start was odd

© KnowledgeTek
rev 5.41E 15.H.31

1394
Sect 17: 1394b High Speed/Long Distance
Loop Free Build
# Interconnects

<table>
<thead>
<tr>
<th>Media</th>
<th>Distance</th>
<th>Max Speed</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STP</td>
<td>4.5 m</td>
<td>S1600</td>
<td>Shielded Twisted Pair</td>
</tr>
<tr>
<td>MMF</td>
<td>100 m</td>
<td>S1600</td>
<td>Glass Multi-mode Fiber, 50 micron</td>
</tr>
<tr>
<td>POF</td>
<td>50 m</td>
<td>S200</td>
<td>Plastic Optical Fiber</td>
</tr>
<tr>
<td>HPCF</td>
<td>100 m</td>
<td>S200</td>
<td>Hard Polymer Clad Fiber</td>
</tr>
<tr>
<td>UTP</td>
<td>100 m</td>
<td>S100</td>
<td>Unshielded Twisted Pair, Category 5</td>
</tr>
</tbody>
</table>
Similar cable to 1394a but with different connectors

Two PHY modes defined
  1394b - beta only
  Bilingual - 1394a and 1394b

Two connectors keyed to identify PHY connection
  Beta cables can fit into bilingual sockets
  Bilingual cables cannot fit into beta only sockets

Cable length defined are:
  2 meters - 30 gauge signal wires; 26 gauge power wires
  4.5 meters - 25 gauge signal wires; 22 gauge power wires
STP Connectors

Beta - 6 mm
Bilingual - 4 mm

Viewed from front of plug face

<table>
<thead>
<tr>
<th>Pin</th>
<th>Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TPB*</td>
</tr>
<tr>
<td>2</td>
<td>TPB</td>
</tr>
<tr>
<td>3</td>
<td>TPA*</td>
</tr>
<tr>
<td>4</td>
<td>TPA</td>
</tr>
<tr>
<td>5</td>
<td>TPA return</td>
</tr>
<tr>
<td>6</td>
<td>Power</td>
</tr>
<tr>
<td>7</td>
<td>Reserved</td>
</tr>
<tr>
<td>8</td>
<td>Power Ground</td>
</tr>
<tr>
<td>9</td>
<td>TPB return</td>
</tr>
</tbody>
</table>

1394b High Speed/Long Distance
# STP Cable Assemblies

<table>
<thead>
<tr>
<th>Plug 1</th>
<th>Plug 2</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beta</td>
<td>Beta</td>
<td>1394b</td>
</tr>
<tr>
<td>1394-1995 6 pin</td>
<td>Bilingual</td>
<td>1394-1995</td>
</tr>
<tr>
<td>1394a 4 pin</td>
<td>Bilingual</td>
<td>1394a</td>
</tr>
</tbody>
</table>

- Beta Beta 1394b
- 1394-1995 Bilingual
- 1394a Bilingual
Glass Optical Fiber

Characteristics:

1Gbd or 2Gbd
2 meters to 100 meters range
Uses VCSEL, wavelength 830 – 860 nm (Vertical Cavity Surface Emitting Laser)
S400 beta, S800, S1600
Range, regardless of speed: 2m to 100m
Rise/Fall time (20% to 80%) 0.26ns
50 micron MMF (Multi-Mode Fiber)
Connector is LC duplex
Dimensions and interface spec of the FOCUS 10 addendum of the TIA/EIA 604
POF/HPCF

Characteristics

Uses 650 nanometer light emitting diode
NRZ encoding, “1” indicated by high light intensity
Data rate is S100 beta and S200 beta
Connector is PN, defined in IEC61754-16 and IEC 61753-AA
BER < 10^-12

POF

1000 micron step index multimode fiber
Distance is 50 meters

HPCF

225 micron graded index multimode fiber
Distance is 100 meters
UTP Category 5

Characteristics

S100 beta to 100 meters
UTP Category 5
BER < $10^{-12}$
ISO/IEC 11801 for category 5 and ISO/IEC 11801 chapter 7 for 100 ohm balanced connection
Connector is 8 pin IEC 603-7
NRZ, full duplex
# Cat 5 connector

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TPB</td>
</tr>
<tr>
<td>2</td>
<td>TPB*</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>TPA</td>
</tr>
<tr>
<td>8</td>
<td>TPA*</td>
</tr>
</tbody>
</table>
PHY-Link Interface
# 1394b Remote Command Packet

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHY ID</td>
<td>Node for this command</td>
</tr>
<tr>
<td>Type</td>
<td>Extend PHY packet type (8 indicates command packet)</td>
</tr>
</tbody>
</table>
| E-cmd | 0 - NOP  
1 - Initiate Standby with connected port  
2 - Restore from standby with connected port  
3-7 reserved |
| Port  | Which port to execute this command |
| cmd   | 0 - NOP  
1 - Transmit Disable Notify, then disable port  
2 - Initiate suspend  
3 - Reserved  
4 - Clear this port's fault bit  
5 - Enable port  
6 - Resume port  
7 - Use e-cmd |
1394b Link/Phy Connection

Link Hardware

Data [0:7]
Control [0:1]
LReq
Pint
PClk
LClk
LPS
Link on

PHY Chip

OSC

TPA
TPB

Direct

Beta Mode

Electrical Isolation

Power

Regulator

1394

Sect 17: 1394b High Speed/Long Distance
## 1394b Link/Phy Connection Definitions

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>[0:7] for S400</td>
</tr>
<tr>
<td>Control</td>
<td>What is the meaning of the data lines</td>
</tr>
<tr>
<td>LReq</td>
<td>Serial command to the Phy</td>
</tr>
<tr>
<td>PClk</td>
<td>98.304MHz clock, must be generated in PHY</td>
</tr>
<tr>
<td>LClk</td>
<td>PHY clock returned by link</td>
</tr>
<tr>
<td>LPS</td>
<td>Link power status</td>
</tr>
<tr>
<td>Link On</td>
<td>Commands Link to Power On</td>
</tr>
<tr>
<td>Direct</td>
<td>Indicates Link and Phy are directly connected</td>
</tr>
<tr>
<td>Pint</td>
<td>PHY interrupt to link</td>
</tr>
</tbody>
</table>
## Extended PHY Registers

<table>
<thead>
<tr>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000b</td>
</tr>
<tr>
<td>0001b</td>
</tr>
<tr>
<td>0010b</td>
</tr>
<tr>
<td>0011b</td>
</tr>
<tr>
<td>0100b</td>
</tr>
<tr>
<td>0101b</td>
</tr>
<tr>
<td>0110b</td>
</tr>
<tr>
<td>0111b</td>
</tr>
<tr>
<td>1000b</td>
</tr>
<tr>
<td>1111b</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical ID</td>
<td>Root</td>
<td>CPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gap Count</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extended = 7</td>
<td>Total Ports</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max speed</td>
<td>Res</td>
<td>Delay</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>C</td>
<td>Jitter</td>
<td>1394 Power class</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Watchdog</td>
<td>ISBR</td>
<td>Loop</td>
<td>Pwr fail</td>
<td>Timeout</td>
<td>Bias</td>
<td>enab acc</td>
<td>enab multi</td>
</tr>
<tr>
<td>Max legacy path speed</td>
<td>B-link</td>
<td>Reserved</td>
<td>Standby</td>
<td>Reset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Page select</td>
<td>Res</td>
<td>Port select</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Register 0 (page select)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Register 7 (page select)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Every port has 8 pages
Every page has 8 registers

Register numbers

Extended Registers
### Extended PHY registers - Page 0

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000b</td>
<td>A Status</td>
<td>B Status</td>
<td>Ch</td>
<td>Con</td>
<td>RX OK</td>
<td>Dis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1001b</td>
<td>Negotiated speed</td>
<td>Int enbl</td>
<td>Fault</td>
<td>Standby fault</td>
<td>Disable Scrambl</td>
<td>Beta only</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010b</td>
<td>DC con</td>
<td>Max port speed</td>
<td>LPP</td>
<td>Cable speed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1011b</td>
<td>Reserved</td>
<td>Reserved</td>
<td>Beta Mode</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1100b</td>
<td>Port Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1101b</td>
<td>Reserved</td>
<td>Loop Disable</td>
<td>Standby Disable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1110b</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1111b</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Extended Register - Page 0

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AStat</td>
<td>0 invalid, 1</td>
</tr>
<tr>
<td>BStat</td>
<td>Valid only on DS port, Same encoding as AStat</td>
</tr>
<tr>
<td>Con</td>
<td>Connected and operating speed negotiation complete</td>
</tr>
<tr>
<td>RX OK</td>
<td>DS mode - Receiving a TPBias, Beta mode - receiving a continuous electrically valid signal</td>
</tr>
<tr>
<td>Standby fault</td>
<td>Error was detected during transition to standby or active</td>
</tr>
<tr>
<td>Dis Scrambler</td>
<td>Used for test only</td>
</tr>
<tr>
<td>Beta only</td>
<td>Port not capable of DS mode</td>
</tr>
<tr>
<td>DC Con</td>
<td>Port has detected a DC connection to its peer</td>
</tr>
<tr>
<td>LPP</td>
<td>Local plug present</td>
</tr>
<tr>
<td>Con unreliable</td>
<td>Beta mode speed negotiation has failed</td>
</tr>
<tr>
<td>Loop Disable</td>
<td>Port is disabled to prevent a loop, cleared on bus reset and disconnection</td>
</tr>
<tr>
<td>Standby</td>
<td>Port is in standby mode</td>
</tr>
<tr>
<td>Hard disable</td>
<td>If port is in disable mode, forces re-negotiation of the operating mode and speed</td>
</tr>
</tbody>
</table>
### Extended PHY registers - Page 1

<table>
<thead>
<tr>
<th>Compliance level</th>
<th>00h</th>
<th>01h</th>
<th>02h</th>
<th>1394a</th>
<th>1394b</th>
<th>all others</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Compliance level**
- **Reserved**
- **Vendor ID**
- **Product ID**

**Sect 17: 1394b High Speed/Long Distance**
## PIL - FOP Interface packet

<table>
<thead>
<tr>
<th>any</th>
<th>PP1</th>
<th>PP2</th>
<th>DB1</th>
<th>DB2</th>
<th>DE</th>
<th>DE</th>
<th>any</th>
</tr>
</thead>
</table>

- Packet prefix ➔ Data bytes ➔ Data end ➔

### Sect 17: 1394b High Speed/Long Distance
Review

1. Contrast the 1394 a Phy with 1394 b.
2. How does 86/10b ensure synchronization between sender and receiver?
3. Why do we care about disparity?
4. How does BOSS mode arbitration work?
5. Why use BOSS arbitration?
6. What are the new interconnects?
7. Name the benefit of each new interconnect.
Appendix A

IEEE 1394-1995

Table of Contents

Note: This Table of Contents is to be used with the IEEE Standard for a High Performance Serial Bus, IEEE Std 1394-1995 which was copyrighted by IEEE in 1995. This Table of Contents is not part of the copyrighted document, but is provided to assist the user to locate information in the document more efficiently. This Table of Contents was compiled by Hugh Curley.
1. Overview
1.1 Scope
1.2 References
1.3 Document organization
1.4 Serial Bus applications
  1.4.1 Alternate bus
  1.4.2 Low cost peripheral bus
  1.4.3 Bus bridge
1.5 Service model
1.6 Document notation
  1.6.1 Mechanical notation
  1.6.2 Signal naming
  1.6.3 Size notation
  1.6.4 Numerical values
  1.6.5 Packet formats
  1.6.6 Register formats
  1.6.7 C++ code notation
  1.6.8 State machine notation
  1.6.9 CSR, ROM, and field notation
  1.6.10 Register specification format
  1.6.11 Reserved registers and fields
  1.6.12 Operation description priorities
1.7 Compliance
  1.7.1 CSR Architecture compliance
  1.7.2 Serial Bus physical layers
2. Definitions and Abbreviations
  2.1 Conformance glossary
  2.2 Technical glossary
3. Summary description (informative)
  3.1 Node and module architectures
  3.2 Topology
    3.2.1 Cable environment
    3.2.2 Backplane environment
  3.3 Addressing
  3.4 Protocol architecture
    3.4.1 Data Transfer Services
3.5 Transaction layer
  3.5.1 Transaction layer services
  3.5.2 Lock subcommands
  3.5.3 Subaction queue independence
3.6 Link layer
  3.6.1 Link layer services
  3.6.2 Link and transaction layer interactions
  3.6.2.1 Unified transactions
  3.6.2.2 Split transactions
  3.6.2.3 Subaction concatenation
  3.6.2.4 Retries
  3.6.3 Asynchronous arbitration
  3.6.4 Isochronous arbitration
3.7 Physical layer
  3.7.1 Data bit transmission and reception
  3.7.2 Fair arbitration
  3.7.3 Cable physical layer
    3.7.3.1 Cable configuration
    3.7.3.1.1 Bus initialize
    3.7.3.1.2 Tree identify
    3.7.3.1.3 Self identify
    3.7.3.2 Normal arbitration
    3.7.3.3 Speed signaling
    3.7.3.4 Cable media interface
    3.7.4 Backplane physical layer
      3.7.4.1 Backplane arbitration
      3.7.4.2 Urgent arbitration
      3.7.4.3 Backplane media interface
3.8 Bus management
4. Cable physical layer specification
  4.1 Cable physical layer services
    4.1.1 Cable physical layer bus management services
      for the management layer
    4.1.1.1 PHY control request
    4.1.1.2 PHY control confirmation
    4.1.1.3 PHY event indication
    4.1.2 PHY layer arbitration services for the link layer
      4.1.2.1 PHY arbitration request
      4.1.2.2 PHY arbitration confirmation
      4.1.3 PHY layer data services for the link layer
        4.1.3.1 PHY clock indication
        4.1.3.2 PHY data request
        4.1.3.3 PHY data indication

Appendix A: IEEE Standards Table of Contents
4.2 Cable physical connection specification
  4.2.1 Media attachment
  4.2.1.1 Connectors
  4.2.1.1.1 Connector plug
  4.2.1.1.2 Connector plug terminations
  4.2.1.1.3 Connector socket
  4.2.1.1.4 Positive retention
  4.2.1.1.5 Contact finish on plug and socket contacts
  4.2.1.1.6 Termination finish on plug and contact socket terminals.
  4.2.1.1.7 Shell finish on plugs and sockets.
  4.2.1.2 Cables
  4.2.1.2.1 Cable material
  4.2.1.2.2 Cable assemblies
  4.2.1.3 Connector and cable assembly performance criteria
  4.2.1.3.1 Performance group A: Basic mechanical dimensional conformance & electrical functionality when subjected to mechanical shock & vibration
  4.2.1.3.2 Performance group B: Low level contact resistance when subjected to thermal shock & humidity stress
  4.2.1.3.3 Performance group C: Insulator integrity when subjected to thermal shock & humidity stress
  4.2.1.3.4 Performance group D: Contact life & durability when subjected to mechanical cycling & corrosive gas exposure
  4.2.1.3.5 Performance group E: Contact resistance & unmating force when subjected to temperature life stress
  4.2.1.3.6 Performance group F: Mechanical retention and durability
  4.2.1.3.7 Performance group G: General tests
  4.2.1.4 Signal propagation performance
  4.2.1.4.1 Signal impedance
  4.2.1.4.2 Signal pairs attenuation
  4.2.1.4.3 Signal pairs velocity of propagation
  4.2.1.4.4 Signal pairs relative propagation skew
  4.2.1.4.5 Power pair characteristic impedance
  4.2.1.4.6 Power pair DC resistance
  4.2.1.4.7 Crosstalk
  4.2.1.4.8 Shield AC coupling
  4.2.2 Media signal interface
  4.2.2.1 Signal amplitude
  4.2.2.2 Common mode voltage
  4.2.2.3 Speed signaling
  4.2.2.4 Arbitration signal voltages
  4.2.2.5 Input impedance
  4.2.2.6 Noise
  4.2.2.7 Power and ground
  4.2.2.8 Driver and receiver fault protection.
  4.2.3 Media signal timing
  4.2.3.1 Data rate
  4.2.3.2 Data signal rise and fall times
  4.2.3.3 Jitter and skew
  4.2.3.4 Self-ID packet
  4.2.3.5 Link-on packet
  4.2.3.6 PHY configuration packet
  4.2.3.7 Cable physical layer line states
  4.2.3.8 Cable physical layer timing constants
  4.2.3.9 Gap timing
  4.2.3.10 Cable physical layer node constants
  4.2.3.11 Node variables
  4.2.3.12 Port variables
  4.2.4 Cable physical layer operation
  4.2.4.1 Data transmission and reception
  4.2.4.1.1 Cable environment data transmission
  4.2.4.1.2 Cable environment data reception and repeat
  4.2.4.2 Cable environment arbitration
  4.2.4.2.1 Bus Reset
  4.2.4.2.2 Tree identify
  4.2.4.2.2.1 Tree-ID state machine notes
  4.2.4.2.2.2 Tree-ID actions and conditions
  4.2.4.2.3 Tree-ID actions and conditions
  4.2.4.3 Bus reset state machine notes
  4.2.4.4 Bus reset actions and conditions
  4.2.4.5 Tree-ID state machine notes
  4.2.4.6 Tree-ID actions and conditions

Appendix A: IEEE Standards Table of Contents

© Knowledge Tek rev 5.41b 15.G31
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.2.3</td>
<td>Self identify</td>
<td>103</td>
</tr>
<tr>
<td>4.4.2.3.1</td>
<td>Self-ID state machine notes</td>
<td>103</td>
</tr>
<tr>
<td>4.4.2.3.2</td>
<td>Self-ID actions and conditions</td>
<td>104</td>
</tr>
<tr>
<td>4.4.2.4</td>
<td>Normal arbitration</td>
<td>106</td>
</tr>
<tr>
<td>4.4.2.4.1</td>
<td>Normal arbitration state machine notes</td>
<td>108</td>
</tr>
<tr>
<td>4.4.2.4.2</td>
<td>Normal arbitration actions and conditions</td>
<td>110</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Gap timing</td>
<td>126</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Arbitration sequence</td>
<td>127</td>
</tr>
<tr>
<td>5.3.4.1</td>
<td>Arbitration number</td>
<td>127</td>
</tr>
<tr>
<td>5.3.4.2</td>
<td>Priority</td>
<td>127</td>
</tr>
<tr>
<td>5.3.4.3</td>
<td>Format of arbitration sequence</td>
<td>128</td>
</tr>
<tr>
<td>5.4.4</td>
<td>Backplane physical layer operation</td>
<td>128</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Arbitration</td>
<td>129</td>
</tr>
<tr>
<td>5.4.1.1</td>
<td>Fairness intervals</td>
<td>129</td>
</tr>
<tr>
<td>5.4.1.2</td>
<td>Fair arbitration</td>
<td>130</td>
</tr>
<tr>
<td>5.4.1.3</td>
<td>Urgent arbitration</td>
<td>131</td>
</tr>
<tr>
<td>5.4.1.4</td>
<td>Arbitration by the cycle master</td>
<td>132</td>
</tr>
<tr>
<td>5.4.1.5</td>
<td>Isochronous arbitration</td>
<td>132</td>
</tr>
<tr>
<td>5.4.1.6</td>
<td>Immediate arbitration</td>
<td>132</td>
</tr>
<tr>
<td>5.4.2.1</td>
<td>Backplane environment packet transmission and reception</td>
<td>133</td>
</tr>
<tr>
<td>5.4.2.2</td>
<td>Backplane environment packet reception</td>
<td>134</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Backplane PHY reset</td>
<td>135</td>
</tr>
<tr>
<td>5.5.1.1</td>
<td>Command reset</td>
<td>135</td>
</tr>
<tr>
<td>5.5.1.2</td>
<td>Bus reset</td>
<td>135</td>
</tr>
<tr>
<td>5.5.2</td>
<td>Backplane PHY initialization</td>
<td>135</td>
</tr>
<tr>
<td>5.2</td>
<td>Backplane physical connection specification</td>
<td>117</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Media attachment</td>
<td>118</td>
</tr>
<tr>
<td>5.2.1.1</td>
<td>Distribution of nodes</td>
<td>118</td>
</tr>
<tr>
<td>5.2.1.2</td>
<td>Fault detection and isolation</td>
<td>118</td>
</tr>
<tr>
<td>5.2.1.3</td>
<td>Live insertion</td>
<td>118</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Media signal interface</td>
<td>118</td>
</tr>
<tr>
<td>5.2.2.1</td>
<td>Definition of logic states</td>
<td>119</td>
</tr>
<tr>
<td>5.2.2.2</td>
<td>Bit rates</td>
<td>119</td>
</tr>
<tr>
<td>5.2.2.3</td>
<td>Transition times</td>
<td>119</td>
</tr>
<tr>
<td>5.2.2.4</td>
<td>Noise rejection</td>
<td>119</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Media signal timing</td>
<td>119</td>
</tr>
<tr>
<td>5.2.3.1</td>
<td>Backplane transmit data timing</td>
<td>119</td>
</tr>
<tr>
<td>5.2.3.2</td>
<td>Backplane receive data timing</td>
<td>120</td>
</tr>
<tr>
<td>5.2.3.3</td>
<td>Backplane and transceiver skew</td>
<td>121</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Backplane physical layer timing</td>
<td>122</td>
</tr>
<tr>
<td>5.2.4.1</td>
<td>Arbitration Clock Rate</td>
<td>122</td>
</tr>
<tr>
<td>5.2.4.2</td>
<td>Bus synchronization and propagation delay</td>
<td>122</td>
</tr>
<tr>
<td>5.2.4.3</td>
<td>Arbitration bit timing</td>
<td>123</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Coding</td>
<td>125</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Backplane physical layer signals</td>
<td>125</td>
</tr>
</tbody>
</table>
6.1.3 Link layer isochronous data services for application layers 142
6.1.3.1 Link isochronous control request 142
6.1.3.2 Link cycle synch indication 142
6.1.3.3 Link isochronous request 142
6.1.3.4 Link isochronous indication 143
6.2 Link layer facilities 143
6.2.1 Primary packets 144
6.2.2 Asynchronous packets 145
6.2.2.1 Asynchronous packets with no-data payload 146
6.2.2.1.1 Read request for data quadlet 146
6.2.2.1.2 Write response 147
6.2.2.2 Asynchronous packet formats with data quadlet payload 147
6.2.2.2.1 Read request for data block 148
6.2.2.2.2 Write request for data quadlet 148
6.2.2.2.3 Cycle start 148
6.2.2.2.4 Read response for data quadlet 150
6.2.2.3 Asynchronous packet formats with data block payload 150
6.2.2.3.1 Read request for data block 151
6.2.2.3.2 Lock request 152
6.2.2.3.3 Read response for data block 153
6.2.2.3.4 Lock response 153
6.2.3 Isochronous packets 154
6.2.3.1 Isochronous data-block packet format 154
6.2.4 Primary packet components 155
6.2.4.1 Reserved fields, codes, and values 155
6.2.4.2 Destination address 155
6.2.4.2.1 Destination ID 156
6.2.4.2.2 Destination offset 156
6.2.4.3 Transaction label (tl) 156
6.2.4.4 Retry code (rt) 157
6.2.4.5 Transaction code (tcode) 157
6.2.4.6 Priority (pri) 158
6.2.4.7 Source ID 158
6.2.4.8 Data length 158
6.2.4.9 Extended transaction code 158
6.2.4.10 Response code (rcode) 159
6.2.4.11 Data field 159
6.2.4.12 Tag 159
6.2.4.13 Channel 159
6.2.4.14 Synchronization code (sy) 159
6.2.4.15 CRCs 160
6.2.4.15.1 Definitions 160
6.2.4.15.2 CRC generation equations 160
6.2.4.15.3 CRC checking 160
6.2.5 Acknowledge packets 161
6.2.5.1 Acknowledge packet format 161
6.2.5.2 ACK packet components 161
6.2.5.2.1 Reserved acknowledge codes 161
6.2.5.2.2 Acknowledge code 162
6.2.5.2.3 Acknowledge parity 162
6.3 Link layer operation 162
6.3.1 Overview of link layer operation (informative) 163
6.3.1.1 Communication with the PHY layer 163
6.3.1.2 Sending an asynchronous packet 163
6.3.1.3 Receiving an asynchronous packet 163
6.3.1.4 Sending an acknowledge concatenated to an asynchronous packet. 164
6.3.1.5 Isochronous cycles 164
6.3.1.6 Sending isochronous packets 164
6.3.1.7 Receiving an isochronous packet 164
6.3.2 Cycle synch event 165
6.3.3 Details of link layer operation 166
6.3.3.1 Link initialization 167
6.3.3.2 Asynchronous operation 167
6.3.3.3 Isochronous operation 169
6.4 Link layer reference code 171
7. Transaction layer specification 173
7.1 Transaction layer services 173
7.1.1 Transaction layer bus management services for Serial Bus management 173
7.1.1.1 Transaction control request. 173
7.1.1.2 Transaction control confirmation 174
7.1.1.3 Transaction event indication 174
7.1.2 Transaction layer data services for applications and bus management. 174
7.1.2.1 Transaction data request 174
7.1.2.2 Transaction data confirmation 175
7.1.2.3 Transaction data indication 175
7.1.2.4 Transaction data response 176

Appendix A: IEEE Standards Table of Contents
7.2 Transaction facilities
7.2.1 Split transaction timer 176
7.2.2 Transaction retry limit 176

7.3 Transaction operation
7.3.1 Overview of transaction layer operations (informative) 177
7.3.1.1 Read transactions 177
7.3.1.2 Write transactions 177
7.3.1.3 Lock transactions 178
7.3.1.4 Error handling 178
7.3.2 Transaction completion definitions 179
7.3.2.1 Unified transaction 179
7.3.2.2 Split transaction 179
7.3.2.3 Concatenated transaction 179
7.3.2.4 Broadcast transaction 179
7.3.2.5 Pending transaction 179
7.3.3 Details of transaction layer operation
7.3.3.1 Outbound transaction state machine 180
7.3.3.2 Inbound transaction state machine 183
7.3.3.2.1 Inbound transaction state machine initialization 184
7.3.3.2.2 Responding to a transaction request 185
7.3.3.2.3 Responding to a transaction response 186
7.3.4 Transaction types 187
7.3.4.1 Read transactions 187
7.3.4.2 Write transactions 187
7.3.4.3 Lock transactions 187
7.3.5 Retry protocols 188
7.3.5.1 Single-phase retry protocol 188
7.3.5.1.1 Initialization of inbound retry 189
7.3.5.1.2 Inbound-retry operation 189
7.3.5.1.3 Initialization of outbound retry 189
7.3.5.1.4 Outbound-retry operation 190
7.3.5.2 Dual-phase retry protocol 190
7.3.5.2.1 Initialization of inbound retry 191
7.3.5.2.2 Inbound-retry operation 192
7.3.5.2.3 Initialization of outbound retry 196
7.3.5.2.4 Outbound-retry operation 196

8. Serial Bus management specification
8.1 Serial Bus management summary
8.1.1 Node control 199
8.1.2 Isochronous resource manager (cable environment) 199
8.1.3 Isochronous resource manager (backplane environment) 199
8.1.4 Bus manager (cable environment) 199

8.2 Serial Bus management services
8.2.1 Serial Bus control request 200
8.2.2 Serial Bus control confirmation 201
8.2.3 Serial Bus event indication 201

8.3 Serial Bus management facilities
8.3.1 Node capabilities taxonomy 203
8.3.1.1 Repeater (cable environment) 203
8.3.1.2 Transaction capable 203
8.3.1.3 Isochronous capable 203
8.3.1.4 Cycle master capable 204
8.3.1.5 Isochronous resource manager capable 204
8.3.1.6 Bus manager capable (cable environment) 204
8.3.2 Command and status registers
8.3.2.1 Reset conditions 205
8.3.2.2 CSR Architecture core registers
8.3.2.2.1 STATE_CLEAR register 206
8.3.2.2.2 STATE_SET register 208
8.3.2.2.3 NODE_IDS register 208
8.3.2.2.4 RESET_START register 209
8.3.2.2.5 INDIRECT_ADDRESS and INDIRECT_DATA registers 209
8.3.2.2.6 SPLIT_TIMEOUT register 209
8.3.2.2.7 ARGUMENT, TEST_START and TEST_STATUS registers 210
8.3.2.2.8 UNITS_BASE, UNITS_BOUND, MEMORY_BASE and MEMORY_BOUND registers 210
8.3.2.2.9 INTERRUPT_TARGET and INTERRUPT_MASK registers 210
8.3.2.2.10 CLOCK_VALUE, CLOCK_TICK_PERIOD, CLOCK_STROBE_ARRIVED and CLOCK_INFO registers 210
8.3.2.2.11 MESSAGE_REQUEST and MESSAGE_RESPONSE registers 210
8.3.2.3 Serial Bus-dependent registers 210
8.3.2.3.1 CYCLE_TIME register 211
8.3.2.3.2 BUS_TIME register
8.3.2.3.3 POWER_FAIL_IMMINENT register
8.3.2.3.4 POWER_SOURCE register
8.3.2.3.5 BUS_TIMEOUT register
8.3.2.3.6 BUS_MANAGER_ID register
8.3.2.3.7 BANDWIDTH_AVAILABLE register
8.3.2.3.8 CHANNELS_AVAILABLE register
8.3.2.3.9 MAINT_CONTROL register
8.3.2.3.10 MAINTUTILITY register
8.3.2.4 Unit registers
8.3.2.4.1 TOPOLOGY_MAP registers (cable environment)
8.3.2.4.2 SPEED_MAP registers (cable environment)
8.3.2.5 Configuration ROM
8.3.2.5.1 Company ID
8.3.2.5.2 Minimal ROM format
8.3.2.5.3 General ROM format
8.3.2.5.4 Bus Info Block
8.3.2.5.5 Root Directory
8.3.2.5.5.1 Module Vendor Id entry
8.3.2.5.5.2 Node Capabilities entry
8.3.2.5.5.3 Node Unique Id entry
8.3.2.5.6 Unit Directories
8.3.2.5.6.1 Unit Power Requirements entry (cable environment)
8.3.2.5.7 Root Leaves
8.3.2.5.7.1 Node Unique Id leaf
8.3.3 Serial Bus management variables
8.4 Serial Bus management operations
8.4.1 Bus configuration procedures (backplane environment)
8.4.1.1 Unmanaged bus (backplane environment)
8.4.1.2 Determination of the isochronous resource manager (backplane environment)
8.4.1.3 Determination of the cycle master (backplane environment)
8.4.2 Bus configuration procedures (cable environment)
8.4.2.1 Unmanaged bus (cable environment)
8.4.2.2 Prior isochronous traffic (cable environment)
8.4.2.3 Determination of the isochronous resource manager (cable environment)
8.4.2.4 Reallocation of prior isochronous resources (cable environment).
8.4.2.5 Determination of the bus manager (cable environment)
8.4.2.6 Determination of the cycle master (cable environment)
8.4.2.7 Power management by the isochronous resource manager (cable environment)
8.4.2.8 Allocation of new isochronous resources (cable environment)
8.4.3 Isochronous management (cable environment)
8.4.3.1 Bandwidth allocation
8.4.3.2 Channel allocation
8.4.3.3 Bandwidth set-aside
8.4.4 Power management (cable environment)
8.4.4.1 PHY power management
8.4.4.2 Link power management
8.4.4.3 Unit power management
8.4.4.4 Power management by the bus manager
8.4.4.5 Power management by the isochronous resource manager
8.4.5 Speed management (cable environment)
8.4.5.1 Accessing the speed map
8.4.6 Topology management (cable environment)
8.4.6.1 Accessing the topology map
8.4.6.2 Gap count optimization
8.5 Bus configuration state machines (cable environment)
8.5.1 Candidate cycle master states
8.5.2 Candidate isochronous resource manager states
8.5.3 Candidate bus manager states
Annex A. Cable environment system properties (normative)
A.1 External shielded cable interconnects
A.1.1 Definitions
A.1.2 Specifications
A.2 Internal unshielded interconnects
A.2.1 Definitions
A.2.2 Specifications
A.3 Cable power sourcing and connection
A.3.1 Definitions
A.3.2 Specifications
A.4 Electrical isolation requirements (summary)
A.5 Examples 247
A.6 Special isolation schemes using PHY/PHY interface
Annex B. External connector positive retention (Normative)
Annex C. Internal device physical interface (normative)

C.1 Overview
C.2 Electrical interface for internal devices
C.2.1 Power requirements
C.2.2 Bus signal requirements
C.2.3 Miscellaneous signals
C.2.4 Signal descriptions
C.3 Internal unitized device connectors
C.3.1 Internal unitized plug
C.3.2 Internal unitized receptacles
C.3.3 Connector cable receptacles
C.3.3.1 Cable receptacle termination
C.3.4 Cable
C.3.4.1 Flat ribbon cable
C.3.4.2 Discrete wire
C.3.5 Contact finish on mating surfaces of plug and receptacle contacts
C.3.6 Termination finish on plug and receptacle contact
C.3.7 Connector performance criteria
C.3.7.1 Test sample preparation
C.3.7.2 Performance group A: Basic mechanical conformance & electrical functionality when subjected to mechanical shock and vibration
C.3.7.3 Performance group B: Low level contact resistance when subjected to thermal shock and humidity stress
C.3.7.4 Performance group C: Insulator integrity when subjected to thermal shock and humidity stress
C.3.7.5 Performance group D: Contact life and durability when subject to mechanical cycling and corrosive gas exposure
C.3.7.6 Performance group E: Contact resistance and mating & unmating force when subjected to temperature life stress
C.3.7.7 Performance group F: Mechanical retention and durability
C.3.7.8 Performance group G: General tests

Annex D. Backplane physical layer timing formulae (Informative)

D.1 Backplane propagation delay
D.2 Backplane arbitration timing
D.2.1 Synchronization timing
D.2.2 Arbitration sample timing
D.2.3 Arbitration hold timing
D.2.4 Arbitration bit timing
D.3 Backplane gap timing
D.3.1 Acknowledge gap
D.3.1.1 Occurrence of acknowledge gap
D.3.1.2 Acknowledge gap timing
D.3.2 Subaction gap and arbitration reset gap
D.3.2.1 Occurrence of subaction and arbitration reset gaps
D.3.2.2 Subaction gap and arbitration reset gap timing
D.3.3 Arbitration gap scenarios
D.3.3.1 Scenario 1: acknowledge gap
D.3.3.2 Scenario 2: subaction gap
D.3.3.3 Scenario 3: arbitration reset gap

D.4 Backplane environment skew

Annex E. Cable operation and implementation examples (informative)

E.1 Timing formulae for cable environment gap control
E.1.1 Arbitration phase
E.1.2 Data transfer phase
E.1.3 Acknowledgment phase
E.1.4 Between packet gap times
E.2 Cable environment jitter budget
E.3 Cable physical layer configuration example
E.3.1 Bus initialize
E.3.2 Tree identify
E.3.3 Self identify
E.3.4 Topology Construction

Annex F. Backplane physical implementation example (informative)
L.5 Sample preparation
  L.5.1 Panel mounted connector sample
  L.5.2 Measure Sample Zo with TDR
  L.5.3 Cable mounted connector sample
L.6 Procedure
L.7 "Noise floor" plot
L.8 Documentation
  L.8.1 Plots and magnetic files
  L.8.2 Test report
L.9 Performance

369
369
370
370
370
370
370
371
371
371
372
372
Appendix C: Isochronous Connection Management
Isochronous Connection Management

Establishment of an Isochronous Stream
Talker, Listener(s), Channel Number, Bandwidth

Two Methods:
Plug Control Registers (PCR)
Streams (Asynchronous Commands to establish)

Refer to Appendix C for details of these two methods
Isochronous Connection Management
Method 1

Plug Control Registers (PCR)
Defined in 1394a and IEEE 1883

Talker
Output Master PCR (1)
Output PCR (1-32)

Listener
Input Master PCR (1)
Input PCR (1-32)
Isochronous Connection Management

Method 2

Streams

Defined in SBP-2

Stream Command Block ORB
  Controls device

Stream Control ORB
  Controls Flow
    Start/Stop/Pause
    Configure Channels (Reassign Channel numbers)
    Update Channel Mask (Assign Channels)
Isochronous Connection Management
Plug Control Registers

Old way - physically plug together

New way - logically plug together
### Isochronous Connection Management

**Plug Control Registers**

**Talker**

<table>
<thead>
<tr>
<th>Output</th>
<th>Master</th>
<th>Plug</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SPD</th>
<th>Broadcast Base</th>
<th>reserved</th>
<th>X Spd</th>
<th>Output Plugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 = S100</td>
<td></td>
<td>0 = S800</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1 = S200</td>
<td></td>
<td>1 = S1600</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2 = S400</td>
<td></td>
<td>2 = S3200</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3 = S000 XSPD</td>
<td></td>
<td>3 = Reserved</td>
<td></td>
</tr>
</tbody>
</table>

- **Spd**
- **X Spd**

Broadcast base used to determine base Isochronous channel number for broadcasts

Output Plugs Quantity of output plug registers implemented by this node
### Isochronous Connection Management

#### Plug Control Registers

<table>
<thead>
<tr>
<th>O</th>
<th>b</th>
<th>Point to Point</th>
<th>X Spd</th>
<th>Channel</th>
<th>Spd</th>
<th>Overhead</th>
<th>Payload</th>
</tr>
</thead>
</table>

- **O**: Online
- **b**: Broadcast connection exists
- **Point to Point**: Number of point to point connections for this plug
- **Spd & X Spd**: Same encoding as output master plug
- **Channel**: Channel number for this plug
- **Overhead**: Allocation units of overhead
- **Payload**: Maximum number of data quadlets transmitted in a single Isochronous packet

0 = 1024 quadlets
## Isochronous Connection Management

**Plug Control Register**

### Input Master Plug

<table>
<thead>
<tr>
<th>Spd</th>
<th>Reserved</th>
<th>Input Plugs</th>
</tr>
</thead>
</table>

### Input Plug

<table>
<thead>
<tr>
<th>Ob</th>
<th>Point to Point</th>
<th>R</th>
<th>Channel</th>
<th>Reserved</th>
</tr>
</thead>
</table>

All fields as defined in outplug plug registers
Isochronous Connection Management

Streams

Procedures

- Login management ORB (covered in SBP-2 section)
  - Exchange addresses for STATUS FIFO and command agent

- Create Streams management ORB (Isochronous login)
  - Exchange addresses for status FIFO, command block agent and stream control agent
  - Exchange bandwidth and channel requirement information

- Issue Streams Command Block ORB
  - Contains commands for device giving starting address, etc.

- Issue Stream control ORB to assign channels

- Issue Stream control ORB to start

- Use Isochronous protocol to transfer data
Isochronous Management
Streams

Create Streams (Isochronous login)

```
<table>
<thead>
<tr>
<th>t</th>
<th>Reserved</th>
<th>Channels</th>
<th>Delta Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>req fmt</td>
<td>function</td>
<td>login ID</td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Status FIFO</td>
</tr>
</tbody>
</table>
```

Appendix C: Isochronous Connection Management
Isochronous Connection Management

<table>
<thead>
<tr>
<th>Delta Time</th>
<th>Range = 0 to 7999. Used to shift Isochronous cycles into the future up to 1 second</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>Talker</td>
</tr>
<tr>
<td>channels</td>
<td>Maximum number of Isochronous channels to be used</td>
</tr>
<tr>
<td>Max payload</td>
<td>Sum of data length of all channels for this talker per Isochronous cycle</td>
</tr>
</tbody>
</table>
Isochronous Connection Management

Streams

Create Stream Response

<table>
<thead>
<tr>
<th>Length</th>
<th>Login ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command Block Agent Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stream Control Agent Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Minimum transfer length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>
Isochronous Connection Management

Streams

Stream Command Block ORB

Next ORB

Cycle Mark Offset

Stream Length

Reserved

Command Block

 Appendix C: Isochronous Connection Management
Isochronous Connection Management

Cycle Mark offset and CM specify the location on the device of the first quadlet of ISO data

- CM = 0 Undefined
- CM = 1 Invalid
- CM = 2 First quadlet located at address given in command block (offset = 0)
- CM = 3 First quadlet located at address given in command block plus cycle mark offset
Isochronous Connection Management
Streams

Stream Control ORB

Next ORB

Stream Control Dependent

n

reg

fmt

Reserved

Stream Control

Stream Event

Sy

rpt

Reserved

Reserved

Second Count Hi

Reserved

Second Count

Cycle Count

Reserved

Appendix C: Isochronous Connection Management
Isochronous Connection Management

Streams

Stream Control

0 = Start
1 = Stop
2 = Pause
3 = Update channel mask
4 = Configure channels
5 = Set Error Mode
6 = Query Stream Status
Isochronous Connection Management

Update Channel Mask
64 bit channel mask, set to 1 to represent which channels are used by this node

Configure Channels
Allows numbering of channels as they are recorded
Isochronous Connection Management Notes
Isochronous Connection Management Notes
Appendix D

1394 Device Bay
Device Bay = Set of Related Standards

Device Bay Interface Specification
- Mechanical
- Power Management
- Device Bay Controller
- Device Classes
- Connectors
- Software
- Legacy Support
- Status Indicators

Includes related standards
- Plug and Play
- Advanced configuration and Power Interface
- 1394 High Performance Serial Bus
- Universal Serial Bus
- On Now
- Device Bay Interface Specification
User Friendly PC

Easy Upgrades
No Jumpers, switches, terminators, cables, or configuration
Hot Pluggable

Easy Sharing of devices or data
Can move devices between desktop and laptop computer
Can move devices/data from one user to another

Easy Security
Remove and lock hard disk when leaving

Easy Repair
Remove defective device and replace with new
Device Bay - Device Classes

Storage

Hard disks (Fixed and Removable)
Tape Drives
CD-Rom, DVD-Rom, VCR, Cam Corders, Set Top Boxes

Communication and Connectivity

Modems - POTS, ISDN
LAN
IR, RF
Graphics, Video, Audio, Internet, Intranet

Data Security

User authentication

Non-Compliant Connector

Devices that don’t conform to USB/1394 or Power Requirements
Batteries, etc.
Device Bay
Form Factors

DB 32

32.00 X 146.00 X 178.00 mm
1.260” high X 5.748” wide X 7.008” long
Device Bay
Form factors

DB 20

20.00 X 130.00 X 141.550 mm
.787” high X 5.118” wide X 5.571” long
Device Bay
Form Factors

DB 13

13.00 X 130.00 X 141.50 mm
.512” high X 5.118” wide X 5.571” long
Device Bay
DB 32 Mechanical

Grip area
Never touch area
Ejector Points
Non-compliant connector
Connector

EMI/ESD Pad
Retention Areas

Maximum Weight = 1.40 kg
3.08 lbs
Appendix D: Overview of Device Bay

Device Bay
DB20/13 Mechanical

- Grip Areas
- Device Bay
- DB20/13 Mechanical
- EMI/ESD Pad
- Retention Area
- Ejector Points
- Connector
- Non-Compliant Connector Area

DB20 Maximum Weight = 0.50 kg
1.10 lbs
DB13 Maximum Weight = 0.35 kg
0.77 lbs
Device Bay - Connector

Same connector for all 3 form factors
Blind Mating
Long life (minimum of 2,500 insertions)
Plug is in device, Receptacle in Bay

Single Connector for 1394, USB, and Power
Either PCB or cable mounted
Hot Insertion and Removal (Ground and Vid mate 1st)
# Device Bay Connector - Device Side

<table>
<thead>
<tr>
<th>A14</th>
<th>Ground</th>
<th>Ground</th>
<th>A1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A15</td>
<td>Ground</td>
<td>(1394) TPA</td>
<td>A2</td>
</tr>
<tr>
<td>A16</td>
<td>Ground</td>
<td>(1394) TPA *</td>
<td>A3</td>
</tr>
<tr>
<td>A17</td>
<td>Ground</td>
<td>Ground</td>
<td>A4</td>
</tr>
<tr>
<td>A18</td>
<td>Ground</td>
<td>(1394) TPB</td>
<td>A5</td>
</tr>
<tr>
<td>A19</td>
<td>Ground</td>
<td>(1394) TPB *</td>
<td>A6</td>
</tr>
<tr>
<td>A20</td>
<td>Ground</td>
<td>Ground</td>
<td>A7</td>
</tr>
<tr>
<td>A21</td>
<td>Ground</td>
<td>(1394) PRSN#</td>
<td>A8</td>
</tr>
<tr>
<td>A22</td>
<td>Ground</td>
<td>DEV_ACT#</td>
<td>A9</td>
</tr>
<tr>
<td>A23</td>
<td>Ground</td>
<td>(USB) PRSM #</td>
<td>A10</td>
</tr>
<tr>
<td>A24</td>
<td>Ground</td>
<td>(USB) D+</td>
<td>A11</td>
</tr>
<tr>
<td>A25</td>
<td>Ground</td>
<td>(USB) D-</td>
<td>A12</td>
</tr>
<tr>
<td>A26</td>
<td>Reserved</td>
<td>V id</td>
<td>A13</td>
</tr>
</tbody>
</table>

## GAP

<table>
<thead>
<tr>
<th>B10</th>
<th>Ground</th>
<th>V 3.3</th>
<th>B1</th>
</tr>
</thead>
<tbody>
<tr>
<td>B11</td>
<td>Ground</td>
<td>V 3.3</td>
<td>B2</td>
</tr>
<tr>
<td>B12</td>
<td>Ground</td>
<td>V 3.3</td>
<td>B3</td>
</tr>
<tr>
<td>B13</td>
<td>Ground</td>
<td>V 3.3</td>
<td>B4</td>
</tr>
<tr>
<td>B14</td>
<td>Ground</td>
<td>V 3.3</td>
<td>B5</td>
</tr>
<tr>
<td>B15</td>
<td>Ground</td>
<td>V 3.3</td>
<td>B6</td>
</tr>
<tr>
<td>B16</td>
<td>V 5.0</td>
<td>V 12</td>
<td>B7</td>
</tr>
<tr>
<td>B17</td>
<td>V 5.0</td>
<td>V 12</td>
<td>B8</td>
</tr>
<tr>
<td>B18</td>
<td>V 5.0</td>
<td>V 12</td>
<td>B9</td>
</tr>
</tbody>
</table>

---

© KnowledgeTek
rev 5.41b 15.G.31

Appendix D: Overview of Device Bay
Device Bay
Power

V id
- 3.3 VDC for identification
- Supplied and switched by device bay upon detection of a device presents
- Must be disabled if no device inserted
- Must not be enabled unless operating system enables

V op
- Supplied by device bay but switched by device
- Supplies 12 VDC, 5.0 VDC and 3.3 VDC
Device Bay
Power Sequence

1. Bay may be supplying Vop or not, but is not supplying Vid
2. Device is inserted
3. 1394 PRSN# or USB PRSN# is asserted low
4. Bay supplies Vid
5. Bay supplies Vop
6. Device switches on
Device Bay Controller

Device Bay 0

Device Bay 1

1394
USB
Power
Control

Remove Req
Interlock

Remove Req
Interlock

Appendix D: Overview of Device Bay
Device Bay Controller

Status-        Status Indicator
             Device Inserted
             Removal Request

Interlock-     Prevents Removal

Remove Req-    Removal Request Button

Control-       USB PRSN# (USB Bus Present)
               1394 PRSN# (1394 Bus Present)

DBC            Device Bay Controller
               Node on the 1394 Bus - Repeats to the device connectors
               Controls Bay Hardware
               Controls Bay Power
Device Bay Config ROM - Bus Info Block

Note: This should be upgraded in spec to match 1394a.
**Device Bay Config ROM - Required Root Directory Entries**

<table>
<thead>
<tr>
<th>Root directory length</th>
<th>Root directory CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td>module_vendor_ID</td>
</tr>
<tr>
<td>0C</td>
<td>node_capabilities</td>
</tr>
<tr>
<td>8D</td>
<td>node_unique_ID offset</td>
</tr>
<tr>
<td>D1</td>
<td>unit_directory_offset</td>
</tr>
<tr>
<td>2</td>
<td>node_unique_ID CRC</td>
</tr>
</tbody>
</table>

| node_vendor_ID | chip_ID_lo | chip_ID_hi |

**Bus Info Block Previous**

- **spt**: Split Timeout = 0 (not implemented)
- **drq**: Disable Request = 0
- **64**: Use 64 bit addressing = 1
- **fix**: Use fixed addressing = 1
- **lst**: Support “lost” bit = 1

**Appendix D: Overview of Device Bay**
## Device Bay Config ROM - Required Unit Directory Entries

### Root Directory Previous

<table>
<thead>
<tr>
<th>Unit directory Length</th>
<th>Unit directory CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$12_{16}$</td>
<td>unit_spec_ID (00 80 5F)$_{16}$</td>
</tr>
<tr>
<td>$13_{16}$</td>
<td>unit_sw_version (01 00 00)$_{16}$</td>
</tr>
</tbody>
</table>

Note: Different Standards Body than SCSI
# Device Bay Controller Required CSR's

<table>
<thead>
<tr>
<th>CSR Offset</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>State clear</td>
</tr>
<tr>
<td>04h</td>
<td>State set</td>
</tr>
<tr>
<td>08h</td>
<td>Node ID</td>
</tr>
<tr>
<td>0Ch</td>
<td>Reset Start</td>
</tr>
<tr>
<td>18h-1Ch</td>
<td>Split Timeout (only if DBC can be a requester)</td>
</tr>
<tr>
<td>210h</td>
<td>Busy Timeout (only if DBC supports retries)</td>
</tr>
</tbody>
</table>
## DBC Control Registers

<table>
<thead>
<tr>
<th>Register</th>
<th>Index Offset</th>
<th>Width</th>
<th>R/Lo</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor ID</td>
<td>00h</td>
<td>16</td>
<td>RO</td>
<td>Vendor ID - same as PCI SIG Vendor ID</td>
</tr>
<tr>
<td>Revision ID</td>
<td>04h</td>
<td>8</td>
<td>RO</td>
<td>Vendor chosen revision number</td>
</tr>
<tr>
<td>Subsystem Vendor ID</td>
<td>08h</td>
<td>16</td>
<td>RO</td>
<td>Device Bay Vendor ID</td>
</tr>
<tr>
<td>Subsystem ID</td>
<td>0Ah</td>
<td>16</td>
<td>RO</td>
<td>Subsystem revision ID</td>
</tr>
<tr>
<td>DBCCR</td>
<td>0Ch</td>
<td>32</td>
<td>RO</td>
<td>Device Bay Controller Capabilities Register</td>
</tr>
<tr>
<td>BSTRO</td>
<td>10h</td>
<td>32</td>
<td>RO</td>
<td>Bay 0 Status Register</td>
</tr>
<tr>
<td>BCERO</td>
<td>14h</td>
<td>32</td>
<td>R/W</td>
<td>Bay 0 Control and Enable Reg</td>
</tr>
<tr>
<td>BSTRI</td>
<td>18h</td>
<td>32</td>
<td>RO</td>
<td>Bay 1 Status</td>
</tr>
<tr>
<td>BCERI</td>
<td>1Ch</td>
<td>32</td>
<td>R/W</td>
<td>Bay 1 Control and Enable Reg</td>
</tr>
<tr>
<td>BSTR (N-1)</td>
<td>8(n-1)+10h</td>
<td>32</td>
<td>RO</td>
<td>Bay (n-1) Status</td>
</tr>
<tr>
<td>BCER (N-1)</td>
<td>8(n-1)+14h</td>
<td>32</td>
<td>R/W</td>
<td>Bay (n-1) Control and Enable Reg</td>
</tr>
</tbody>
</table>
## DBC - Bay Status Register

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31-11</td>
<td>Reserved</td>
<td>R/O</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
| 10-8  | Bay Form factor (Not device FF)| R/O    | 000 = DB32
001 = DB20
010 = DB13
011 - 111 = Reserved |
| 7     | Security lock status          | R/O    | 1 = Physical security lock engaged             |
| 6-4   | Bay status                    | R/O    | 000 = Bay Empty
001 = Device Inserted
010 = Device Enabled
011 = Removal Requested
100 = Removal Allowed
101-111 = Reserved |
| 3     | Removal Request               | R/WC   | Eject button has been pressed. Can only be cleared by writing a 1 to it |
| 2     | Device Status Changed         | R/WC   | Device status has changed                       |
| 1     | 1394 PRSN                     | R/O    | 1394 Device present in bay                      |
| 0     | USB PRSN                      | R/O    | USB Device present in Bay                       |

__Appendix D: Overview of Device Bay__
## DBC - Bay Control and Enable Register

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31-8</td>
<td>Reserved</td>
<td>R/O</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
| 7   | Lock Control          | R/W    | $1 =$ Lock Engaged  
$0 =$ Clear Lock                                                              |
| 6-4 | Requested Status      | R/W    | Status as requested by the operating system  
$000 =$ No change to bay state requested  
$001 =$ Change to device inserted  
$010 =$ Change to device enabled  
$011 =$ Change to removal requested  
$100 =$ Change to removal allowed  
$101-111 =$ Reserved |
| 3   | Removal Request       | R/W    | $1 =$ generate an interrupt on removal request                              |
| 2   | Device Status Changed | R/W    | $1 =$ generate an interrupt on device status change                          |
| 1   | Removal               | R/W    | $1 =$ generate an interrupt on device removal action                         |
| 0   | Vid                   | R/W    | $1 =$ enable Vid power                                                      |
Device Bay Notes

Appendix D: Overview of Device Bay
Appendix R:
Reduced Block Command Set (RBC)
## Reduced Block Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>OP Code</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inquiry</td>
<td>12h</td>
<td>SPC-2</td>
</tr>
<tr>
<td>Mode Select</td>
<td>55h</td>
<td>SPC-2</td>
</tr>
<tr>
<td>Mode Sense</td>
<td>5Ah</td>
<td>SPC-2</td>
</tr>
<tr>
<td>Read (10)</td>
<td>28h</td>
<td>RBC</td>
</tr>
<tr>
<td>Start/ Stop Unit</td>
<td>1Bh</td>
<td>RBC</td>
</tr>
<tr>
<td>Synchronize lock</td>
<td>35h</td>
<td>RBC</td>
</tr>
<tr>
<td>Test Unit Ready</td>
<td>00h</td>
<td>SPC-2</td>
</tr>
<tr>
<td>Write (10)</td>
<td>2Ah</td>
<td>RBC</td>
</tr>
<tr>
<td>Write &amp; Verify (10)</td>
<td>2Eh</td>
<td>RBC</td>
</tr>
<tr>
<td>Write Buffer</td>
<td>3Bh</td>
<td>SPC-2</td>
</tr>
</tbody>
</table>

*NOTES:* Read (6) and Write (6) are not included. Request Sense is not used because 1394 (and other serial interfaces) provide autosense.
Reduced Block Commands Inquiry

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OP code = 12h</td>
</tr>
<tr>
<td>1</td>
<td>Reserved</td>
</tr>
<tr>
<td>2</td>
<td>Page or OP code</td>
</tr>
<tr>
<td>3</td>
<td>Reserved</td>
</tr>
<tr>
<td>4</td>
<td>Allocation Length</td>
</tr>
<tr>
<td>5</td>
<td>Control</td>
</tr>
</tbody>
</table>

CMD DT EUPD
## Reduced Block Commands

### Mode Select

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OP code = 55h</td>
</tr>
<tr>
<td>1</td>
<td>PF=1</td>
</tr>
<tr>
<td>2</td>
<td>Reserved</td>
</tr>
<tr>
<td>3</td>
<td>Reserved</td>
</tr>
<tr>
<td>4</td>
<td>Reserved</td>
</tr>
<tr>
<td>5</td>
<td>Reserved</td>
</tr>
<tr>
<td>6</td>
<td>Reserved</td>
</tr>
<tr>
<td>7 (MSB)</td>
<td>Parameter List Length</td>
</tr>
<tr>
<td>8</td>
<td>(LSB)</td>
</tr>
<tr>
<td>9</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

© KnowledgeTek
rev 5.41b  15.G.31

Appendix R: Reduced Block Command Set
# Reduced Block Commands

## Mode Sense

<table>
<thead>
<tr>
<th>Page Code</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td>OP code = 5Ah</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>Reserved</td>
<td>DBD=1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>Page Code</td>
<td>Page Code</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 (MSB)</td>
<td></td>
<td></td>
<td>Allocation Length</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>(LSB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td>Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Page Code:
- 0 - Current-optional
- 1 - Changeable - not supported
- 2 - Default - Mandatory
- 3 - Saved - Mandatory
<table>
<thead>
<tr>
<th></th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PS=1</td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Page code = 3Eh</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Page Length = 8</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WCD</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(MSB)</td>
<td></td>
<td>Logical Block Size</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(LSB)</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(MSB)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Number of Logical Blocks</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(LSB)</td>
</tr>
</tbody>
</table>
WCD reflects the setting of the WCD list in the Synchronize Cache command and is not changeable with mode select.

Logical block size is not changeable

Number of Logical Blocks is changeable
   To discover the default number of blocks, issue Mode sense with PC = Default

   To discover the current number of blocks, issue Mode sense with PC = Saved
Reduced Block Commands
Read (10)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OP code = 28h</td>
</tr>
<tr>
<td>1</td>
<td>Reserved</td>
</tr>
<tr>
<td>2</td>
<td>(MSB)</td>
</tr>
<tr>
<td>3</td>
<td>LBA</td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>(LSB)</td>
</tr>
<tr>
<td>6</td>
<td>Reserved</td>
</tr>
<tr>
<td>7</td>
<td>(MSB)</td>
</tr>
<tr>
<td>8</td>
<td>Transfer Length</td>
</tr>
<tr>
<td>9</td>
<td>Control</td>
</tr>
</tbody>
</table>
### Power Conditions

- **0h**: No change in power conditions
- **1h**: Place device in active state
- **2h**: Place device in Idle state
- **3h**: Place device in Standby state
- **4h**: Reserved
- **5h**: Place device in steep state
- **6h**: Fh - Reserved

---

<table>
<thead>
<tr>
<th></th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>OP code = 1Bh</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1MM</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1MM</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Power Conditions</td>
<td></td>
<td></td>
<td></td>
<td>Load Eject</td>
<td>Start</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Control</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# RBC
Synchronize Cache

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>OP code = 35h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Reserved</td>
<td>WCD</td>
<td>IMM=0</td>
<td>RA=0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(MSB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>LBA=00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(LSB)</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(MSB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(LSB)</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Writes cache data to media
Applies to entire device only

WCD = Write cache disable
RBC
Test Unit Ready

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
|   | OP code = 00h
| 1 |   |   |   |   |   |   |   |
|   | Reserved
| 2 |   |   |   |   |   |   |   |
|   | Reserved
| 3 |   |   |   |   |   |   |   |
|   | Reserved
| 4 |   |   |   |   |   |   |   |
|   | Reserved
| 5 |   |   |   |   |   |   |   |
|   | Control

Status | Sense Key | ASC, ASCQ
---|------|------
00-Good | 0 | 00, 00h
02 | 05h-Illlegal Request | 25h, 00h - Logical Unit not ready
02 | 02 Not Ready | 04h, 01h - Logical Unit not ready
02 | 02 Not Ready | 04h, 01h - Logical Unit becoming ready
02 | 01 Recovered Error | 5D, xxh - SMART Threshold exceeded xxh defines which threshold
# Reduced Block Commands

**Write (10)**

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

- **OP code = 2Ah**

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Reserved**
- **DPO = 0**
- **FUA**
- **Reserved**
- **RA = 0**

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>(MSB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>(LSB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>(MSB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>(LSB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Control**
Reduced Block Command
Write & Verify (10)

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DPO =0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>LBA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(LSB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>(MSB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Transfer Length</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(LSB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td>Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

OP code = 2Eh

Appendix R: Reduced Block Command Set
# RBC Commands

## Write Buffer

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OP code = 3Bh</td>
</tr>
<tr>
<td>1</td>
<td>Reserved</td>
</tr>
<tr>
<td>2</td>
<td>Mode</td>
</tr>
<tr>
<td>3</td>
<td>Buffer ID</td>
</tr>
<tr>
<td>4</td>
<td>Buffer Offset</td>
</tr>
<tr>
<td>5</td>
<td>(LSB)</td>
</tr>
<tr>
<td>6</td>
<td>(MSB)</td>
</tr>
<tr>
<td>7</td>
<td>Parameter List Length</td>
</tr>
<tr>
<td>8</td>
<td>(LSB)</td>
</tr>
<tr>
<td>9</td>
<td>Control</td>
</tr>
</tbody>
</table>

© KnowledgeTek
rev 5.41b 15.G.31

Appendix R: Reduced Block Command Set
## Reduced Block Commands

### Write Buffer Mode

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Write combined header and data</td>
<td>Not Supported</td>
</tr>
<tr>
<td>1h</td>
<td>Vender specific</td>
<td>Vender Specific</td>
</tr>
<tr>
<td>2h</td>
<td>Write Data</td>
<td>Not Supported</td>
</tr>
<tr>
<td>3h</td>
<td>Reserved</td>
<td>Reserved</td>
</tr>
<tr>
<td>4h</td>
<td>Download Microcode</td>
<td>Not Supported</td>
</tr>
<tr>
<td>5h</td>
<td>Download Microcode and Serve</td>
<td>Mandatory</td>
</tr>
<tr>
<td>6h</td>
<td>Download Microcode with Offset</td>
<td>Not Supported</td>
</tr>
<tr>
<td>7h</td>
<td>Download Microcode with Offset and Serve</td>
<td>Not Supported</td>
</tr>
</tbody>
</table>
Appendix Z:
Answers
Answer To Chapter 5's Problem
Tell Us What's Happening - Trace Format

Destination Length

Transaction

Source

Address

3 5 RReq 14

3 5 RRsp

04 00

31 33 39 34

F6 01 00 08

12 34 56 78

9A BC DE F0

All Numbers In Hex
Trace doesn't show Ack Packets

Reason For Request

Key Info Returned

1394

Appendix Z: Answers
Here's What's Happening - Part 1

3 5 RReq 14 FFFF FFFF 0400
5 3 RRsp 04 XX XX XX
31 33 39 34
F8 01 00 08
12 34 56 78
9A BC DE F0

Get Bus Info Block

Got Length - Skip Around

Get ROM Root Length

ROM Root Length = 4 Quads

Read ROM Root

Offset to Unit Dir = 4 Quads

Get Unit Dir Length

Unit Dir Length = 7 Quads
Tell Us What’s Happening - Part 2

3 5 RReq 1C  FFFF  FFFF  0438
5 3 RRsp
12 00 60 9E
13 01 04 83
38 00 60 9E
39 01 04 D8
14 00 0E 00
3A 01 00 08
54 00 40 00

Get Unit Dir
Management Agent at 4000 Quads
Write Management Agent
ORB at Address 0000 1000 0000

3 5 WReq 08  FFFF  F010  0000
FF C3 00 00
10 00 00 00

Read Login Request

5 3 RReq 20  0000  1000  0000
3 5 RRsp
00 00 00 00
00 00 00 00
FF C3 00 00
10 10 00 00
80 00 00 00
00 00 00 00
FF C3 00 00
10 20 00 00

Login Response at 0000 1010 0000
Status at 0000 1020 0000
Tell Us What's Happening - Part 3

5 3 WReq 0C 0000 1010 0000
00 0C 12 34
FF C5 FF FF
F0 10 01 00

Login Response
Login ID = 1234
Command Agent = FFFF F010 0100

5 3 WReq 08 0000 1020 0000
42 00 00 00
10 10 00 00

Posting Status
For ORB from 0000 1000 0000
Resp = Completed

3 5 WReq 08 FFFF F010 0108
FF C3 00 00
10 00 00 00

Write Command Agent
ORB at Address 0000 1000 0000

5 3 RReq 20 0000 1000 0000

Read Command ORB

3 5 RRsp
FF C3 00 00
10 00 00 20
FF C3 00 00
20 00 00 00
82 D0 00 20
12 00 00 00
08 00 00 00
00 00 00 00

Command ORB - Inquiry Command
Next ORB = 0000 1000 0020
Data goes at 0000 2000 0000
Tell Us What’s Happening - Part 4

5 3 RReq 20 0000 1000 0020
3 5 RRsp 80 00 00 00
00 00 00 00
FF C3 00 00
20 00 00 20
82 90 00 20
00 00 00 00
00 00 00 00
00 00 00 00

5 3 WReq 08 0000 2000 0000
0E 00 03 03
00 00 00 00

5 3 WReq 08 0000 1020 0000
01 00 00 00
10 00 00 00

What Condition Is The Target In ?

Read Command ORB

Command ORB - Test Unit Ready
No Next ORB
Data goes at 0000 2000 0020

Inquiry Data Returned

Post Status
For ORB from 0000 1000 0000
Resp = Completed