1394

KnowledgeTek

Volume 2

rev 5.41E 15.H.31

Copyright

Table of Contents

Table of Contents-Appendices

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

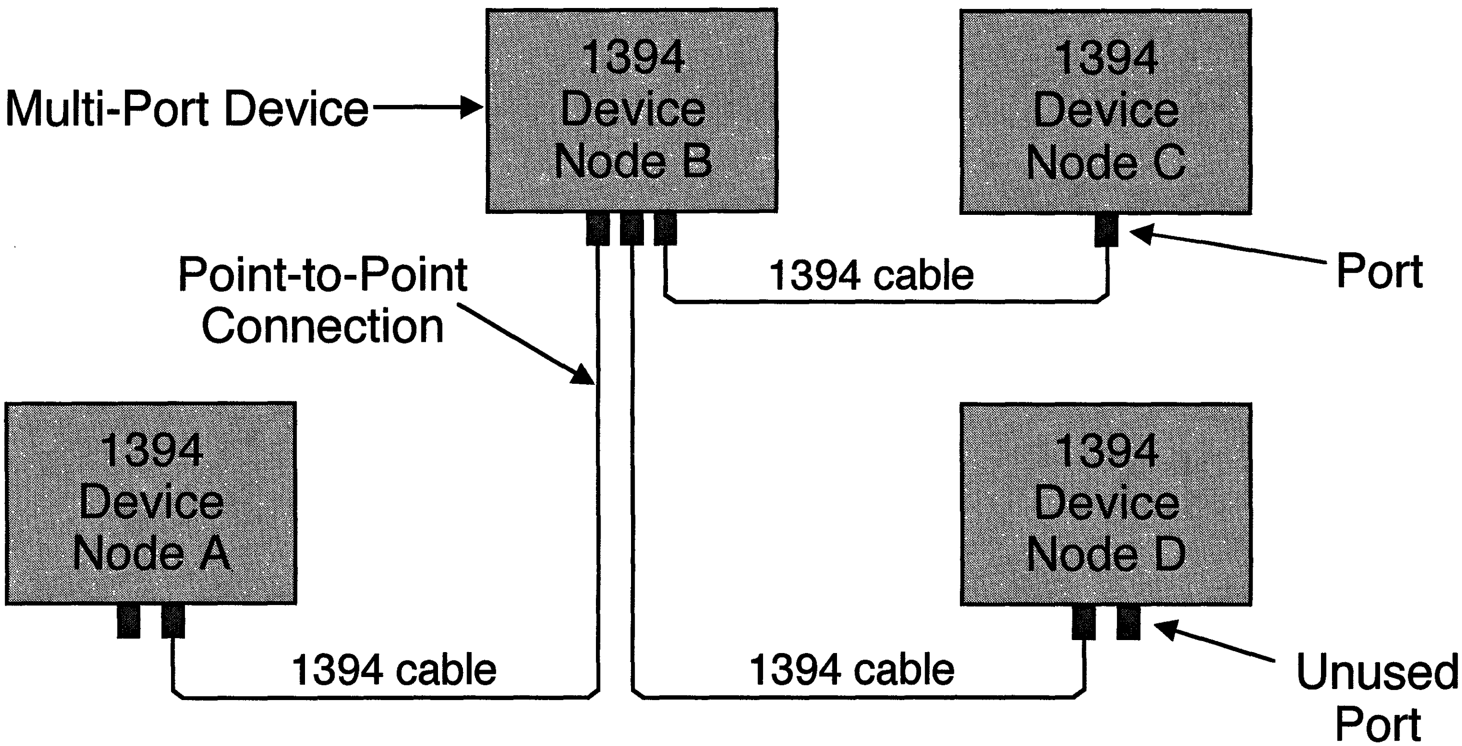
- Two twisted Pairs for data
- No Power

6 Wire Cable

- Two twisted pairs for data transmission
- 2 wires carry power

4.5 meters per hop limit

1394 Physical Topology

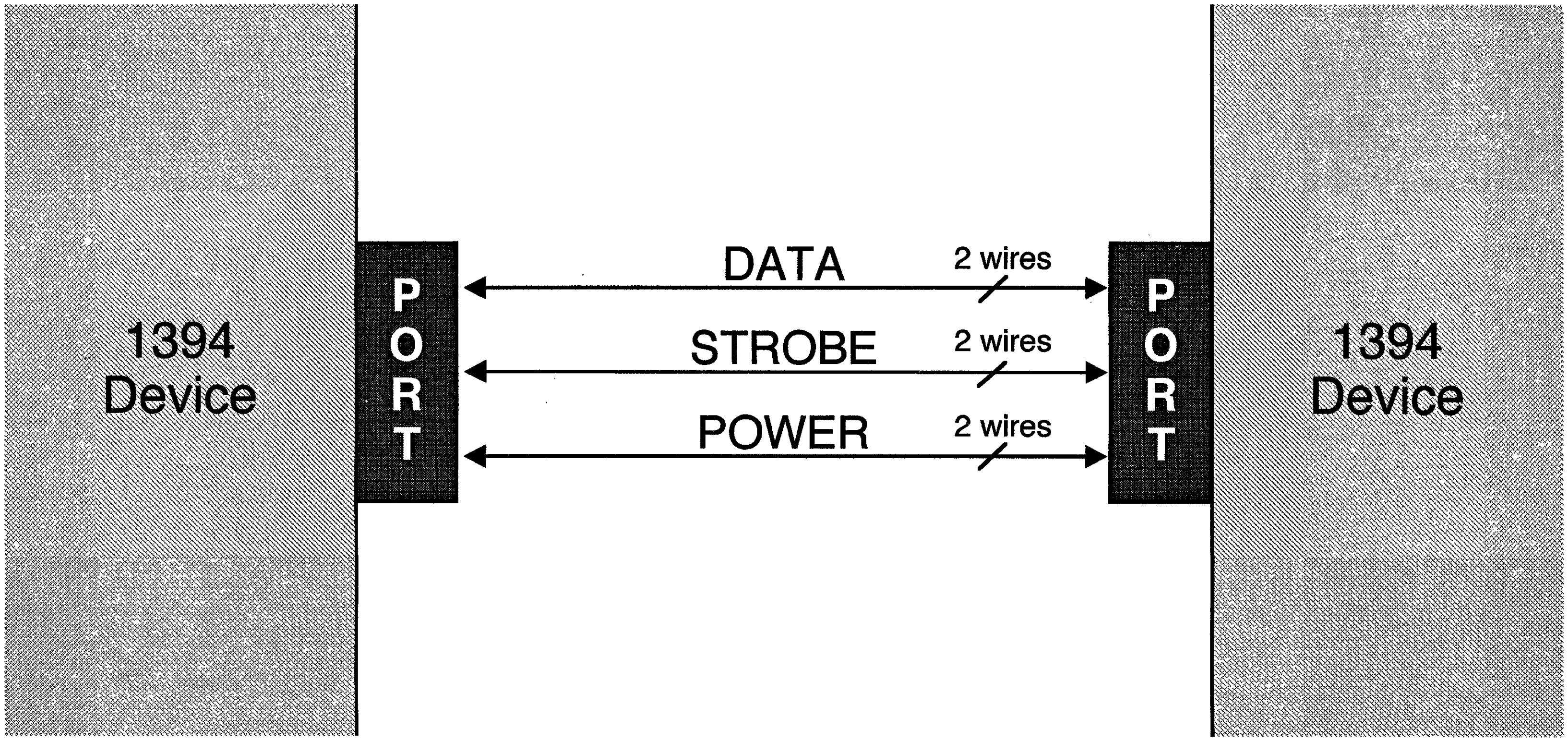


Multi-Port Devices repeat all bus traffic on other ports*

All Devices see all packets

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

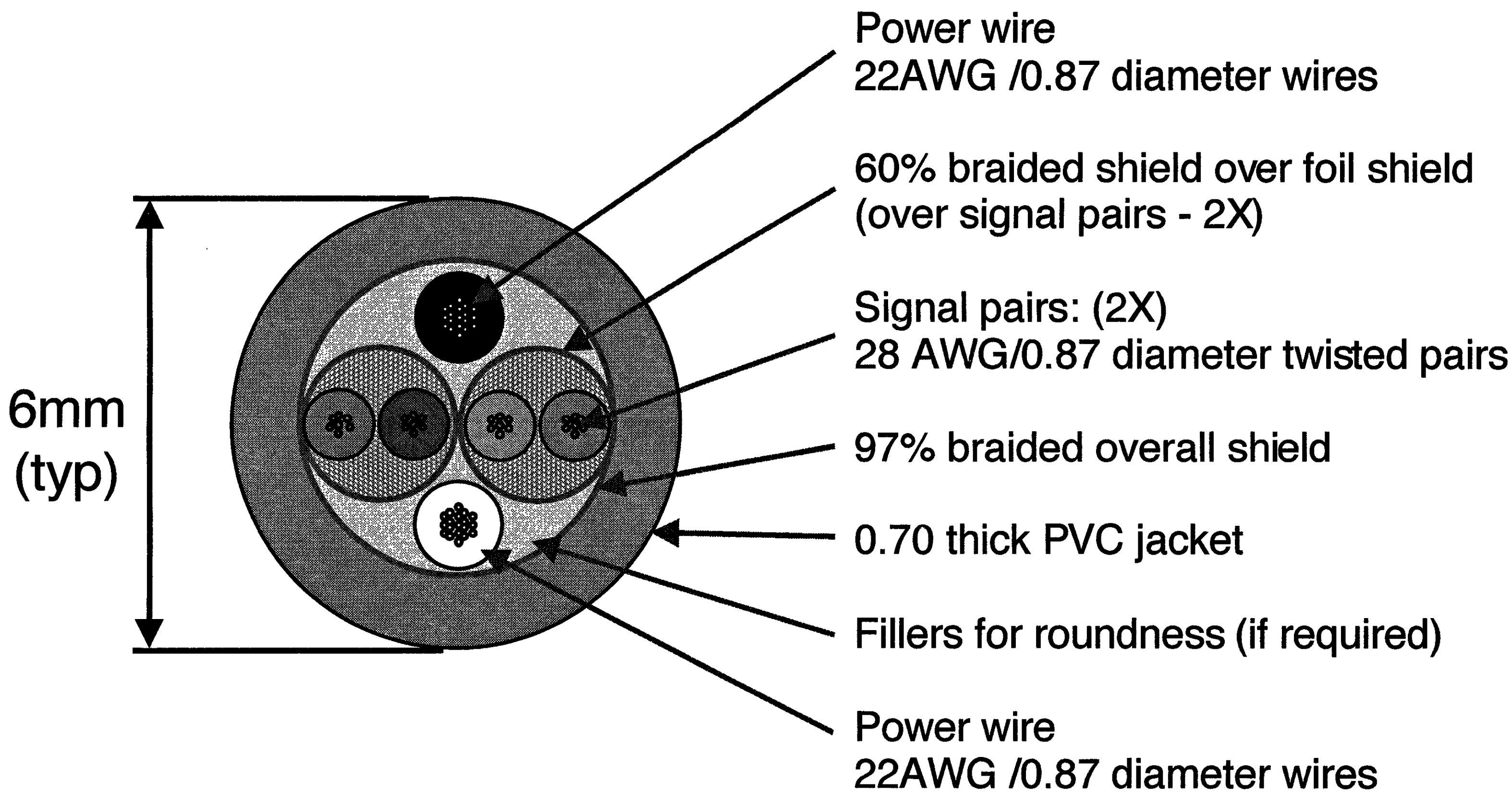
Point-To-Point Connections



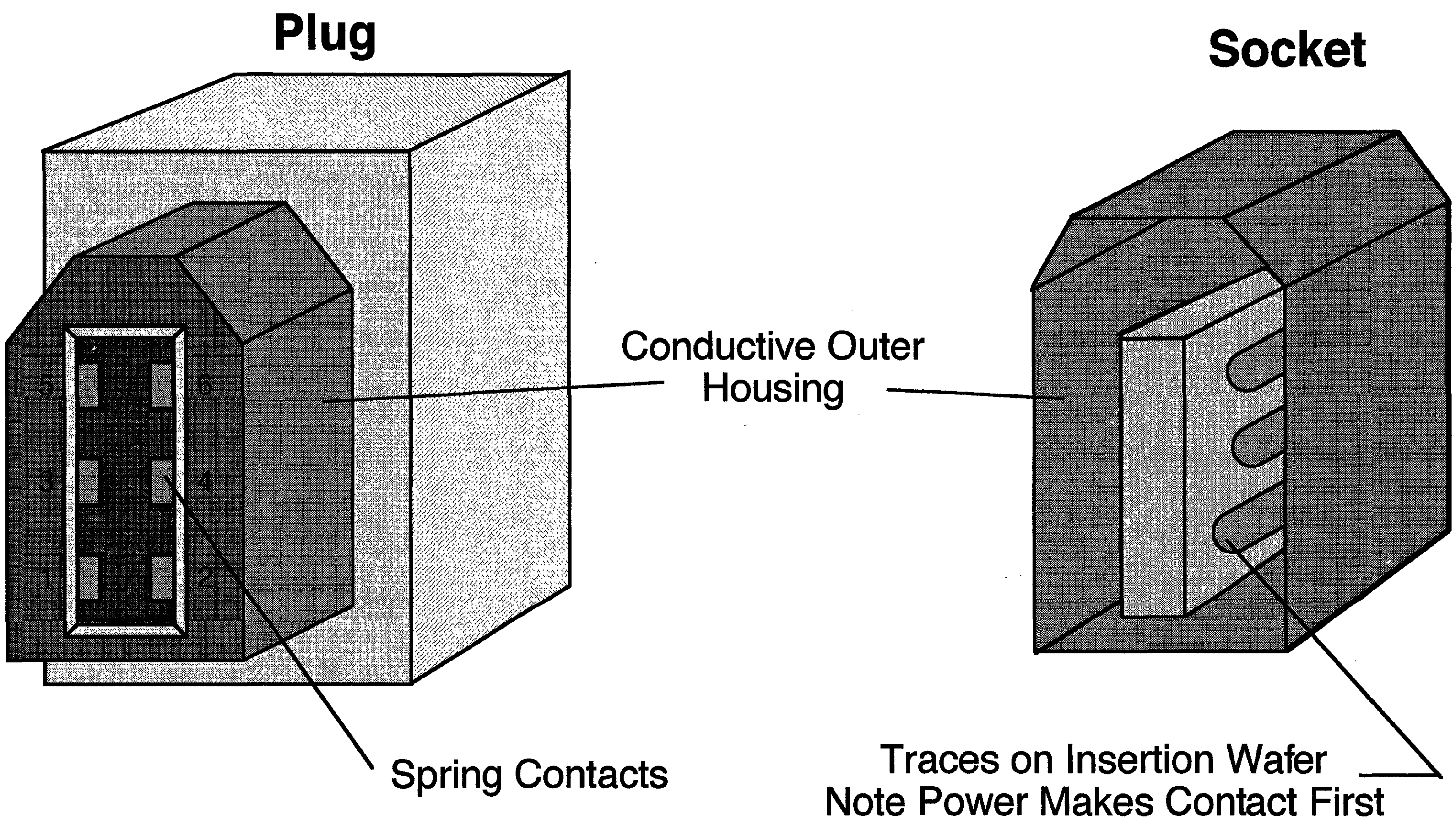
Data fed serially using DATA and STROBE signals

POWER supplies 8-30 Volts (@1.5 Amp)

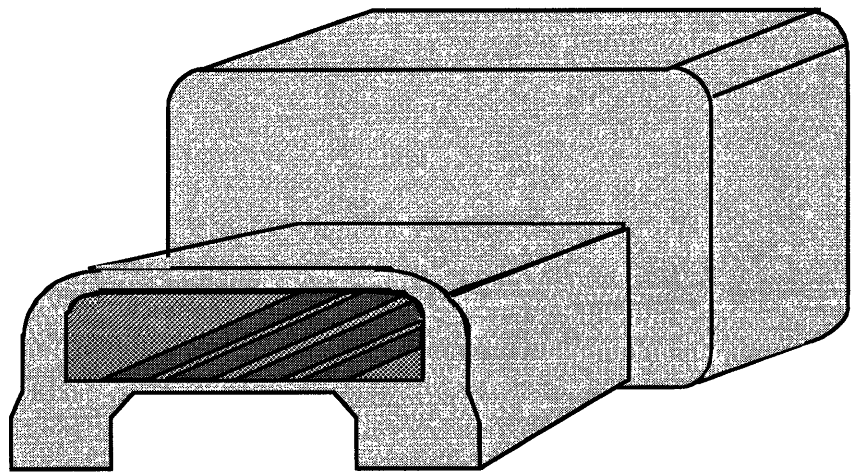
The Cable


6mm
(typ)
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



4 Pin Connector

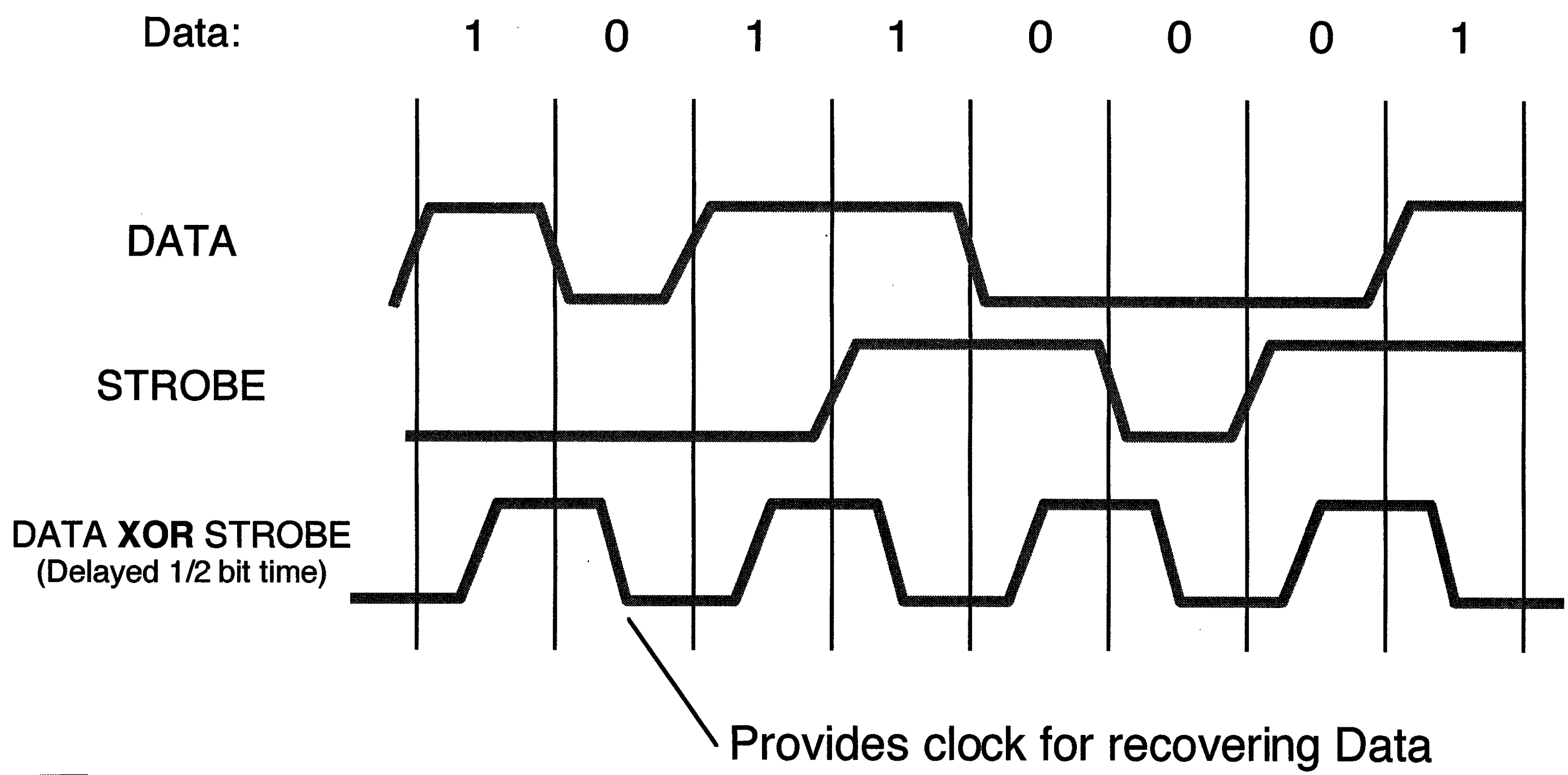


No Power Signals

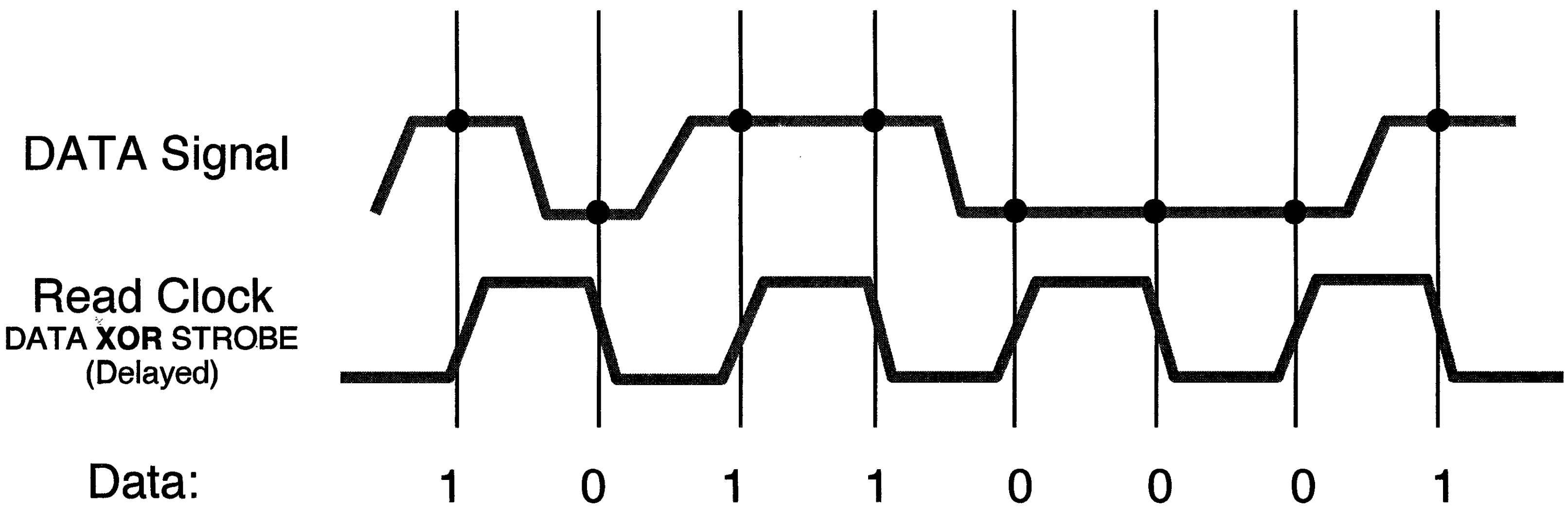
Data/Strobe Encoding

Data is sent via the DATA Signal encoded NRZ (Non-Return to Zero)

STROBE toggles whenever DATA doesn't



Data Recovery

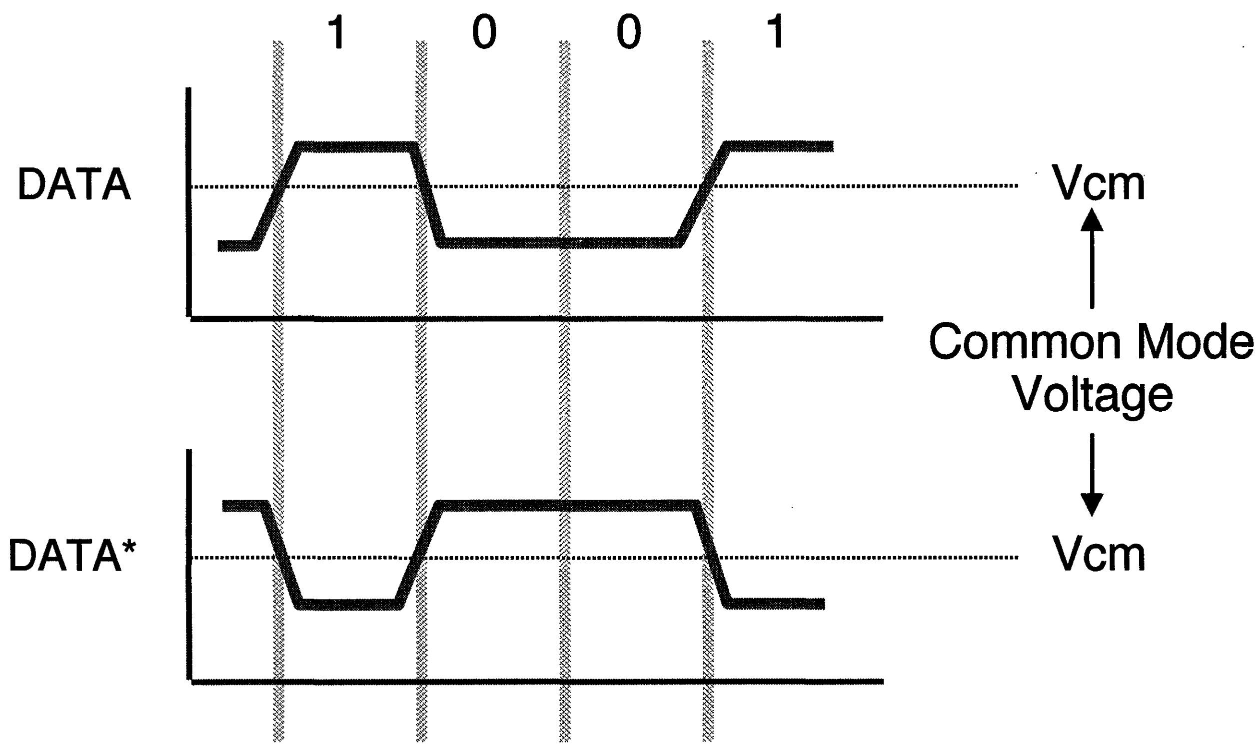


Clock Rates

S100	98.304 Mbps ± 100ppm 10.17 nSec bit time
S200	196.608 Mbps ± 100ppm 5.09 nSec bit time
S400	393.216 Mbps ± 100ppm 2.54 nSec bit time

Differential Signaling

DATA and STROBE signals transmitted differentially

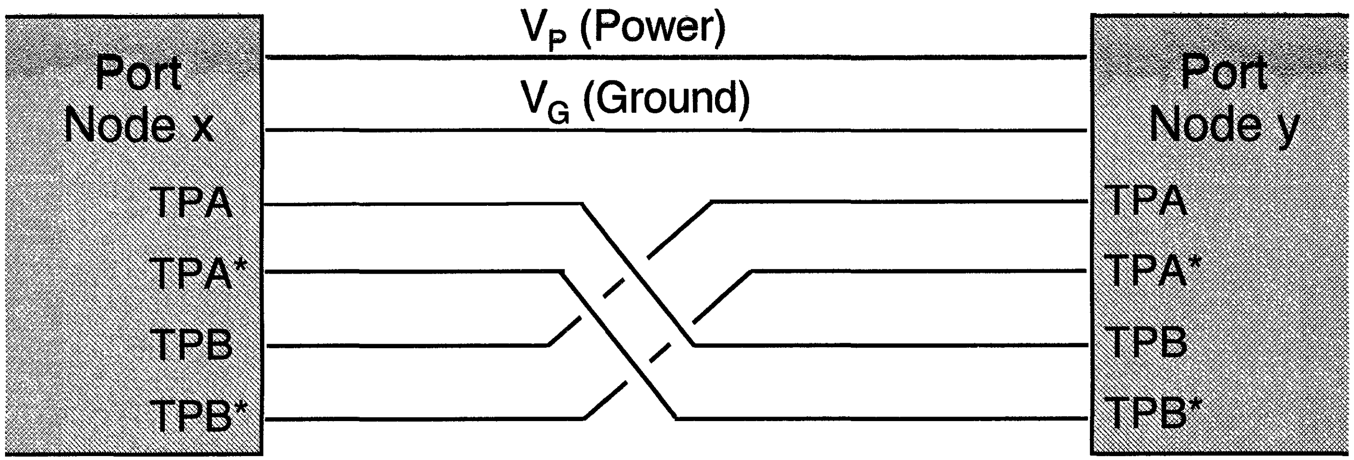


Differential Signal Levels

	S100		S200		S400	
	Max	Min	Max	Min	Max	Min
Transmitting End	265mV	172mV	265mV	172mV	265mV	172mV
Receiving End (Data Transmission)	260mV	142mV	260mV	132mV	260mV	118mV
Receiving End (Arbitration)	260mV	173mV	262mV	171mV	265mV	168mV

Crossover Cable

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

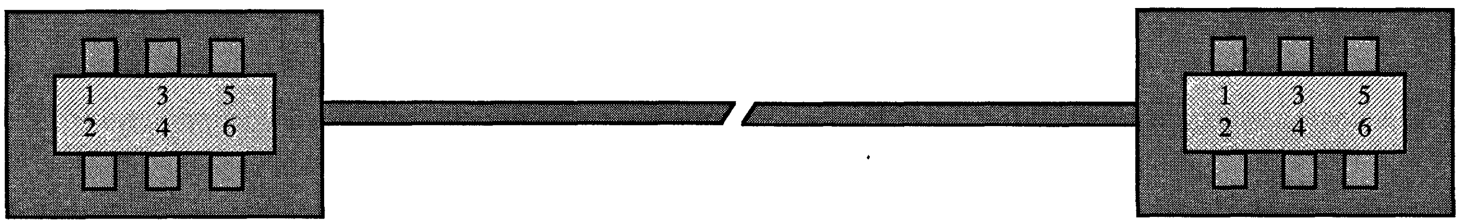


TPA	Transmits STROBE Receives DATA
TPB	Transmits DATA Receives STROBE

Crossover eliminates need for separate upstream/downstream connectors

Similar to Null-Modem cables

Cable



CROSS OVER CABLE IN CONNECTOR OF CABLE

Pin			Pin
1	Power	Power	1
2	Ground	Ground	2
3	TPB*	TPB*	3
4	TPB	TPB	4
5	TPA*	TPA*	5
6	TPA	TPA	6

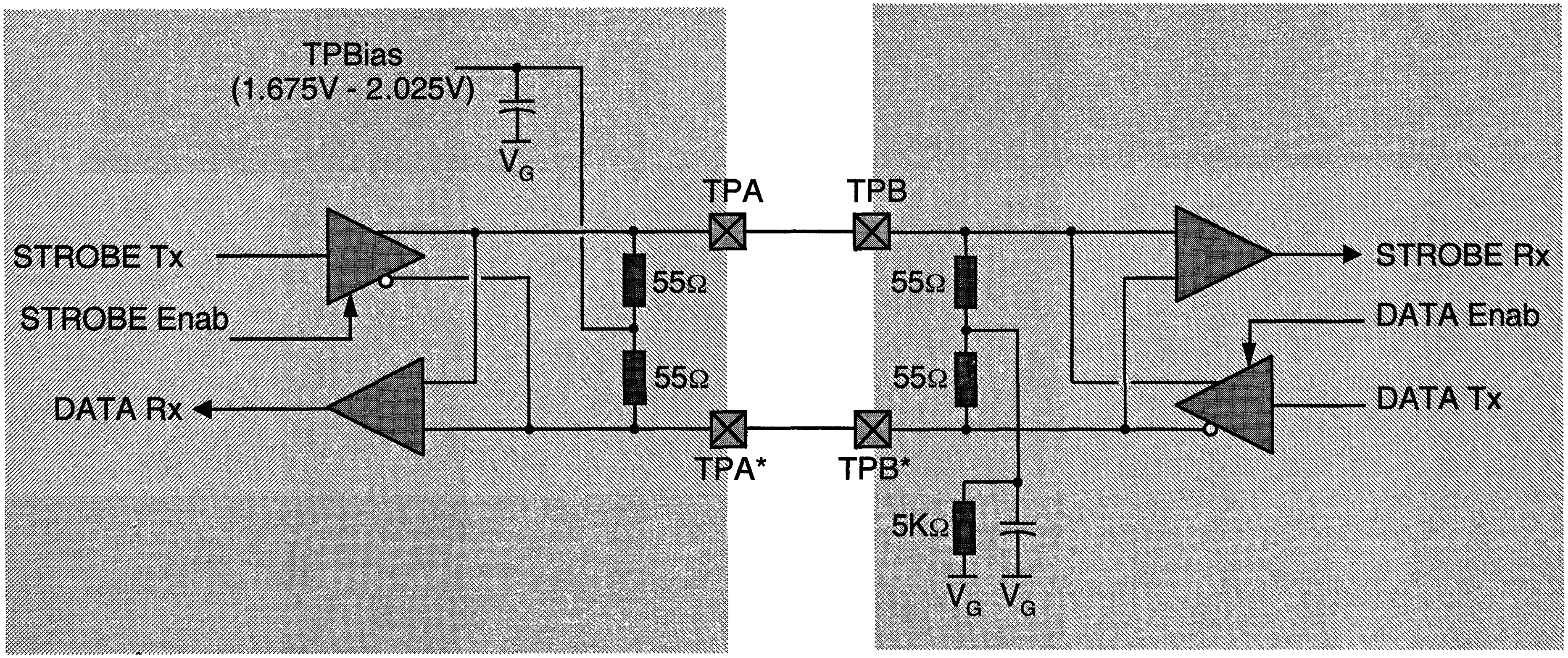
Maximum cable length - 4.5 meters

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

TPA & TPB	Differential Impedance	110Ω ± 6Ω
TPA & TPB	Common Mode Impedance (to VG)	33Ω ± 6Ω
TPA & TPB	Cable Attenuation	2.3dB max @ 100MHz 3.2dB max @ 200MHz 5.8dB max @ 400MHz
VP & VG	DC Resistance	0.333Ω max

Port Basic Interface

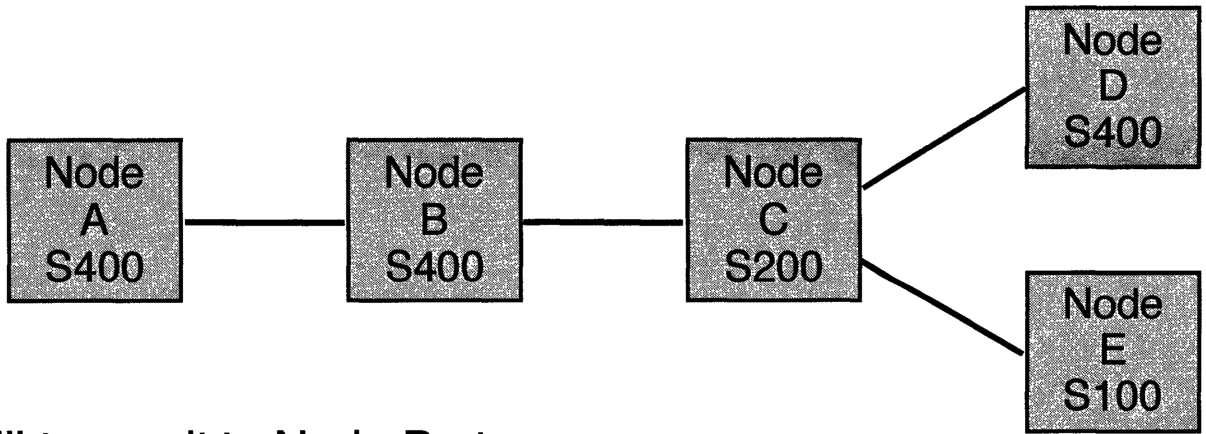


TPBias sets Common Mode Voltage

Receiving End Terminated (Either Direction)

CMOS Driver - 4mAmp

Speed Sensing



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

- S100 0mA
- S200 3.5mA
- S400 9mA

These values approximate

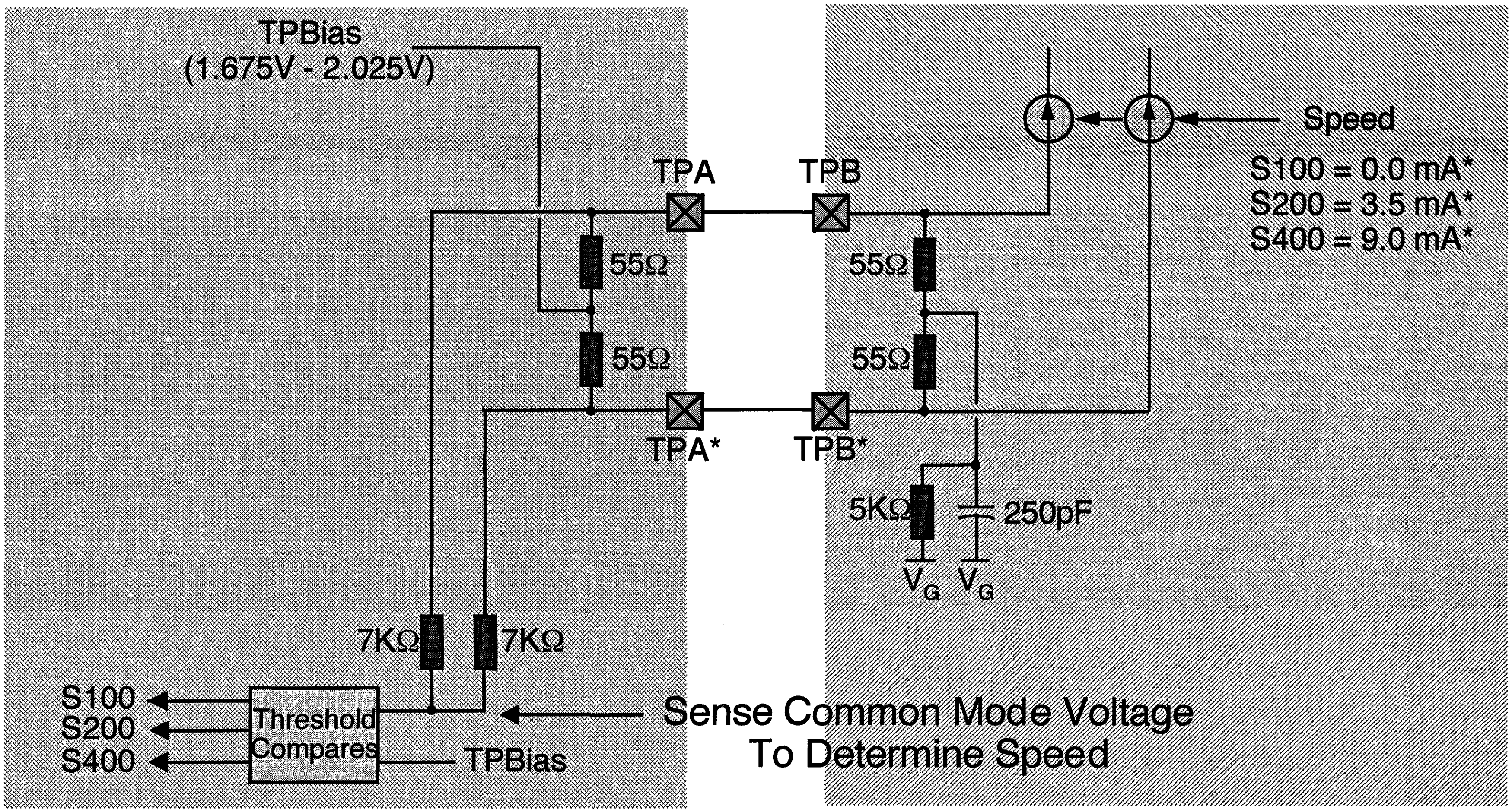
TPA Port senses reduced Common Mode Voltage

- S100 1.665V - 2.015V
- S200 1.438V - 1.665V
- S400 1.092V - 1.438V

Bi-Directional - Each Port knows neighbors speed

- Don't need to sense for faster than your own speed

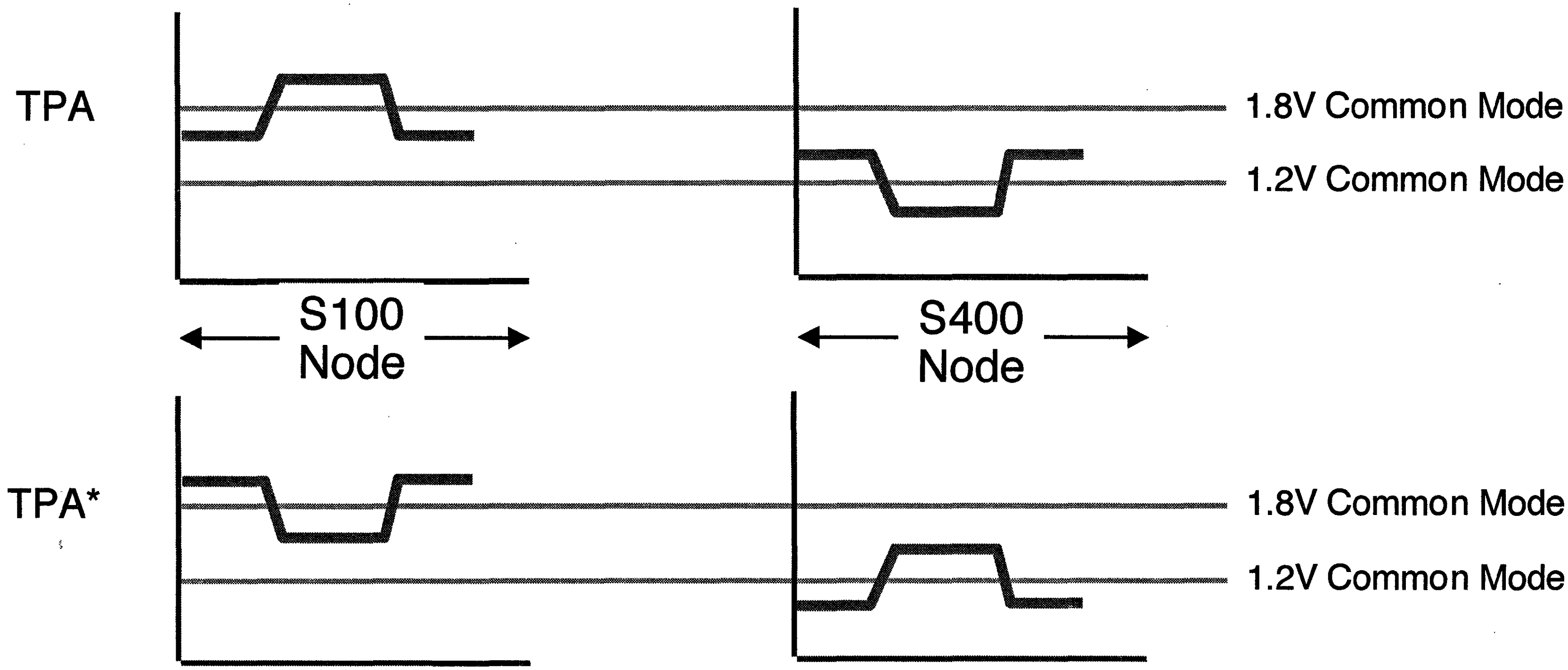
Speed Sensing



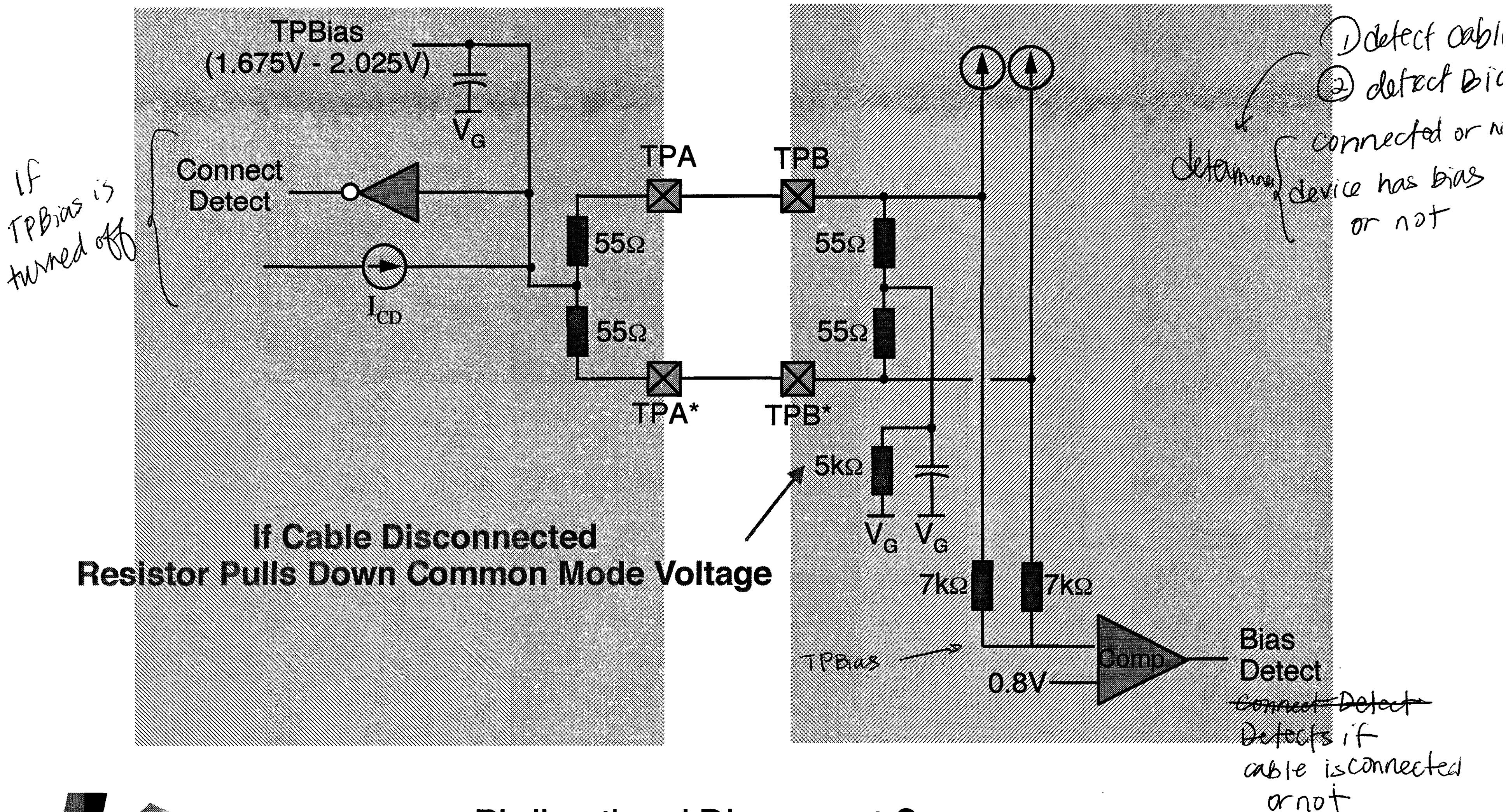
* Spec: S100 = +0.44mA to -0.81mA
S200 = -2.53mA to -4.84mA
S400 = -8.10mA to -12.40mA

TX

Speed Sensing Example

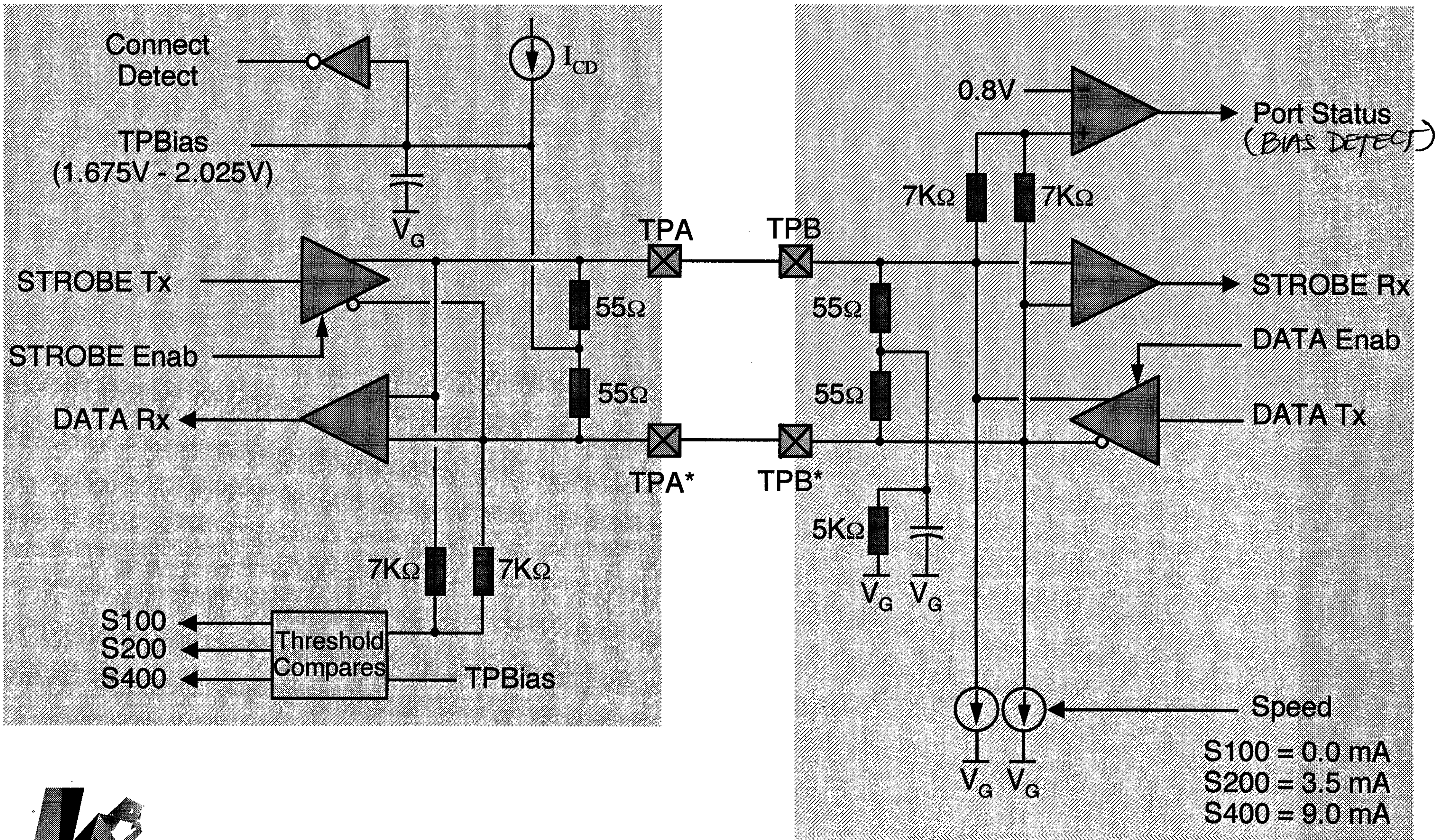


TP Bias and Disconnect Sensing



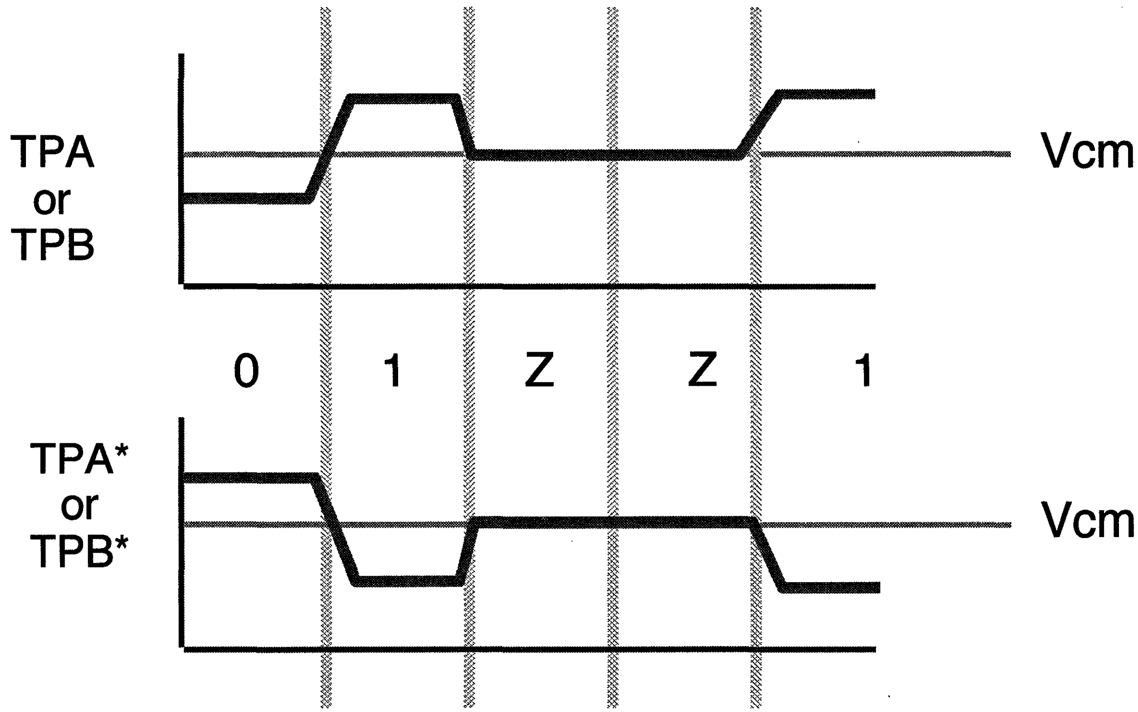
Bi-directional Disconnect Sense

Port Interface with Speed & Disconnect Sensing



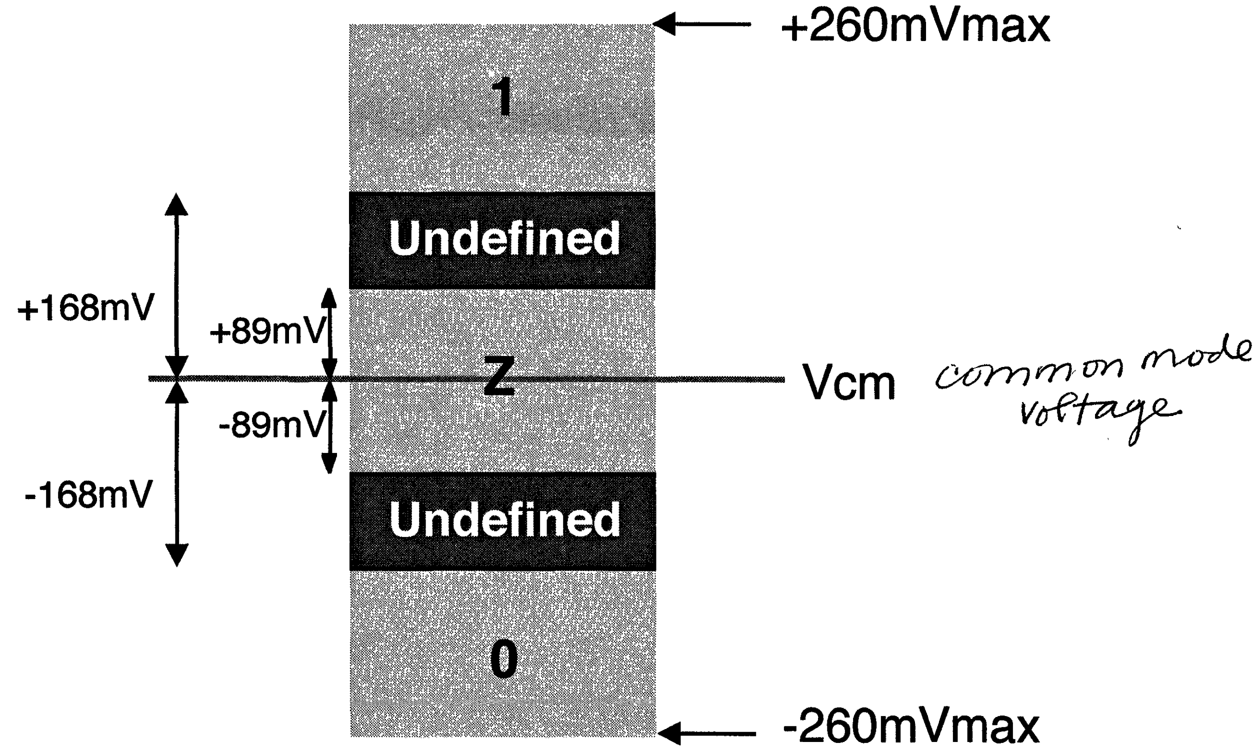
Three Level Signaling

During Arbitration, a third signal level is used:



Three Level Signals

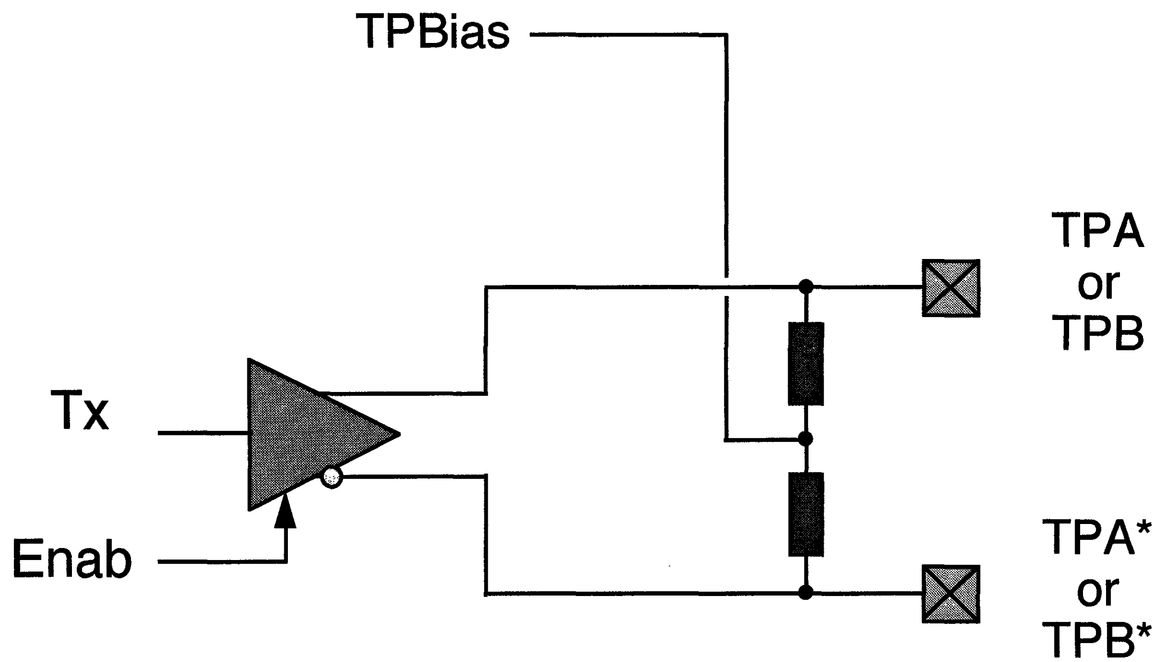
TPA - TPA*
or
TPB - TPB*
At Receiver



Drive Z signal by disabling transmitter driver

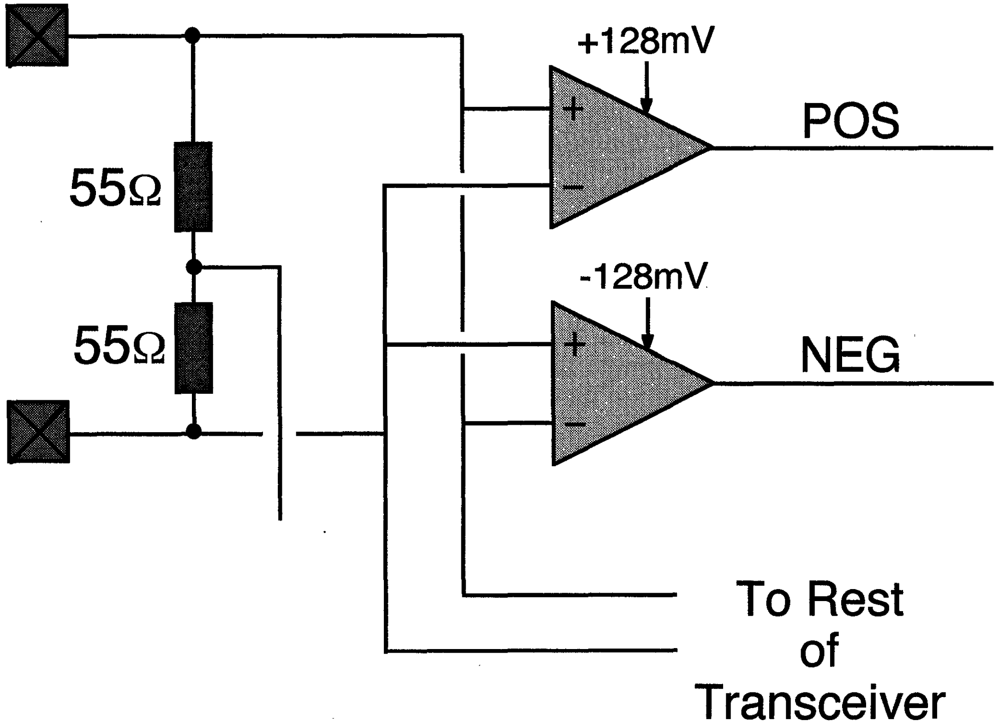
Detect Z by window comparison

Driving Three Level Signals



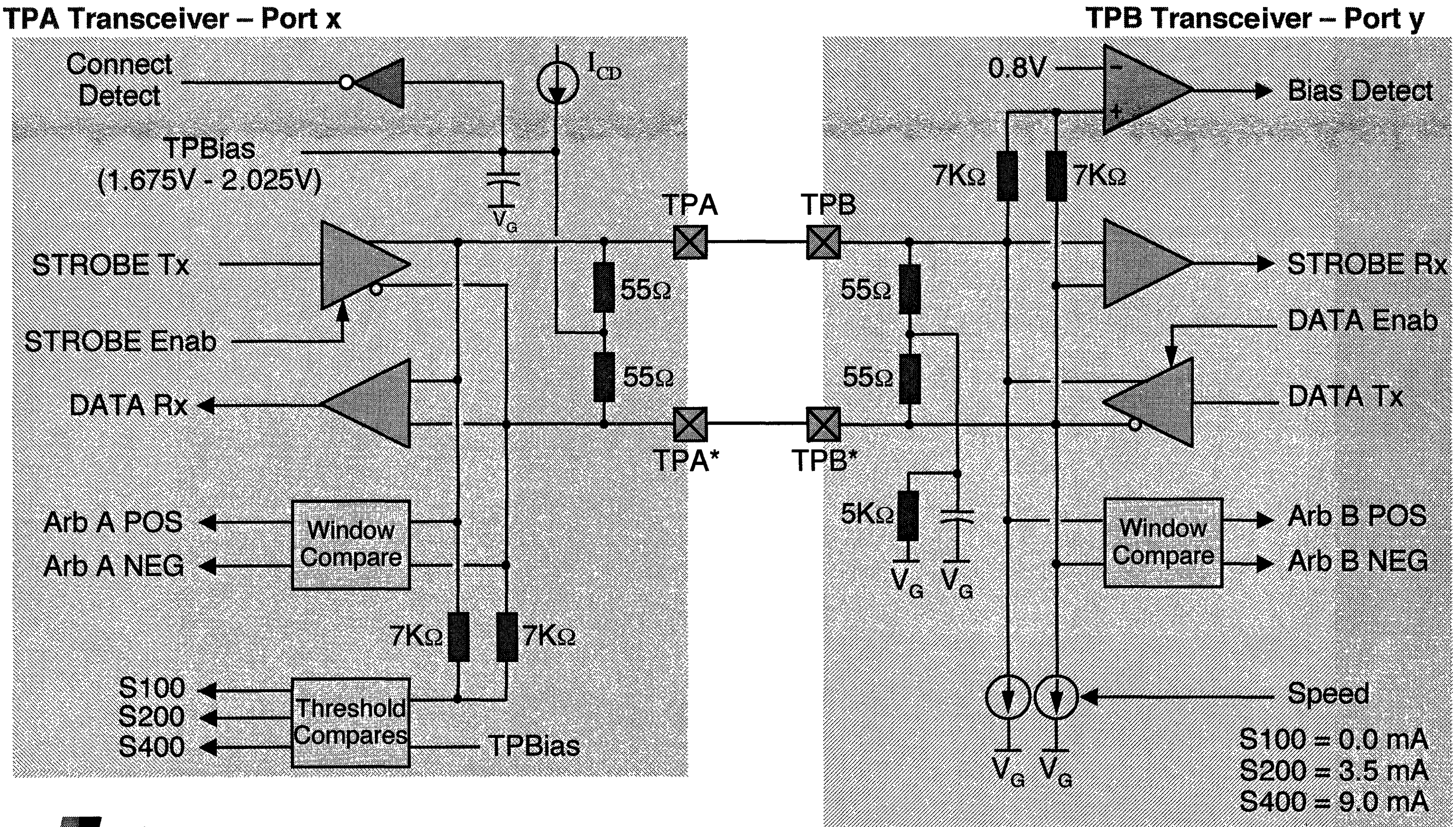
Signal	Tx	Enab
0	0	1
1	1	1
Z	X	0

Receiving Three Level Signals

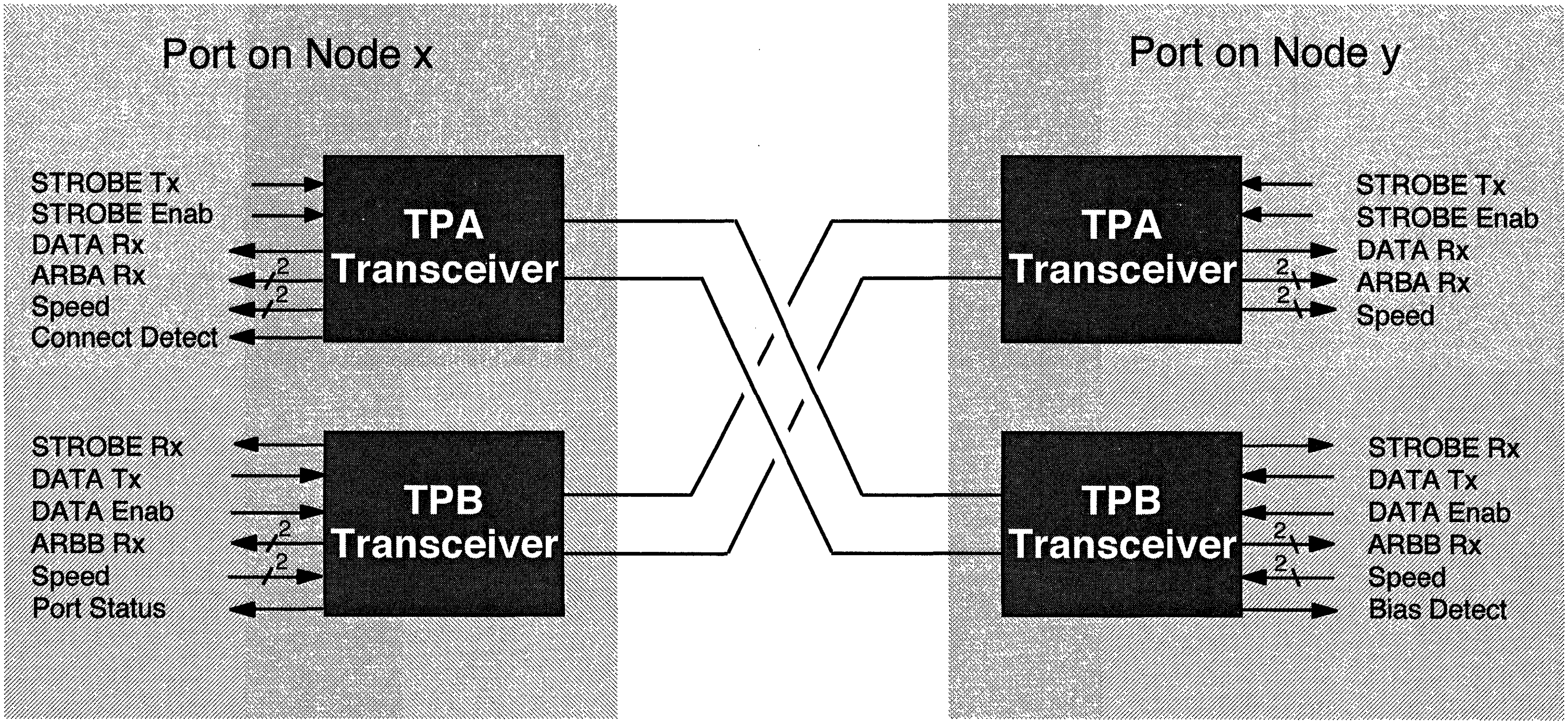


POS	NEG	Meaning
False	False	Z
False	True	0
True	False	1
True	True	Bad

1394 Port Transceiver



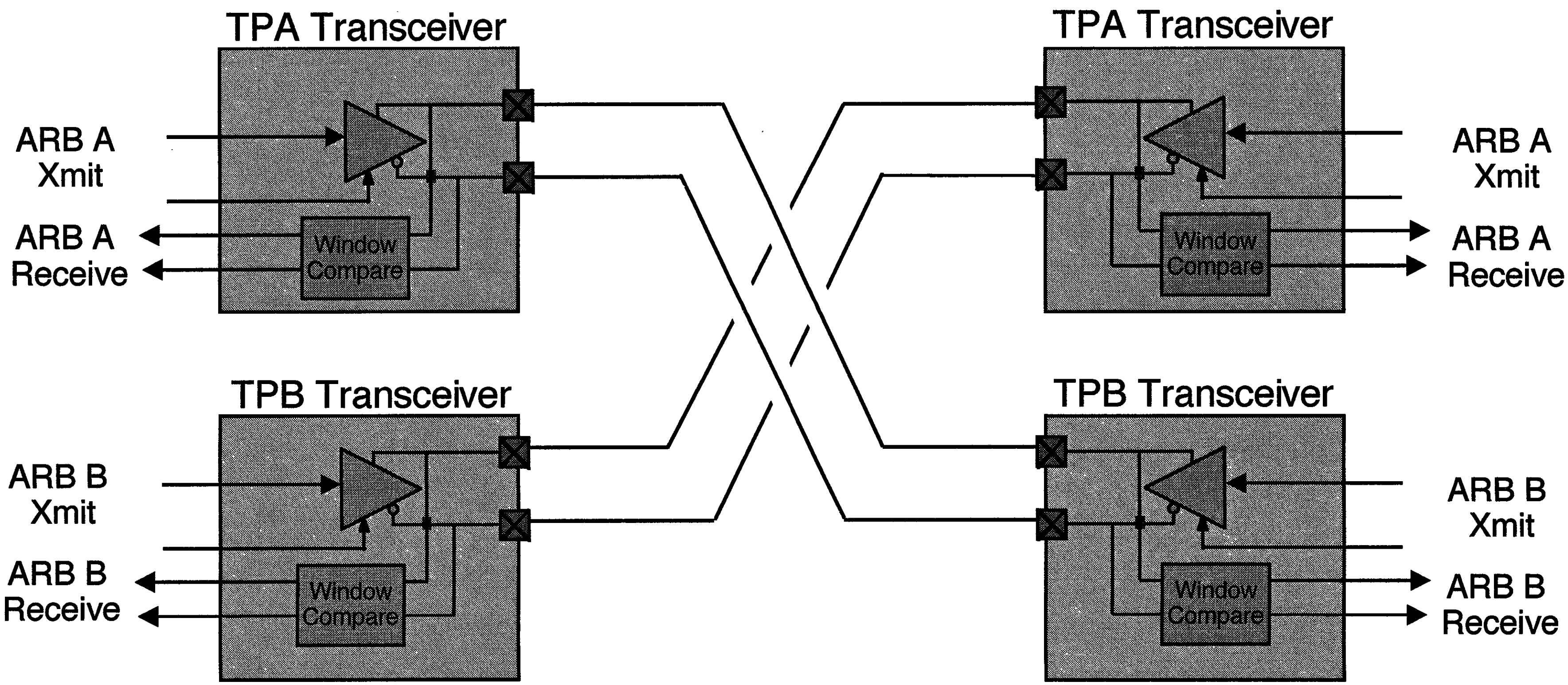
Transceiver Big Picture



Arbitration Signaling

Signal contained on both TPA & TPB

- Both nodes drive both!
- Both nodes receive both!



Effect of Two Drivers

Arb A Transmits	Arb B Transmits	Cable
Z	Z	Z
Z	0	0
Z	1	1
0	Z	0
0	0	0
0	1	Z
1	Z	1
1	0	Z
1	1	1

Z = Not driven

Arbitration Signaling Deduction

If I am sending	And I see on cable	He must have been sending
Z	Z	Z
Z	0	0
Z	1	1
0	Z	1
0	0	0 or Z
0	1	broke
1	Z	0
1	0	broke
1	1	1 or Z

this page is intentionally blank

Arbitration Signal Encoding - Transmit

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

^

Arbitration Signal Encoding - Receive

Arb A Rx	Arb B Rx	Signal Received
Z	Z	Idle
Z	0	Parent Notify or Request Cancel
Z	1	Ident Done
0	Z	Self ID Grant or Request
0	0	Root Contention or Grant or Rx suspend
0	1	Parent Handshake or Data End
1	Z	Child Handshake or Rx disable notify
1	0	Data Prefix
1	1	Bus Reset

V

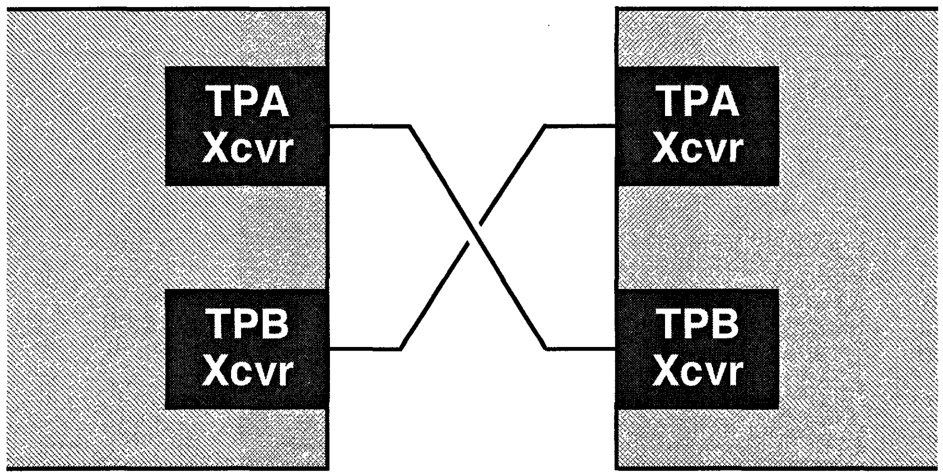
Notice Cable Twist
Data Prefix Transmit: A = 0 B = 1
Data Prefix Received: A = 1 B = 0

Bi-directional Signaling

Arbitration Signal Encoder chosen carefully

When both sides transmit - cable still has correct signal

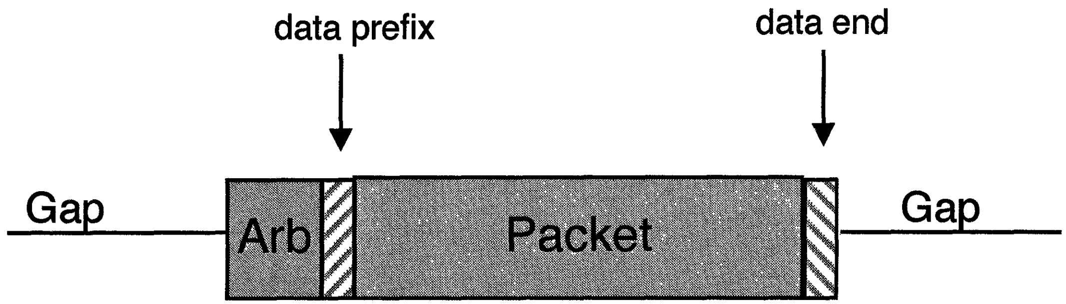
Example



Transmit	0	→ ←	Z	Transmit
Data Prefix	1		0	Request
Receive	0	← →	1	Receive
Data End	1		0	Data Prefix

This works. Data Prefix should nullify Request

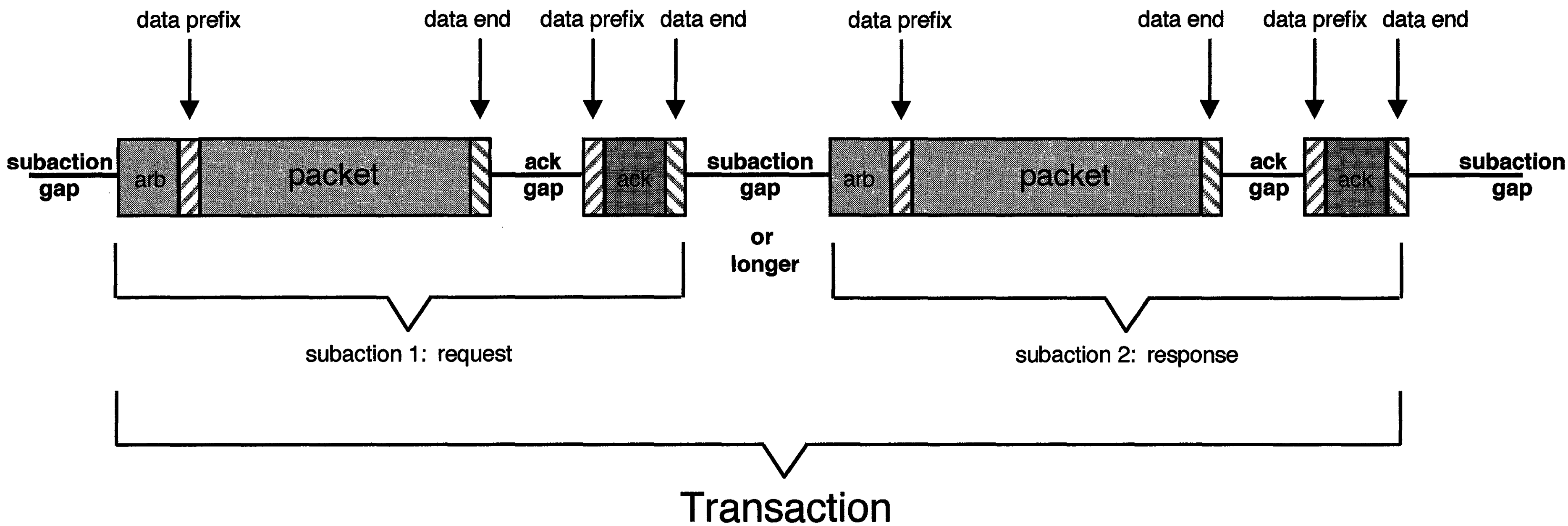
Sending Packets



Gap	Bus is in Idle state (Send A=Z B=Z) 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 - 1.63μSec)
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} = Base Rate Bit Time $\approx$ 10 nSec

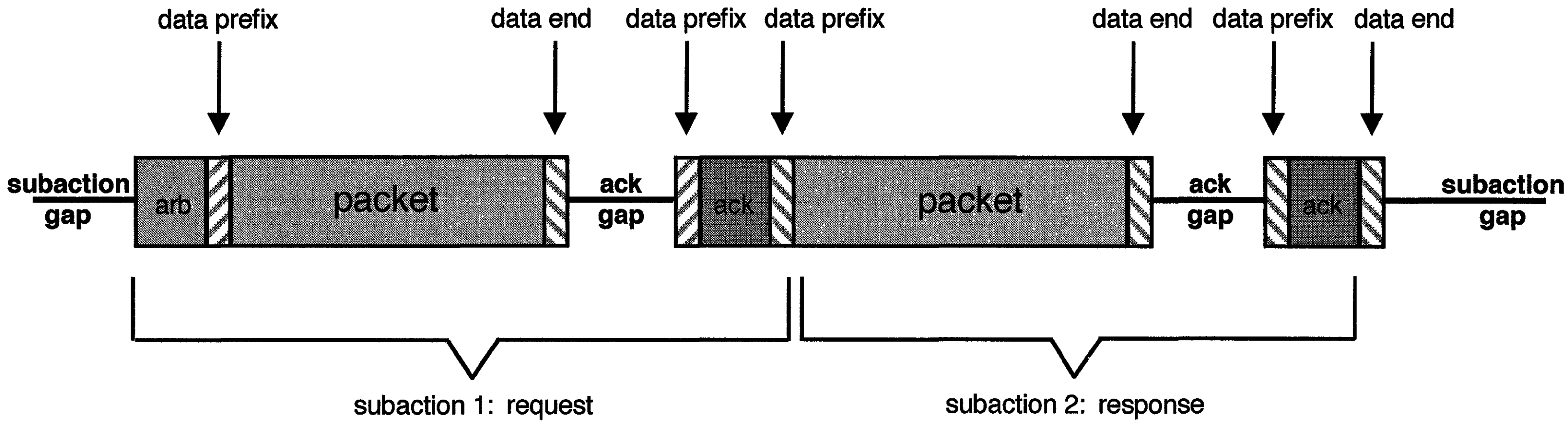
Asynchronous Subactions



Ack Gap — 4 T_{BR} (40 - 50 nSec)

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

Concatenated Asynchronous Subactions



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

Physical Notes

Section 9

Arbitration

Subjects Covered

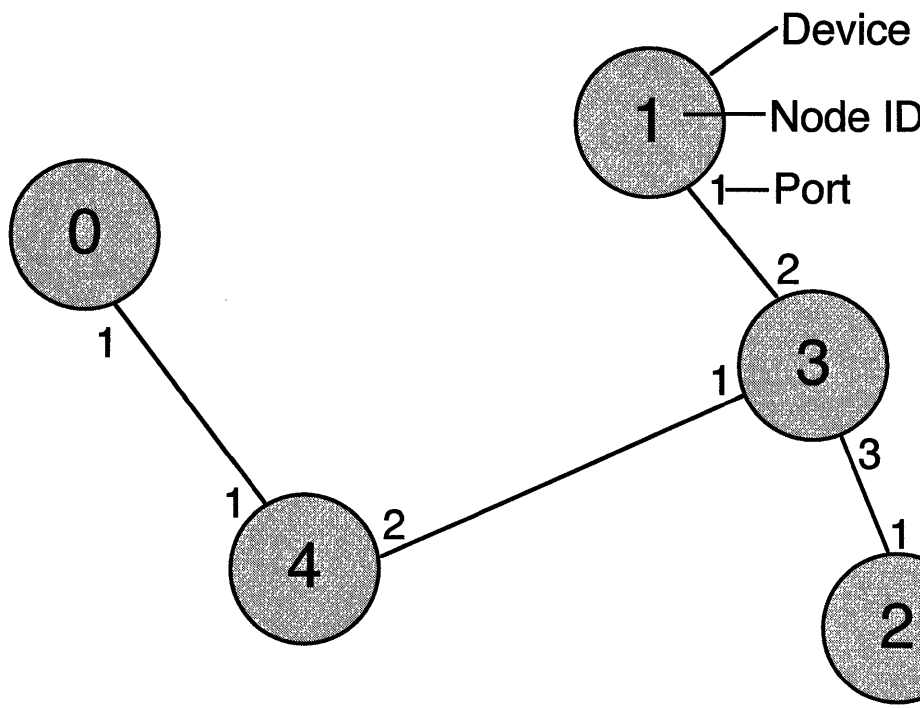
Normal arbitration

Arbitration enhancements

Fairness

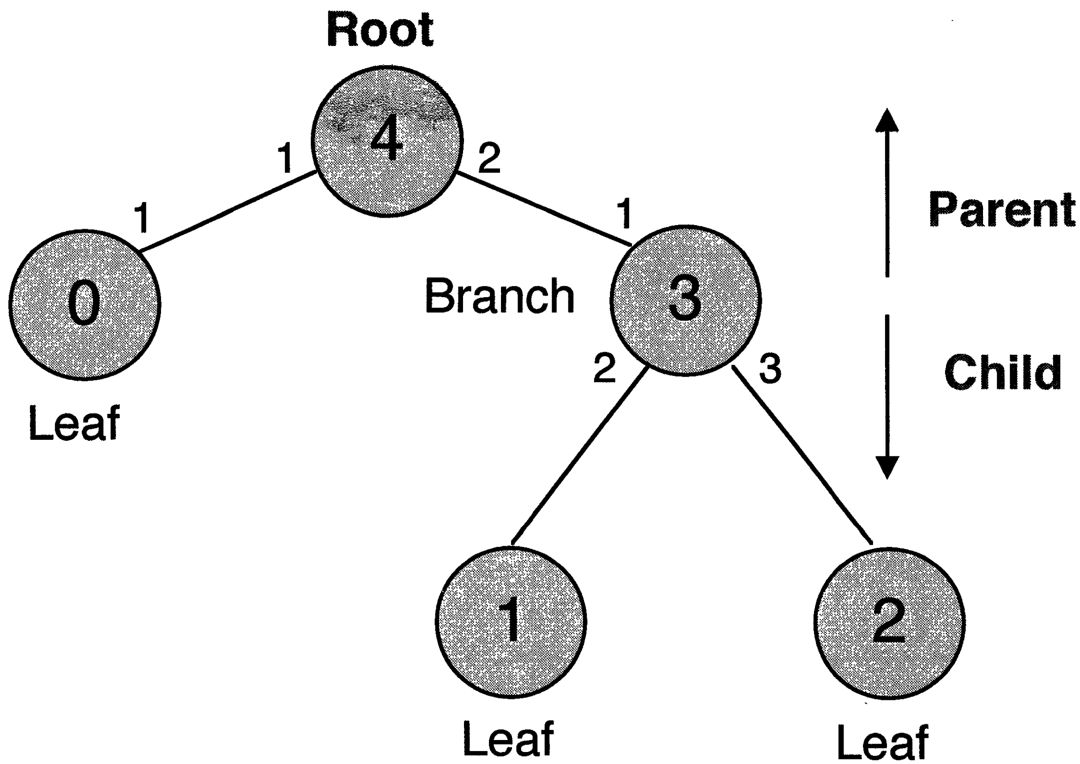
Priority register

Physical Topology



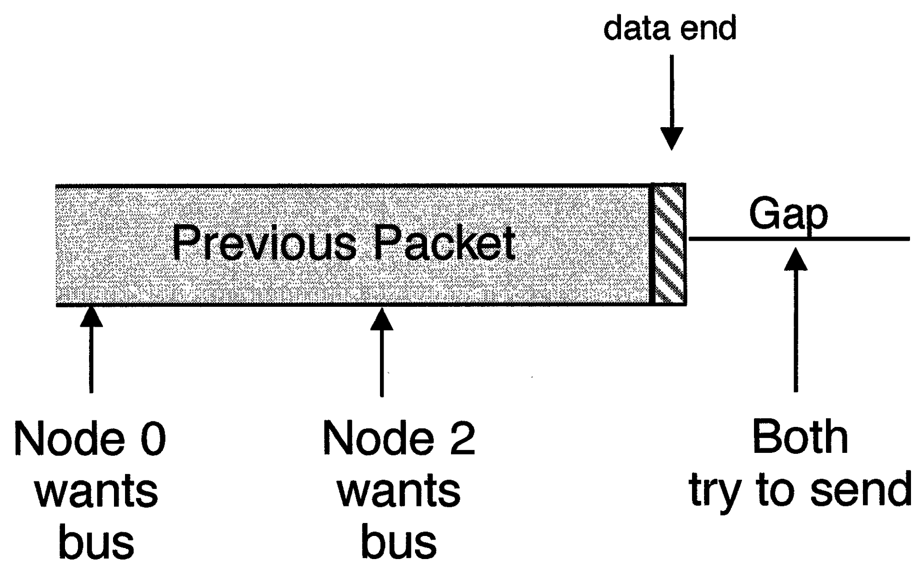
Nodes are numbered automatically during Configuration
Highest numbered node is the Root

Logical Topology



Leaf - Only one connection
Branch - Two or more connected ports
Root - Leaf or branch with no parent

The Need For Arbitration



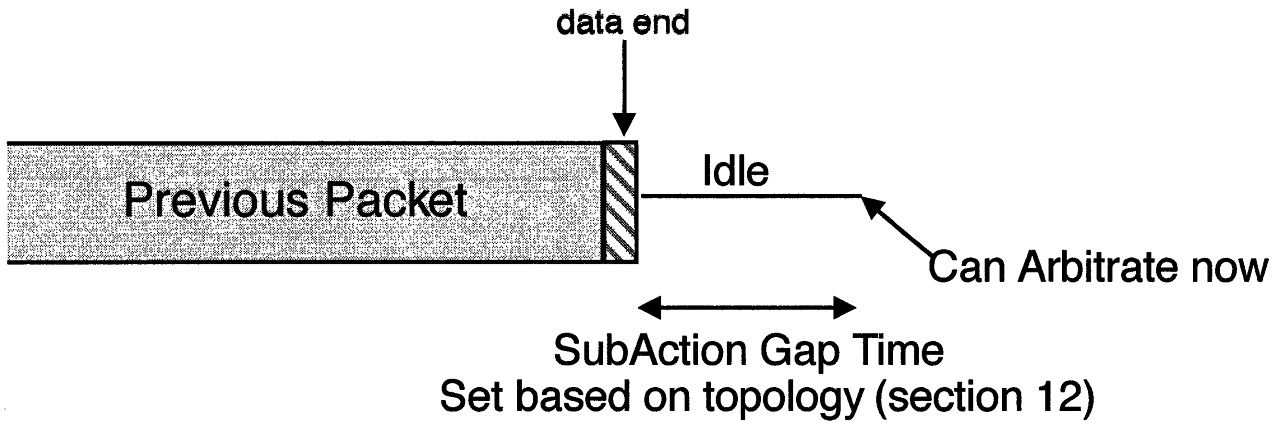
Instead - Nodes go through Arbitration before transmitting

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

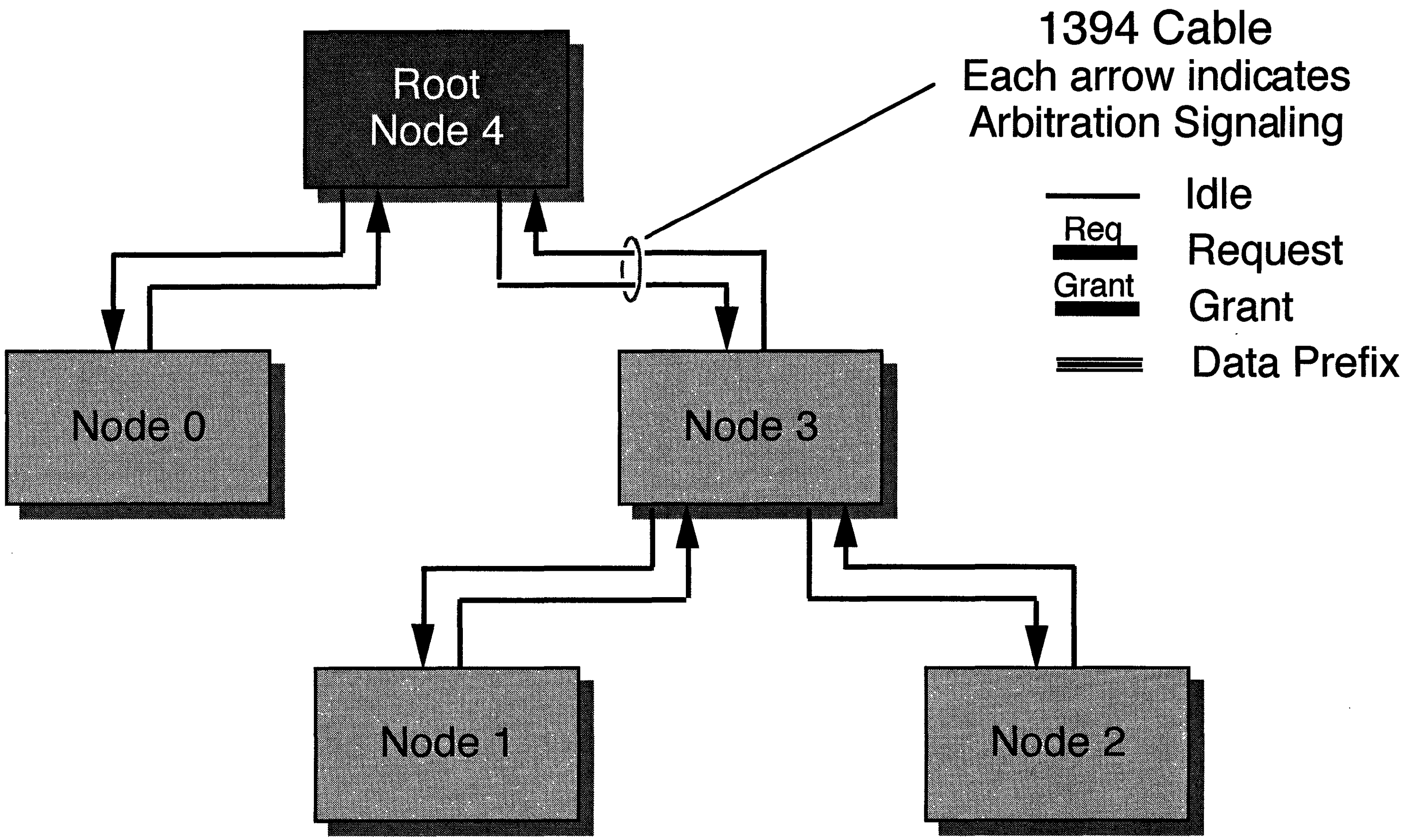
Multiport Nodes

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

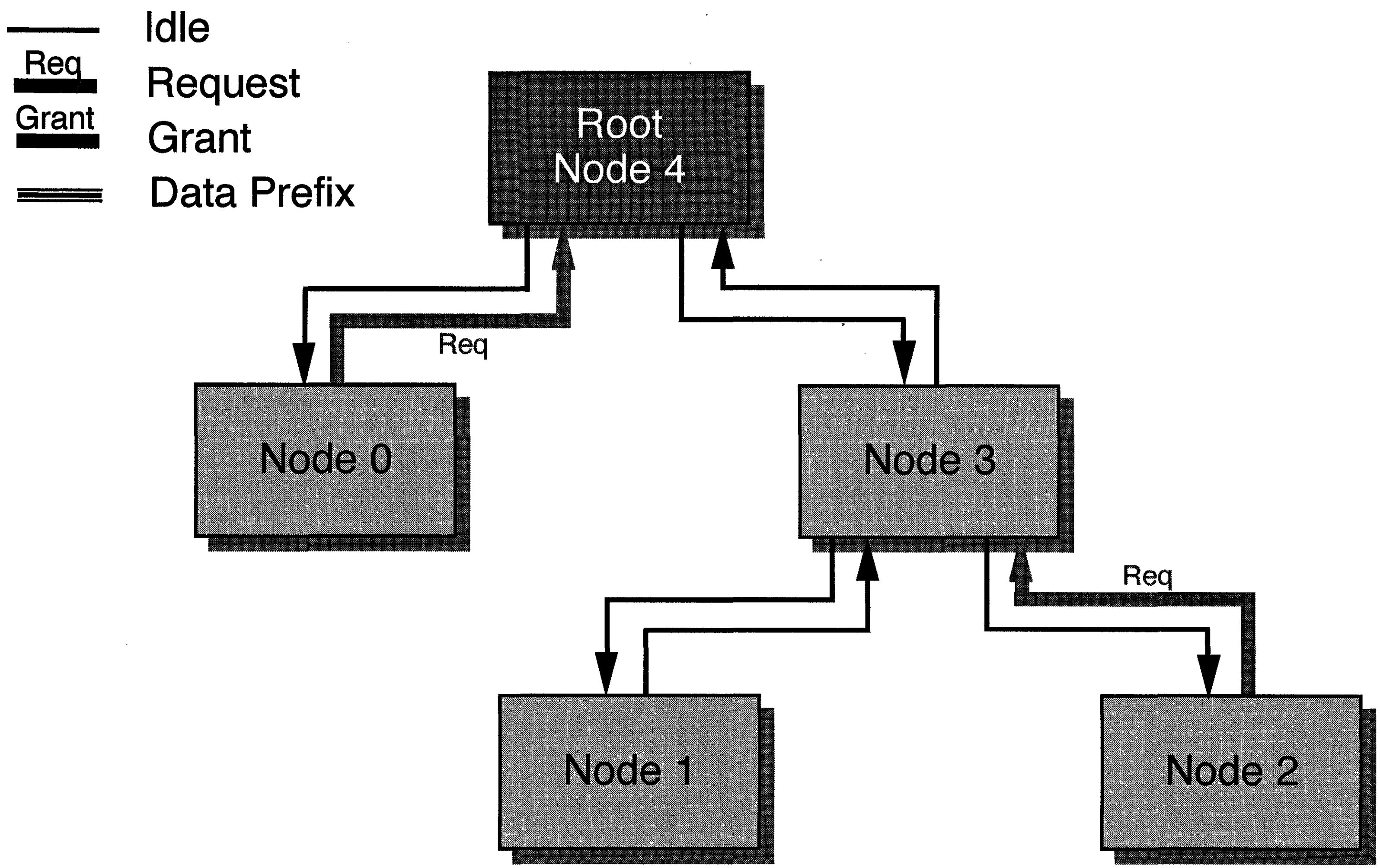
First Request Wins

Arbitration Example

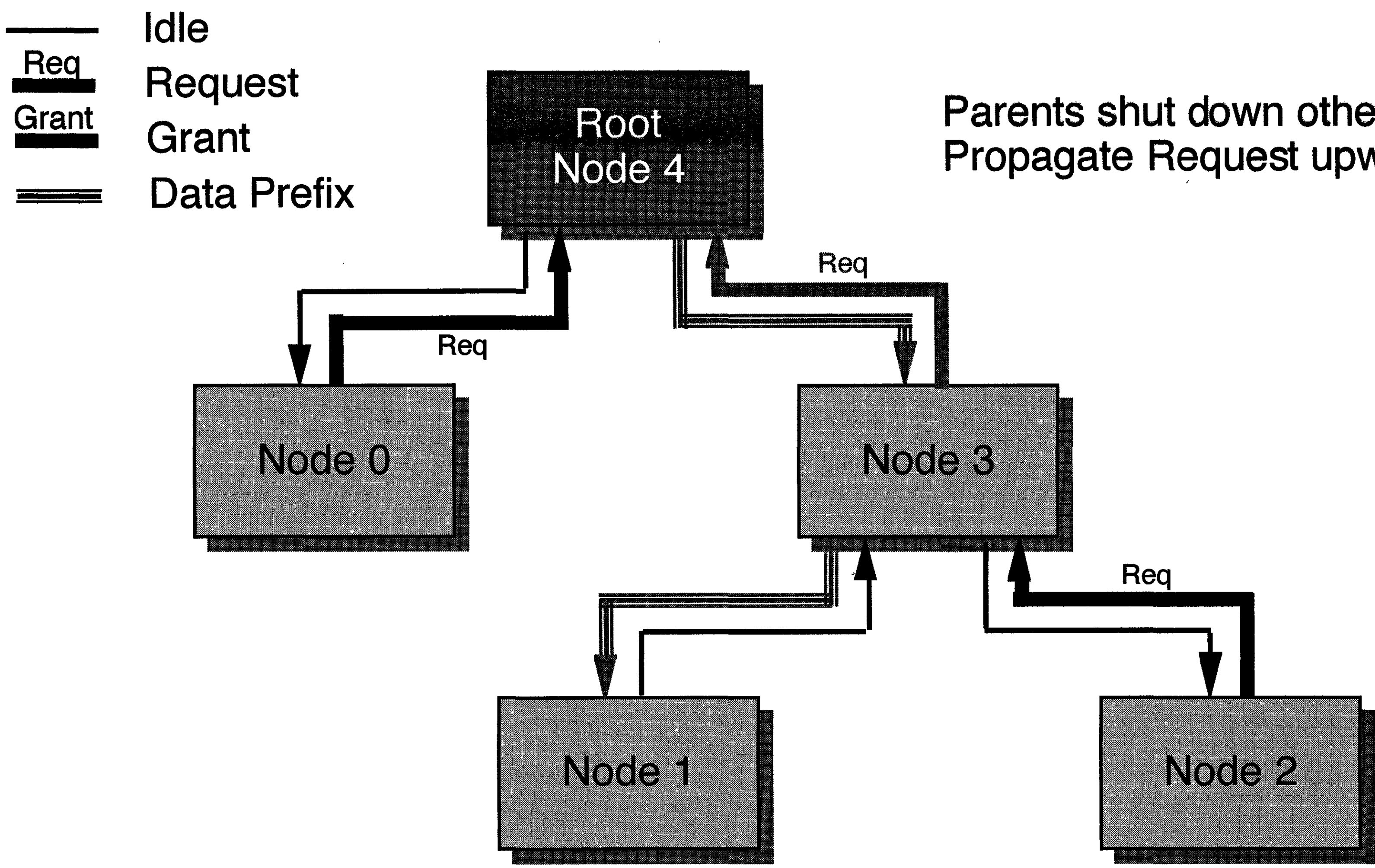


Node 0 and Node 2 want to send a packet

Arbitration Example - Initial Requests

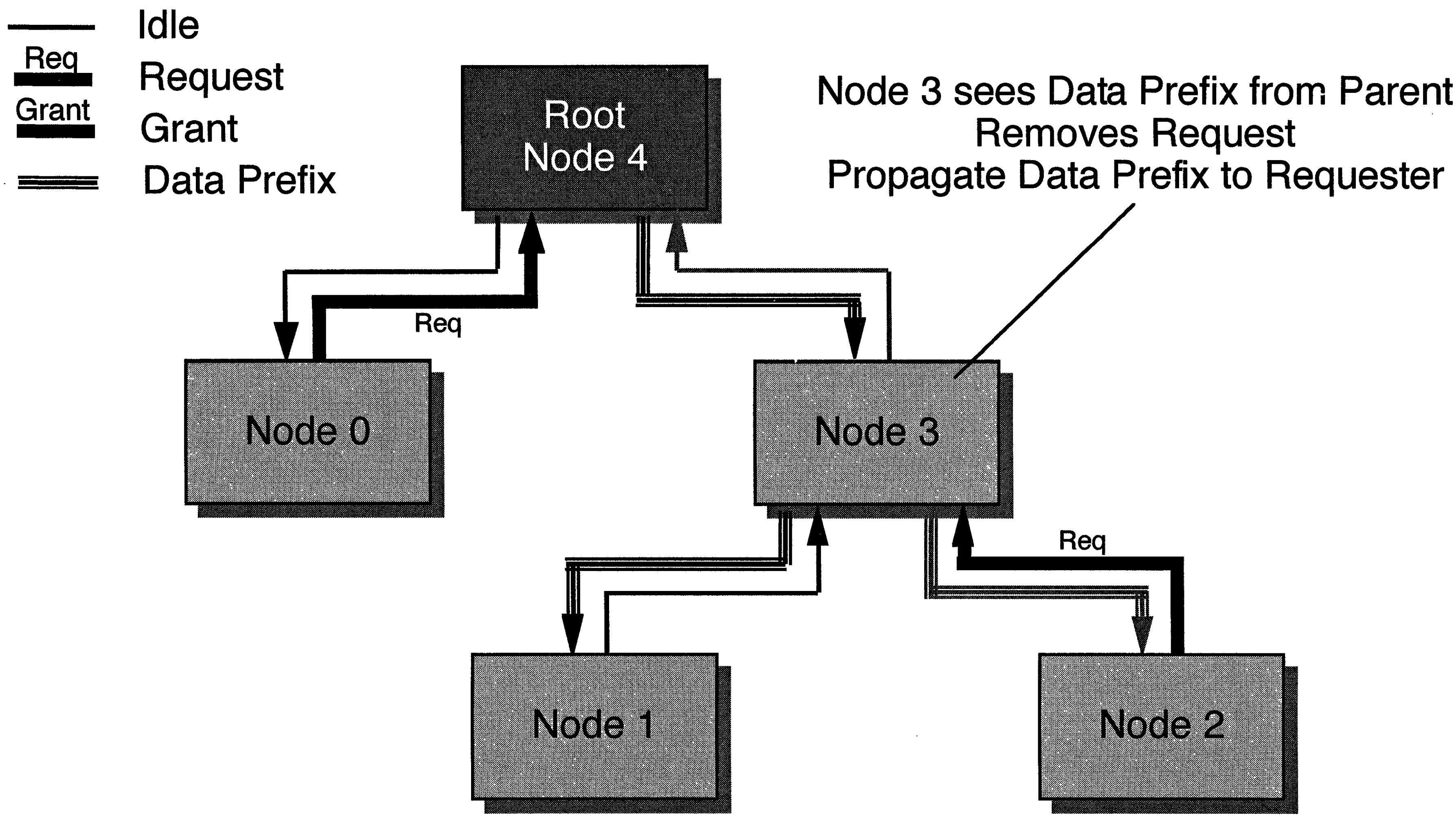


Arbitration Example - Parents Act

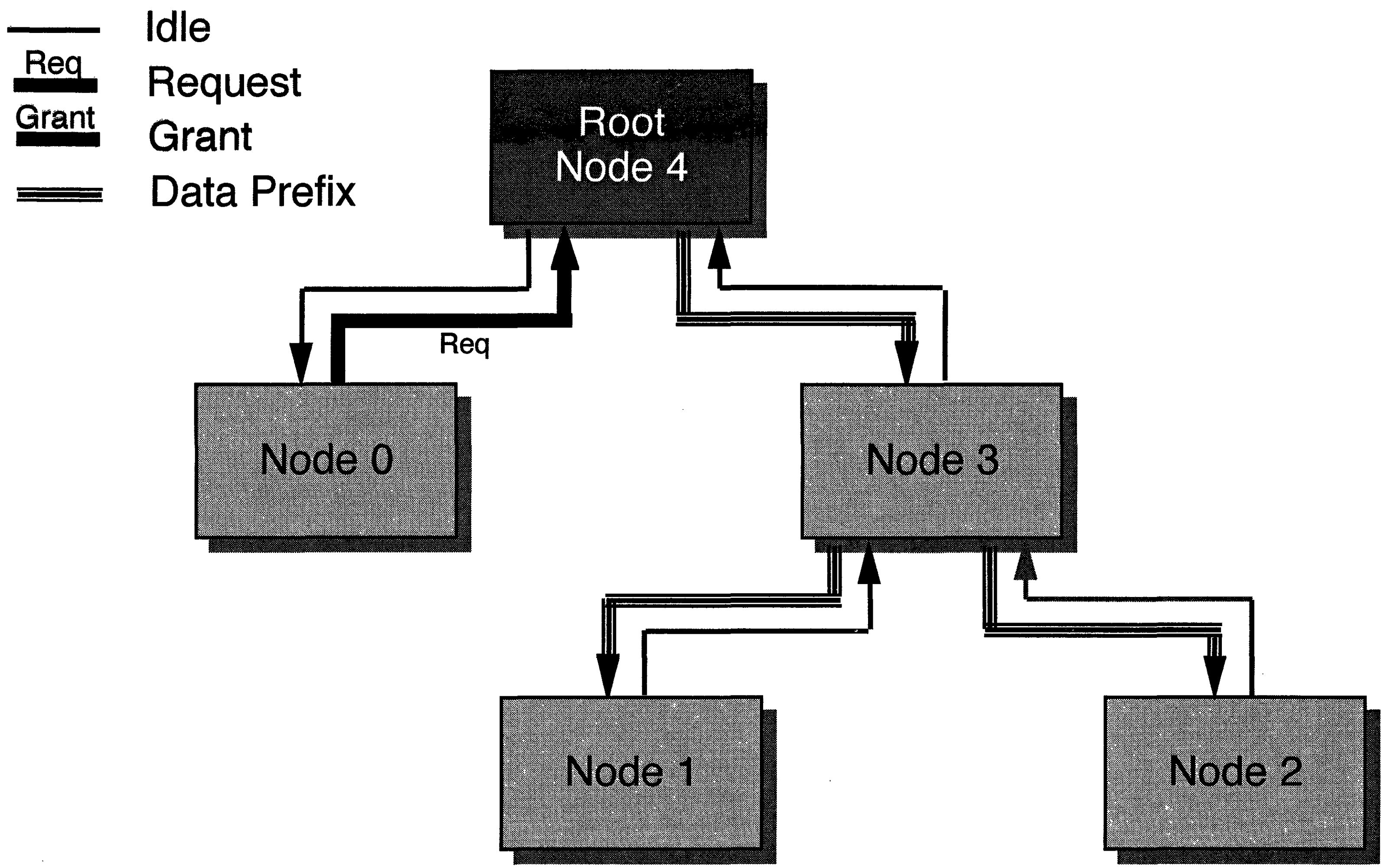


Parents shut down other children
Propagate Request upward

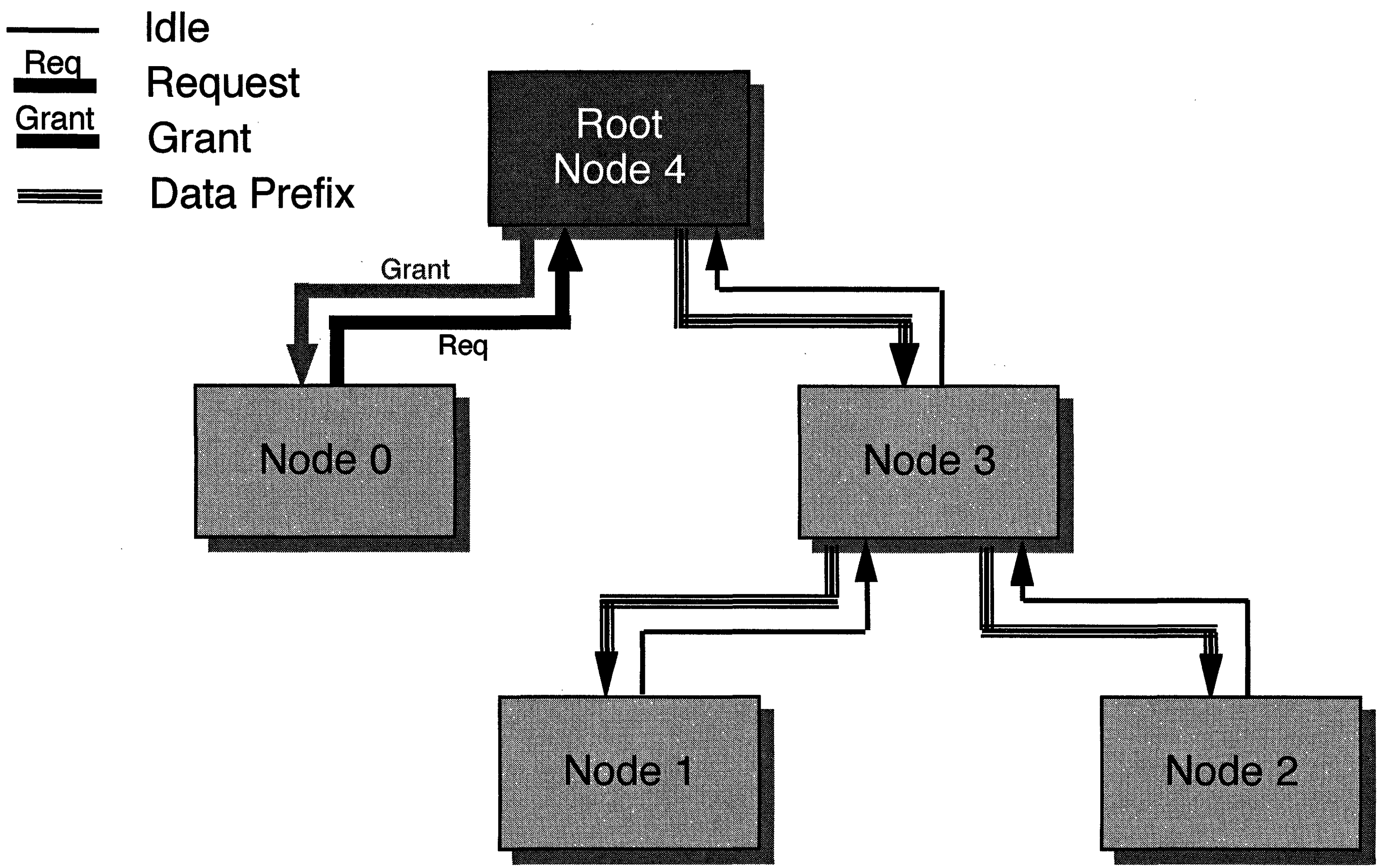
Arbitration Example - Losers Remove Request



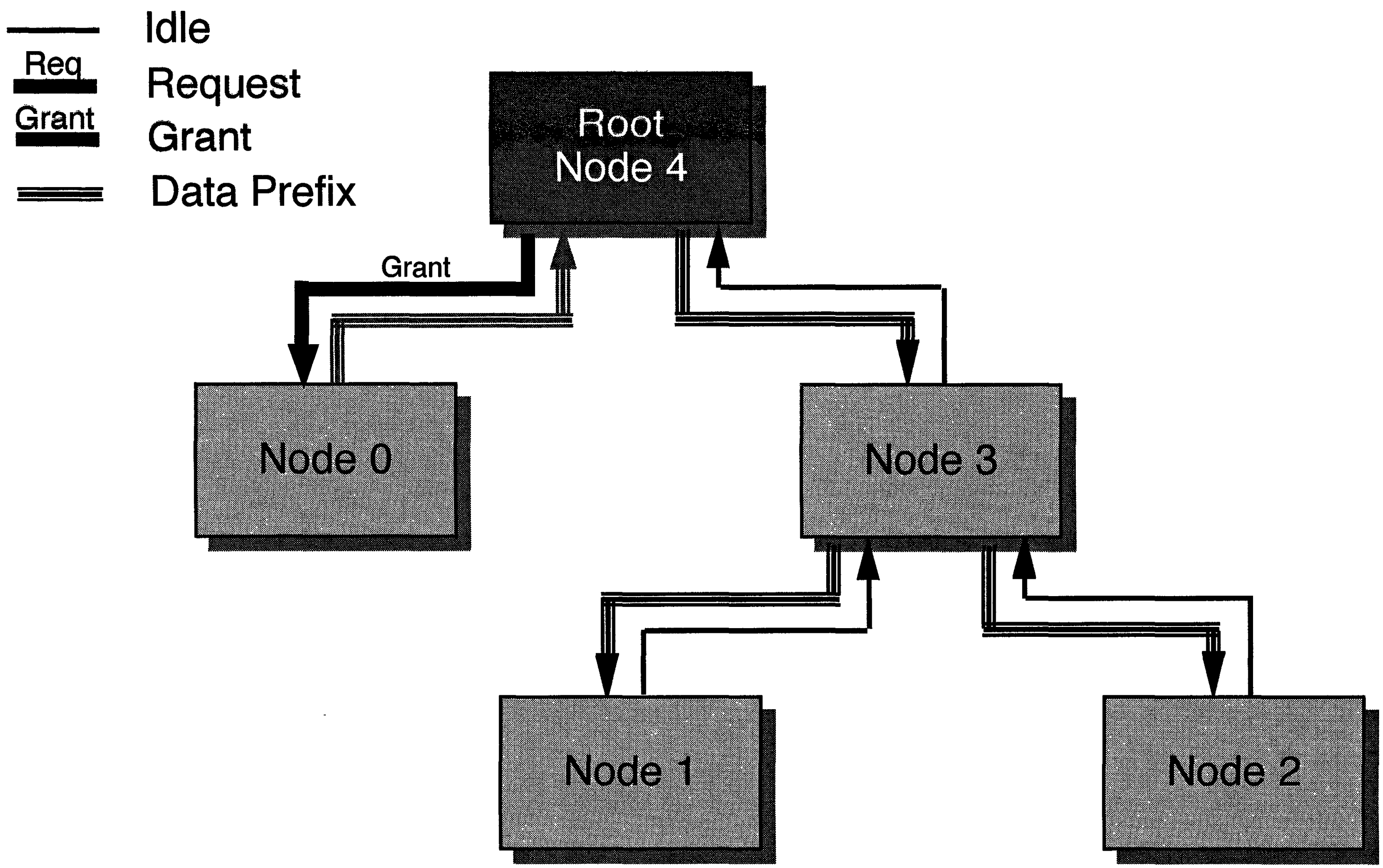
Arbitration Example - Losers Remove Request



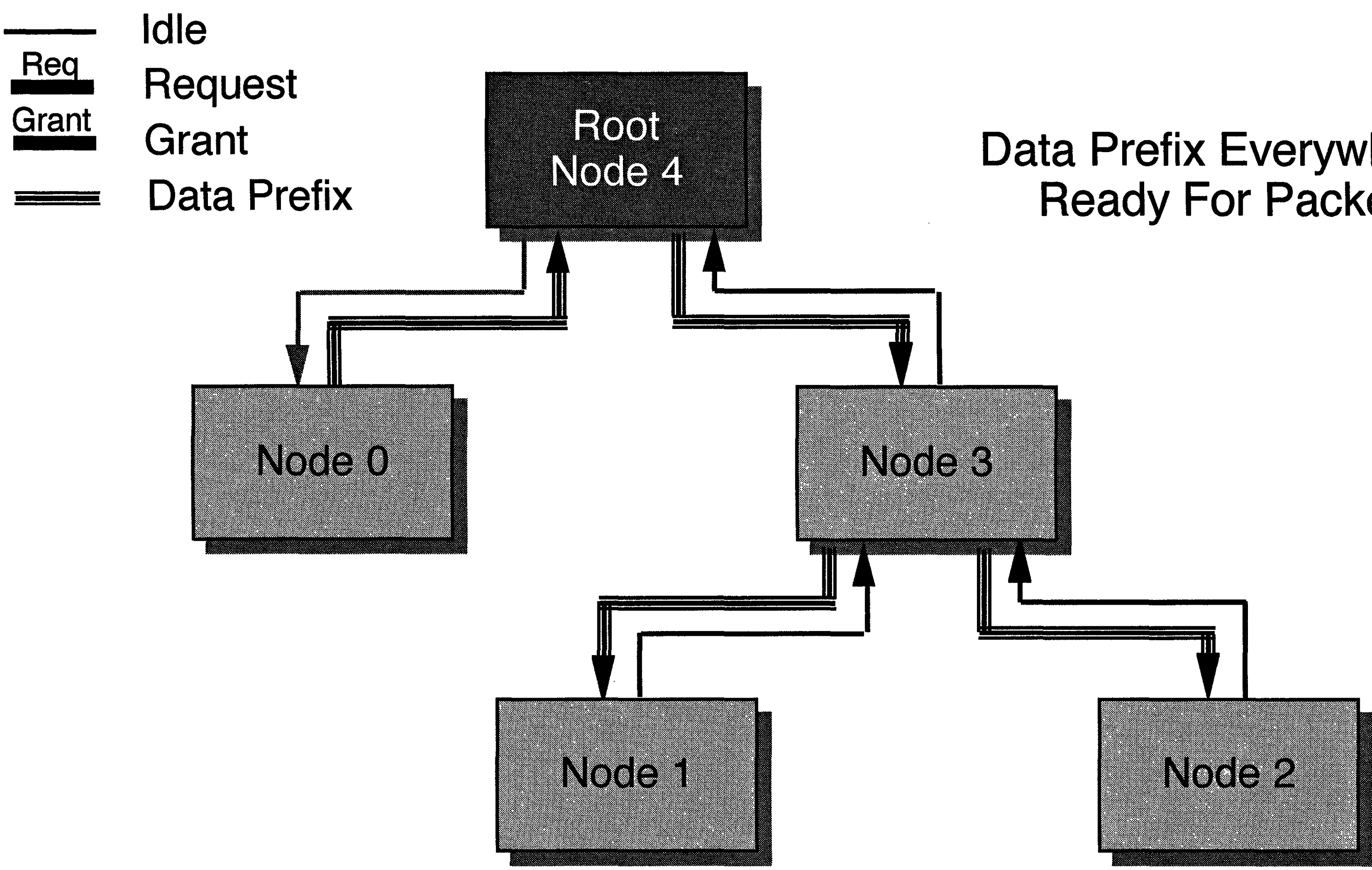
Arbitration Example - Root Issues Grant



Arbitration Example - Winner Issues Data Prefix



Arbitration Example - Root Removes Grant



Data Prefix Everywhere
Ready For Packet

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~~ obsolete map of the bus

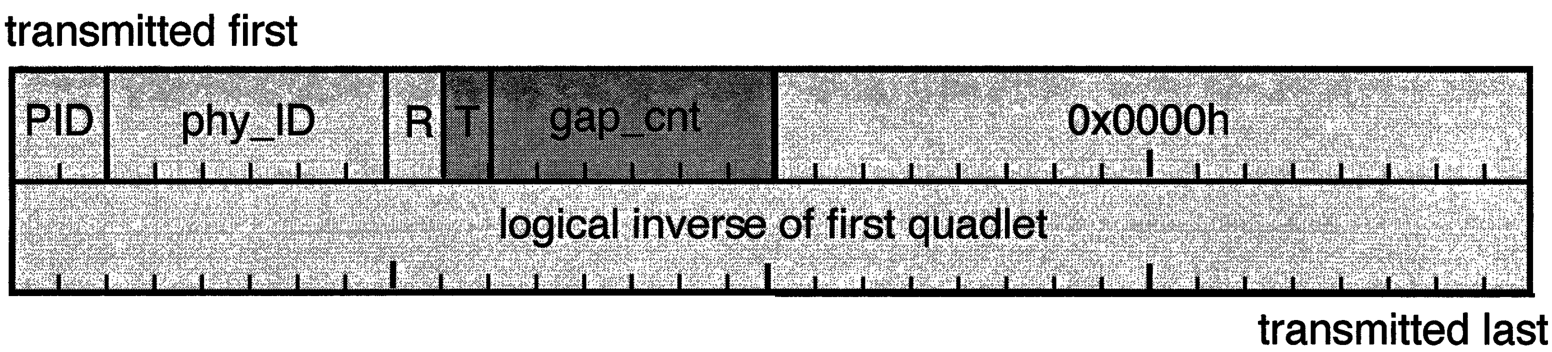
Computes worst case propagation times

Determines optimum timings

- Data Prefix
- Sub-Action Gap
- Arbitration Reset

Sends timings throughout bus via PHY packet

PHY Configuration Packet Format



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

Normal	Node Requests to use the bus
Token	Node closest to root arbitrates and passes grant to it's children. It will jump on with Fly-by arbitration.
Fly-By (eliminates arbitration) packet must be going to parent	Permits a transmitted packet to be concatenated to the end of a primary packet of which no ACK is permitted. (PHY, ACK, BRDCAST, ISOCHRONOUS) ↳ but can't conatenate another onto it
ACK Accelerated	PHY can arbitrate immediately following an observed ACK packet. Savings = subaction gap time.

defined in 1394-1995
now obsolete

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

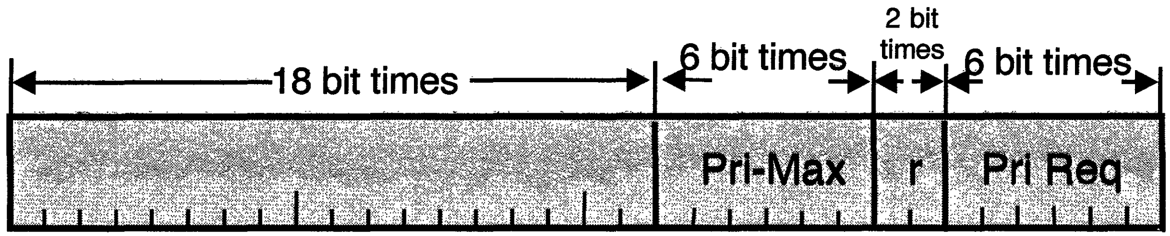
- Fly-by arbitration permits an asynchronous packet to be concatenated to an ACK, or an isochronous packet to be concatenated to a cycle start on another isochronous packet.

- 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



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
- ≠ 0 Priority requests, in excess of fairness defined in 1394-1995

additional accesses per fairness interval

Transaction Codes Eligible

extra accesses only for requests.

Responses don't follow fairness rules - responses are automatically fair since every response is to a request.

- 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

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?

1394 Arbitration Notes

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 example Fuji laser printer reduced 12MB buffer to 1/2MB buffer
- No ACK, No retry

Asynchronous Review

Translated means “not synchronized with time”

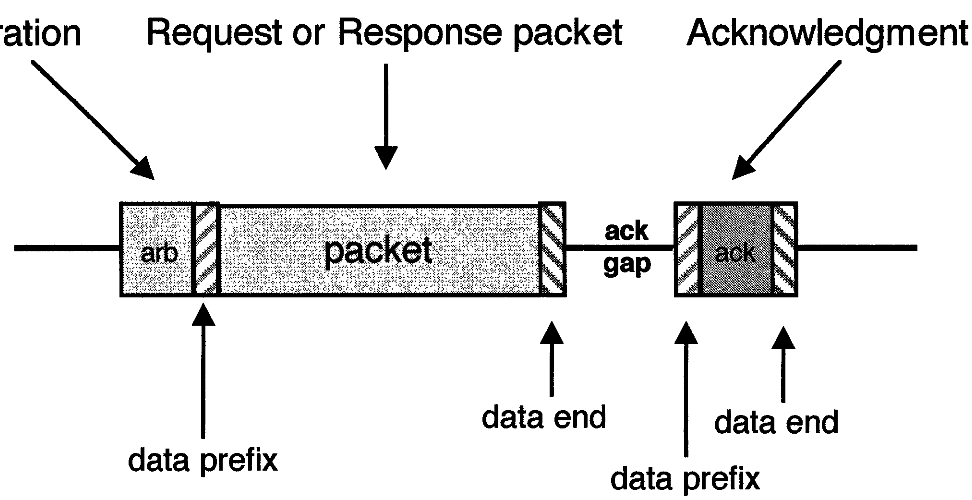
Guaranteed delivery

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

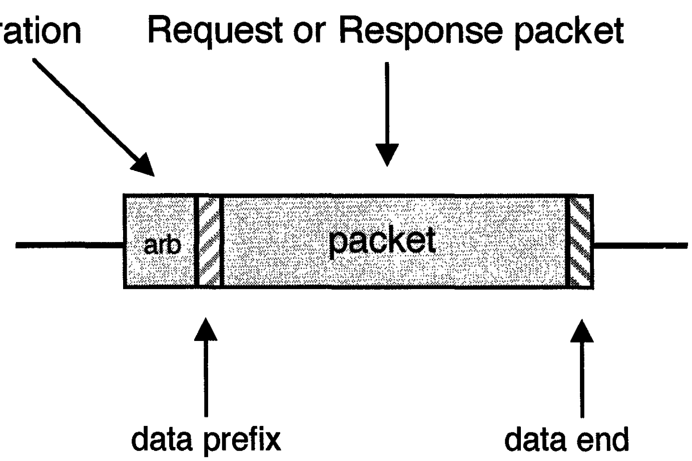
Guaranteed timing

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

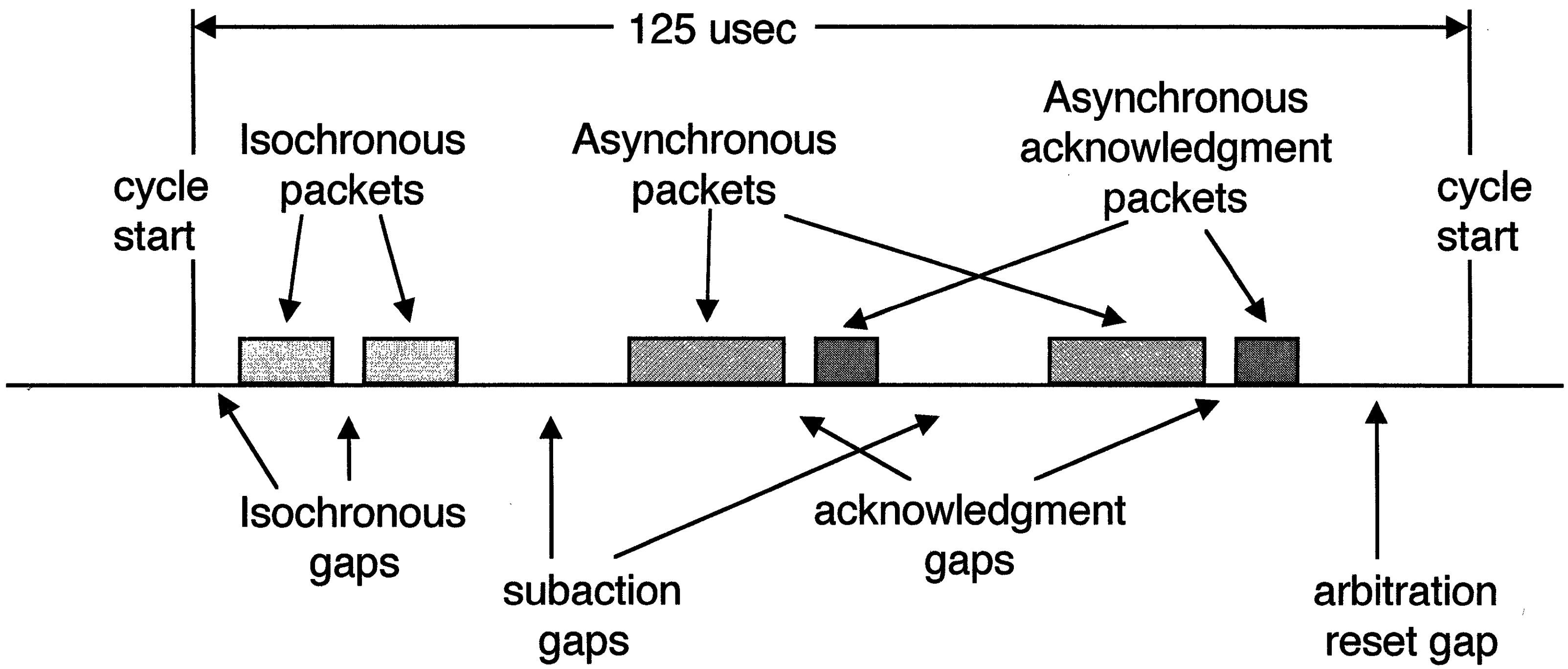
- Asynchronous nodes now see Sub-Action Gaps

Isochronous Cycle and Fairness Interval are independent

for asynch

may end/start @ same time
but don't have to.

Isochronous Cycle



Cycle Start Packet Format

Destination_ID = FFFFh	tl = 0	rt=0	tcode=8	pri=F
Source_ID	F F F F			
Cycle Time Register Address: F 0 0 0 0 2 0 0				
Cycle_Time_Data				
Header_CRC				

synchronized
if late but
always same amount
of time late.

Indicates start of Isochronous Cycle

Places Cycle_Time_Data ⟶ Cycle_Time_Register

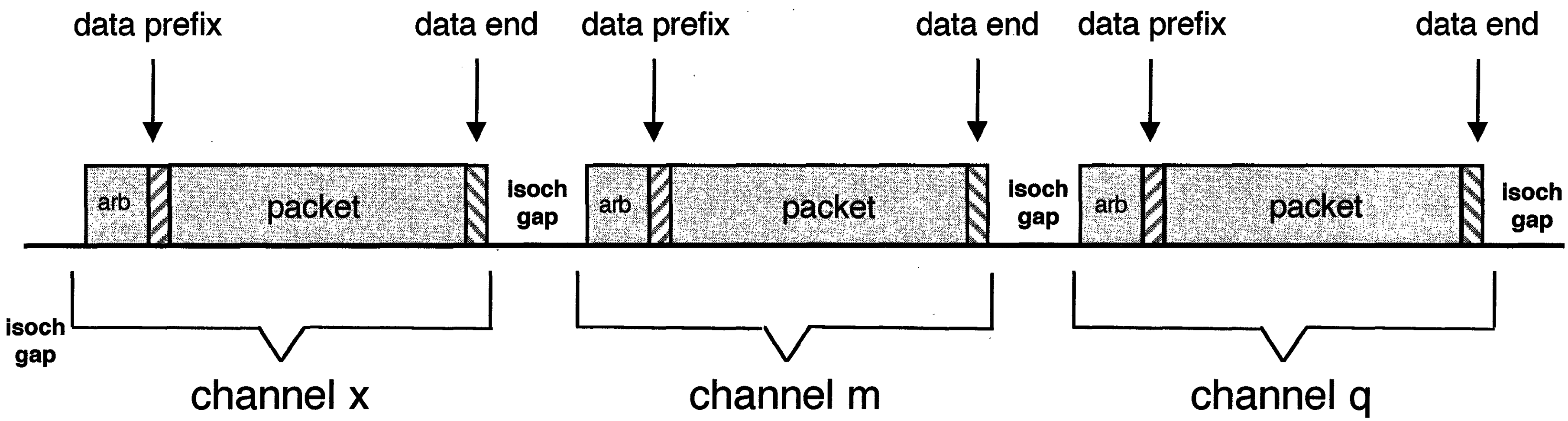
(For each Isochronous capable node)

Cycle Start Packet Contents

Destination_ID	Always set to FFFFh
Source_ID	Node ID of Cycle Master (Root)
tl	Transaction Label (set to 0)
tcode	Transaction Code (8 = Cycle Start)
pri	Priority, for backplane environment Set to Fh
Cycle Time Register Address	Always set to FFFF F000 0200h
Cycle Time Data	Time at transmission of packet

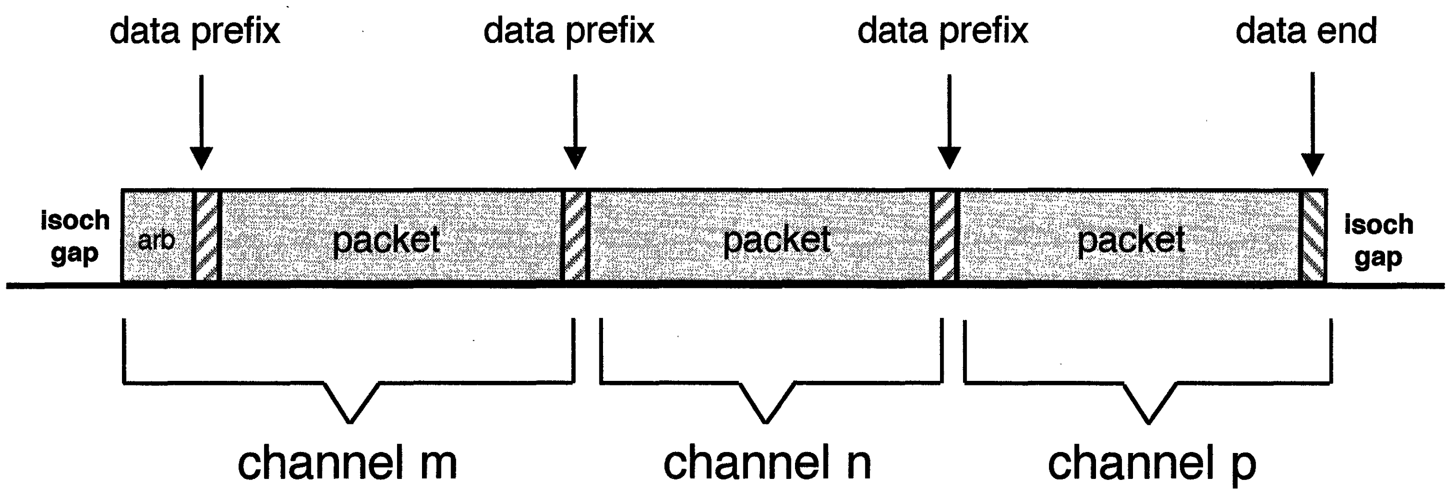
V

Example Isochronous Subactions



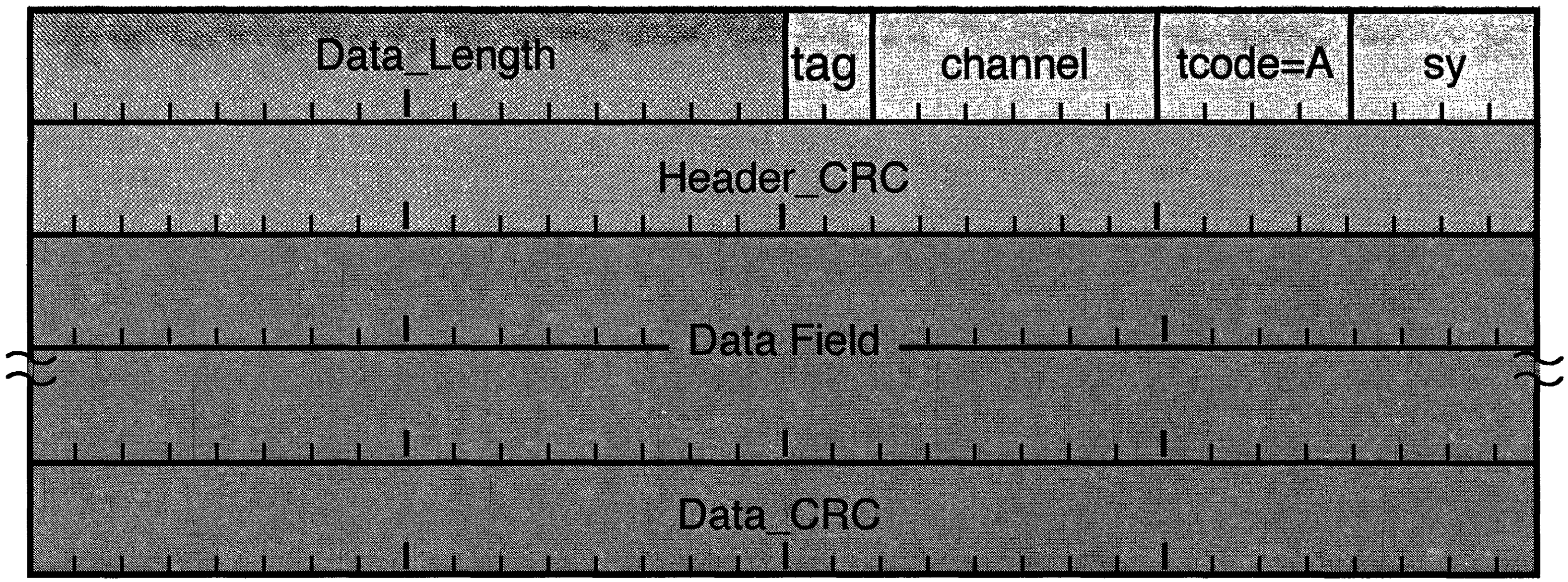
3 different iso talkers on bus

Example Concatenated Isochronous Subactions



1 talker - arbitrated in non - each packet sent to a different channel
3 talkers - use fly by arbitration
2 talkers - combo of the 2 above

Isochronous Data-Block Packet Format



Note that packet is different from the asynchronous packet format

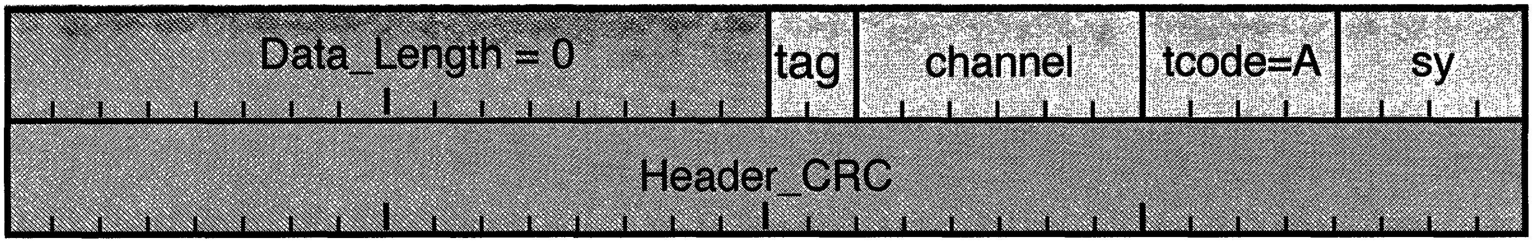
Look @ tcode first

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)

V

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

ycle start packet can be delayed 83 µs

ycle start packet must see gap

must be pulled back in - (longest it will take is 4 cycles) by not allowing any async devices back on.

Data Rate	Maximum Payload Size (bytes)	Comment
S25	128	TTL backplane
S50	256	BTL and ECL backplane
S100	512	cable base rate
S200	1024	
S400	2048	
S800	4096	1394b
S1600	4096	1394b

↳ controls time → isochronous

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 RMnning 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

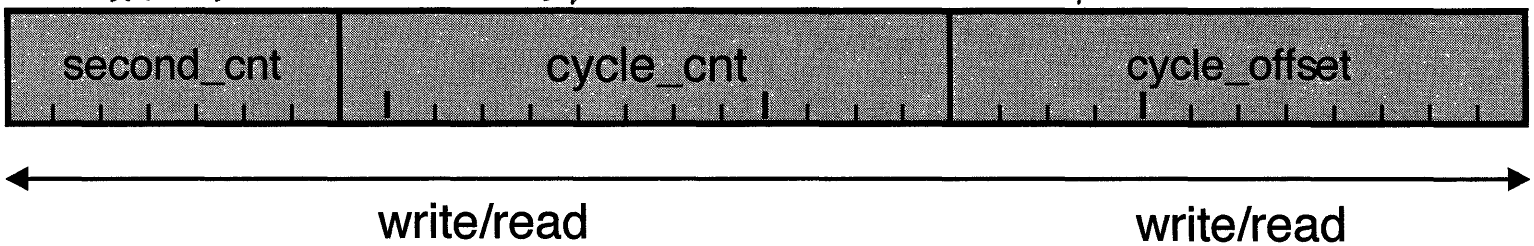
- Describe Isochronous Capabilities in Bus_Info_Block

31h "1"						33h "3"	39h "9"		34h "4"		
IR	C	I	B	PMC	Resv	cyc clk acc	max rec	reserved	G	r	Link Speed
Node vendor ID									chip ID high		
chip ID low											

Cycle Time Register - in isochronous mode

At Address NNNN FFFF F000 0200h:

seconds cycles parts of a cycle



cycle_offset	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)	
second_cnt	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 μSec

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

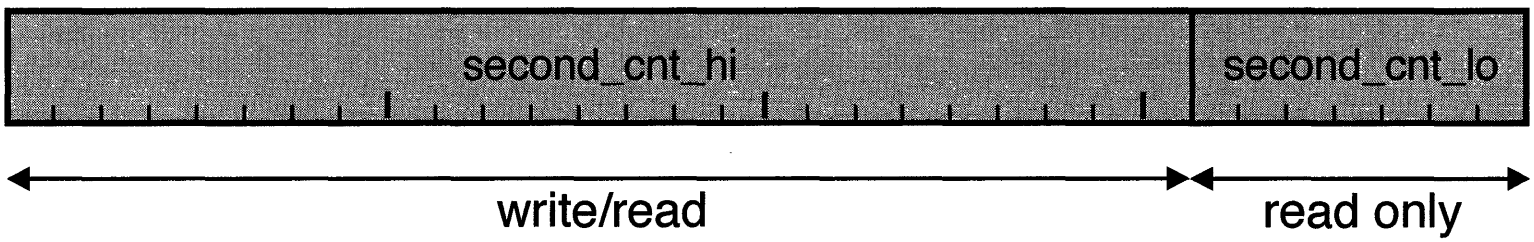
- At CSR offset 0204h
- Keeps universal bus time in seconds
- Rolls over every 136 years
- Set by the Bus Manager

Indicates Cycle Master Capable in Config ROM

<table>
<tr><td colspan="6">31h “1”</td><td>33h “3”</td><td colspan="3">39h “9”</td><td colspan="3">34h “4”</td></tr>
<tr><td>IR</td><td>C</td><td>I</td><td>B</td><td>PMC</td><td>Resv</td><td>cyc clk acc</td><td>max rec</td><td>R</td><td>max ROM</td><td>G</td><td>r</td><td>Link Speed</td></tr>
<tr><td colspan="10">Node vendor ID</td><td colspan="3">chip ID high</td></tr>
<tr><td colspan="13">chip ID low</td></tr>
</table>

Bus Time Register

Address NNNN FFFF F000 0204h:



second_cnt_lo	The second_cnt field of the Cycle_Time Register
second_cnt_hi	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)

<table>
<tr><td colspan="6">31h "1"</td><td>33h "3"</td><td colspan="3">39h "9"</td><td colspan="3">34h "4"</td></tr>
<tr><td>IR</td><td>C</td><td>I</td><td>B</td><td>PMC</td><td>Resv</td><td>cyc clk acc</td><td>max rec</td><td>R</td><td>max ROM</td><td>G</td><td>r</td><td>Link Speed</td></tr>
<tr><td colspan="10">Node vendor ID</td><td colspan="3">chip ID high</td></tr>
<tr><td colspan="13">chip ID low</td></tr>
</table>

Bandwidth_Available Register

Address NNNN FFFF F000 0220h:

reserved	bw_remaining

bw_remaining — Amount of Isochronous Bandwidth remaining

Measured in allocation units (au)

1 allocation unit = Quadlet time @ S1600

Maximum Bandwidth ($\underbrace{100\ \mu Sec}_{\text{max for isoch transactions}}$) = $\overset{\text{decimal}}{4915}$ au set back by bus reset

Register for information only, no direct control

Reg checked by IRM

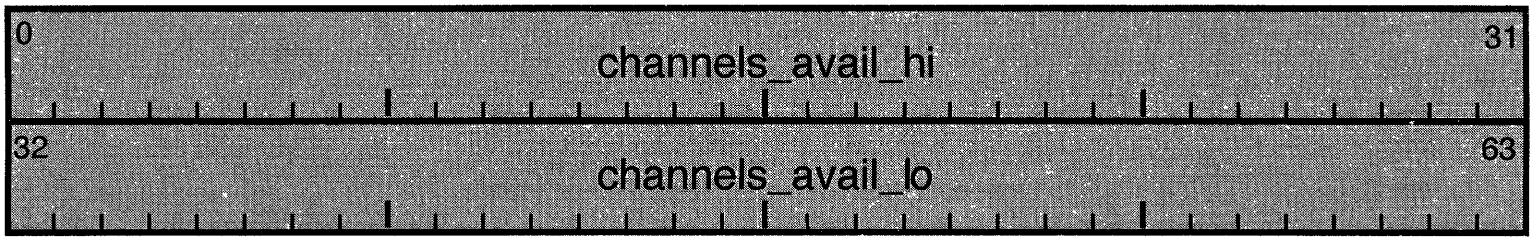
node sends lock transaction to IRM to get aus for its isoch. transaction

if request more than what's left must wait.

NNNN = Node ID

Channels Available Register

Address NNNN FFFF F000 0224h:



Bit map

Bit indicates corresponding channel available

on bus reset
all bits =1 (on)

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:

reserved	bmgr_id

bmgr_id — Bus Manager ID on reset = 3F

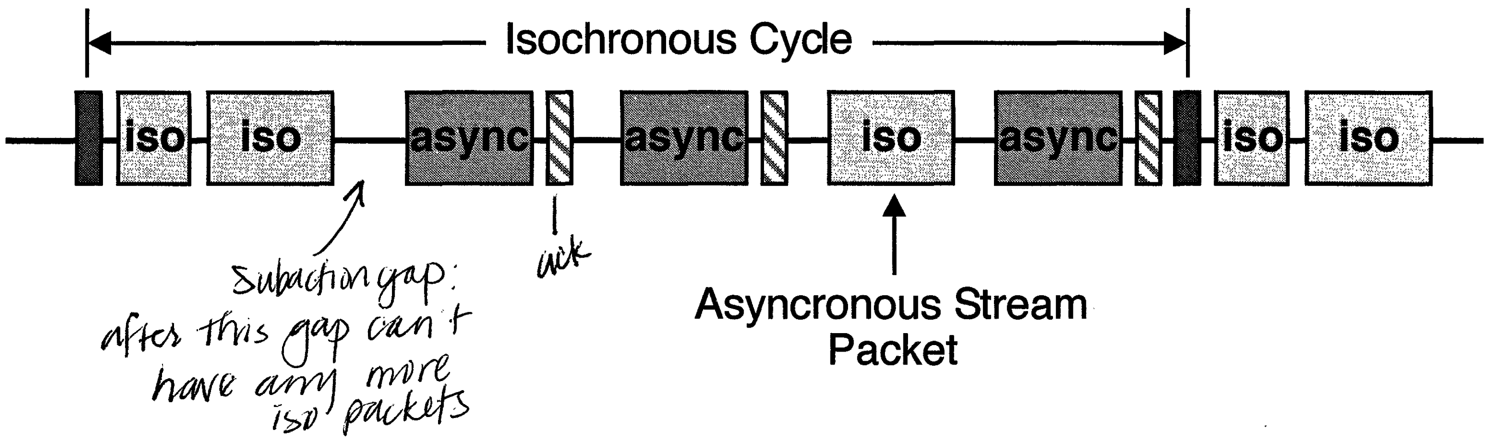
Node ID on this bus of Bus Manager

Must be accessed with a Lock (compare & swap)

NNNN = Node ID

Asynchronous Streams

example of use: store movie to harddrive



Isochronous Format Packet

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

no bandwidth guarantee

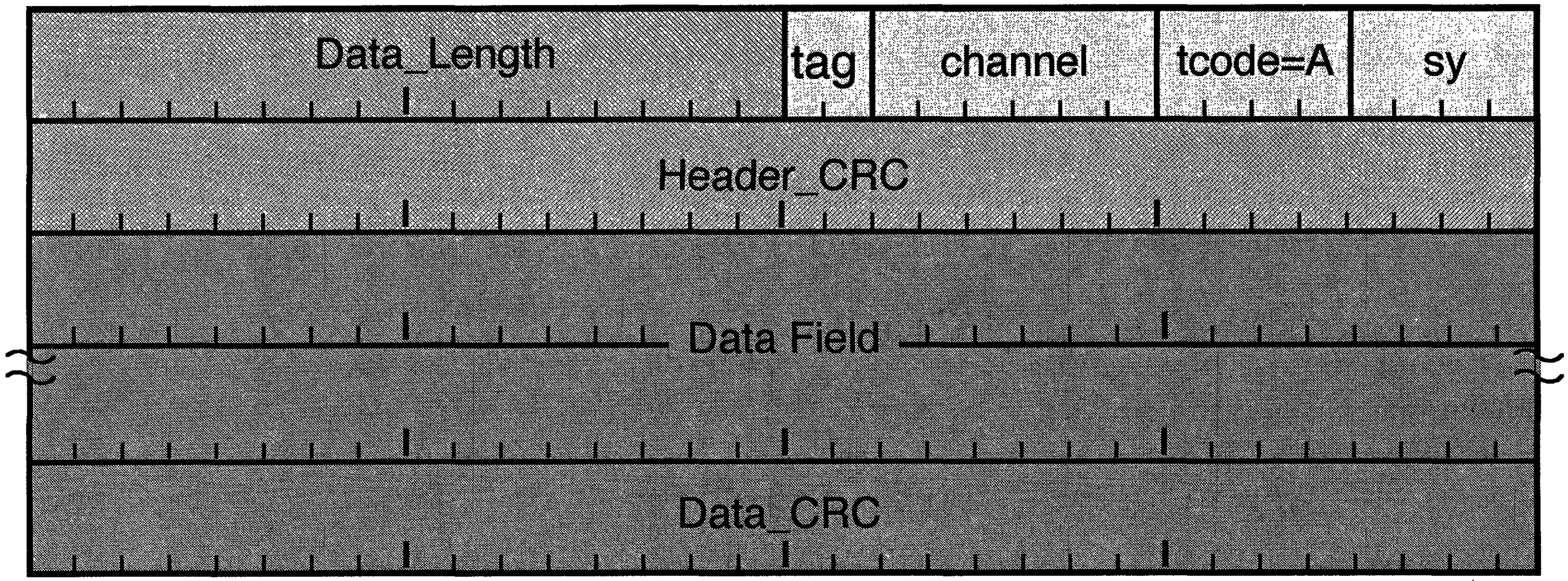
Transmitted during Asynchronous Time

- Arbitration & Fairness

Uses Isochronous Hardware

Does not compete for ISOCH Bandwidth

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

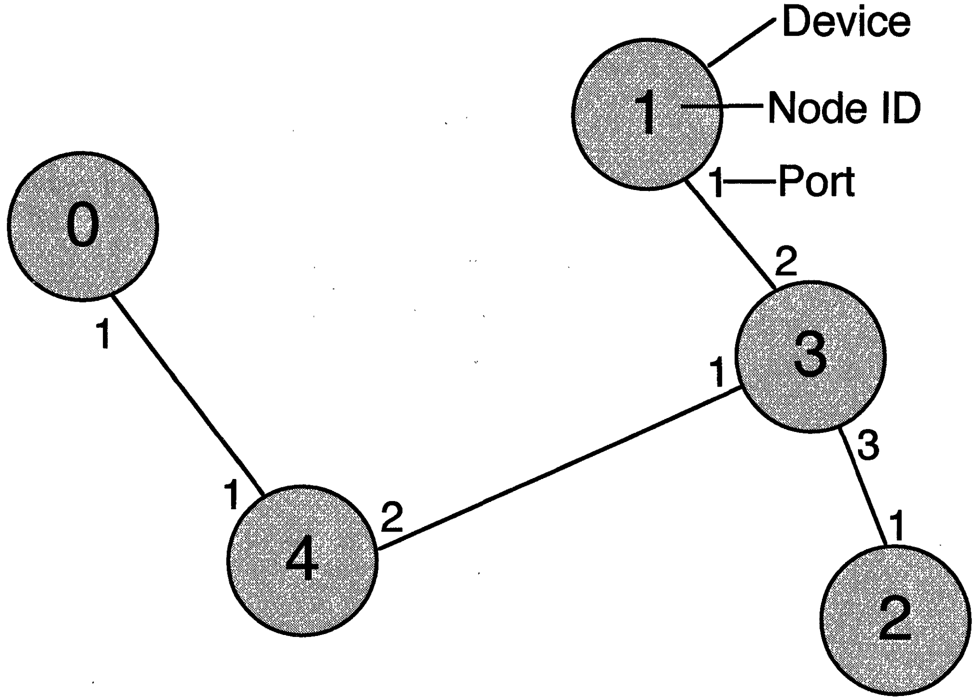
Subjects Covered

Resets

Tree ID

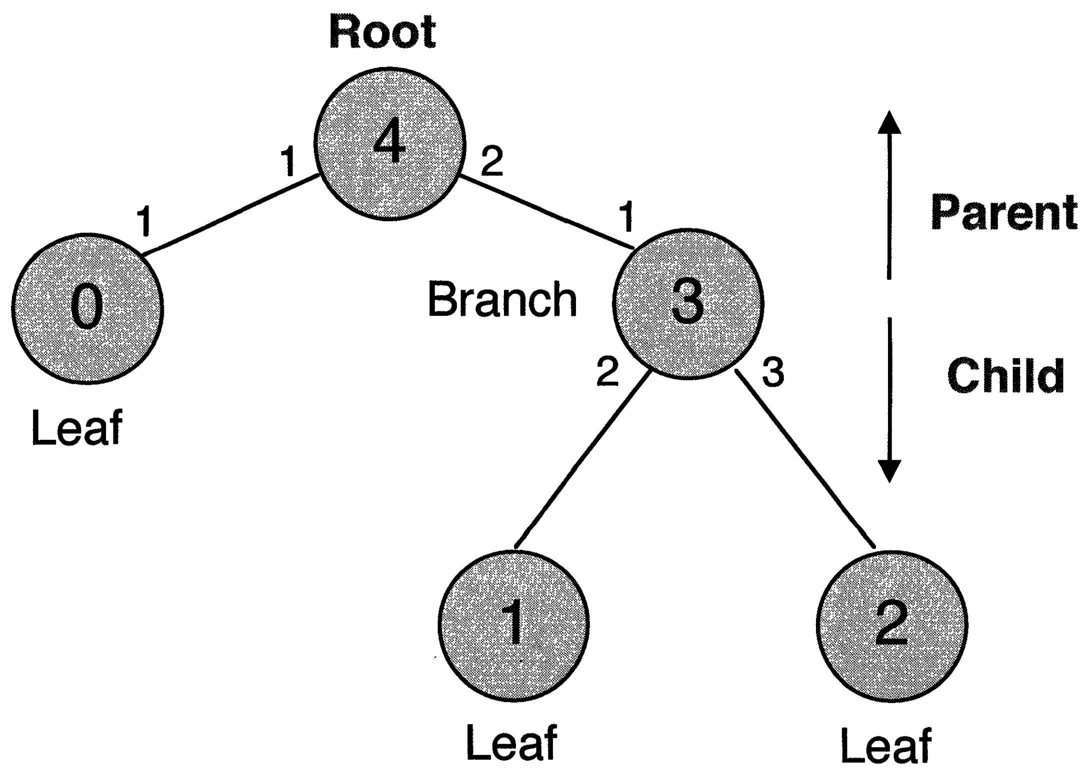
Self ID

Physical Topology (Review)



Numbering and Root determined during Configuration

Logical Topology (Review)



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

Configuration Process

Reset

↓

Tree Identify — Identifies Root

↓

Self Identify — Numbers the nodes
Communicates speed info

↓

Cycle Master Starts — Generates Cycle Start Packets
Prior Isochronous Transfers resume

↓

IRM Identified — Identifies which Node contains
Isochronous Resource Manager

↓

Bus Management
Initialization

Resets

HIERARCHY

Power Reset

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

Bus Reset

HOLD TPA=TPB=1 for 168 μs

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

propopagated thru bus

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 ?**

Connect Detect

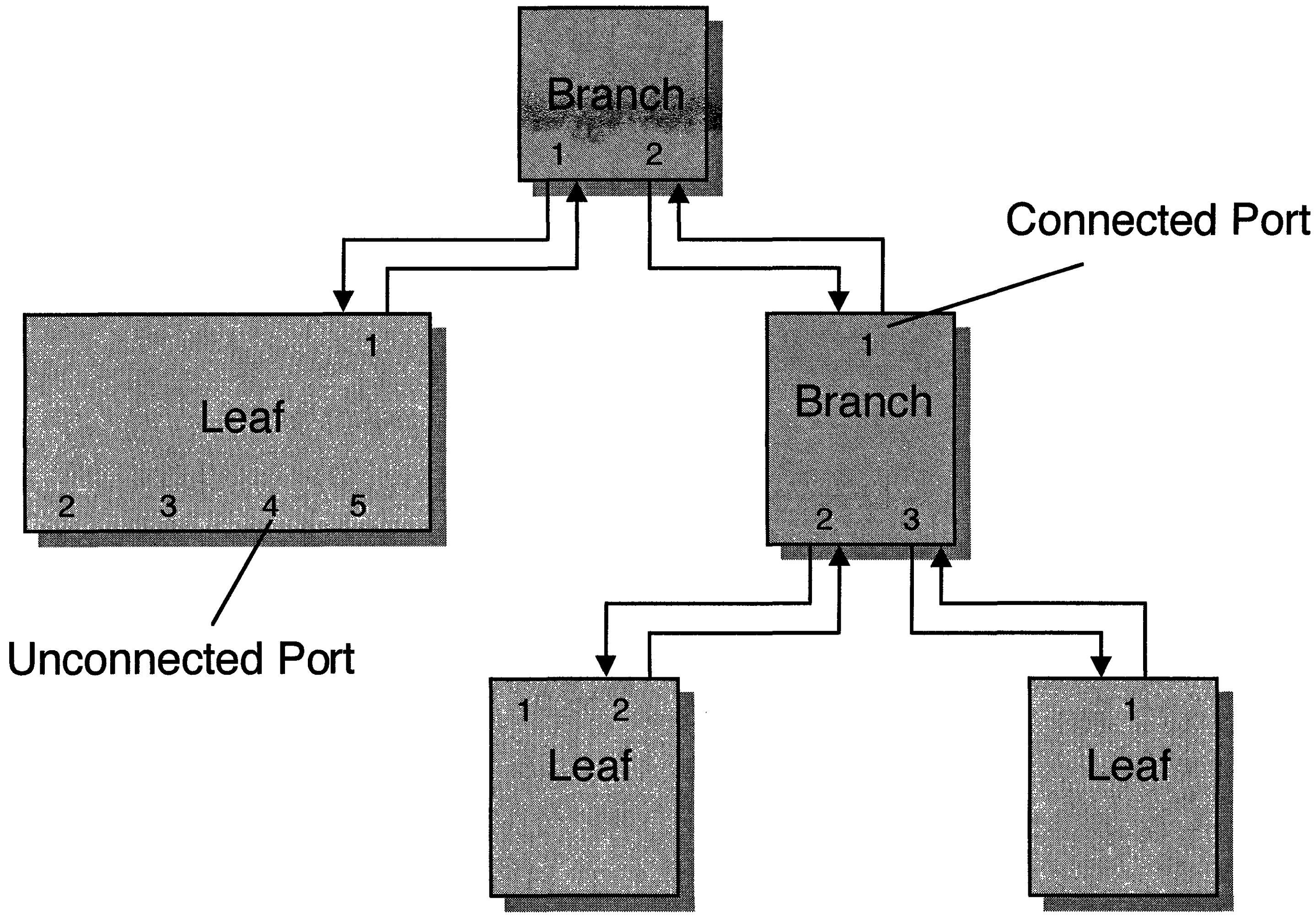
Two categories of Nodes:

Leaf	Only one Port connected
Branch	More than one Port connected

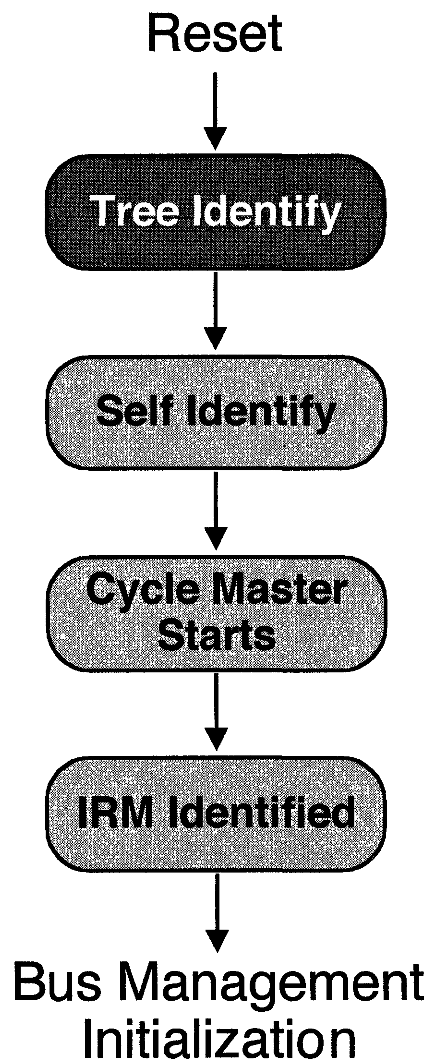
Nodes do not know their ID

Root is unknown

Example Topology



Tree Identify



Identifies Root

Tree Identify Strategy

Leaf Nodes

- Transmit Parent_Notify through only port
- Wait to receive Child_Notify
- Transmit Idle

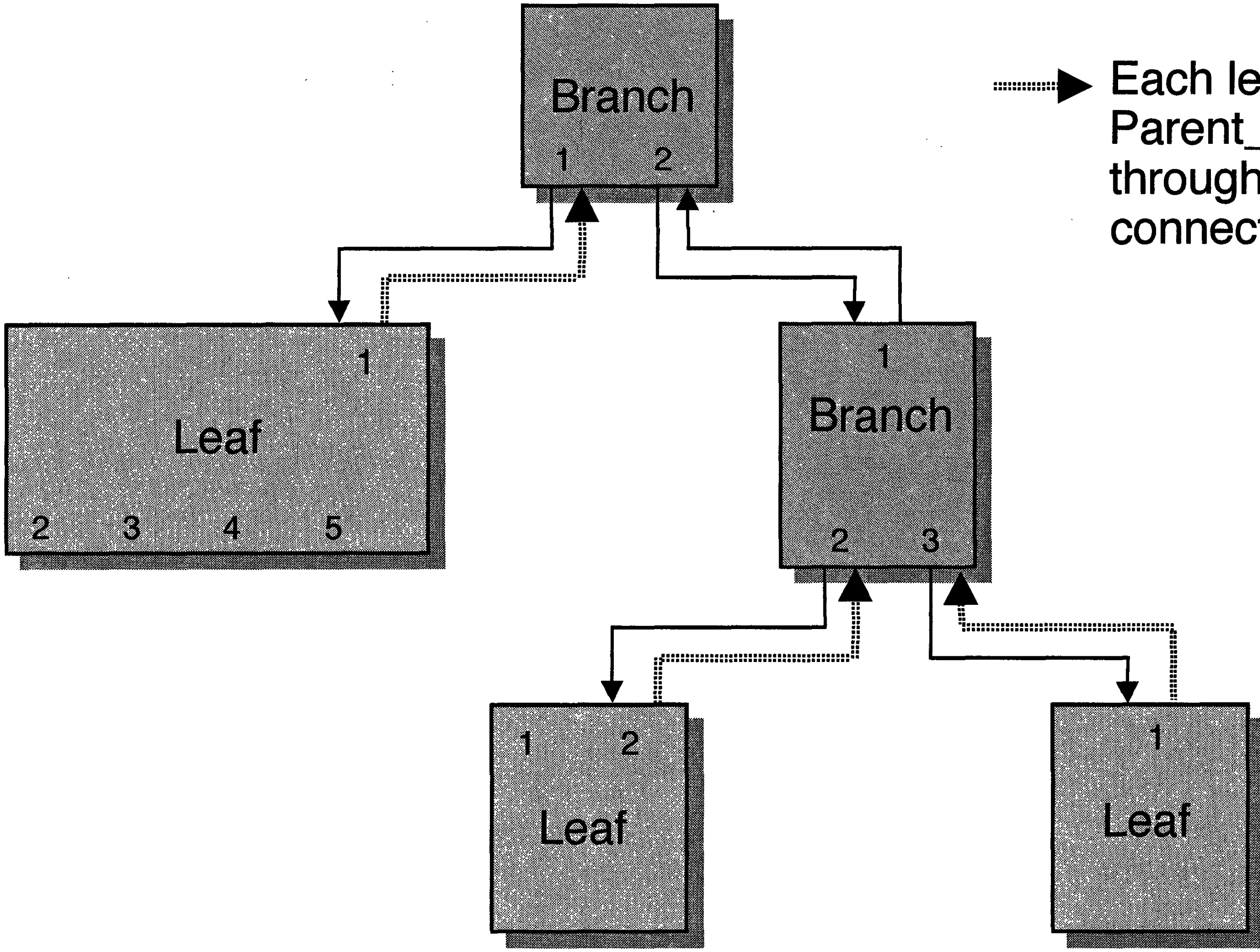
Branch Nodes

- 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

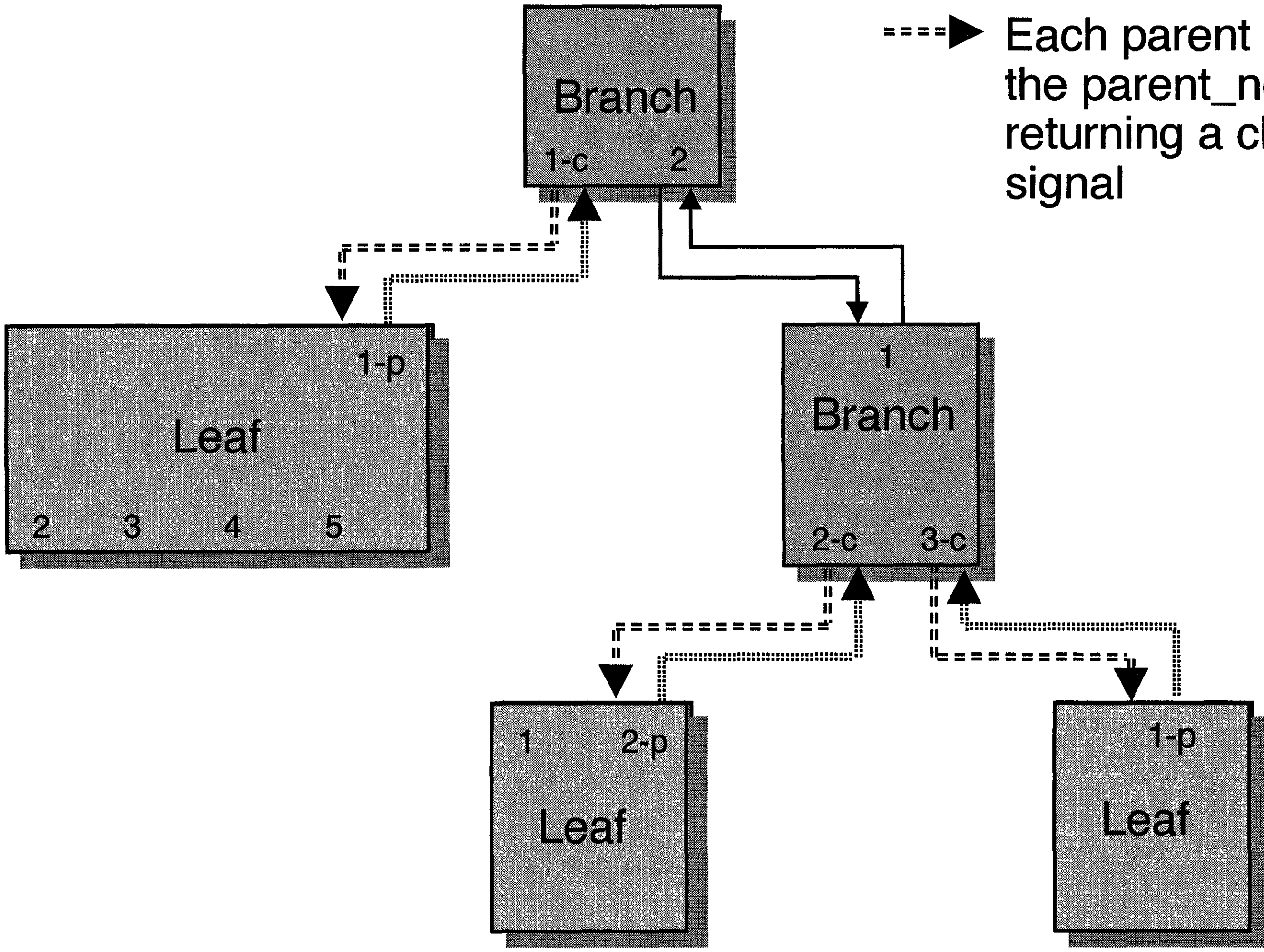
Signal Transmitted	Arb A Tx	Arb B Tx	Comment
Idle	Z	Z	Sent to indicate a gap
Request	Z	0	Sent to parent to request the bus
Grant			Sent to child when bus is granted
Parent Notify	0	Z	Sent to parent during Tree-ID
Data Prefix	0	1	Sent before data packets
Child Notify	1	Z	Sent to child to Ack parent notify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

Tree Identify: Leaf's Send Parent Notify



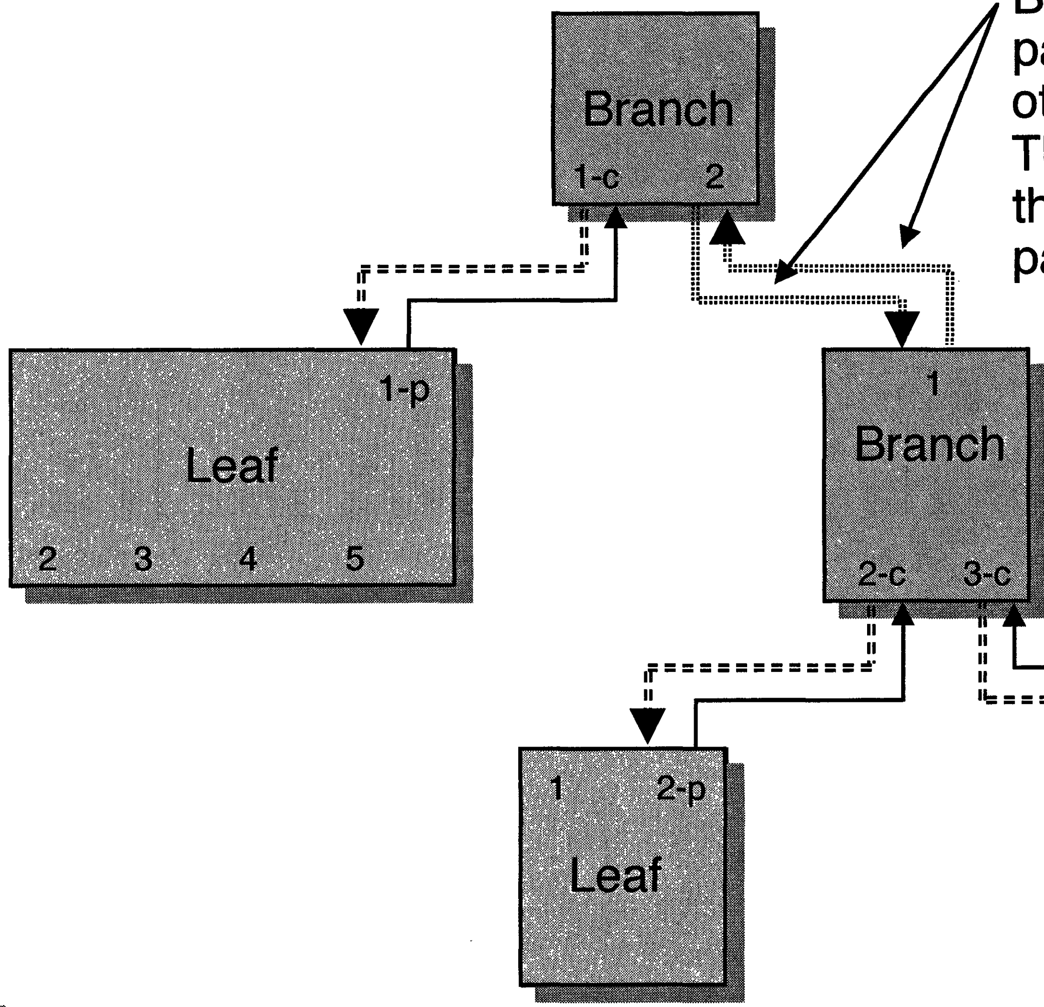
Each leaf sends a Parent_Notify signal through its single connected port

Tree Identify: Parents Acknowledge Children



Each parent acknowledges the parent_notify signal by returning a child_notify signal

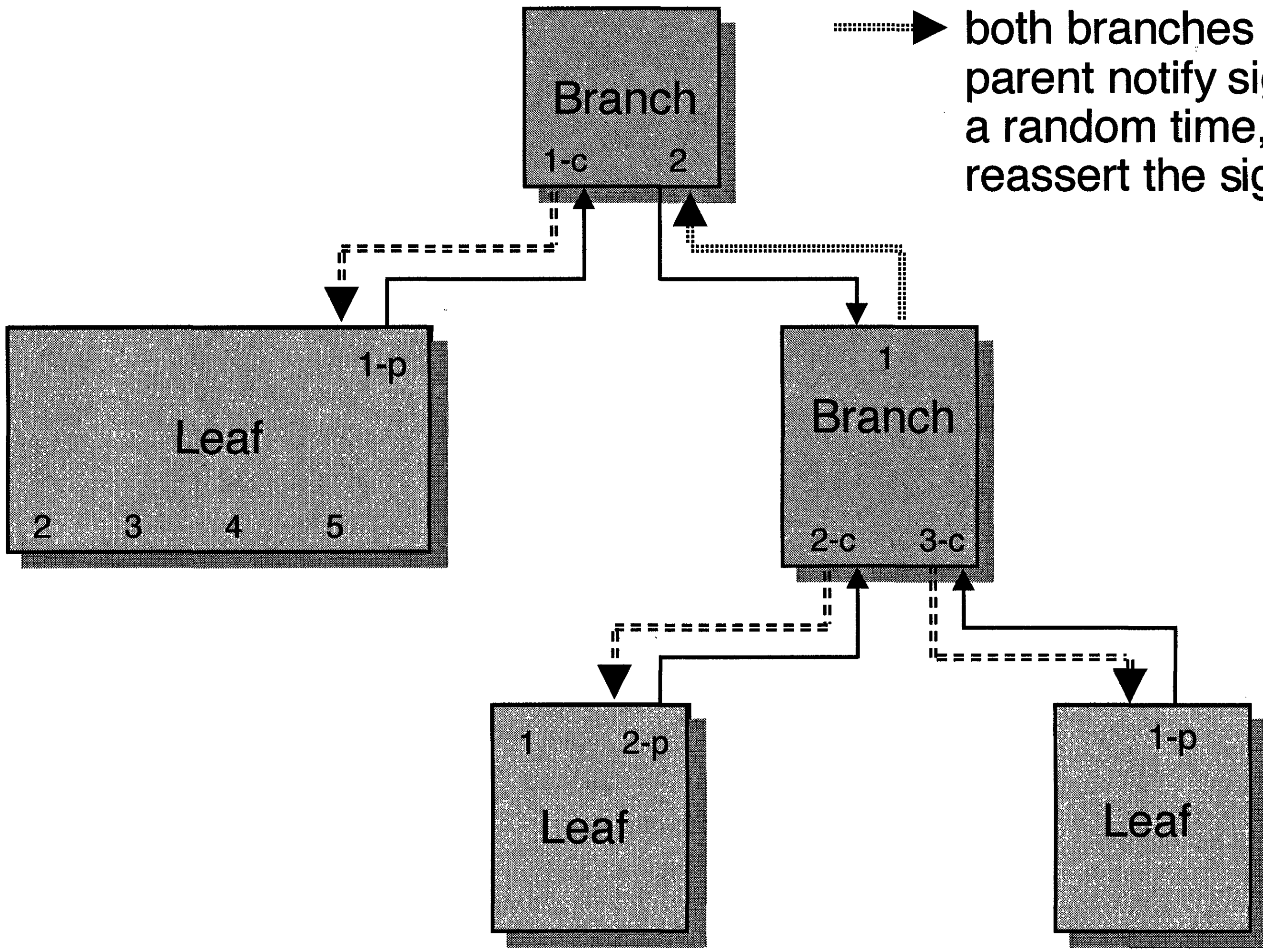
Tree Identify: Branches Notify Deduced Parents



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...

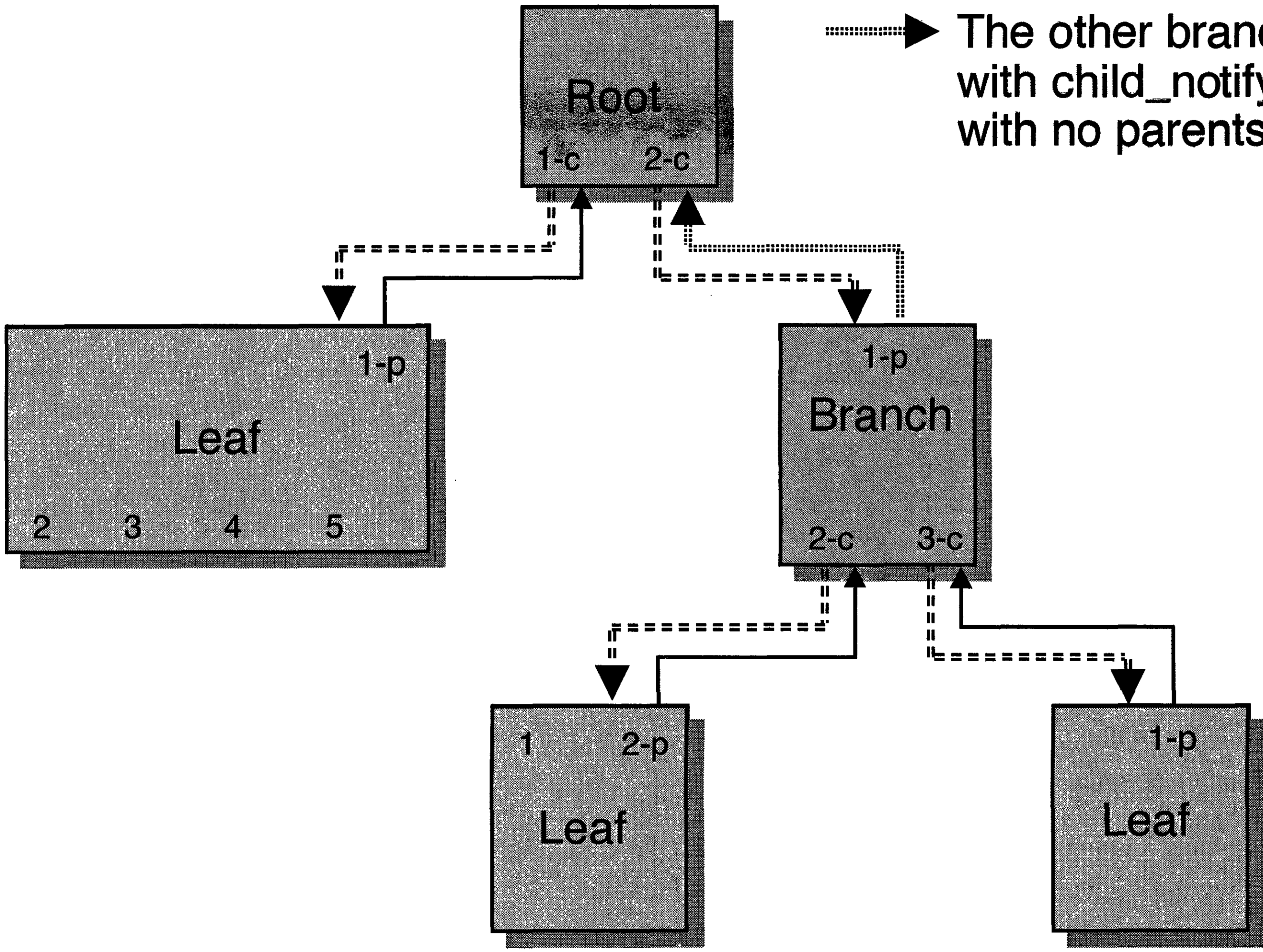
SO THEY WAIT EACH A RANDOM AMOUNT OF TIME (ONE SHORT, ONE LONG)

Tree Identify: Branch/Root Identification



both branches withdraw their parent notify signal and after a random time, one will reassert the signal.

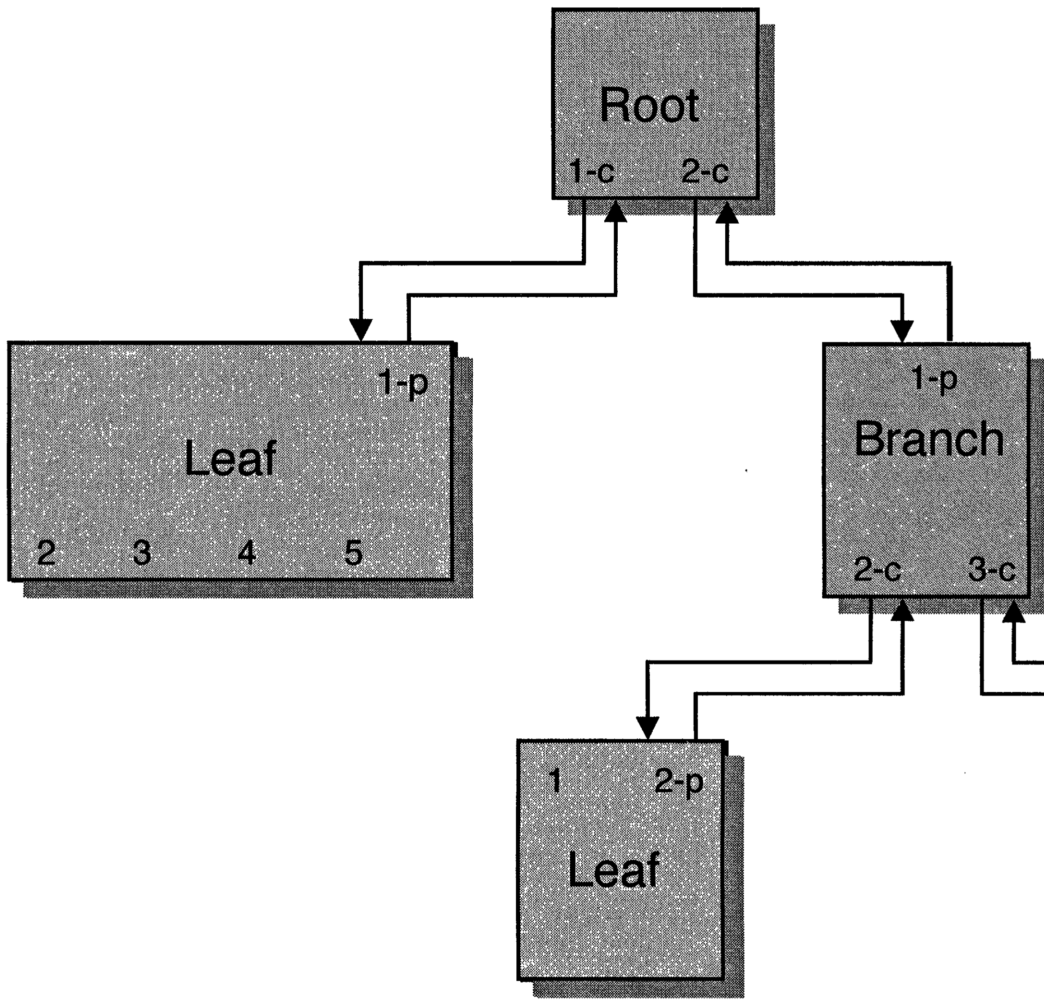
Tree Identify: Branch/Root Identification



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 ?

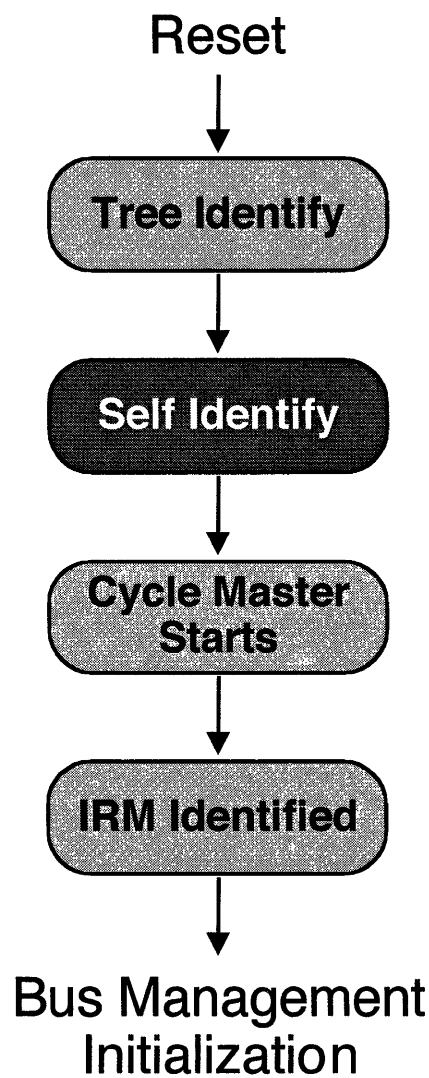
Tree Identify: Complete



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.

ROOT KNOWS IT IS ROOT

Self Identify



Numbers the nodes
Communicates speed information

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

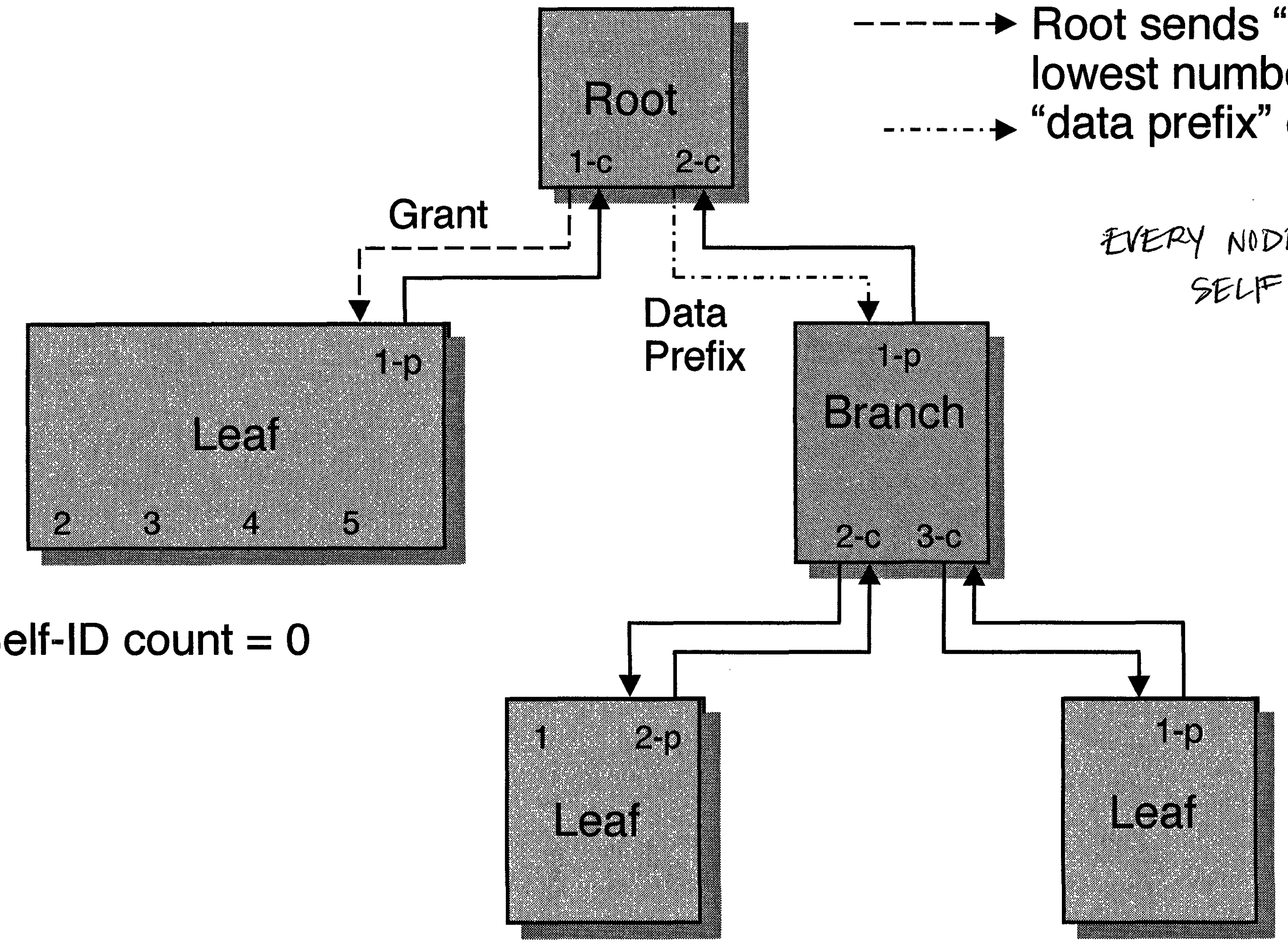
Root

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

Self Identify Signaling

<table>
<tr><th>Signal Transmitted</th><th>Arb A Tx</th><th>Arb B Tx</th><th>Comment</th></tr>
<tr><td>Idle</td><td>Z</td><td>Z</td><td>Sent to indicate a gap</td></tr>
<tr><td>Request</td><td rowspan="2">Z</td><td rowspan="2">0</td><td>Sent to parent to request the bus</td></tr>
<tr><td>Grant</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 rowspan="2">1</td><td rowspan="2">Z</td><td>Sent to child to Ack parent notify</td></tr>
<tr><td>Ident Done</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>
</table>

Self-ID: Root Issues Grant To One Port



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

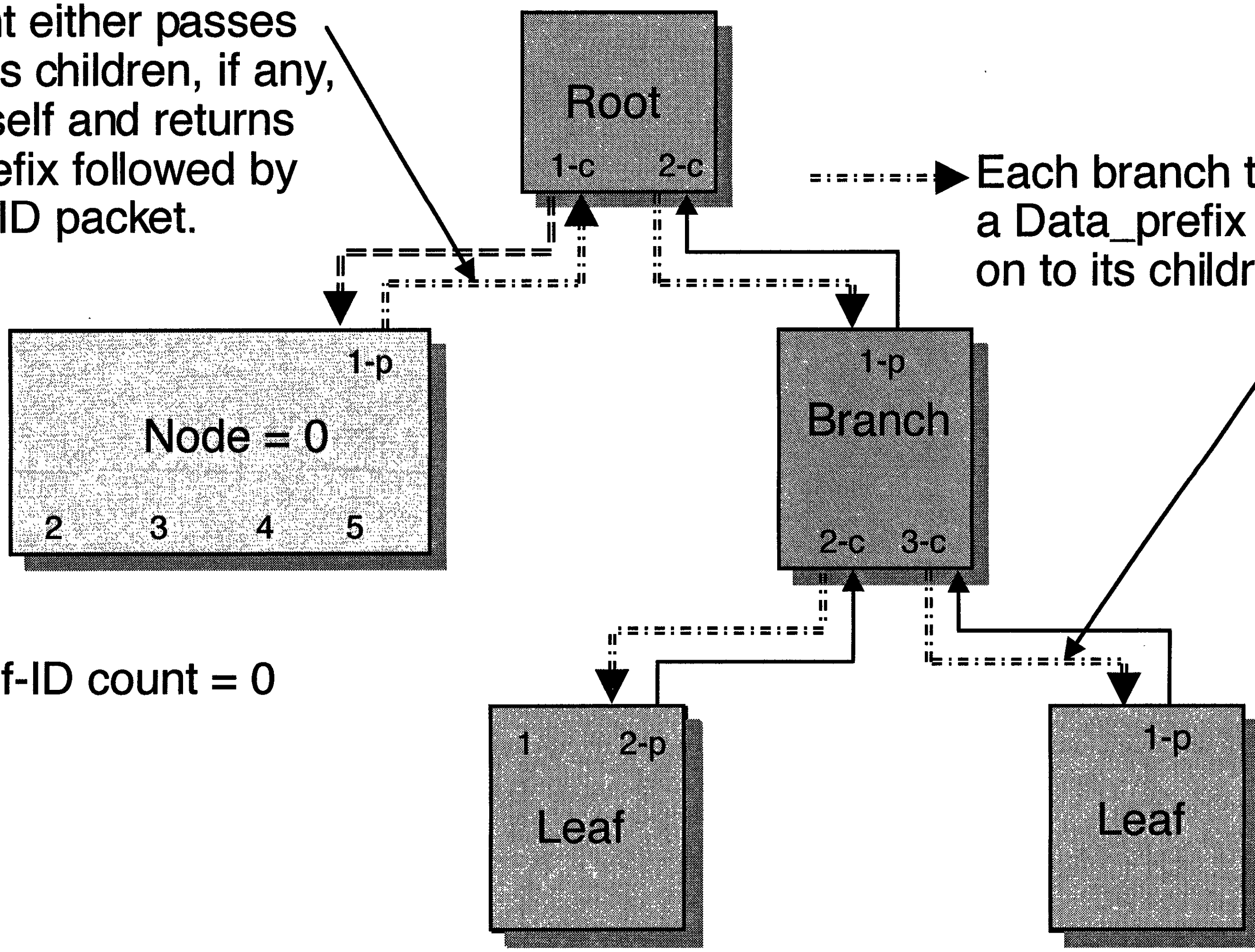
EVERY NODE HAS SELF ID COUNTER

Self-ID count = 0

Self-ID: Node 0 Sends Self-ID Packet

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

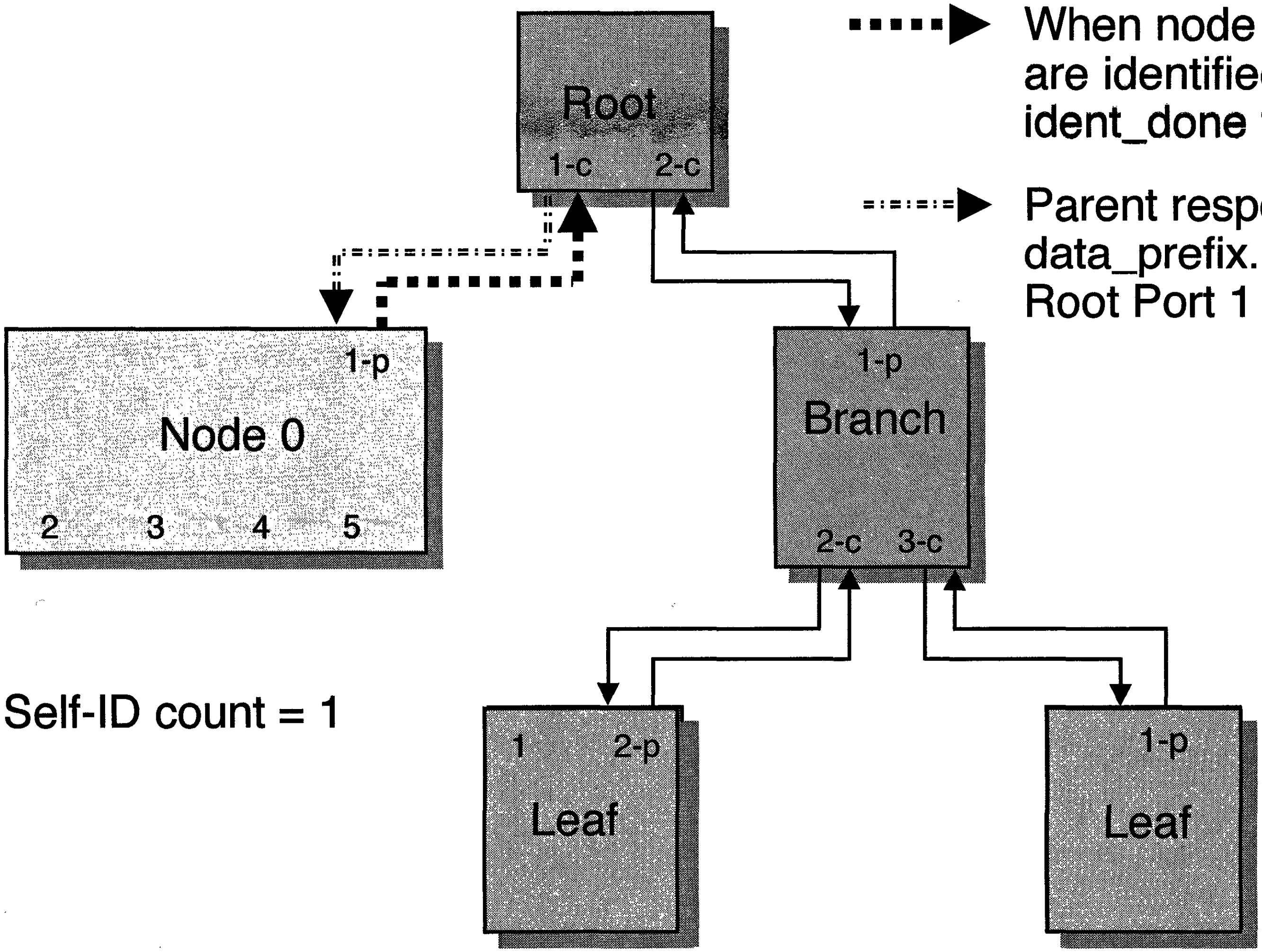
Counts each new Self-ID observed

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

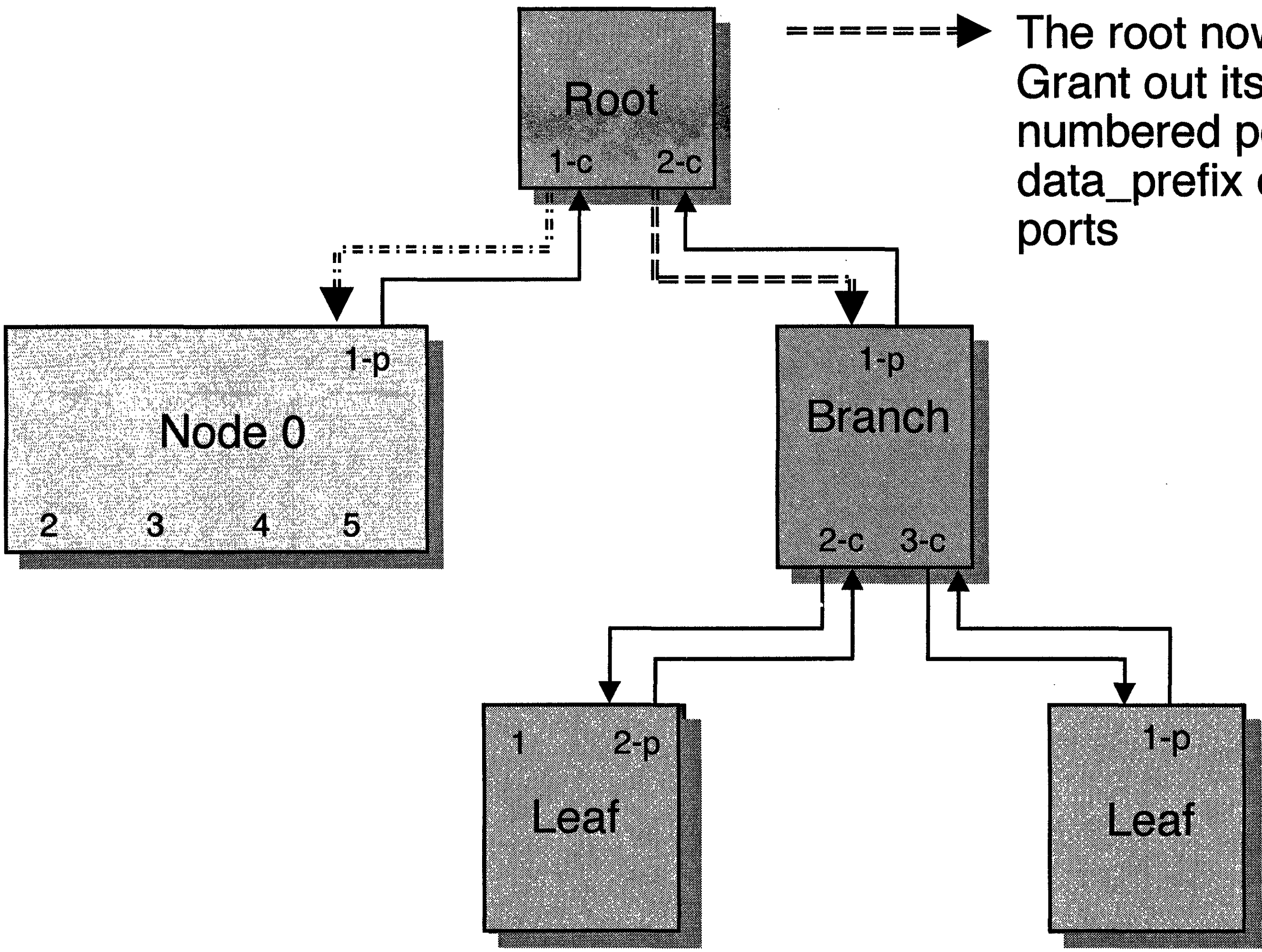


When node and all its children are identified, it sends ident_done to its parent.

Parent responds with data_prefix. In this example Root Port 1 is completed.

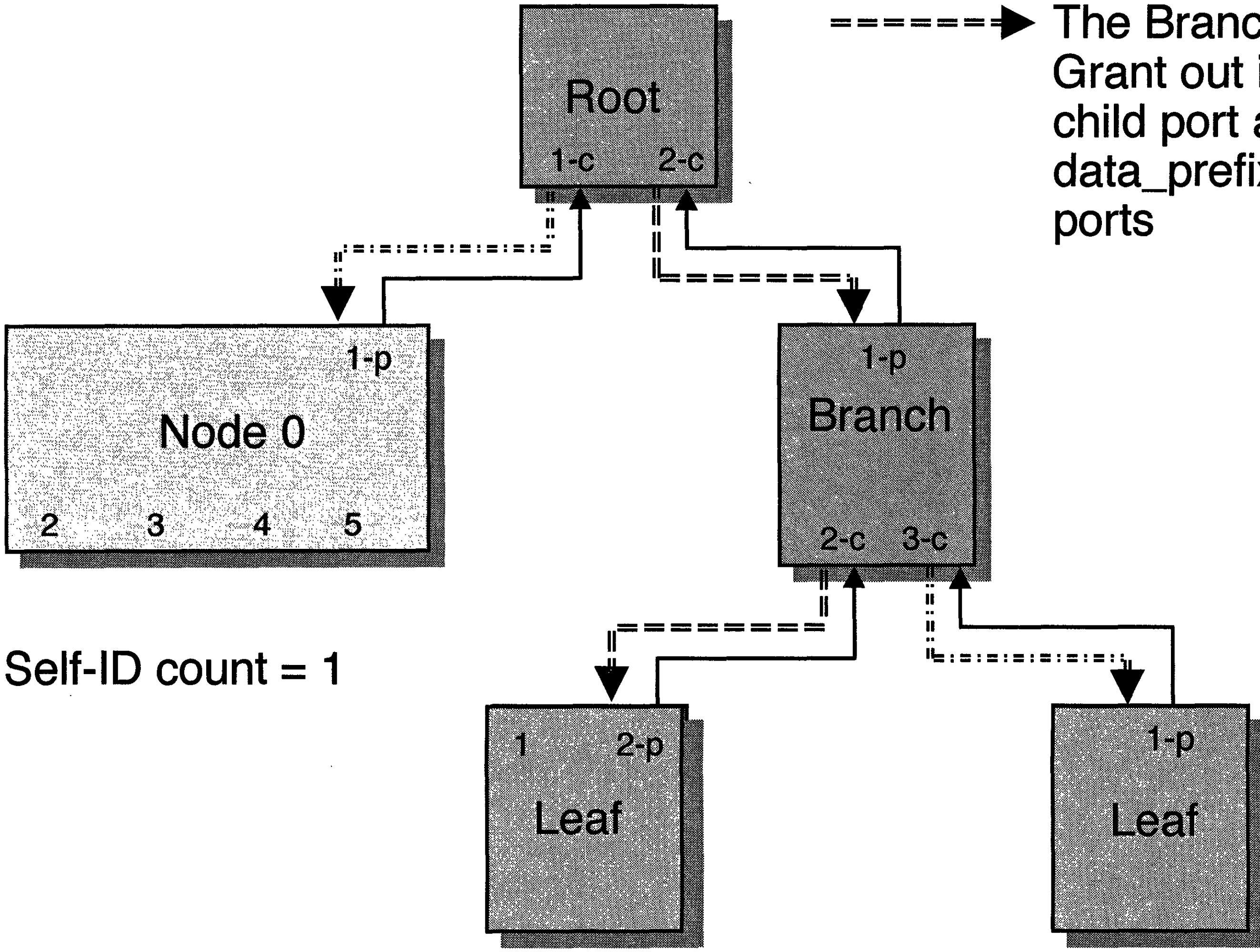
Self-ID count = 1

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

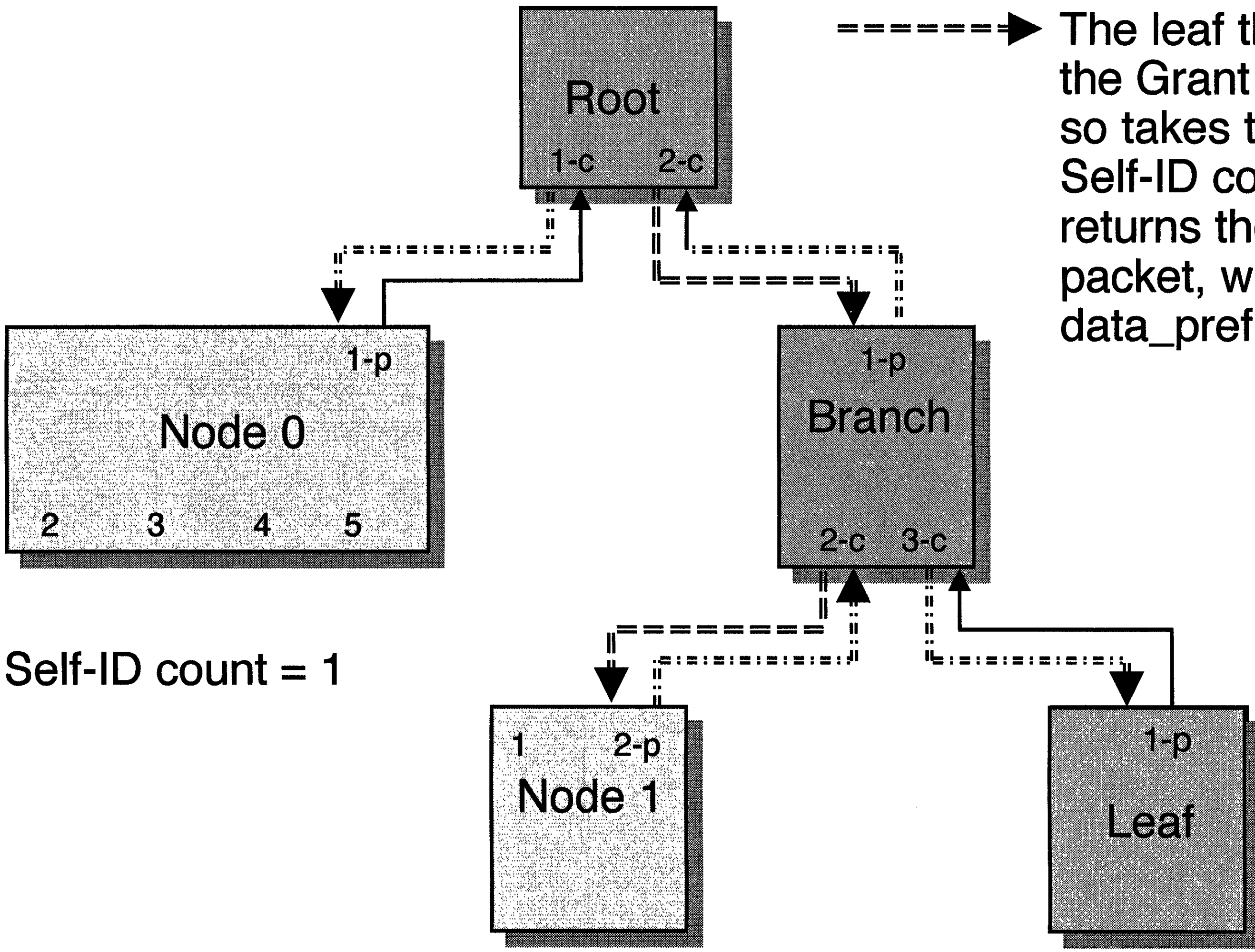
Self-ID: Branch Sends Grant To Lowest Port



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

Self-ID count = 1

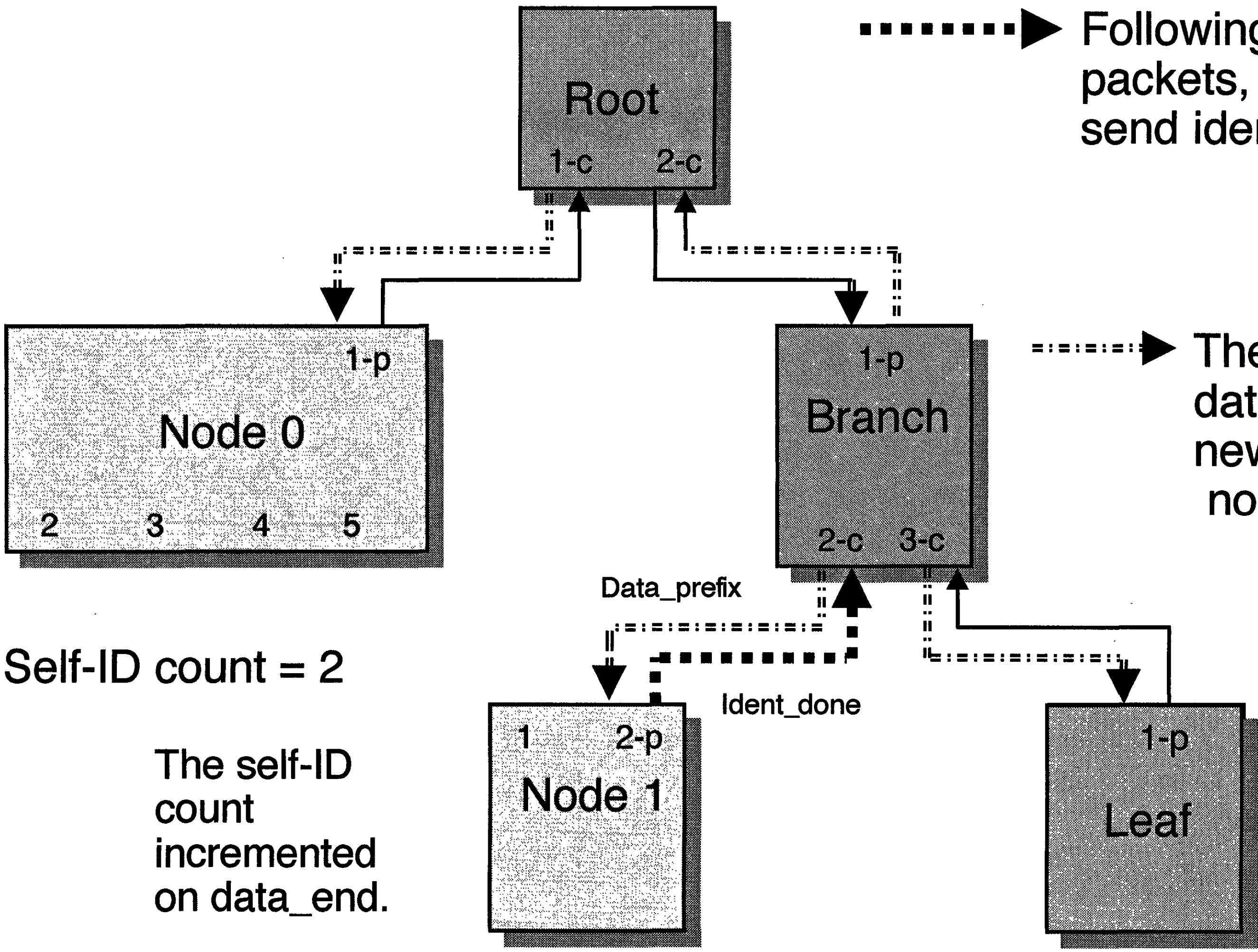
Self-ID: Node 1 Sends Self-ID Packet



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



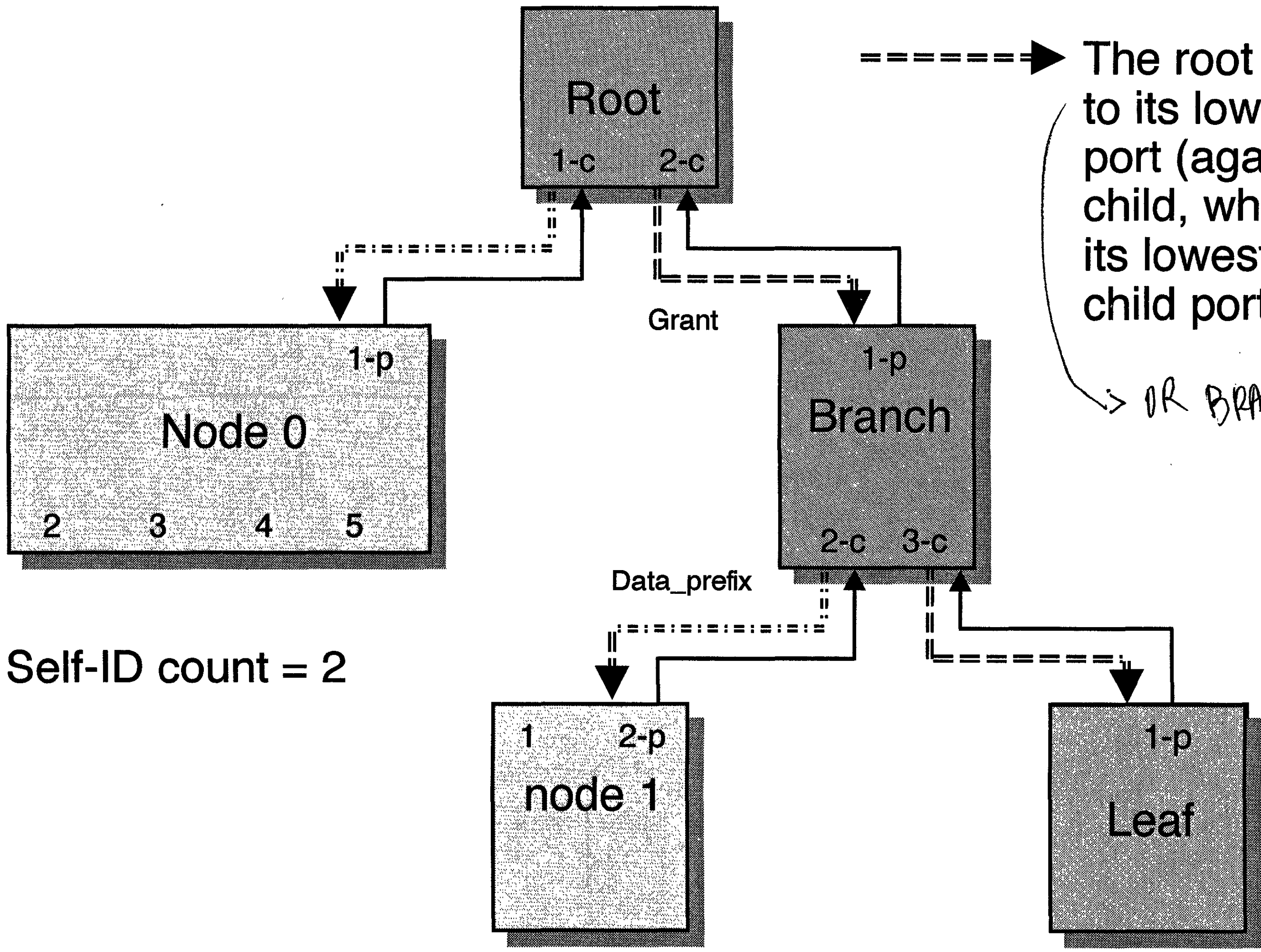
Following the self-ID packets, the node will send ident_done.

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

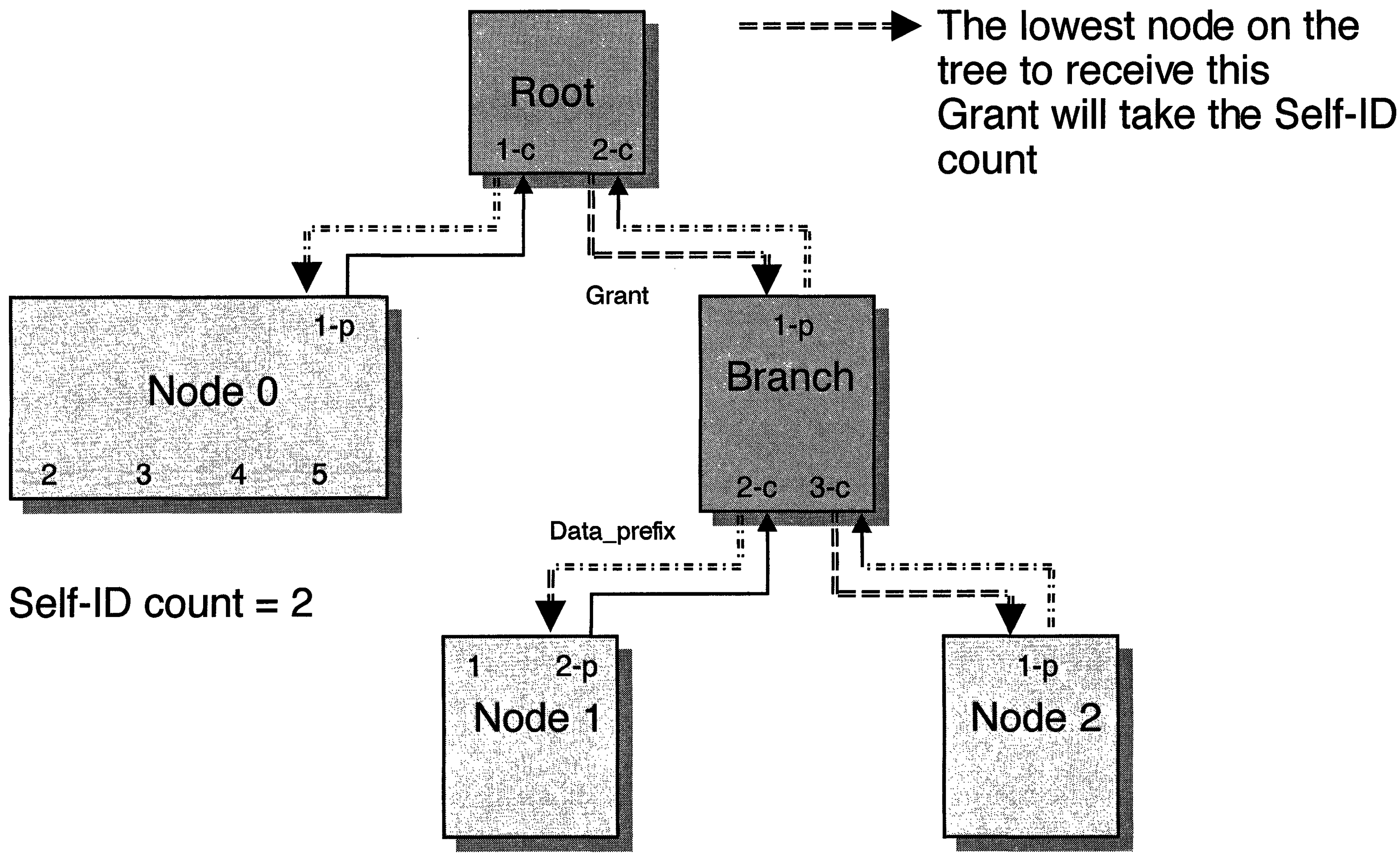


The root issues a Grant to its lowest, unidentified port (again port 2) to its child, who passes it to its lowest unidentified child port.

OR BRANCH @ this point

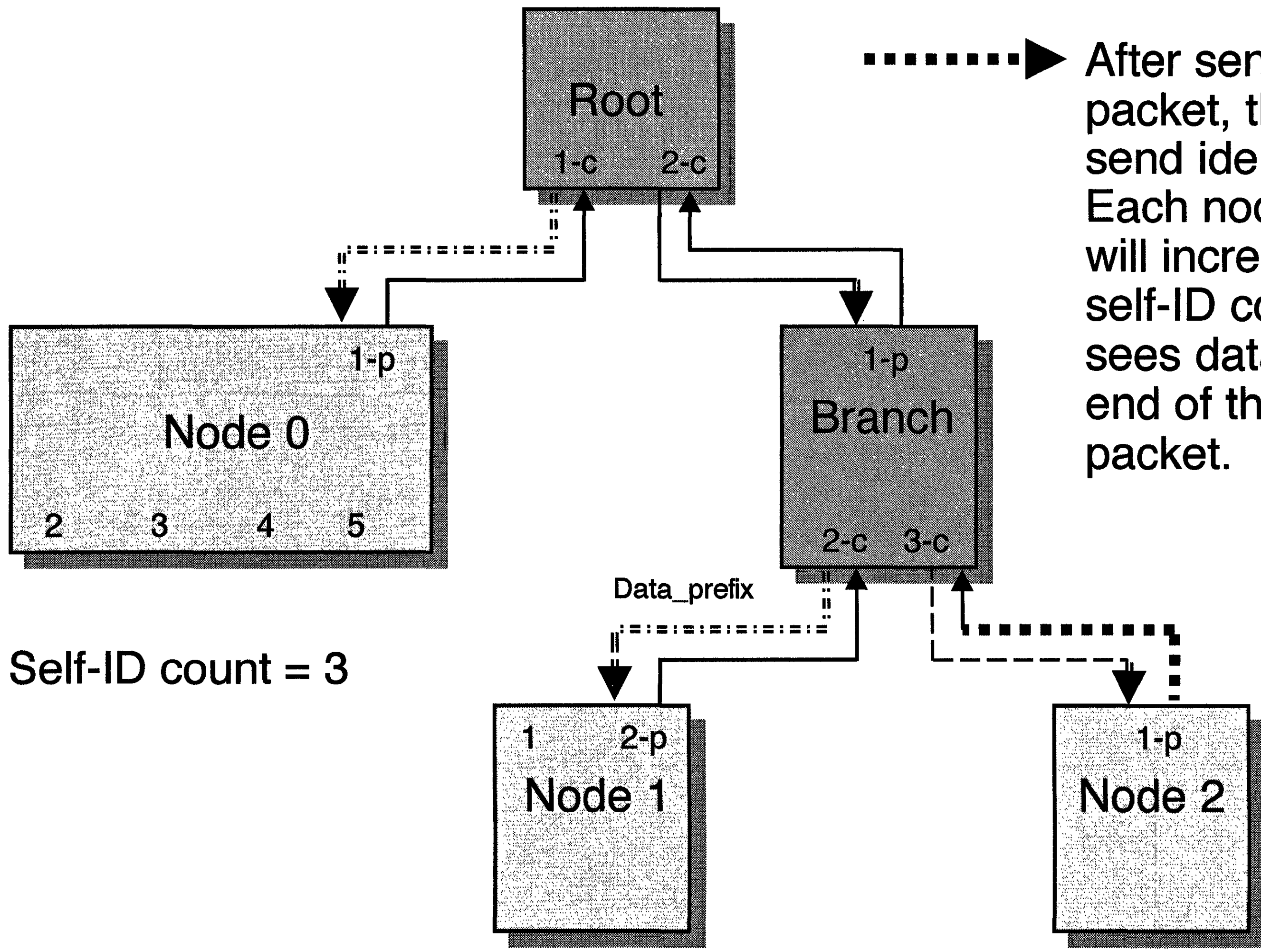
Self-ID count = 2

Self-ID: Node 2 Sends Self-ID Packet



Self-ID count = 2

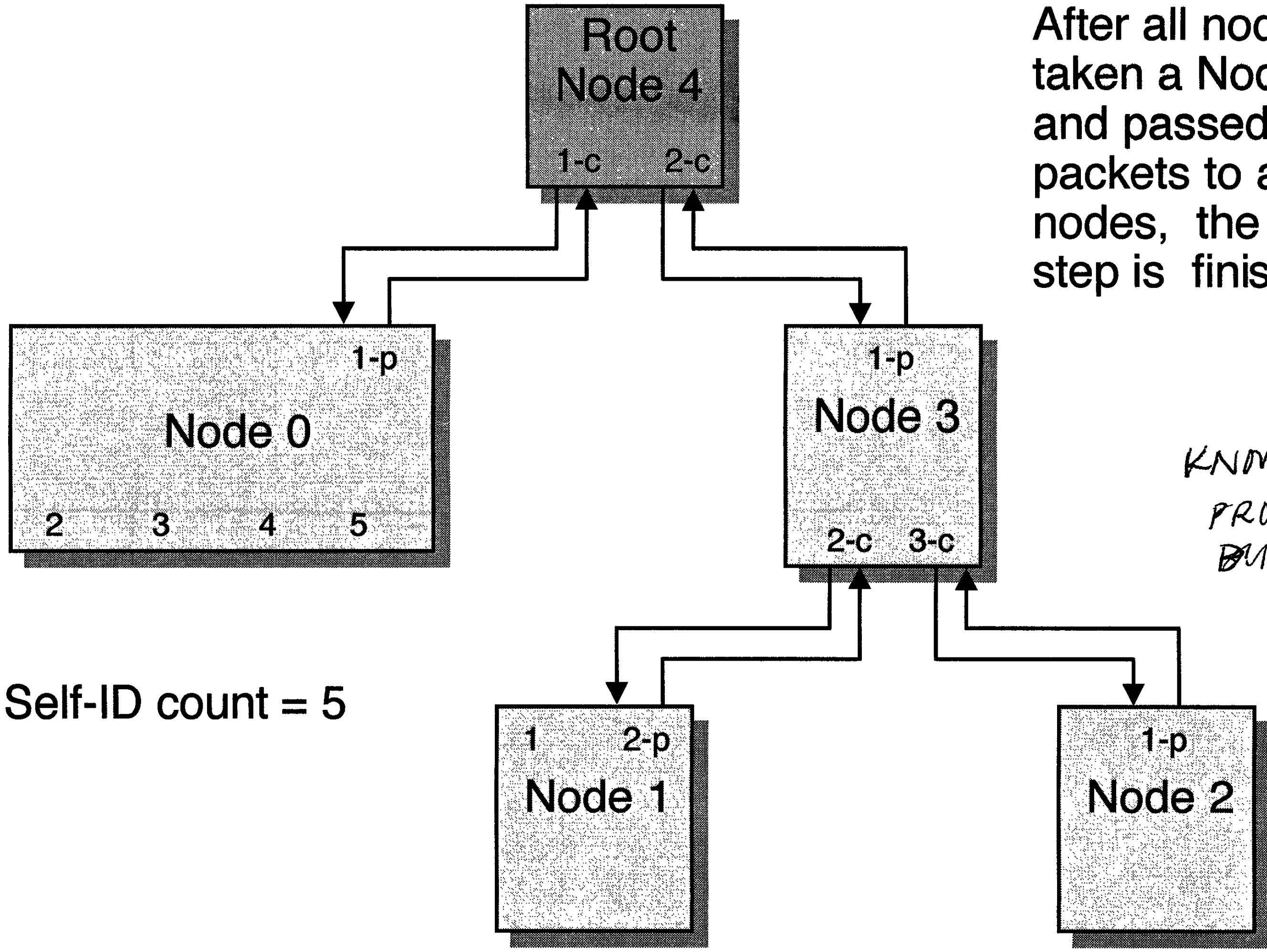
Self-ID: Node 2 Sends Ident_Done



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 count = 3

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.

KNOW WHEN SELF-ID PROCESS DONE WHEN BUS GOES FREE

SELF IDs PACKETS ARE PHY PACKETS

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

HIGHEST NODE OR SELF-ID COUNT -1

This page is intentionally blank

Self-ID Packet Format

First Packet of Self-ID :

MOST PHYs have 3 PORTS

transmitted first

SID	phy_ID	0	L	gap_cnt	sp	* Rsv	c	pwr	p0	p1	p2	i	m
logical inverse of first quadlet													

transmitted last

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

transmitted first

SID	phy_ID	1	n = 0	Rsv	p3	p4	p5	p6	p7	p8	p9	p10	r	m
logical inverse of first quadlet														

transmitted last

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

transmitted first

SID	phy_ID	1	n = 1	Rsv	p11	p12	p13	p14	p15	Reserved
logical inverse of first quadlet										

transmitted last

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
c	Contender for Bus Manager or IRM	
i	Initiated Reset (it's my fault)	
m	More Self-ID Packets (IF MORE PORTS)	

1394-1995 28 ports
1394a - 16 ports n

V

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:

Power Code	Node Power	Power Supplied	Phy Layer Power	Link Layer Power
000	None	None	None	None
001	Self	15W		
010	Self	30W		
011	Self	45W		
100	Bus/Self		3W	None
101	Reserved			
110	Bus		3W	3W
111	Bus		3W	7W

Cycle Master Starts

Reset

↓

Tree Identify

↓

Self Identify

↓

Cycle Master Starts

Generates Cycle Start Packets
Prior Isochronous Transfers resume

↓

IRM Identified

HIGHEST NODE # CAPABLE OF IRM

↓

Bus Management Initialization

Cycle Master Starts

After Self-Identify previous Cycle Master checks to see if it is Root

- If it is not - Turns off

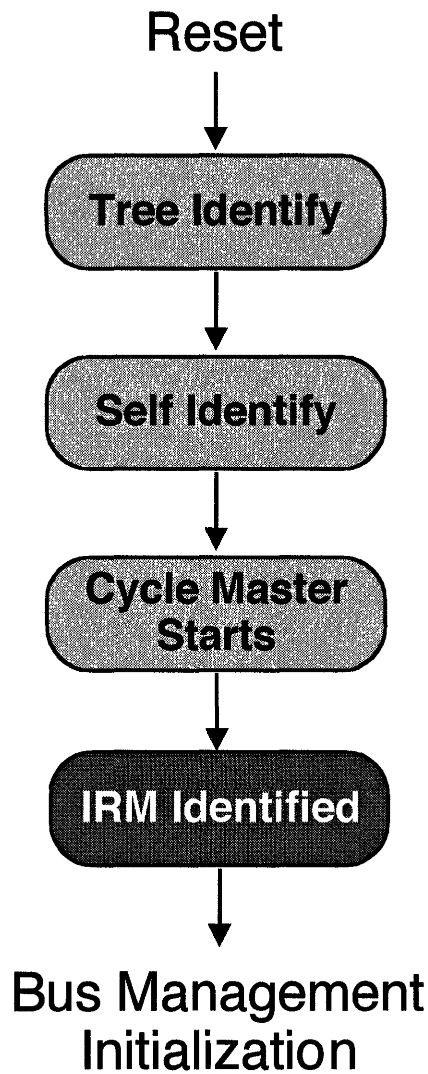
Root turns on Cycle Master

- 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



Identifies which Node contains Isochronous Resource Manager

Isochronous Resource Manager (IRM) Identified

IRM is the highest number node with IRM Capabilities

IRM and other interested nodes monitor Self-ID Packets:

SID	phy_ID	0	L	gap_cnt	sp	del	c	pwr	p0	p1	p2	i	m
logical inverse of first quadlet													

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 EXCEPT FOR ROOT & CYCLE-MASTER IN HW

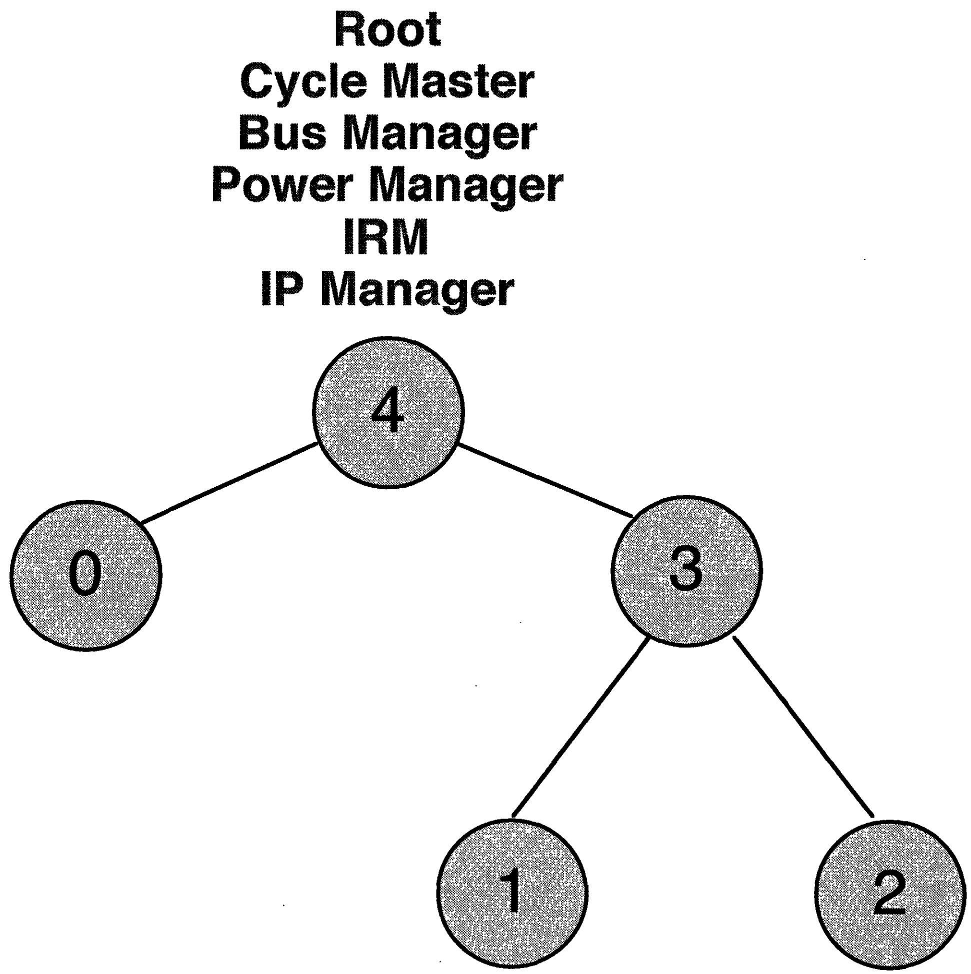
Six standardized managers:

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

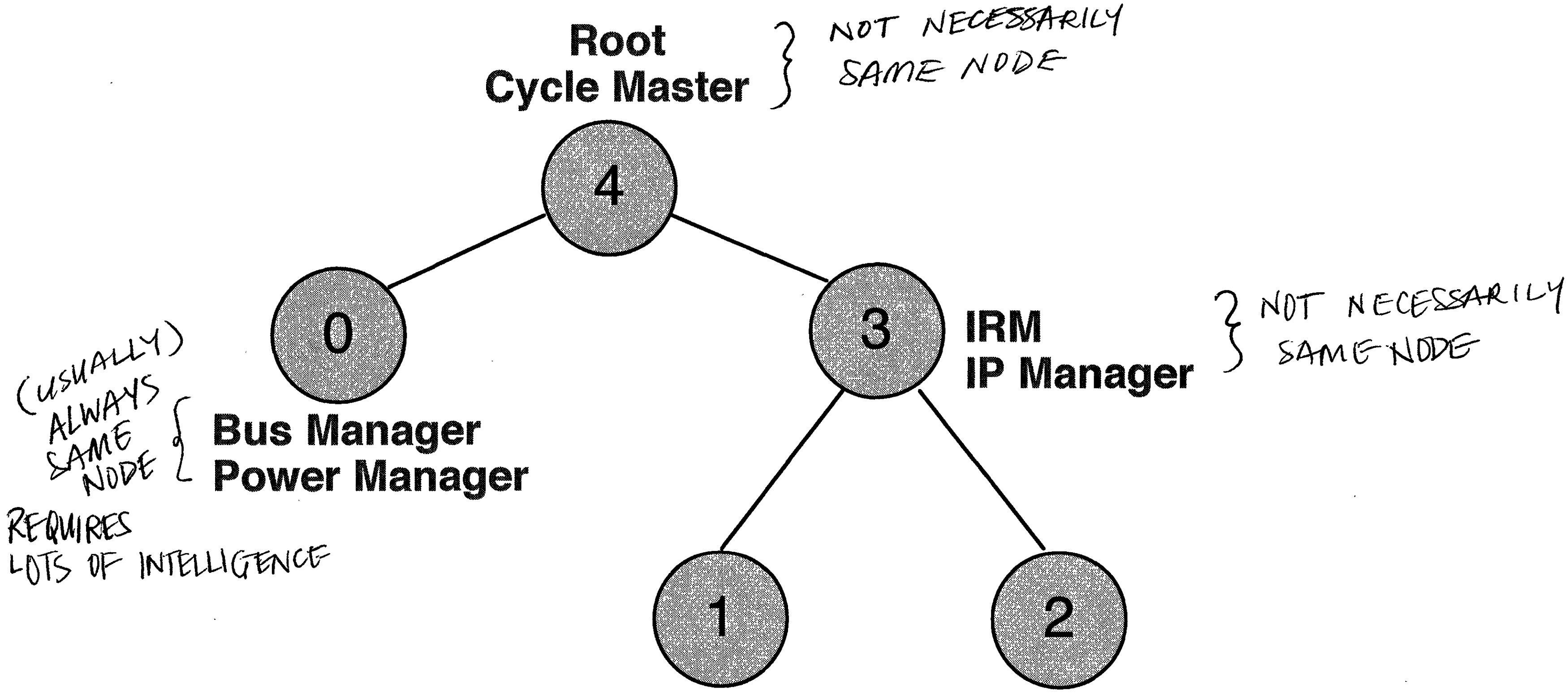
Each of these management functions implemented on a node

- (Not necessarily the same node)

Possible Bus Configuration



Another Possible Bus Configuration



Node Capabilities

Logical
Functions
(Software)

HIERARCHY

Hardware
Functions

Bus Manager Capable
↓
IRM Capable
↓
Cycle Master Capable
↓
Isochronous Capable
↓
Transaction Capable
↓
Repeater

Each level must be capable of all lower level functions

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:
  - $\underbrace{\text{State\_Clear, State\_Set, Node\_IDs, Reset\_Start,}}_{\text{1394 REQUIRED}}$ & Split_Timeout

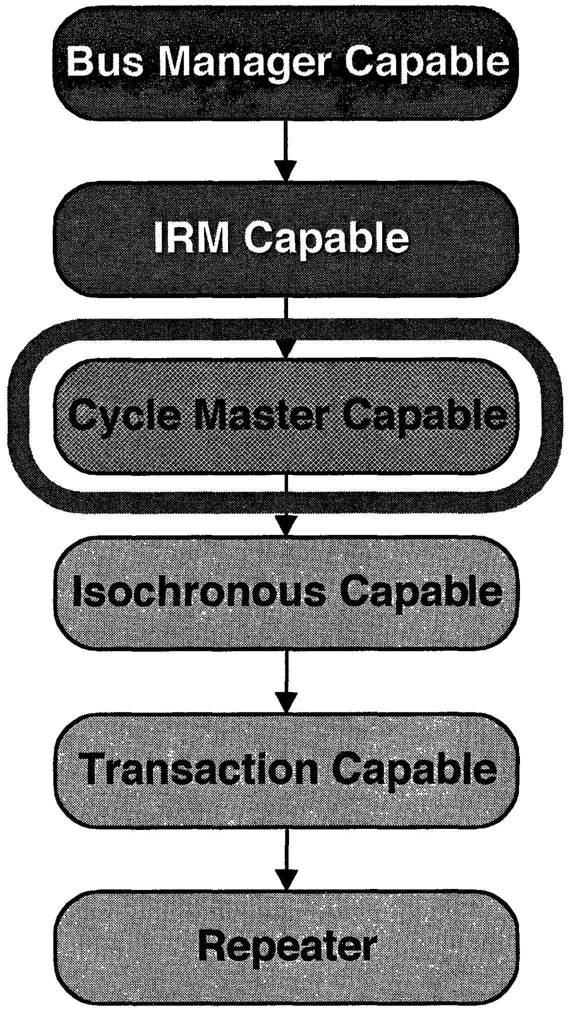
REQUIRED IN SBP-2

Isochronous

- Must implement Cycle_Time Register
- Free RMnning 24.576 MHz clock
- Configuration ROM in General ROM format

HIGH COST ADDITION
(NEED OSCILLATOR & CRYSTAL)

Cycle Master Capable



Cycle Master - NO ADDITIONAL COST REALLY

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

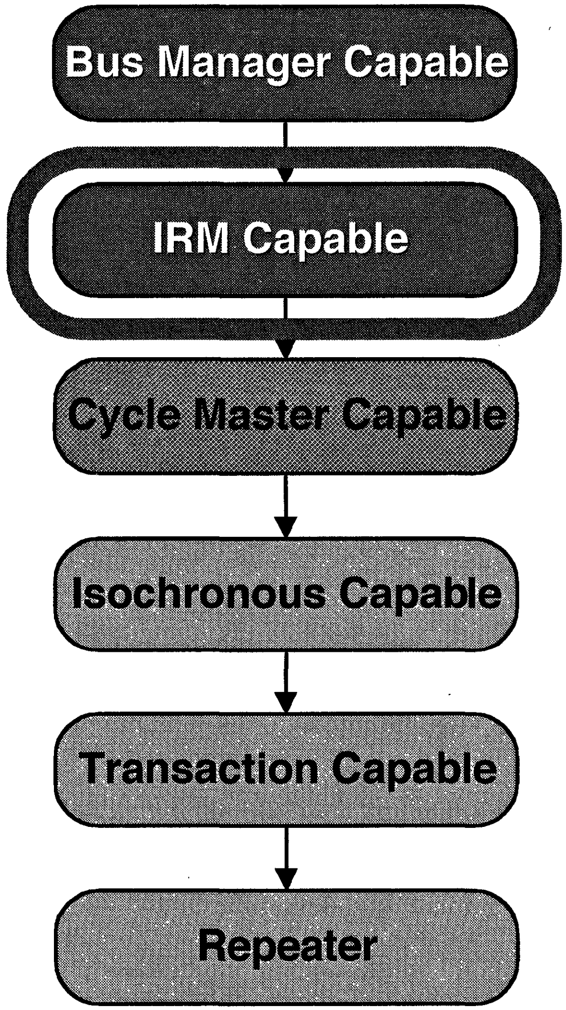
Additional Logical Function Responsibilities

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



Isochronous Resource Manager

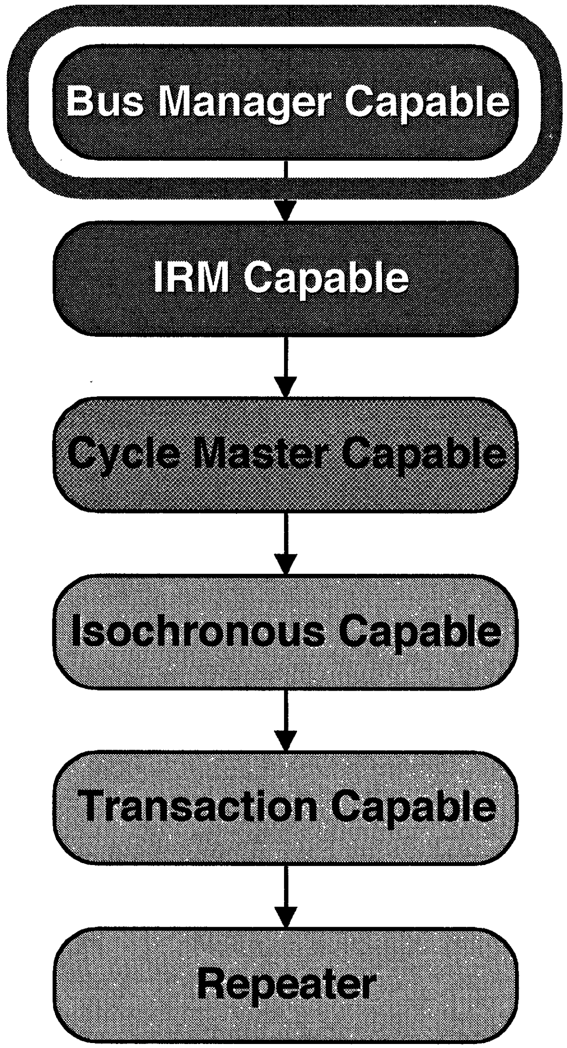
Additional Hardware Responsibilities

- Implement Bus_Manager_ID Register
- Implement Bandwidth_Available register
- Implement Channels_Available register(s) 2 OF THEM

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



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 83 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

all bus manager capable nodes will save all self id packets since don't know which is bus manager yet during self-id process

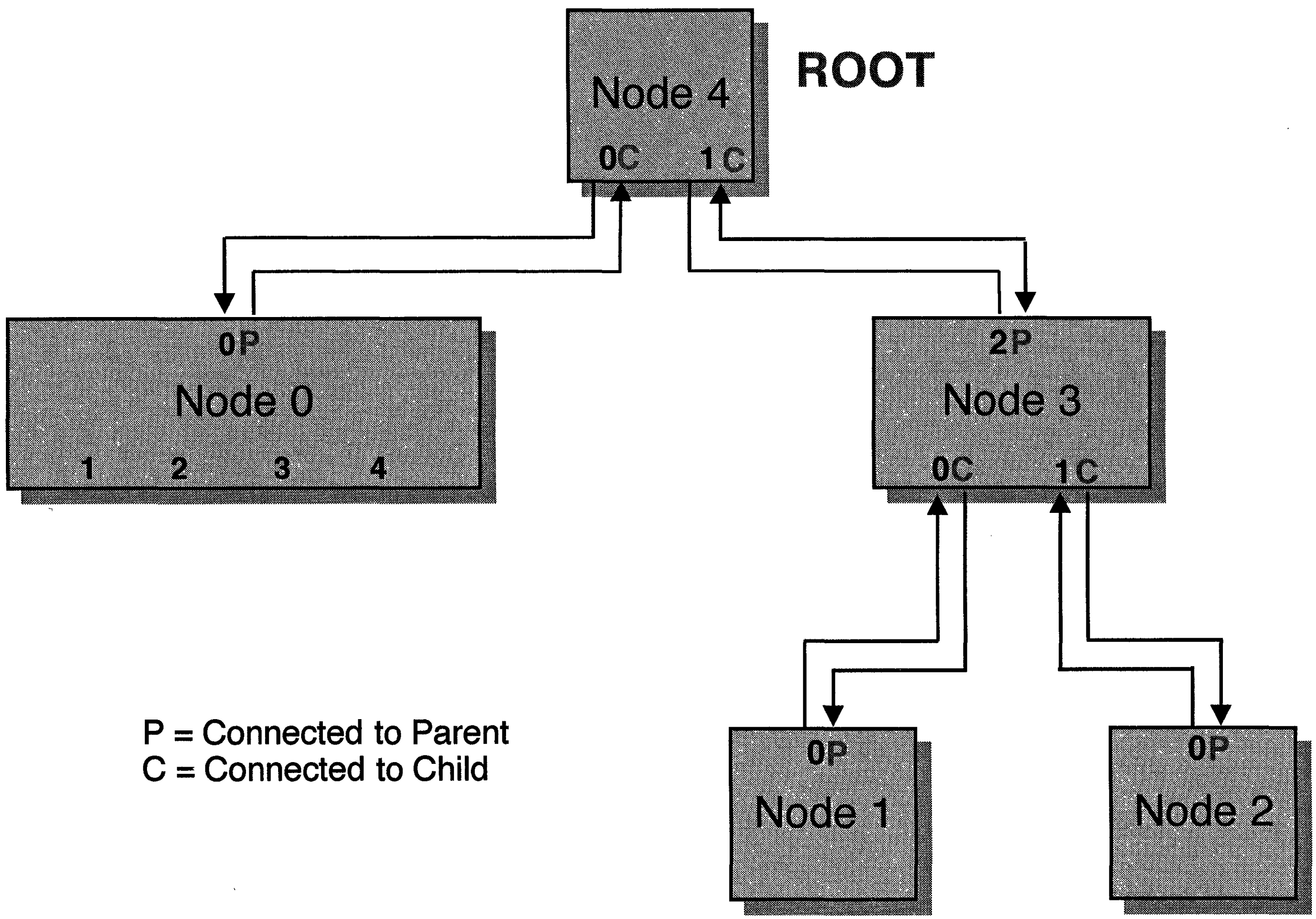
transmitted first

SID	phy_ID	0	L	gap_cnt	sp	res	c	pwr	p0	p1	p2	i	m
logical inverse of first quadlet													

transmitted last

Port Status	Meaning
00	Unimplement 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

Topology Map for Example Topology

WHO IT'S CONNECTED TO

phy_ID	Port 0	Port 1	Port 2	Port 3	Port 4	Port 5:10
00	parent	uncon	uncon	uncon	uncon	no port ...
01	uncon	parent	no port			
02	parent	no port	no port			
03	child	child	parent			
04	child	child	no port			

LEAVES ONLY 1 PORT CONNECTED

ROOT

WORKING BACKWARDS FROM ROOT GO OUT NEXT HIGHEST PORT # (1)

Reconstructing Topology from the Topology Map

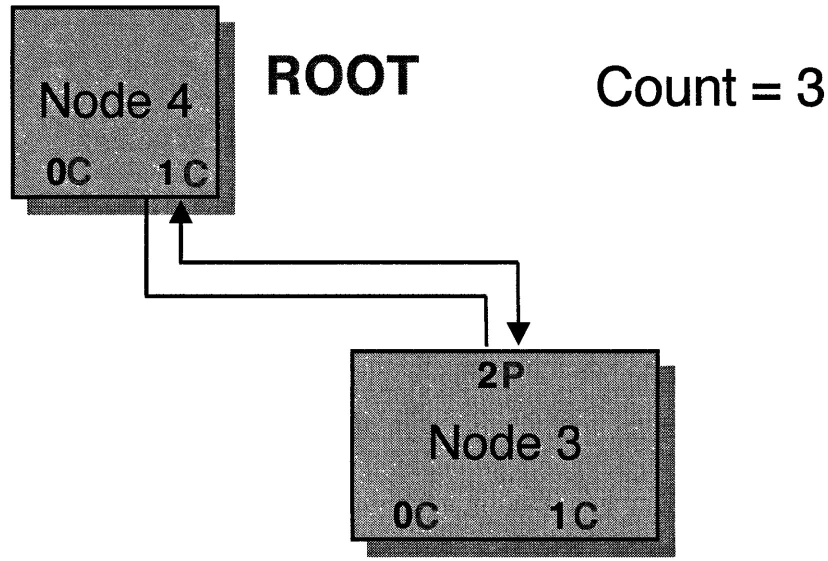
Bottom Up Approach

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

Top Down Approach

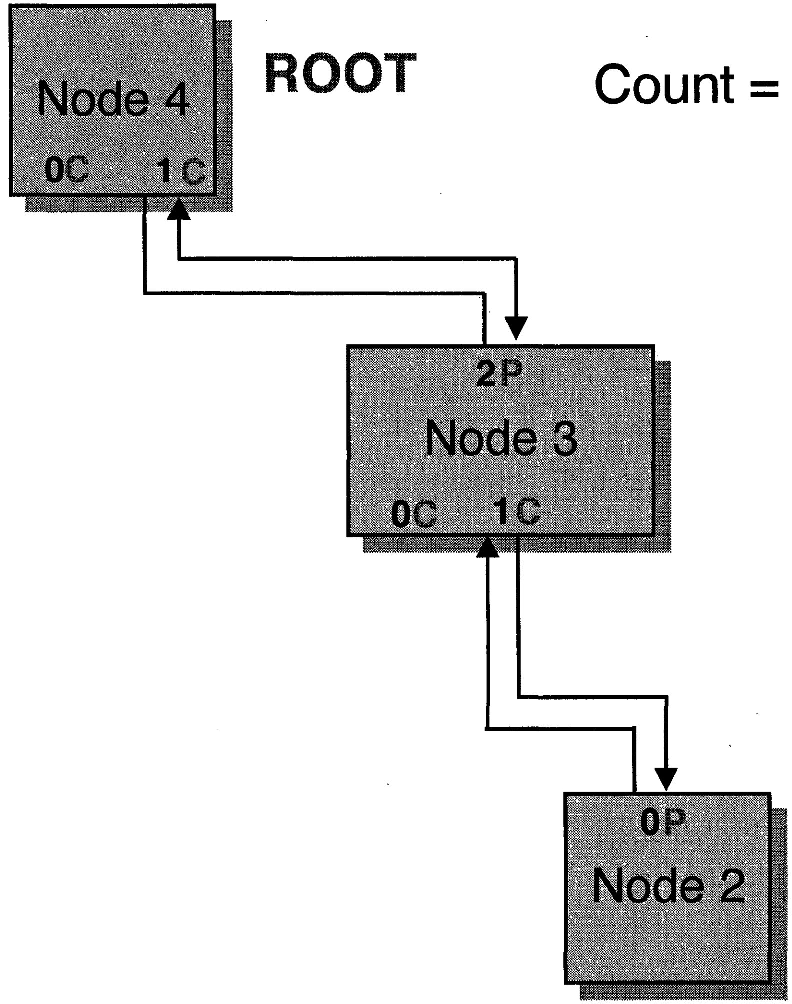
- Recursive Approach
- Start with Root: Count = Root ID
- Evaluate Node by:
  - If this Node is a Leaf, evaluation of this node complete
  - Test each connected Port on this node from highest to lowest
    - Count = Count -1
    - This port connected to Node ID Count
    - Evaluate that Node
  - When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

Reconstructing Topology

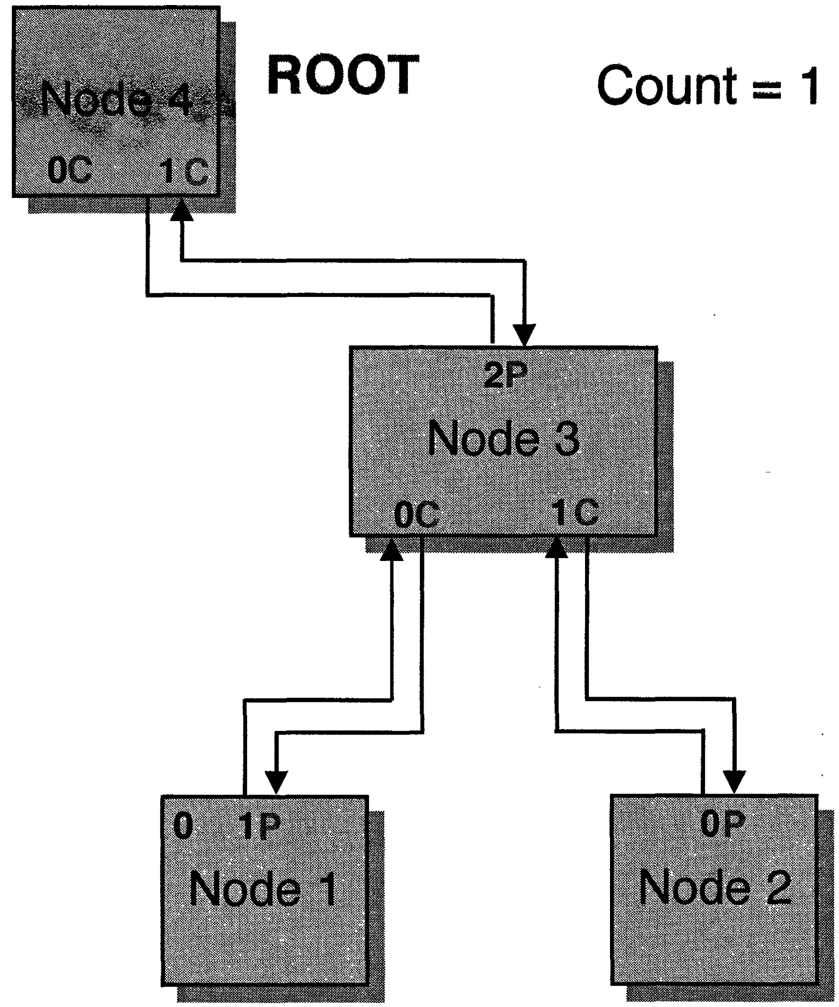


Reconstructing Topology

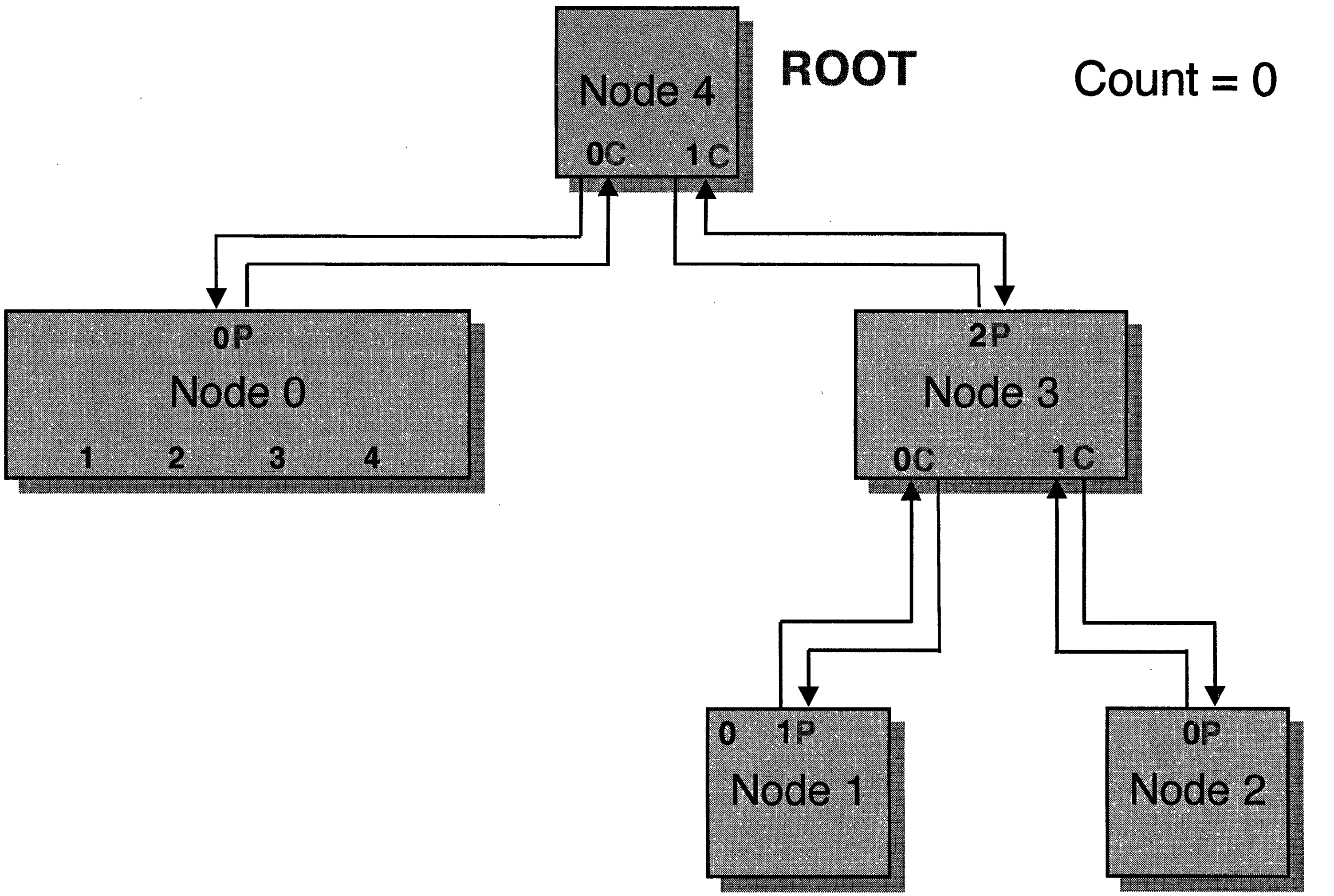
Count = 2



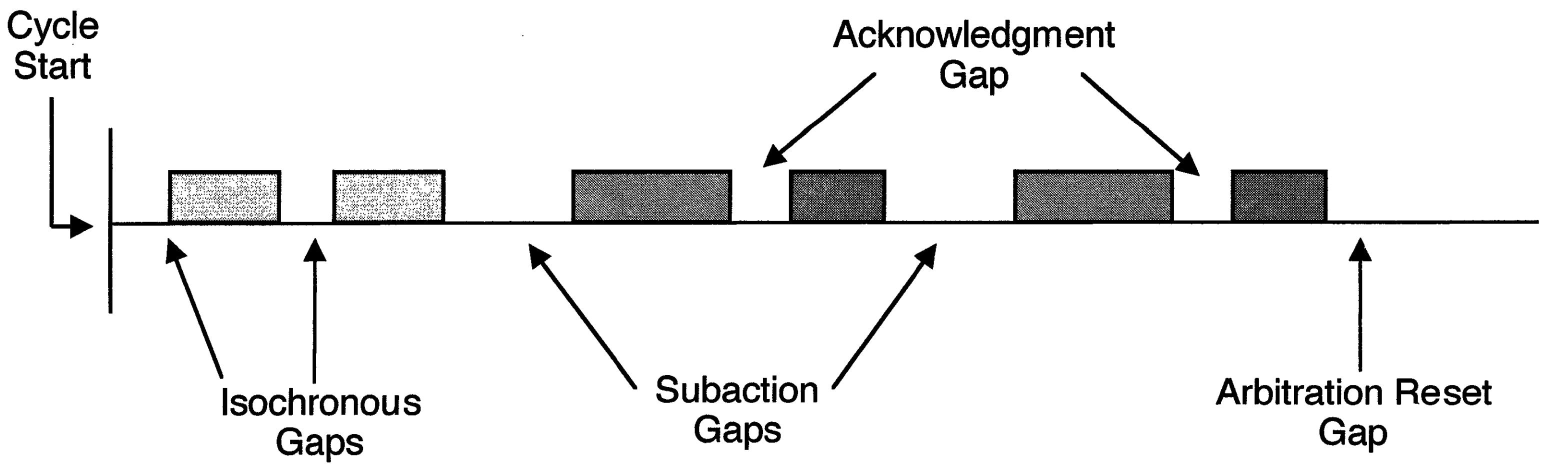
Reconstructing Topology



Reconstructing Topology



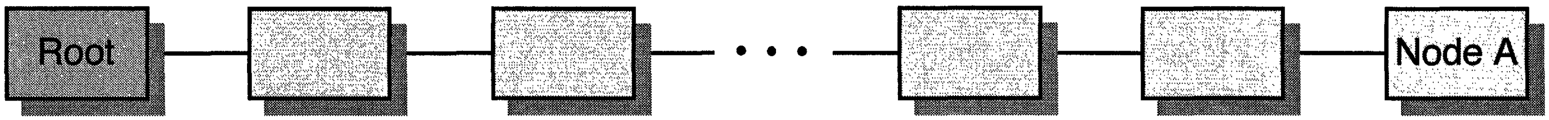
Bus Manager - Gap Timing



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)

NODE TO NODE

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

.167 µS
× 16 hops
2.7 µS
×2 → roundtrip
5.4 µSec

need to add time to prevent race condition

5.4
+ 1.5
6.9 µsec before any one can arbitrate for bus

1394

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

Gap Type	Detection Time	
	Minimum Delay	Maximum Delay
ACK Isoch.	40ns	50ns
Subaction	(27 + gap_count * 16) / base rate	(29 + gap_count * 16) / base rate
Arb. Reset	(51 + gap_count * 32) / base rate	(53 + gap_count * 32) / base rate

Base rate = 98.304 MHz

Setting Gap Timing

FROM 1394-1995

1394a Gap Count	Max. Hops	1394 (1995)	Subaction Gap	Arb. Delay	Total
1	1	2	0.6002	0.0814	0.6816
4	2	4	0.9257	0.1628	1.0885
7	3	6	1.2512	0.2441	1.4959
10	4	8	1.5767	0.3255	1.9023
12	5	10	1.9023	0.4069	2.3092
15	6	12	2.2278	0.4883	2.7161
⋮	⋮				
43	16	33	5.6458	1.3428	6.9855
45	17				
⋮	⋮				
62	23				

All times in microseconds

For Your Reference:

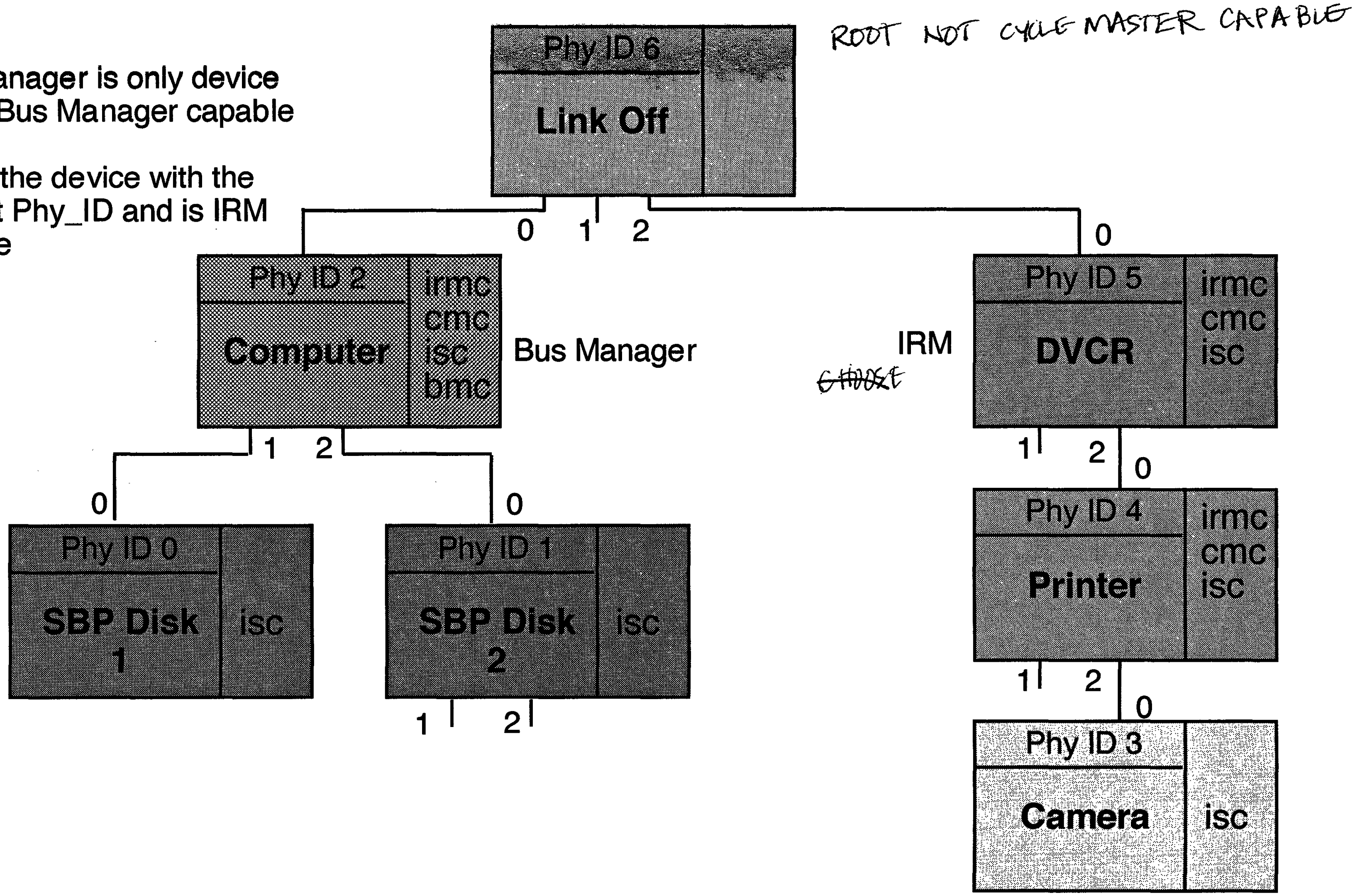
7 Node Bus Configuration Example

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 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				

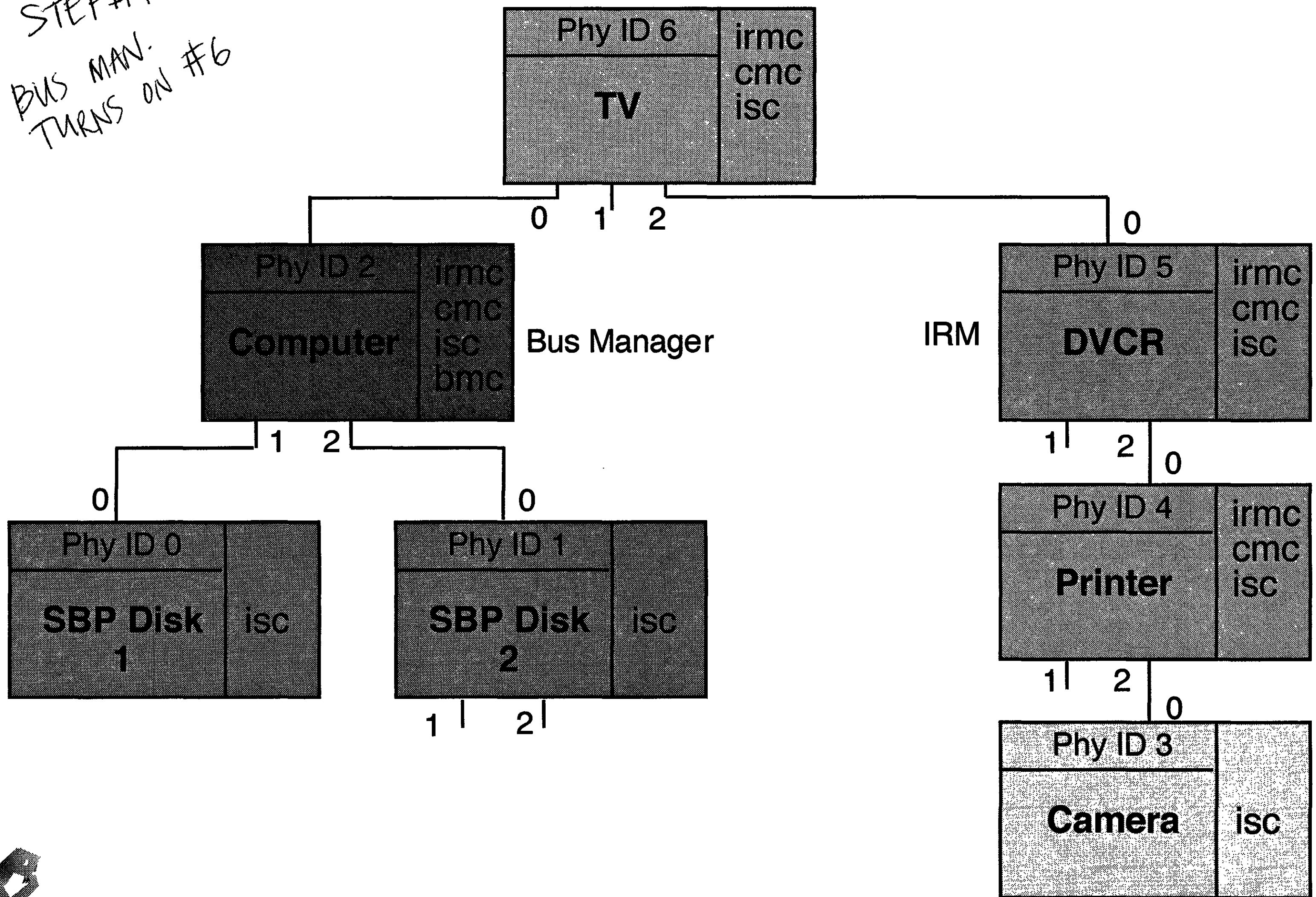
Link On

01	Dest ID	00	00	00
logical inverse of first quadlet				

Bus Configuration with Bus Manager and IRM

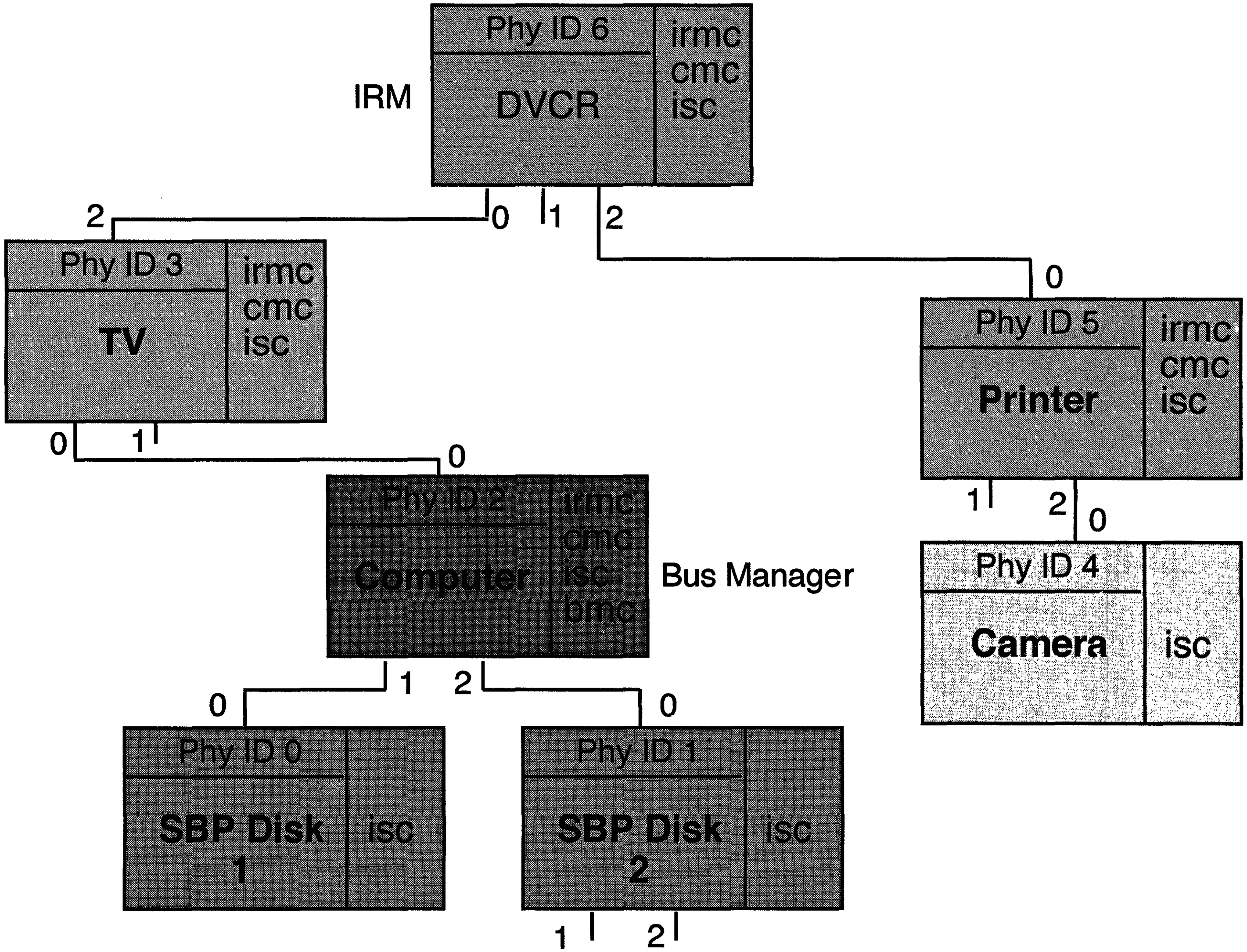
Configuration after Phy_ID 6 Link layer is powered on

STEP#1
BUS MAN.
TURNS ON #6



Bus Configuration with Bus Manager and IRM

Configuration after Second Reset



Bus Management Review

1. Name and define each management responsibility on the bus.

Bus Management Notes

Bus Management Notes

Section 13

Implementation

Subjects Covered

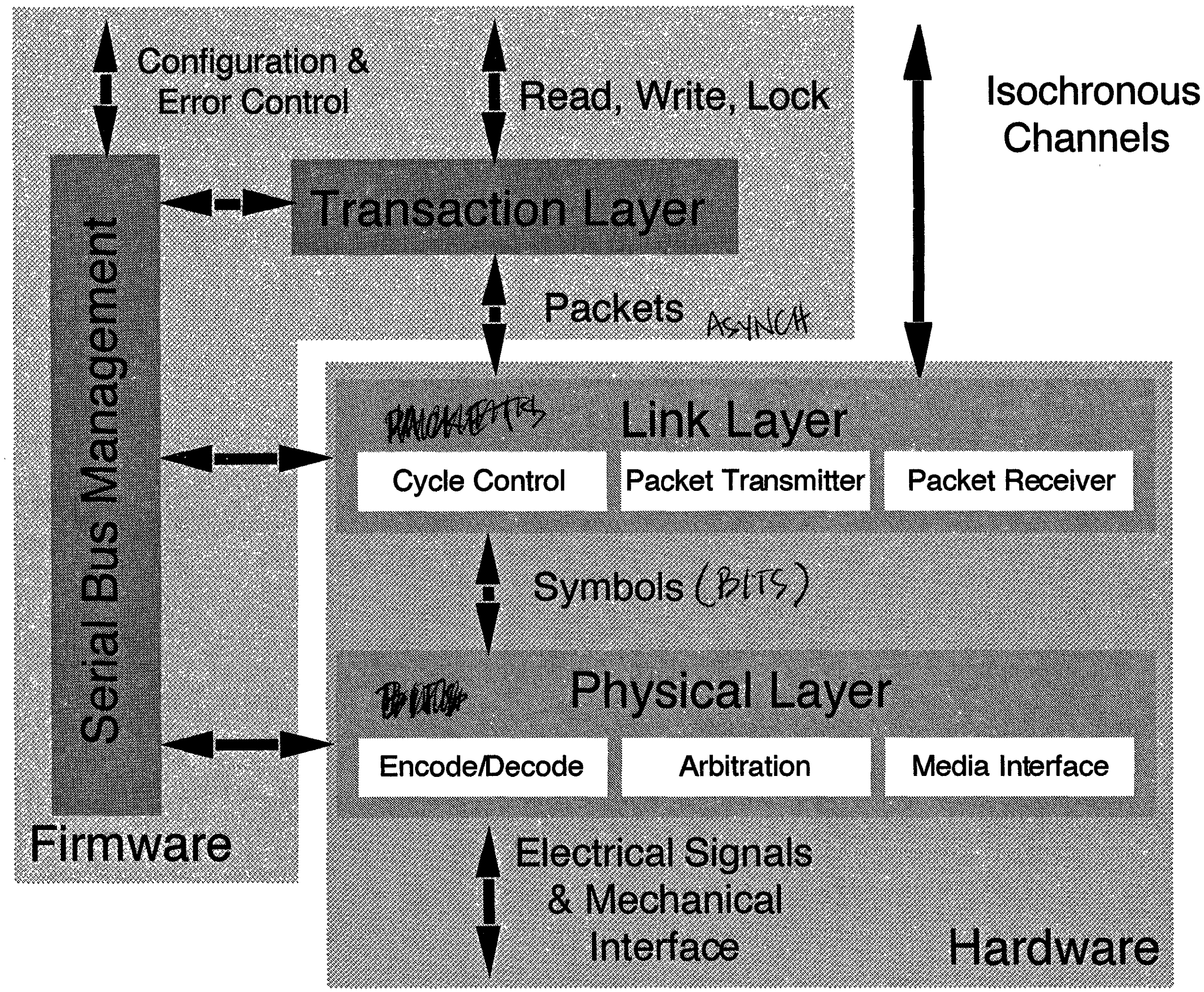
Phy duties and responsibilities

Link duties and responsibilities

Phy - Link communication

Phy registers

1394 Protocol Stack



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 IF ASYNC

Current communication between Phy & Link = 50 MHz ALWAYS

Different complexity depending on functions - SOMETIMES LINK IS INCLUDED IN ULP

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

OTHER ERRORS HANDLED ELSEWHERE

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

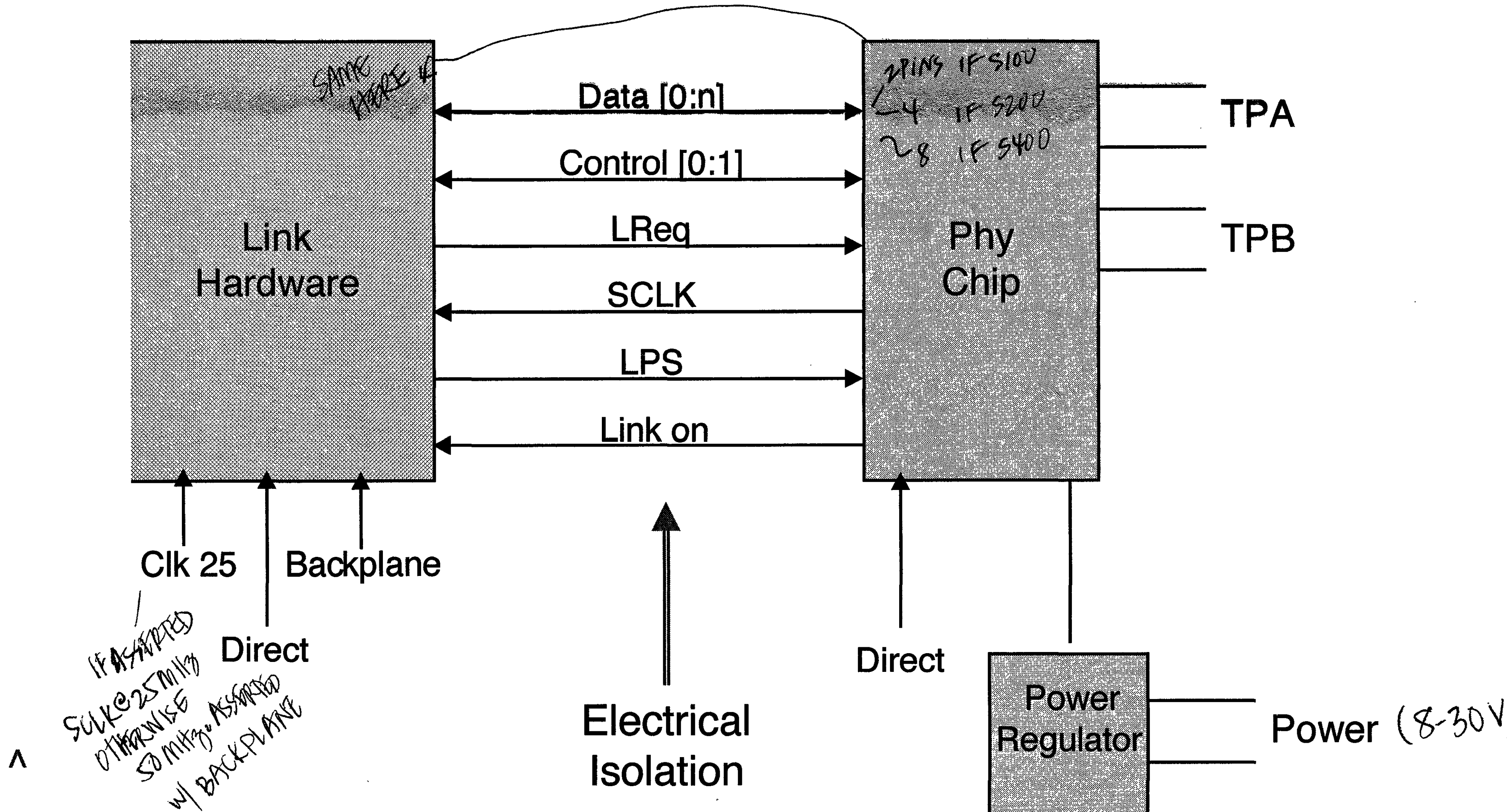
Bus Manager
- Power management
- Speed and Topology maps

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



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	

v

Control [0:1]

Phy is driving

Ctl[0:1]	Name	Comment
00b	Idle	No activity
01b	Status	The PHY is sending status information to the link 2 BITS @ A TIME
10b	Receive	An incoming packet is being transferred from PHY to link
11b	Grant	The link has granted the bus to send a packet

Link is driving

Ctl[0:1]	Name	Comment
00b	Idle	Transmission complete, release the bus
01b	Hold	The link wishes to hold the bus
10b	Transmit	The link is sending a packet to the PHY
11b	--	Unused

LREQ

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

Bits	Name	Comment
0	Start	Indicates start of transfer, always 1
1:3	Request type	Indicates which type of request is being performed
4:6	Request speed	Speed at which the PHY will be sending the packet
7	Stop	Indicates end of transfer, always 0

Bus request for backplane environment

Bits	Name	Comment
0	Start	Indicates start of transfer, always 1
1:3	Request type	Indicates which type of request is being performed
4:5	Request speed	Ignored, set to 0 in backplane environment
6:9	Request priority	Indicates priority or urgent requests (fair requests only)
10	Stop	Indicates end of transfer, always 0

LREQ - Arbitration Control Request

Bits	Name	Comment
0	Start	Indicates start of transfer, always 1
1:3	Request type	Indicates which type of request is being performed
4	Accelerate	0 = Phy may not use accelerated arbitration, 1 = Phy may
5	Stop bit	Indicates end of transfer, always 0

LREQ - Register Requests

Register read request

Bits	Name	Comment
0	Start	Indicates start of transfer, always 1
1:3	Request type	Indicates which type of request is being performed
4:7	Address	Internal PHY address to be read
8	Stop	Indicates end of transfer, always 0

Register write request

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

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

Request Types

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

BACKPLANE ENV ONLY: NOT CONCERN W/ THESE.

Request Speed

LReq[4:6]	Data Rate
000	S100
001	S1600
010	S200
011	S3200
100	S400
110	S800
Others	Reserved

Status

Bits	Name	Description
0	ARB Reset Gap	Used by the link in the dual phase busy/retry state machine
1	Subaction gap	Used by the link to detect end of Isochronous cycle
2	Bus reset start	Phy has entered bus reset state
3	PHY Interrupt	Phy has detected one of the following conditions: Loop Cable power fail Arbitration state machine time-out Bias change on a disabled port
4:7	Address	Address of register being read
8:15	Data	Data in register above

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

Bits	0	1	2	3	4	5	6	7
0000b	Physical ID						Root	CPS
0001b	RHB	IBR	Gap Count					
0010b	Speed		E	Total Ports				
0011b	A Status-0		B Status-0		CH-0	Con-0	Reserved	
0100b	A Status-1		B Status-1		CH-1	Con-1	Reserved	
Total ports +0010b	A status-n		B Status-n		CH-n	Con-n	Reserved	
+0011b	Environment		Register Count					
+0100b	Vendor dependent							

Extended PHY Registers

Root hold off BIT

I am root

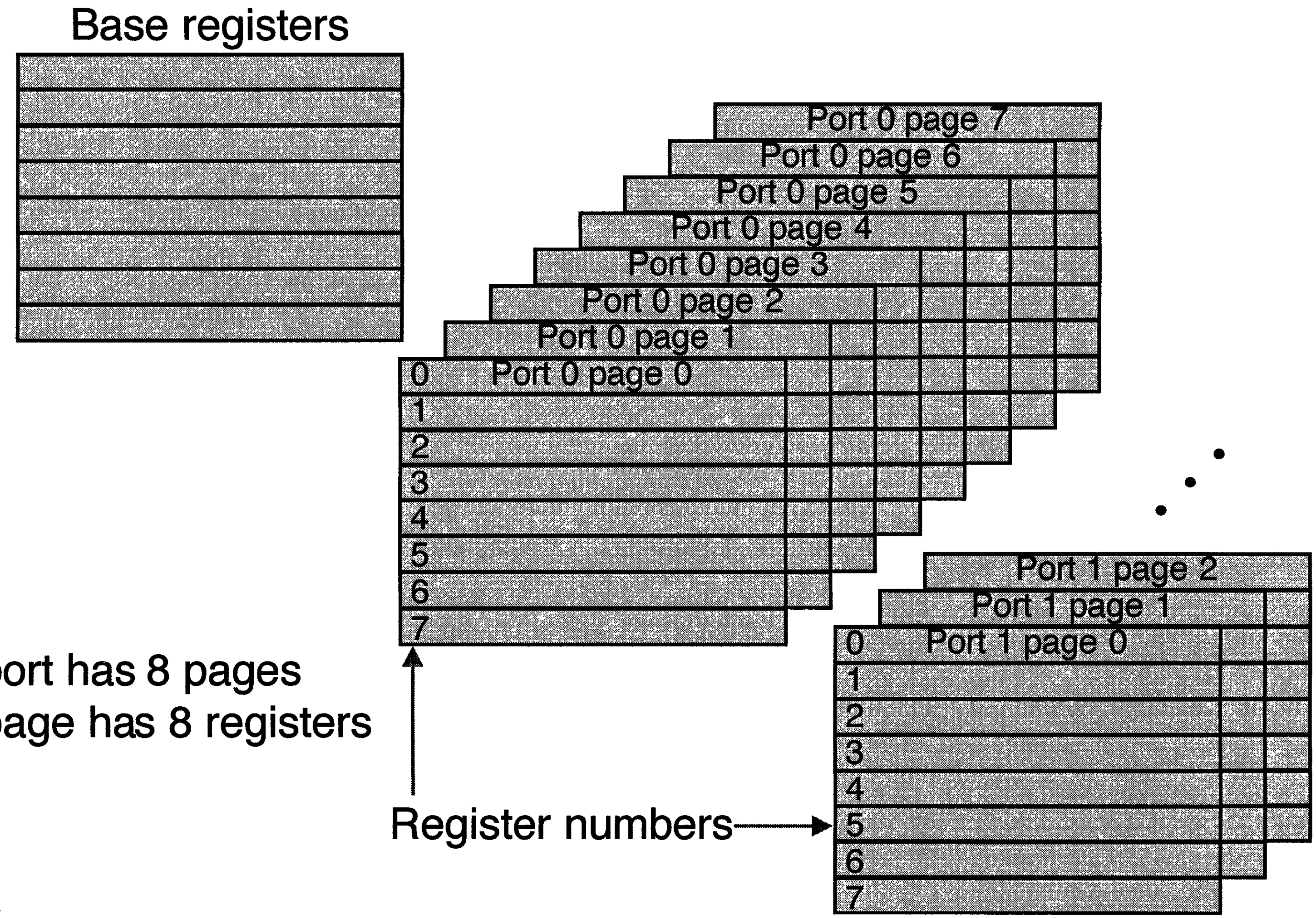
capabilities

Bits	0	1	2	3	4	5	6	7
0000b	Physical ID						Root	CPS
0001b	RHB	IBR	Gap Count					
0010b	Extended = 7			Total Ports				
0011b	Max speed			Rsv	Delay (i) * 20 + 144			
0100b	L	C	Jitter			1394 Power class		
0101b	watch dog	ISBR	Loop	Pwr fail	Timeout	Bias	enab acc	enab multi
0110b	Reserved							
0111b	Page select			Port select				
1000b	Register 0 (page select)							
1111b	Register 7 (page select)							

iate short reset (1.8 μs)

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

Extended Registers

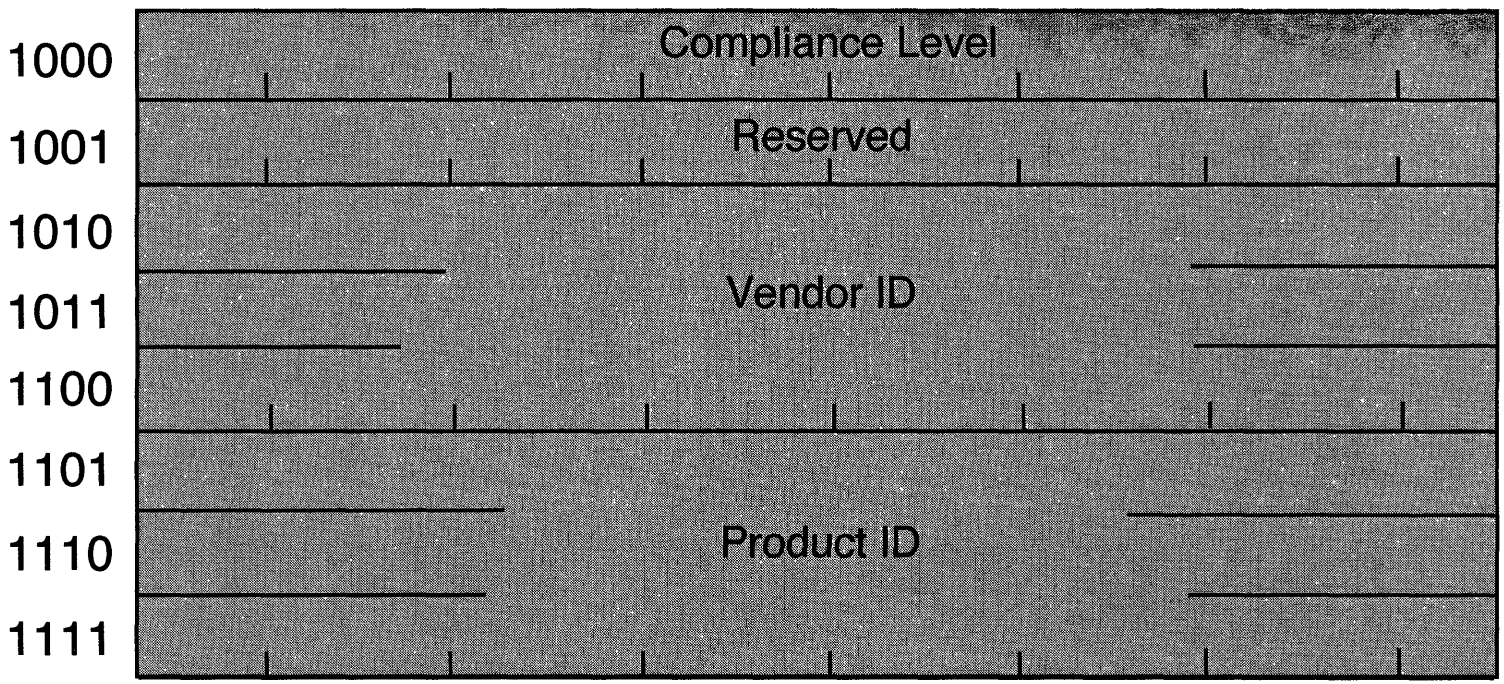


Every port has 8 pages
Every page has 8 registers

Extended PHY registers - Page 0

	0	1	2	3	4	5	6	7
1000b	A Status		B Status		Ch	Con	Bias	Dis
1001b	Negotiated speed			Int Enable	Fault	Reserved		
1010b	Reserved							
1011b	Reserved							
1100b	Reserved							
1101b	Reserved							
1110b	Reserved							
1111b	Reserved							

Extended PHY registers - Page 1



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? 50 MHz
2. Where do I remember the speed of each peer port? PHY reg - negotiated speed
3. Which function breaks SCSI write data into 1394 sized packets? Transaction Layer
4. How many registers are available per port on each Phy? ~~2~~ 64

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

abdicate bit only for if bus manager is not capable of power manager. another node sets this bit.

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

Unit	Link	PHY	Port	Power Usage	Performance
D0	L0	H0	P0	High	High
D1	Reserved	Reserved	Reserved		
D2	L2	H2	P2		
D3	L3	H3	P3	Low	Low

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 it power class, it shall cause a bus reset

Power Classes

to bus

Device Type	Class	Power Supplied	Power Consumed by PHY	Power Consumed by Link
Self	000b	None	None	None
Primary Provider	001b	15 watts	None	None
Primary Provider	010b	30 watts	None	None
Primary Provider	011b	45 watts	None	None
*	100b	None	3 watts	None
Reserved	101b			
Consumer	110b	None	3 watts	3 watts
Consumer	111b	None	3 watts	7 watts

* - 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

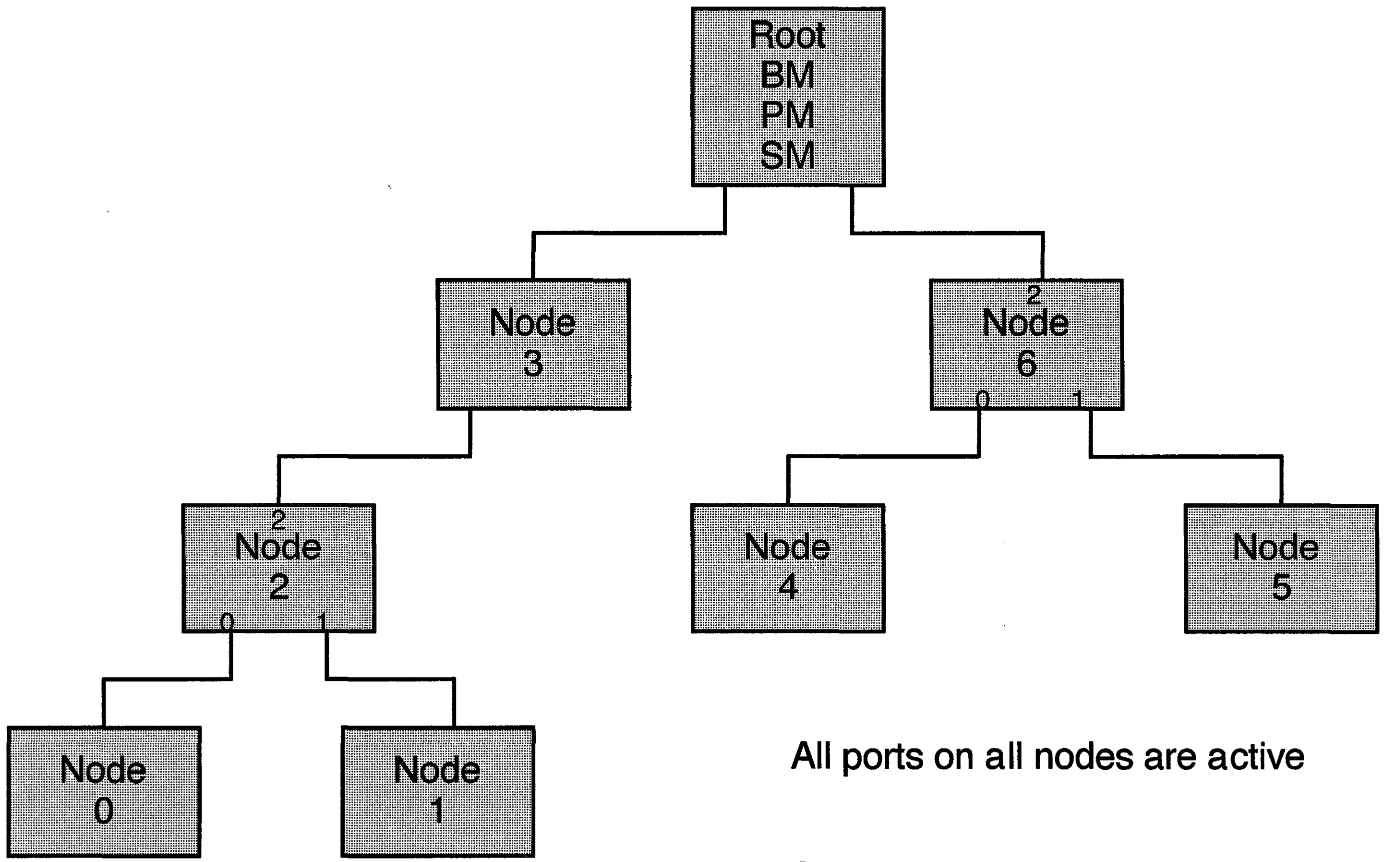
Suspend Target

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

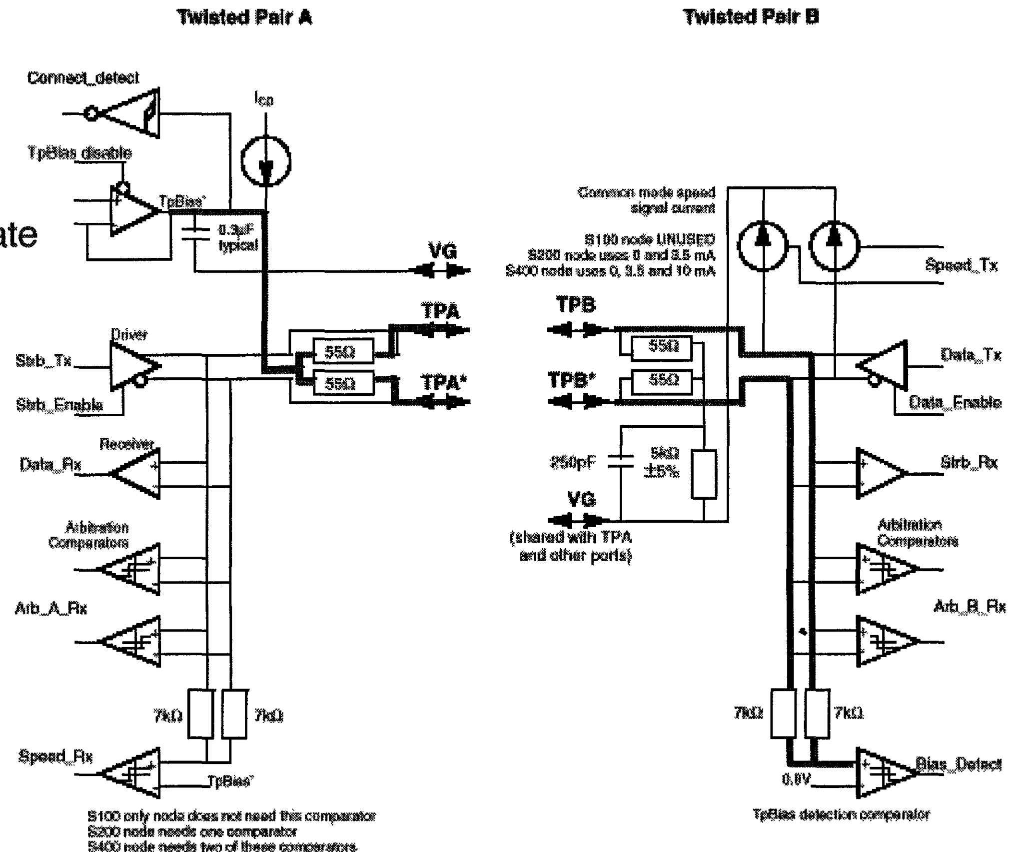
Port detected a RX_DISABLE_NOTIFY

OR

Port no longer detects TpBias

1394a PHY

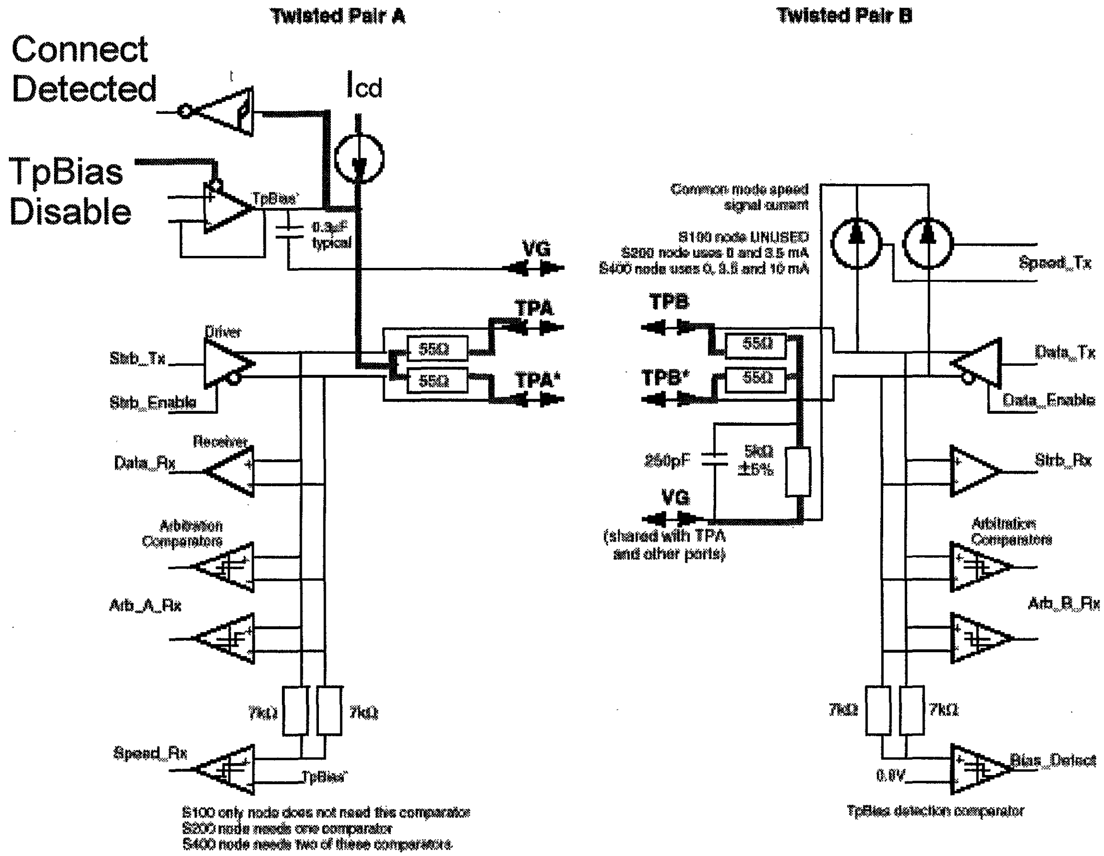
TpBias
Generate



TpBias
Detect

Sensing TpBias from the peer node was called:
"Connected" in 1394-1995, or
"Bias" in 1394a

1394a PHY



Twisted Pair A
Twisted Pair B
Connect Detected
Icd
TpBias Disable
TpBias*
0.3µF typical
VG
TPA
TPA*
Driver
Strb_Tx
Strb_Enable
Receiver
Data_Rx
Arbitration Comparators
Arb_A_Rx
7kΩ
7kΩ
Speed_Rx
TpBias*
S100 only node does not need this comparator
S200 node needs one comparator
S400 node needs two of these comparators
Common mode speed signal current
S100 node UNUSED
S200 node uses 0 and 3.5 mA
S400 node uses 0, 3.5 and 10 mA
Speed_Tx
TPB
TPB*
55Ω
55Ω
55Ω
55Ω
Data_Tx
Data_Enable
250pF
5kΩ ±5%
VG
(shared with TPA and other ports)
Strb_Rx
Arbitration Comparators
Arb_B_Rx
7kΩ
7kΩ
0.8V
Bias_Detect
TpBias detection comparator

PHY Register Map

Contents

Address	0	1	2	3	4	5	6	7
0000_2	Physical_ID						R	PS
0001_2	RHB	IBR	Gap_count					
0010_2	Extended (7)			Total_ports				
0011_2	Max_speed				Delay			
0100_2	LCtrl	Contender	Jitter			Pwr_class		
0101_2	Watchdog	ISBR	Loop	Pwr_fail	Timeout	Port_event	Enab_accel	Enab_multi
0110_2								
0111_2	Page_select				Port_select			
1000_2	$\text{Register0}_{Page\_select}$							
⋮	⋮							
1111_2	$\text{Register7}_{Page\_select}$							

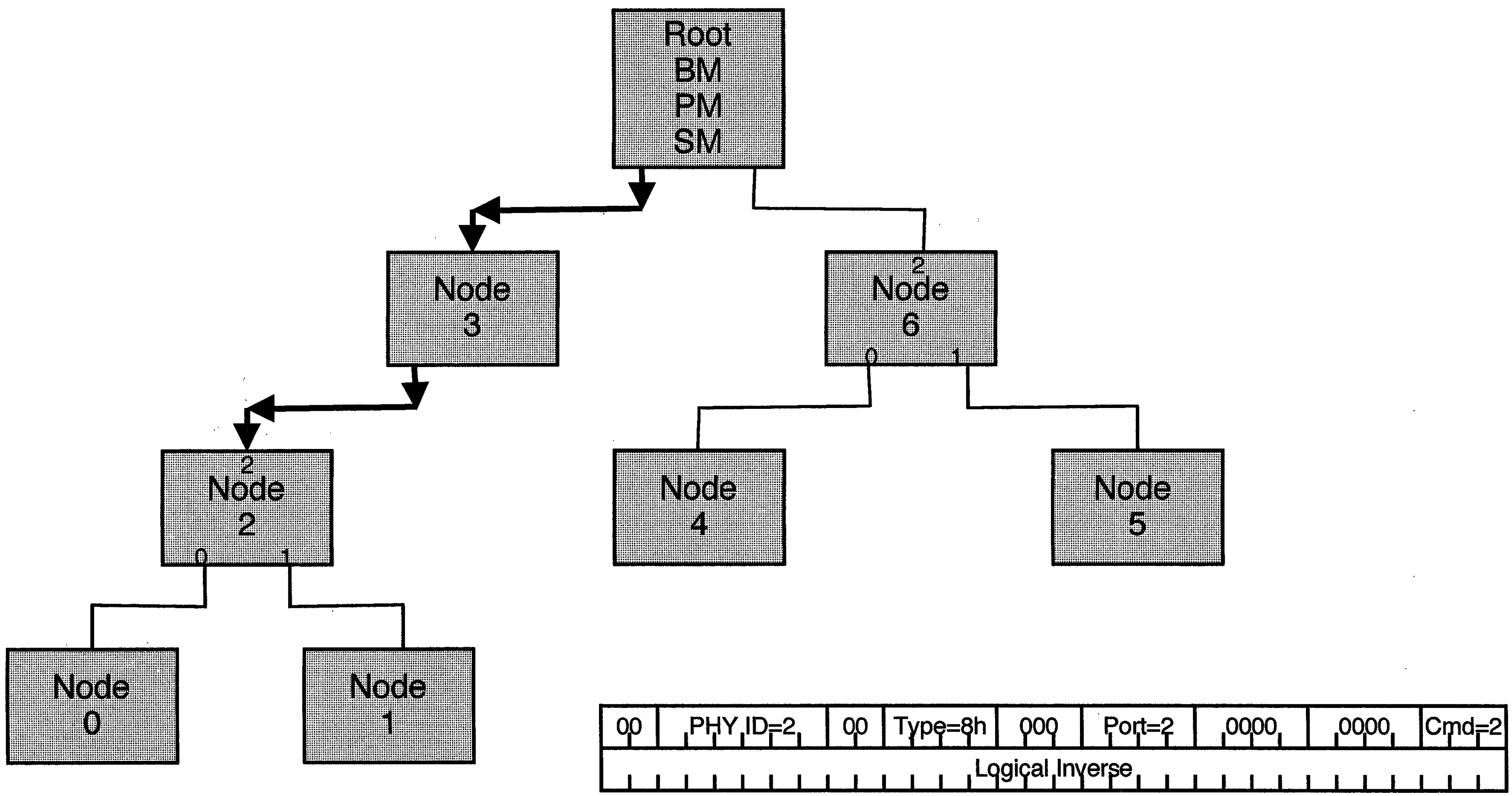
Extended PHY register map for the cable environment

Contents

Address	0	1	2	3	4	5	6	7
1000_2	AStat		BStat		Child	Connected	Bias	Disabled
1001_2	Negotiated_speed			Int_enable	Fault			
1010_2								
1011_2								
1100_2								
1101_2								
1110_2								
1111_2								

PHY register page 0: Port Status page

Suspend Command Packet

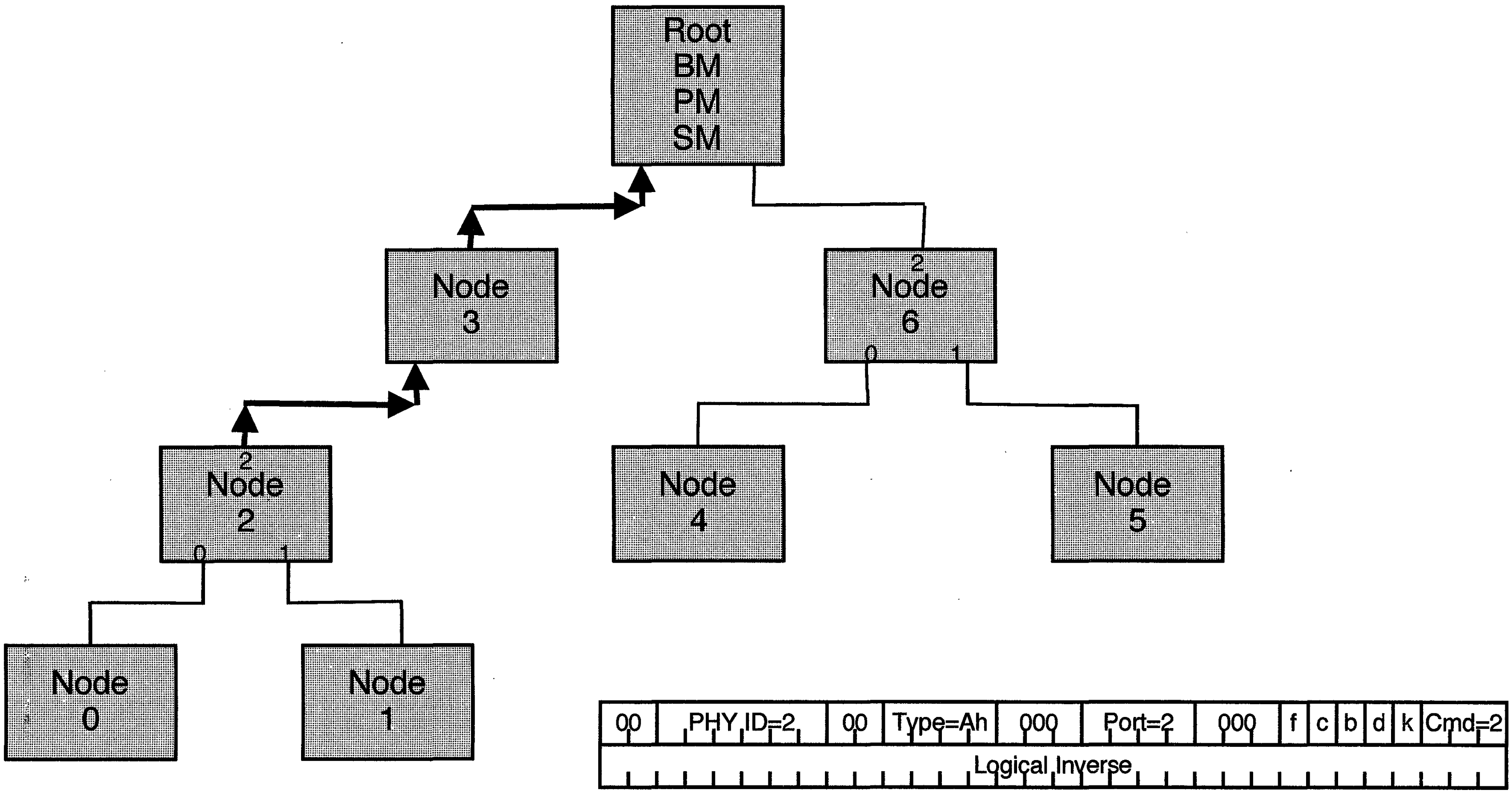


Fields

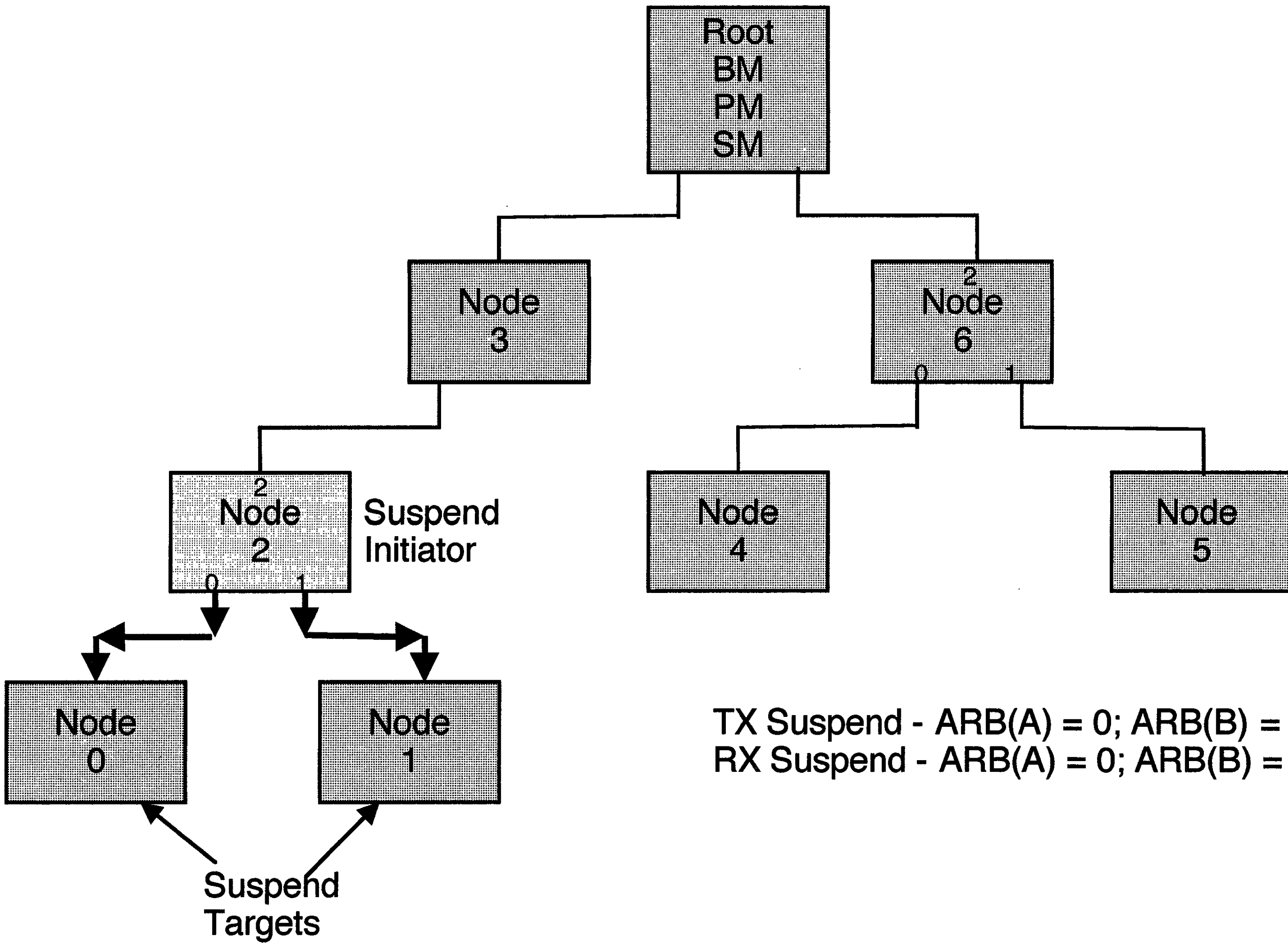
Type (Extended PHY packet)
8 = Command packet
A = Confirmation packet
F = Resume Node

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



Transmit Suspend



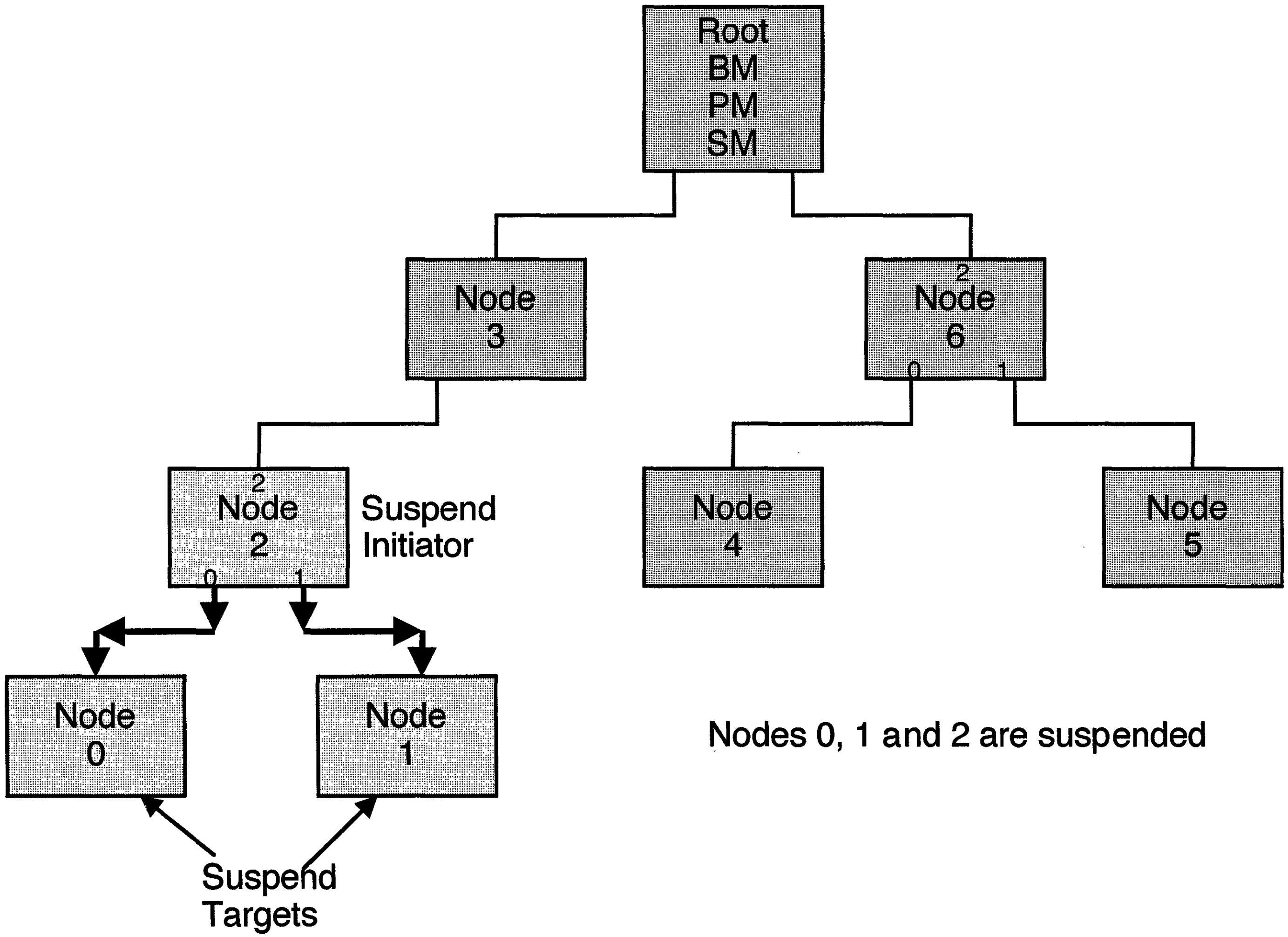
TX Suspend - ARB(A) = 0; ARB(B) = 0
RX Suspend - ARB(A) = 0; ARB(B) = 0

Bias Handshake

TPBIAS
TECT BOTH
DROP. BUT
DN'T

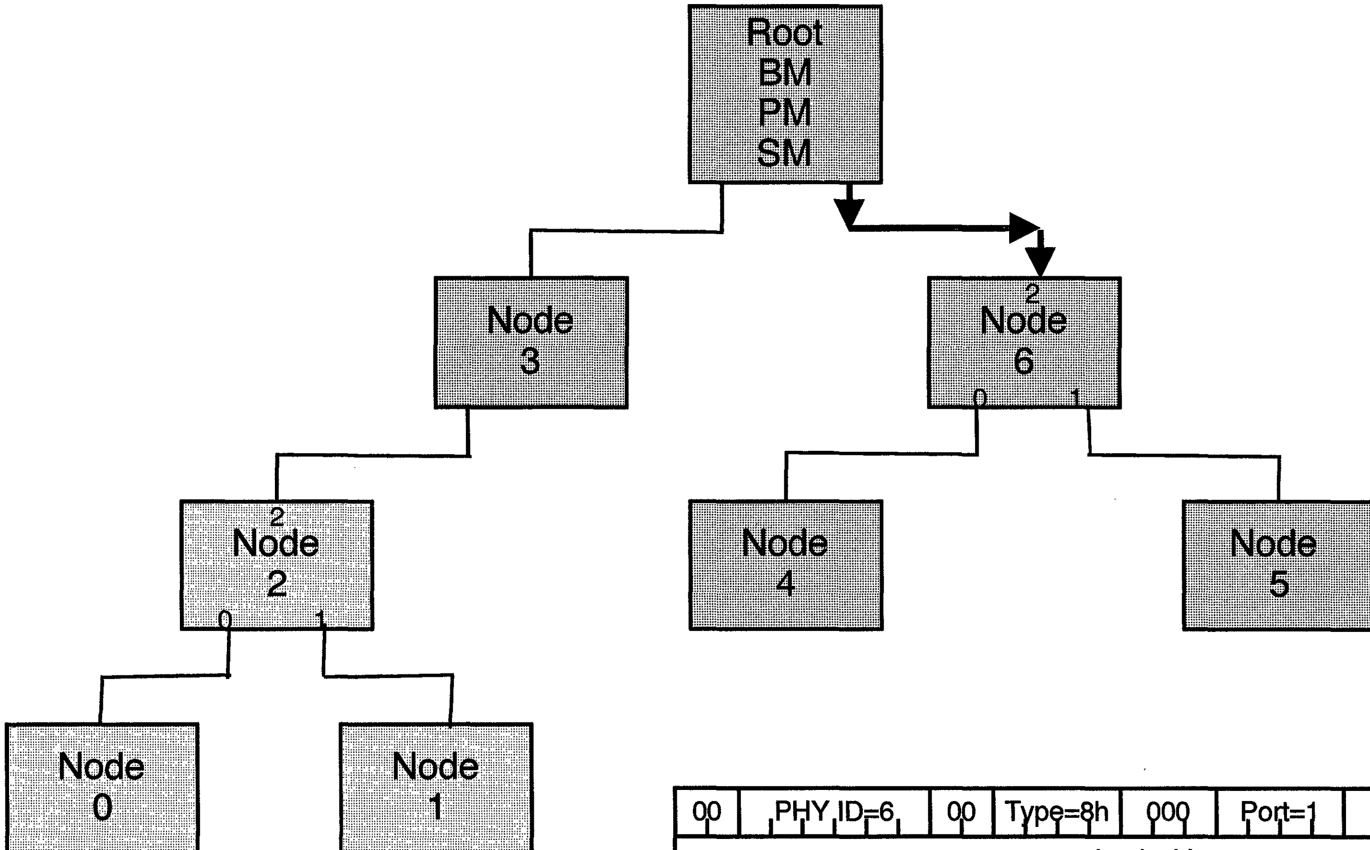
- Suspend Initiator sends TX_SUSPEND to peer port
- Suspend Target receives RX_SUSPEND
- Suspend Target drops TpBias
- Suspend Initiator drives TpBias low until internal Connect Detect circuitry becomes active
- Suspend Initiator disables TpBias and places its output in high impedance state
- Suspend 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

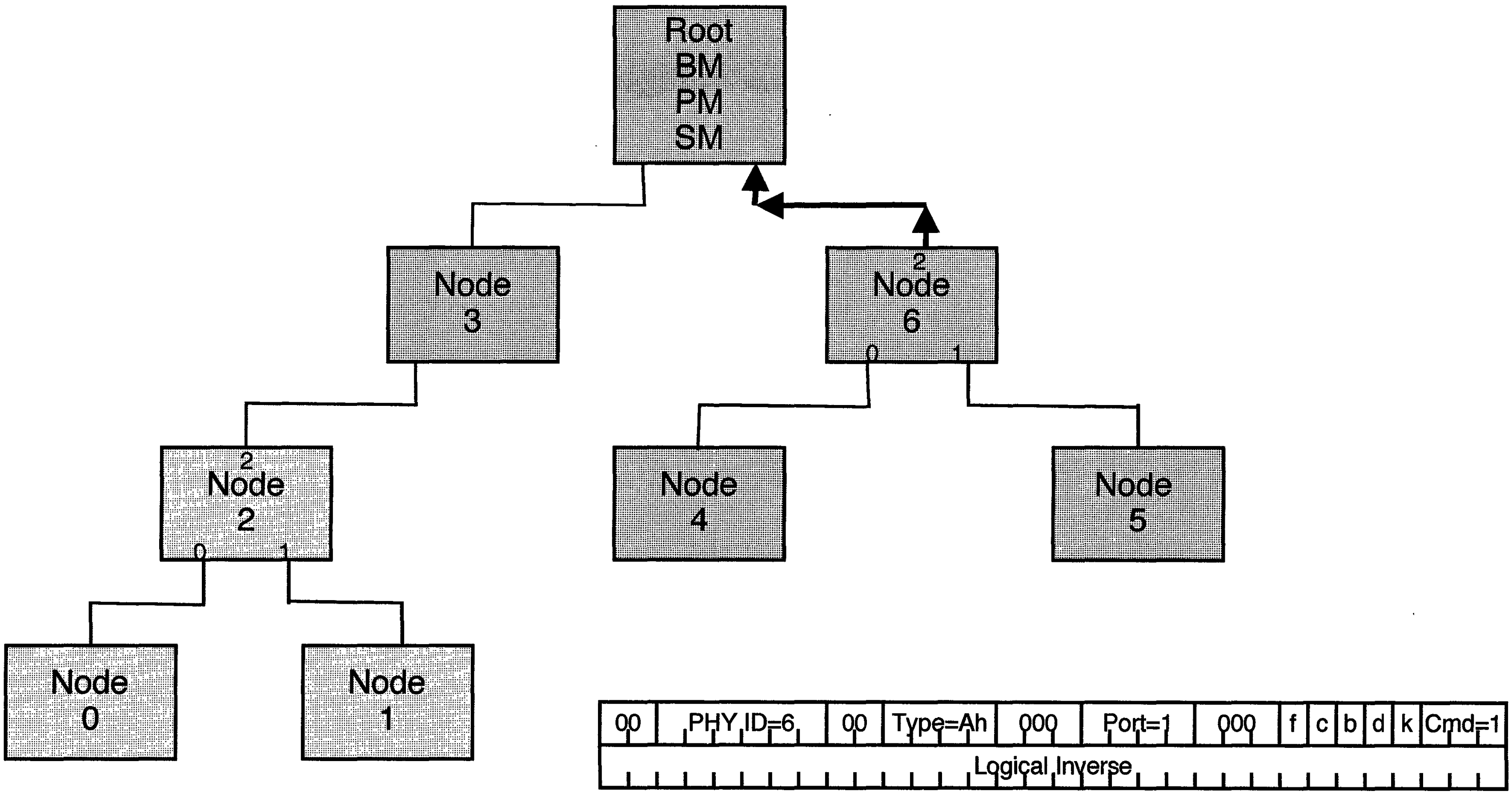
Disable Node 6



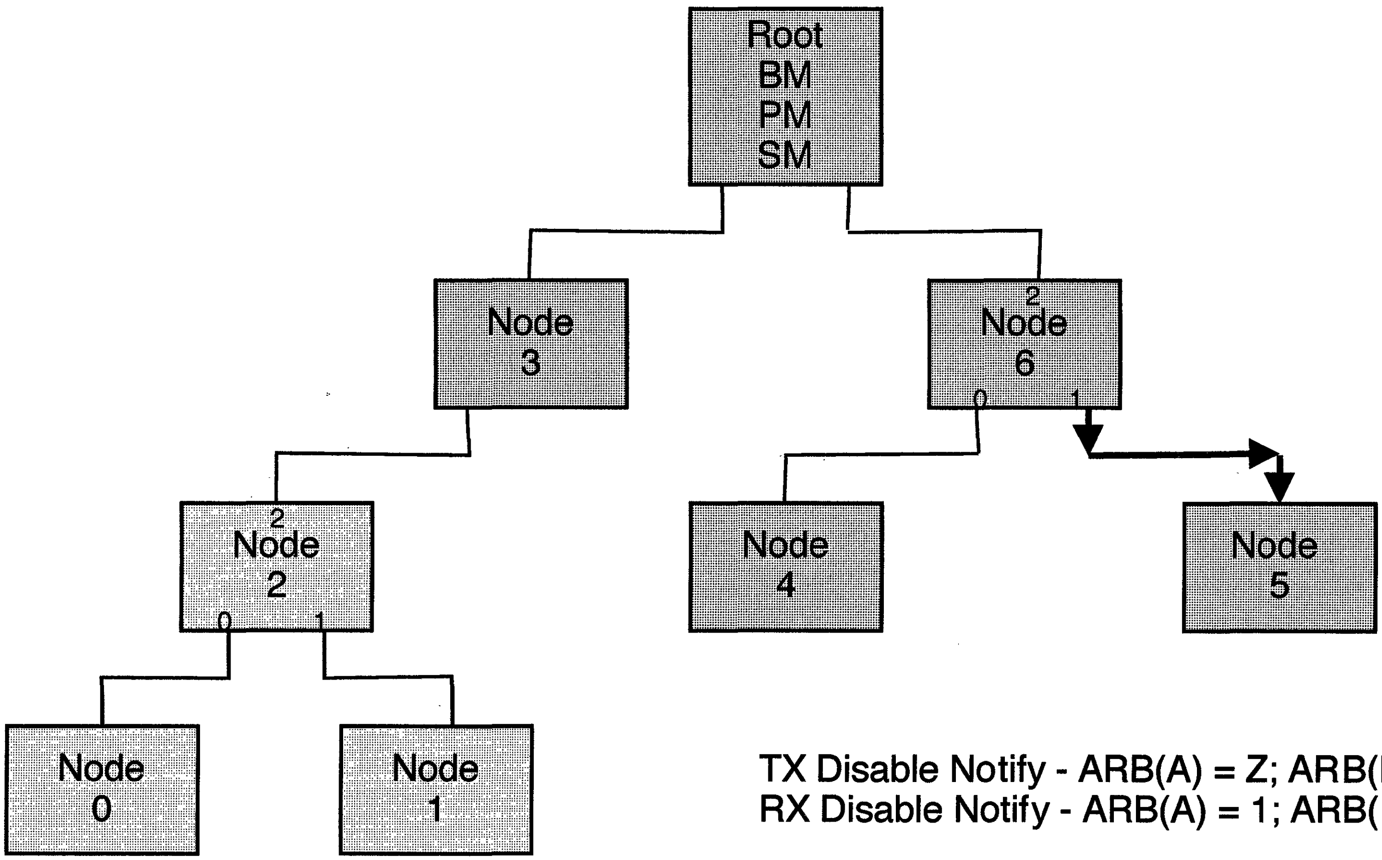
00	PHY_ID=6	00	Type=8h	000	Port=1	0000	0000	Cmd=1
Logical Inverse								

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

Confirmation Packet

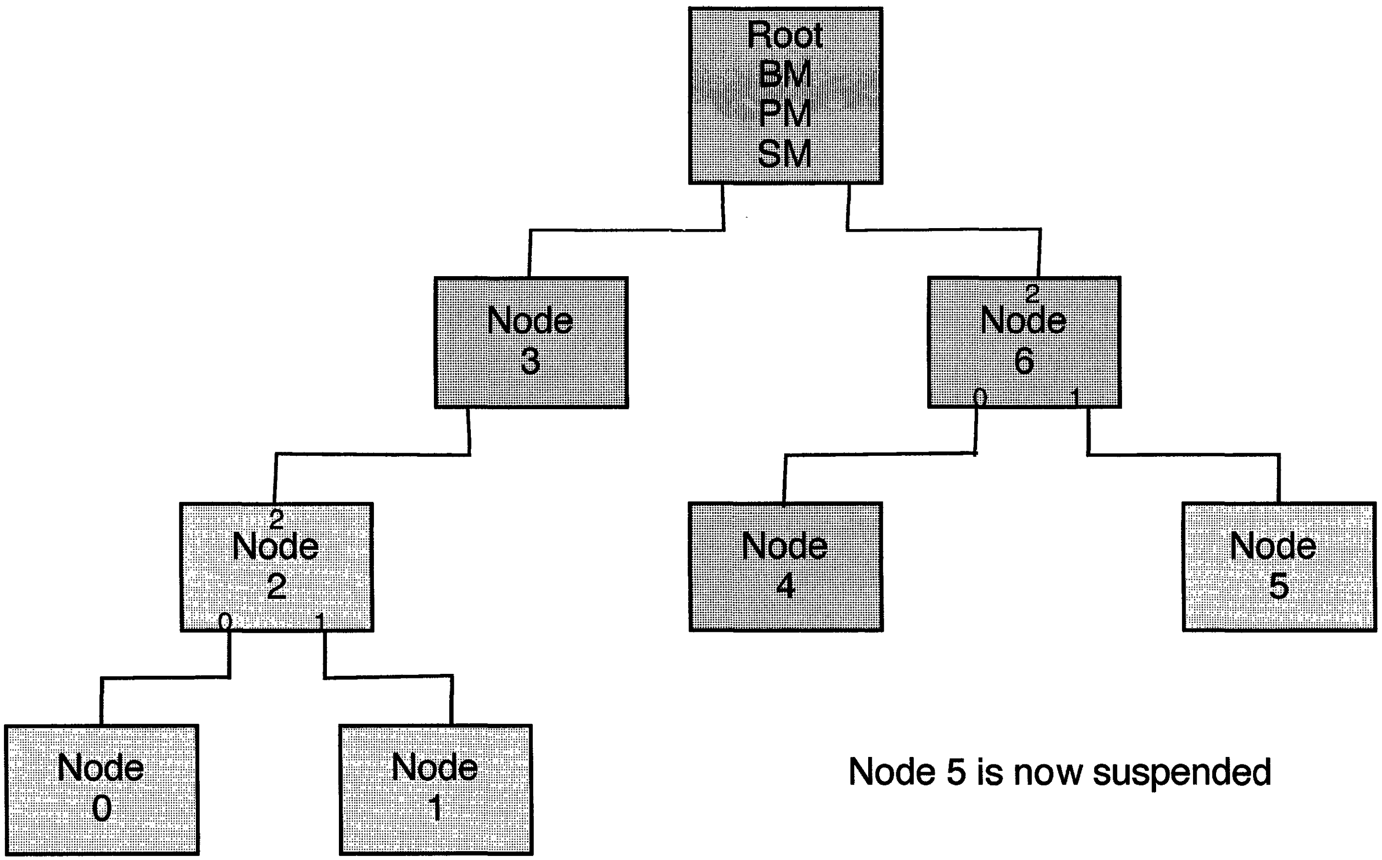


TX Disable



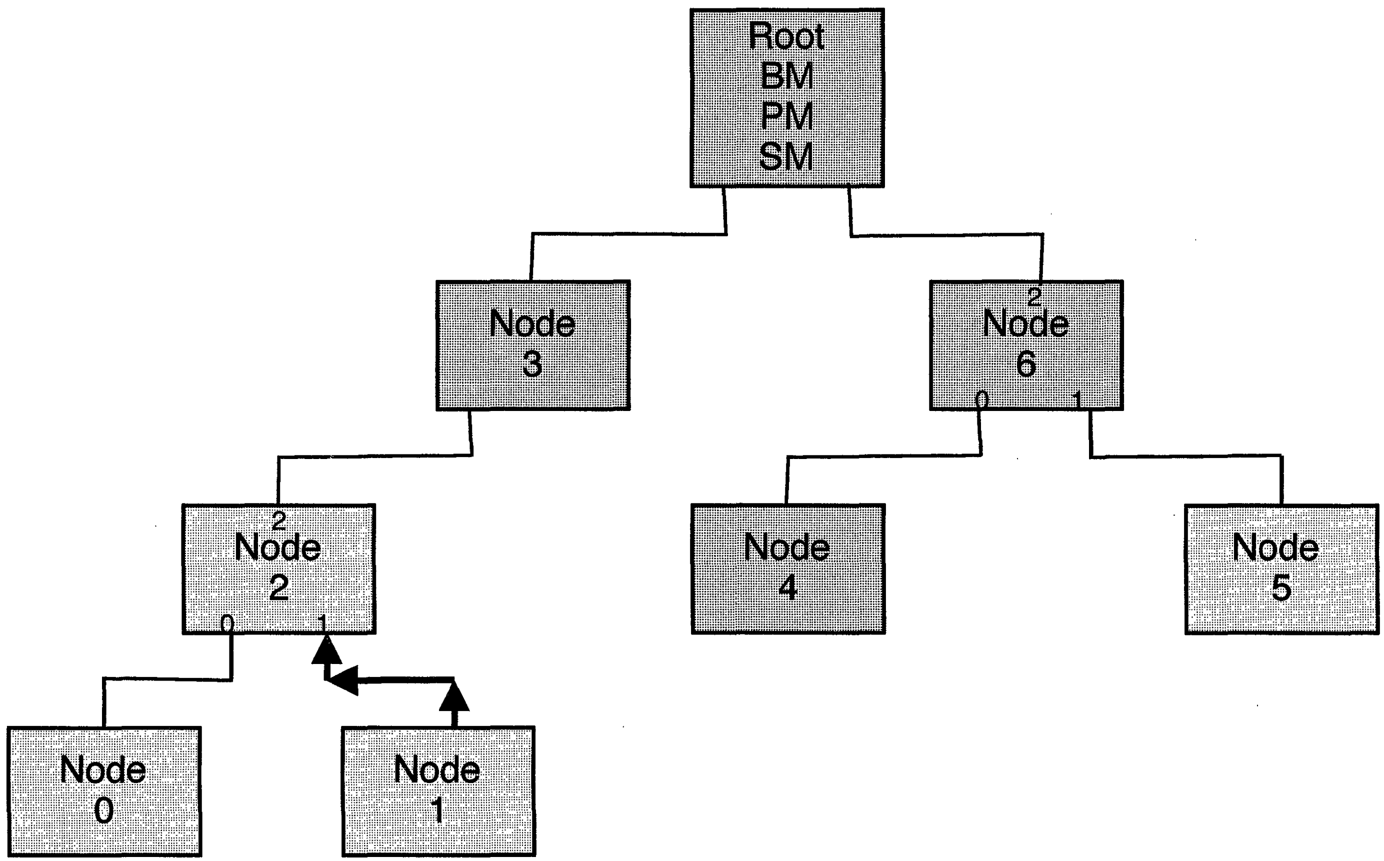
TX Disable Notify - ARB(A) = Z; ARB(B) = 1
RX Disable Notify - ARB(A) = 1; ARB(B) = Z

Disable Node 6 Completed



Node 5 is now suspended

Resume



Root
BM
PM
SM
Node
3
2
Node
6
0
1
2
Node
2
0
1
Node
4
Node
5
Node
0
Node
1

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

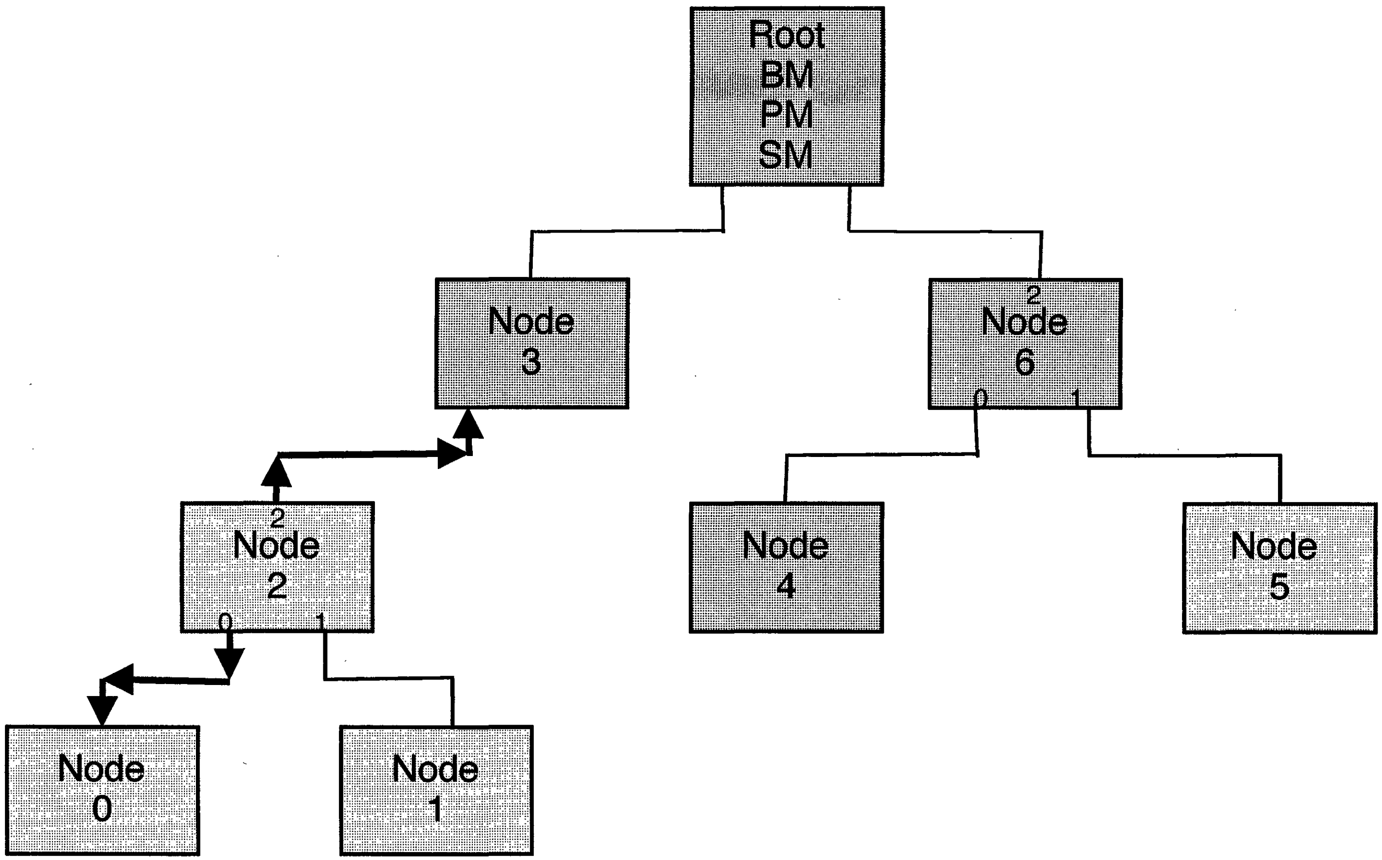
Resume Initiator will detect TpBias from peer

Resume Initiator will issue Bus Reset

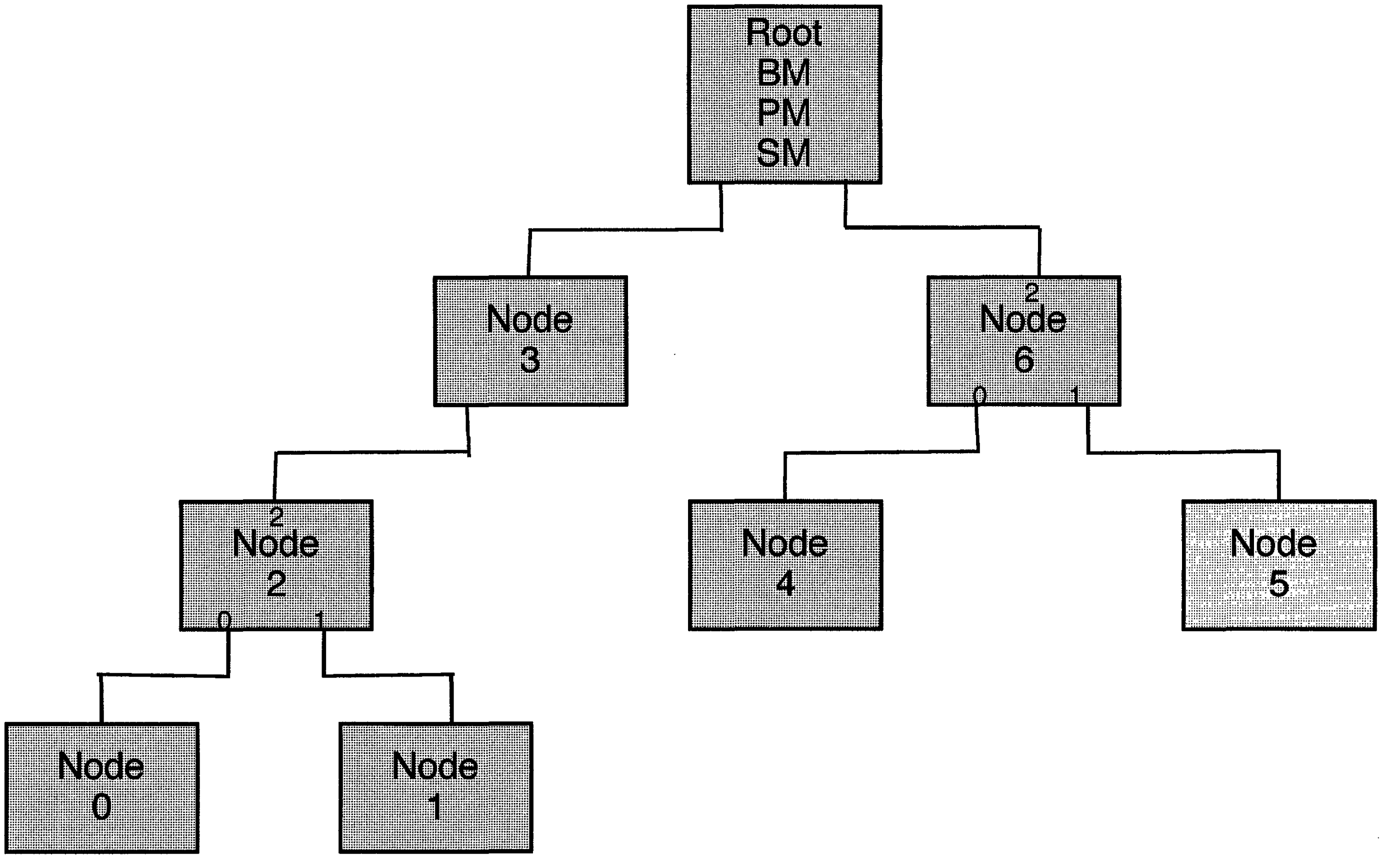
- Boundary nodes will wait 3 Reset_Detect times and issue Short Bus Reset on the active bus
- Other Resume initiators will wait 7 Reset_Detect times and issue regular bus reset

No node will transition to active state until after the reset

Resume Propagated



Resume Completed



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

Power Management Notes

Section 15

1394 Standards

Where to Get the Information

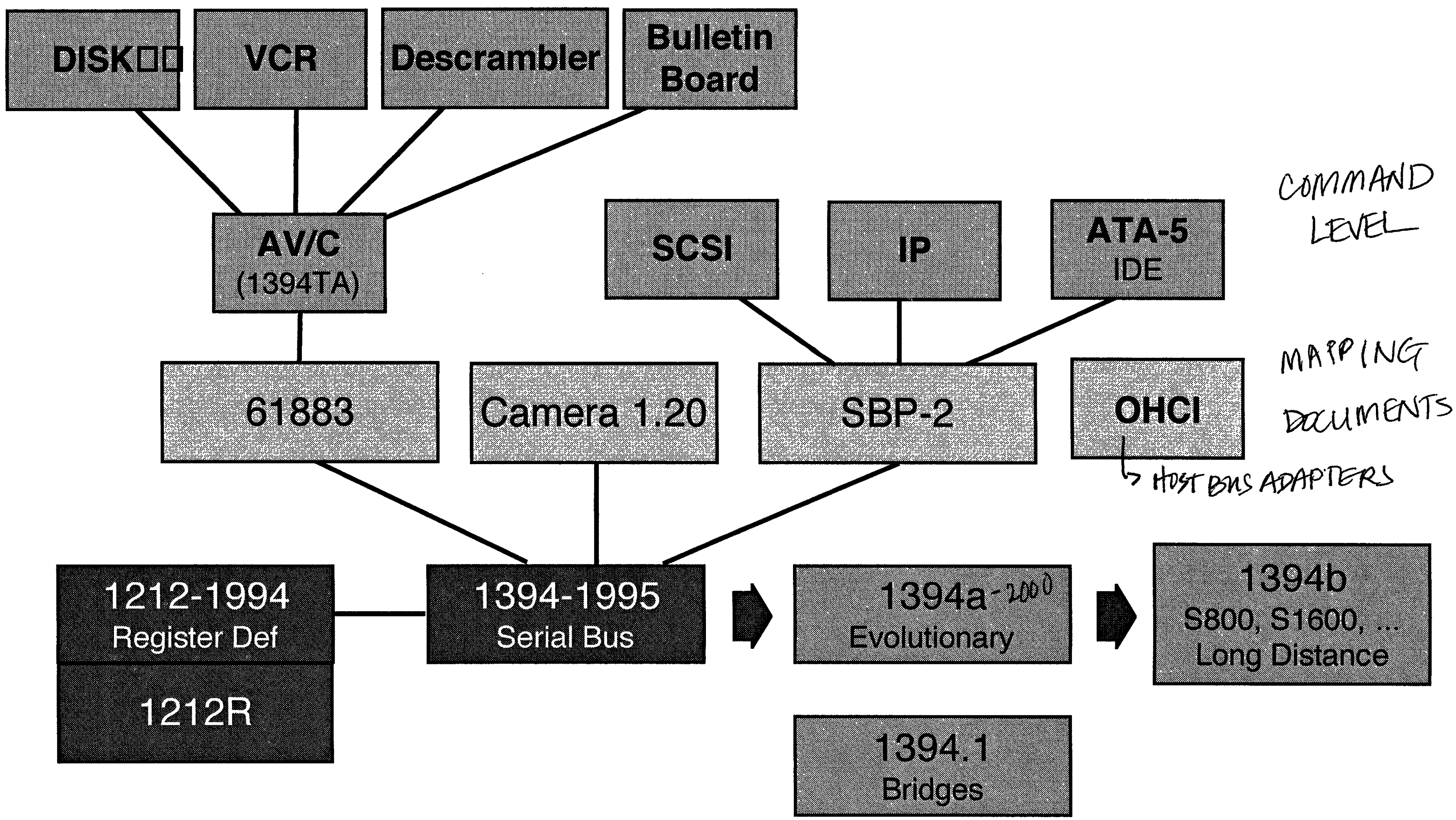
And How to Understand it

Subjects Covered

1394 standards families

Where to get more information

Specifications



Specifications - Transport Level

IEEE 1212-1991	Control and Status Registers
IEEE 1394-1995	High Speed Serial Bus, Approved 1995
IEEE P1394a	Evolutionary improvements to 1394, Working group
IEEE P1394b	800, 1600 Mbps and beyond, Working group working hard
IEEE 1394.1	Bridges, Working in progress

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 SCSI
- www.t13.org 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.catc.com
- CHIPS { 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

advanced info; before it's in spec

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

Specifications Review

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

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 120 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)

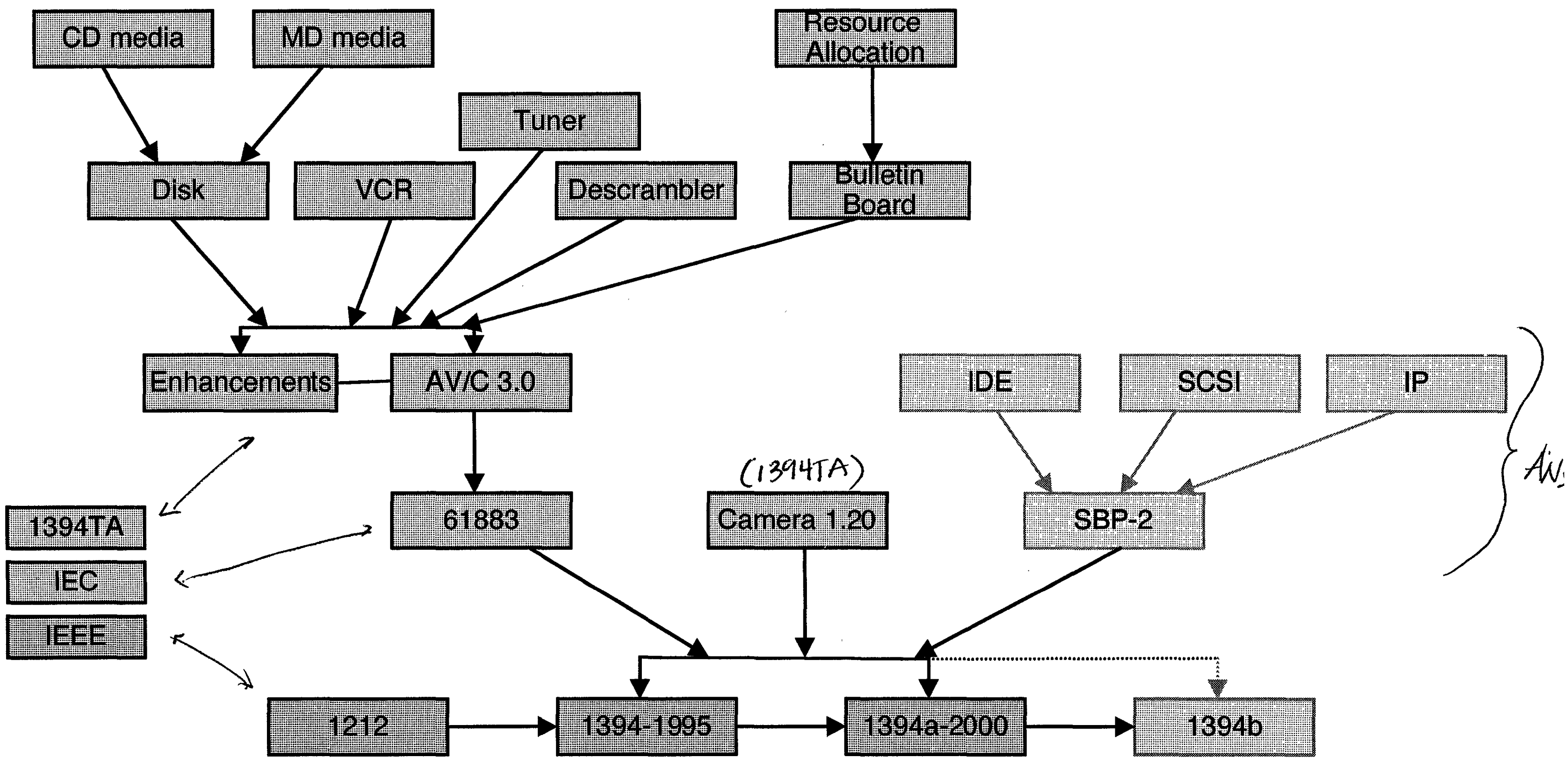
Moving data

- 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


CD media
MD media
Resource Allocation
Tuner
Disk
VCR
Descrambler
Bulletin Board
Enhancements
AV/C 3.0
IDE
SCSI
IP
(1394TA)
1394TA
IEC
IEEE
61883
Camera 1.20
SBP-2
1212
1394-1995
1394a-2000
1394b

61883 / 1394 Compliance

CABLE PHY LINK TRANSACTION MANAGEMENT

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

Nodes shall be IRM capable. STATE_CLEAR.cmstr bit required

(CYCLE MAS CAP. & ISO CAPABLE) IMPLIED

Nodes shall implement plug control registers

Nodes shall implement the following registers:

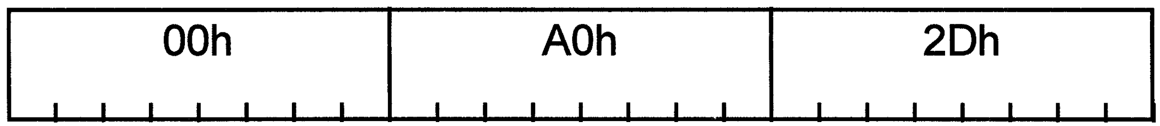
- Cycle Time
- Bus Time
- Bus Manager ID
- Bandwidth Available
- Channels Available

Nodes shall implement General Configuration ROM

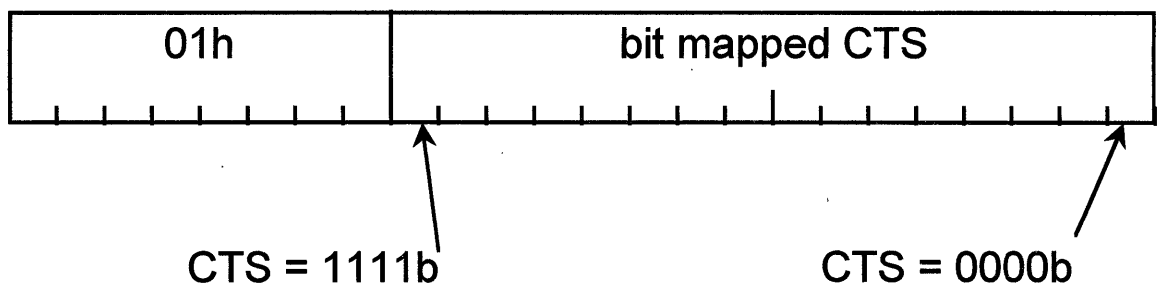
- Unit Directory - See next page

61883 Unit Directory Entries

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



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



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

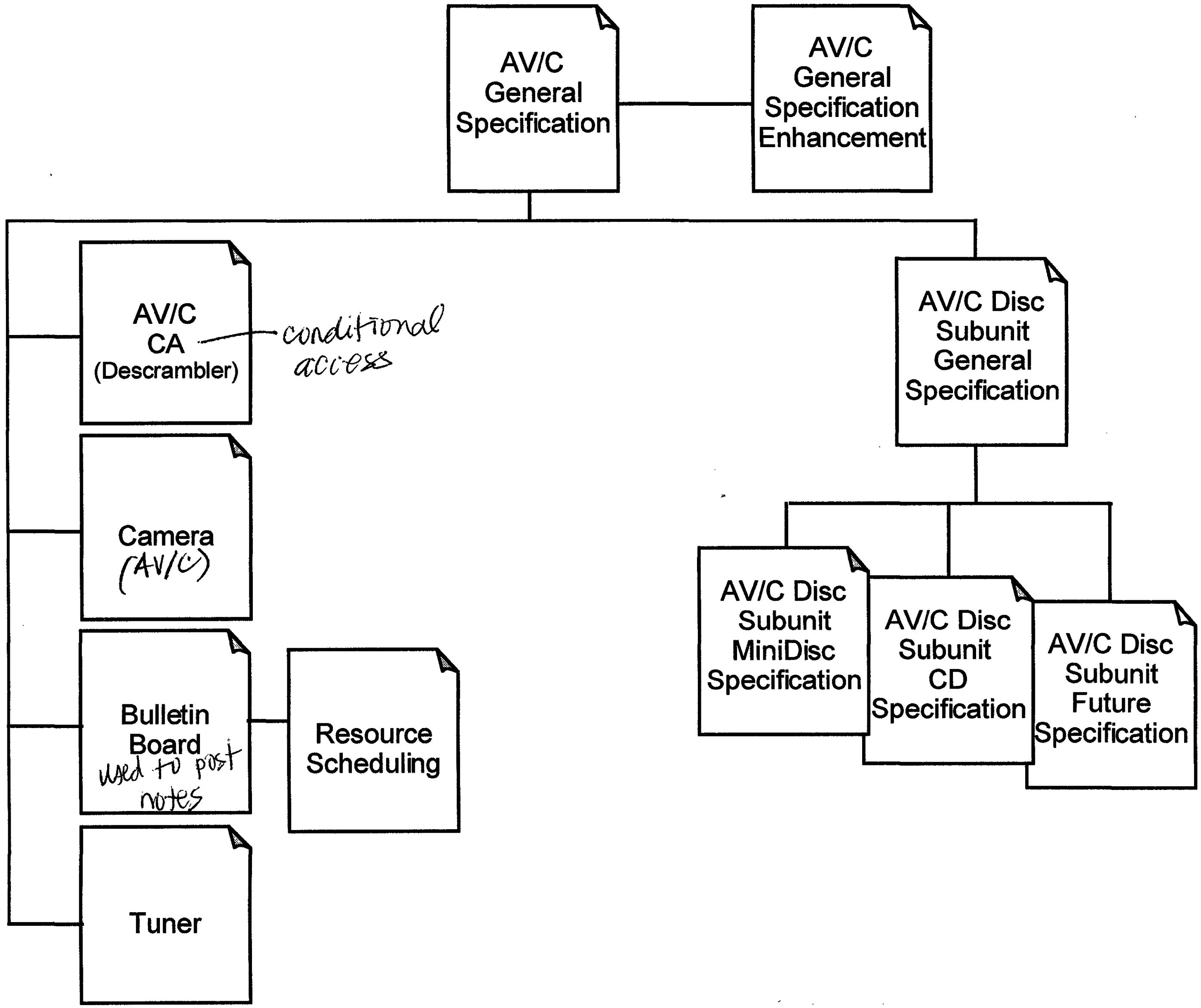
Reference - IEC 61883 (1883) Standards

Standards

Project	Order Number	Title	
61883-1	100C/182/FDIS	General	
61883-2	100C/183/FDIS	SD-DVCR Transmission	standard def TV
61883-3	100C/184/FDIS	HD--DVCR Transmission	high def TV
61883-4	100C/185/FDIS	MPEG data Transmission	
61883-5	100C/186/FDIS	SDL-DVCR Transmission	standard def compressed TV

IEC Website - www.iec.ch

AV/C Document Structure



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

Spd	Reserved	XSpd	Input Plugs

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

O	b	Point-to-point	r	Channel	Reserved

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

protected
can be taken
away by one ~~of~~
who established
it only.

Isochronous Connection Management
Output Master PCR

Spd	Broadcast Base	Reserved	XSpd	Output Plugs

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

O	b	Point-to-point	XSpd	Channel	Spd	Overhead	Payload

O	Online
b	Broadcast connection exists
Point-to-point	Number of point-to-point connections for this plug
Spd	Speed (0 = S100, 1 = S200, 2 = S400, 3 = XSpd)
XSpd	Speed (0 = S800, 1 = S1600, 2 = S3200, 3 = Reserved)
Channel	Channel number for this plug
Overhead	Allocation units of overhead
Payload	Maximum data quadlets in a single isochronous packet (0 = 1024 quadlets)

FOR THIS CHANNEL

↳ 4KB data

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

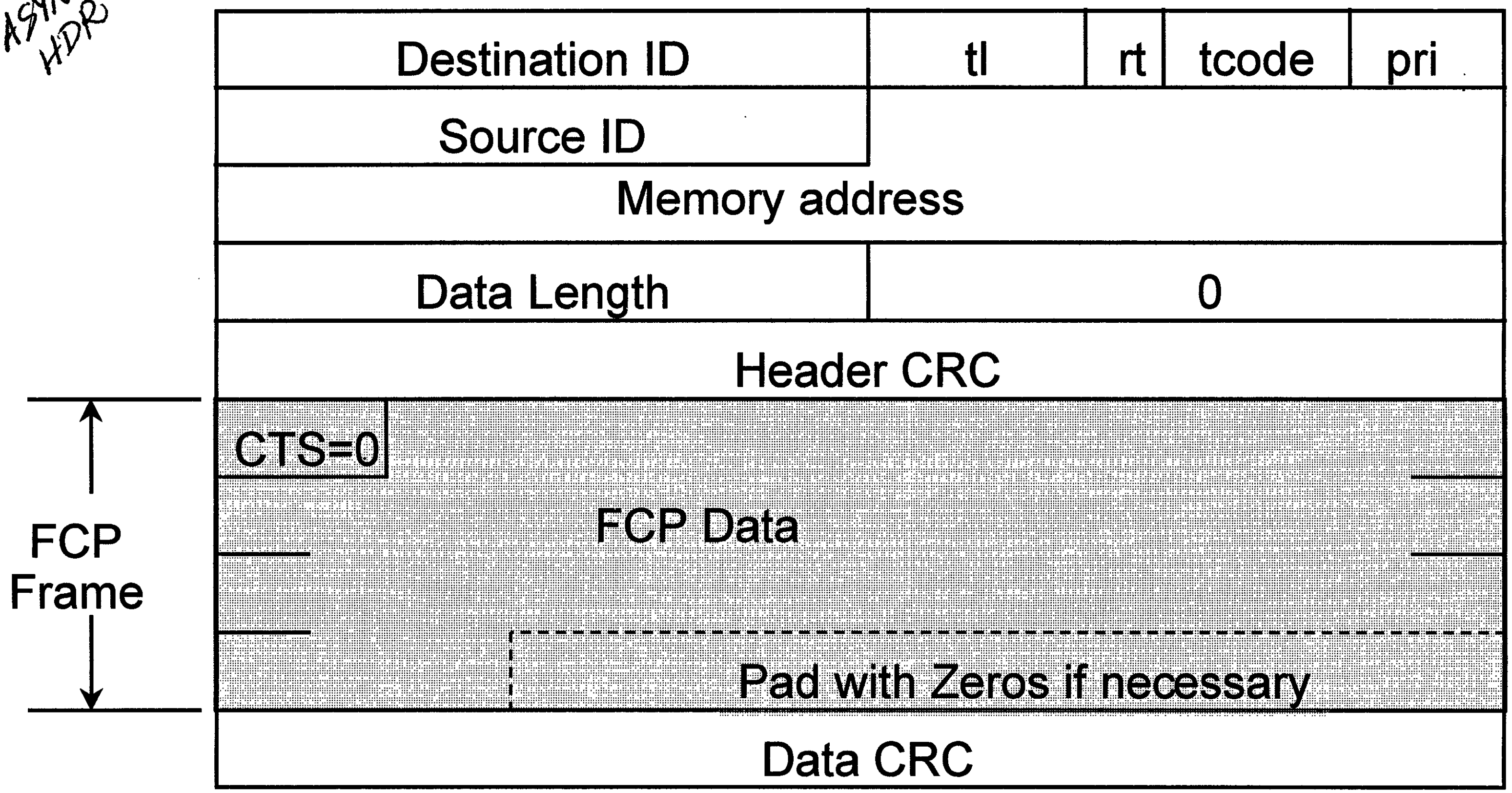
Control	Set a feature to a certain value
Status	Tell me the current setting of a feature
Notify	Tell me if a certain event occurs
Inquiry	Tell me if you support this feature and/or parameters

FCP Command/Response Buffer

FFFF F000 0000	1212 defined CSRs
	1394 defined CSRs
FFFF F000 0400	Configuration ROM
FFFF F000 0800	
FFFF F000 0900	PCRs
FFFF F000 0A00	
FFFF F000 0B00	FCP CMD Buffer
FFFF F000 0D00	FCP RSP Buffer
FFFF F000 0F00	

1394 Frame with FCP Frame

ASYNC HDR



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	

V

AV/C Command Frame

FCP DATA

Transmitted first

CTS	ctype	SU type	SU ID	OpCode	Operand(0)
Operand(1)		Operand(2)		Operand(3)	Operand(4)
...					
Operand(n)		Pad with Zeros if necessary			

Transmitted last

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

Transmitted first

<table>
<tr><td>CTS</td><td>resp</td><td>SU type</td><td>SU ID</td><td colspan="2">OpCode</td><td colspan="2">Operand(0)</td></tr>
<tr><td colspan="2">Operand(1)</td><td colspan="2">Operand(2)</td><td colspan="2">Operand(3)</td><td colspan="2">Operand(4)</td></tr>
<tr><td colspan="8">...</td></tr>
<tr><td colspan="2">Operand(n)</td><td colspan="6">Pad with Zeros if necessary</td></tr>
</table>

Transmitted last

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

V

AV/C Command Frame - Definitions

ctype	Command type		
		0	Control
		1	Status
		2	Inquiry
		3	Notify
		4-7	Reserved
		8-F	Reserved for response code
resp	Response code		
		0-7	Reserved for command type
		8	Function not implemented
		9	Accepted
		A	Rejected
		B	In transition
		C	Implemented/Stable
		D	Changed
		E	Reserved
		F	Interim

AV/C Command Frame - Definitions

SU type	SubUnit type		
		0	Video monitor
		1-2	Reserved
		3	Disc recorder or player
		4	Tape recorder or player
		5	Tuner
		6	Reserved
		7	Video camera
		8-1B	Reserved
		1C	Vendor unique
		1D	Reserved
		1E	Extended to next byte
		1F	Unit
SU ID	SubUnit ID		
		0-4	Instance number which one of that type
		5	Extended to next byte
		6	Reserved
		7	Ignored

AV/C Command Frame - Definitions

Examples:

Second video camera

0	0	1	1	1			0	0	1

Subunit Type = 7   Subunit ID = 1

Fifth VCR

0	0	1	0	0			1	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

Range	Addressing Mode
0 - F	Units and Subunits
10 - 3F	Units
40 - 7F	Subunits
80 - 9F	Reserved
A0 - BF	Units and Subunits
C0 - DF	Subunits
E0 - FF	Reserved

Unit Commands

Notify – tell me if event occurred

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

Common Subunit Commands

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

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

VCR Subunit Commands

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

VCR Subunit Commands (continued)

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

VCR Subunit Commands (continued)

Command	OpCode	Support Level C	S	N	Comments
TIME CODE	51h	R	M	-	Search or inquire about specified medium location
TRANSPORT STATE	DOh	-	M	O	Report current state of transport mechanism
WIND	C4h	*	-	-	Control transport mechanism motion when not in playback or record
WRITE MIC	62h	O	O	-	Store data in MIC

MEMORY IN CARTRIDGE

Playback Modes

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

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

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

Camera Subunit Commands

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

Disk Subunit Commands under AV/C

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

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

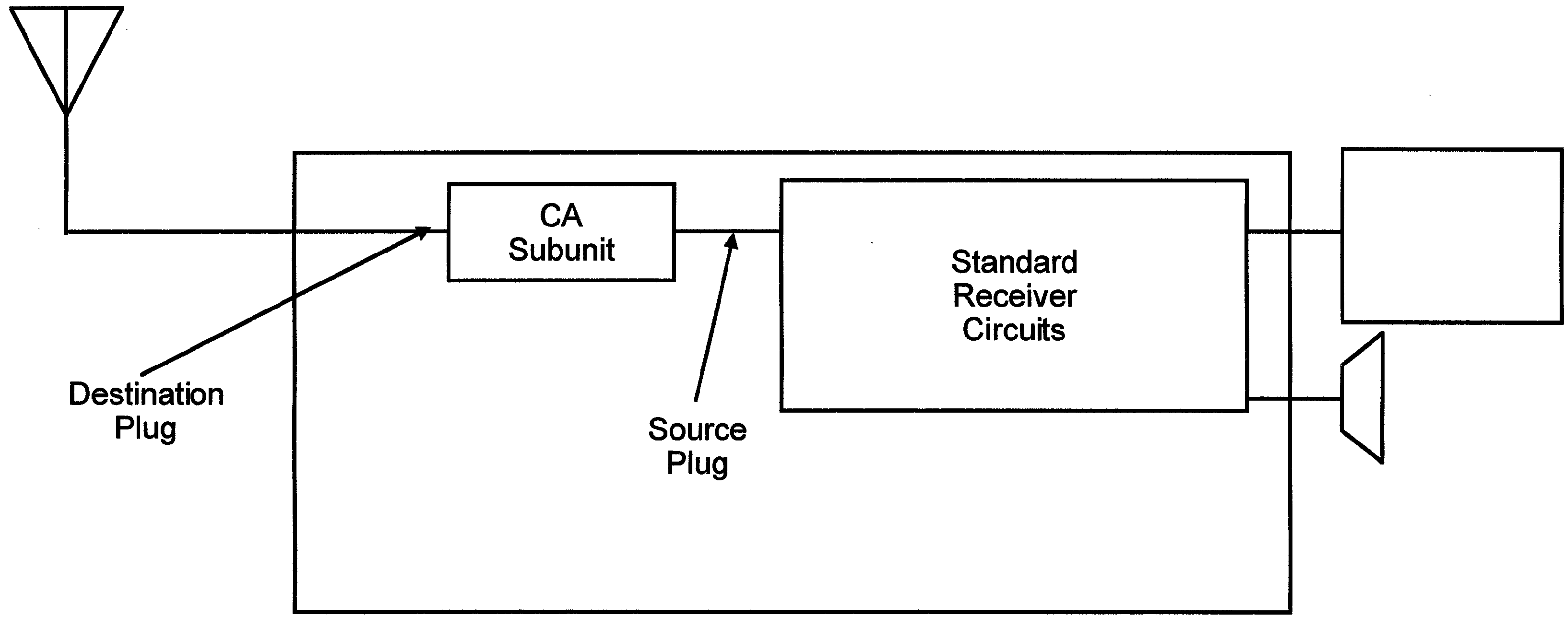
Bulletin Boards

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.

CA Subunit



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

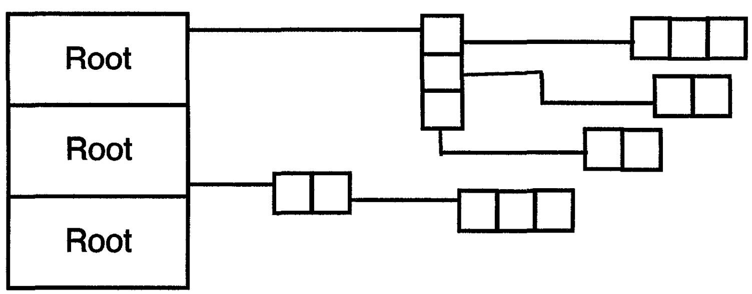
Created when manufactured or with 'Create Descriptor' command

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

Opcode	Open Descriptor (08h)
Operand	Descriptor ID (MSB)
"	"
"	Descriptor ID (LSB)
"	Subfunction
"	Reserved

Which descriptor and how much

Offset	Contents
00	Descriptor Type
01	Descriptor Type
:	Specific info
:	
:	

Descriptor block

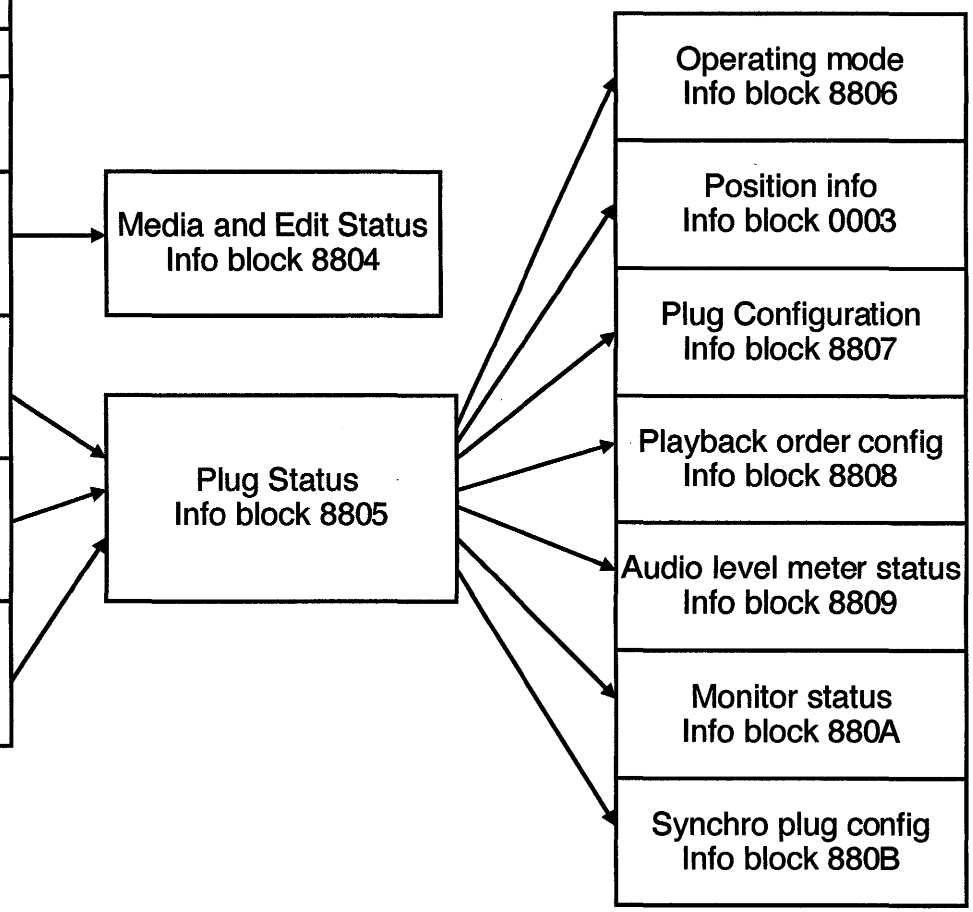
00	Subunit Identifier Descriptor
10	Object List Descriptor - specified by list ID
11	Object List Descriptor - specified by list type
20	Object Entry Descriptor - specified by object position
21	Object Entry Descriptor - specified by object ID
80	Disk Status Descriptor

Response to Read Disk Status Descriptor

Full Descriptor read			Information Block read	
Disk Status	80h	Descriptor type	Disk Status	80h
Reference Method	00h	Descriptor Type Specific Reference	Reference Method	01h
			Info Block ref path Level 0	xxh
				xxh
				xxh
			Info Block ref path Level 1	xxh
				xxh
				xxh
			Info Block ref path Level 2	xxh
				xxh
				xxh

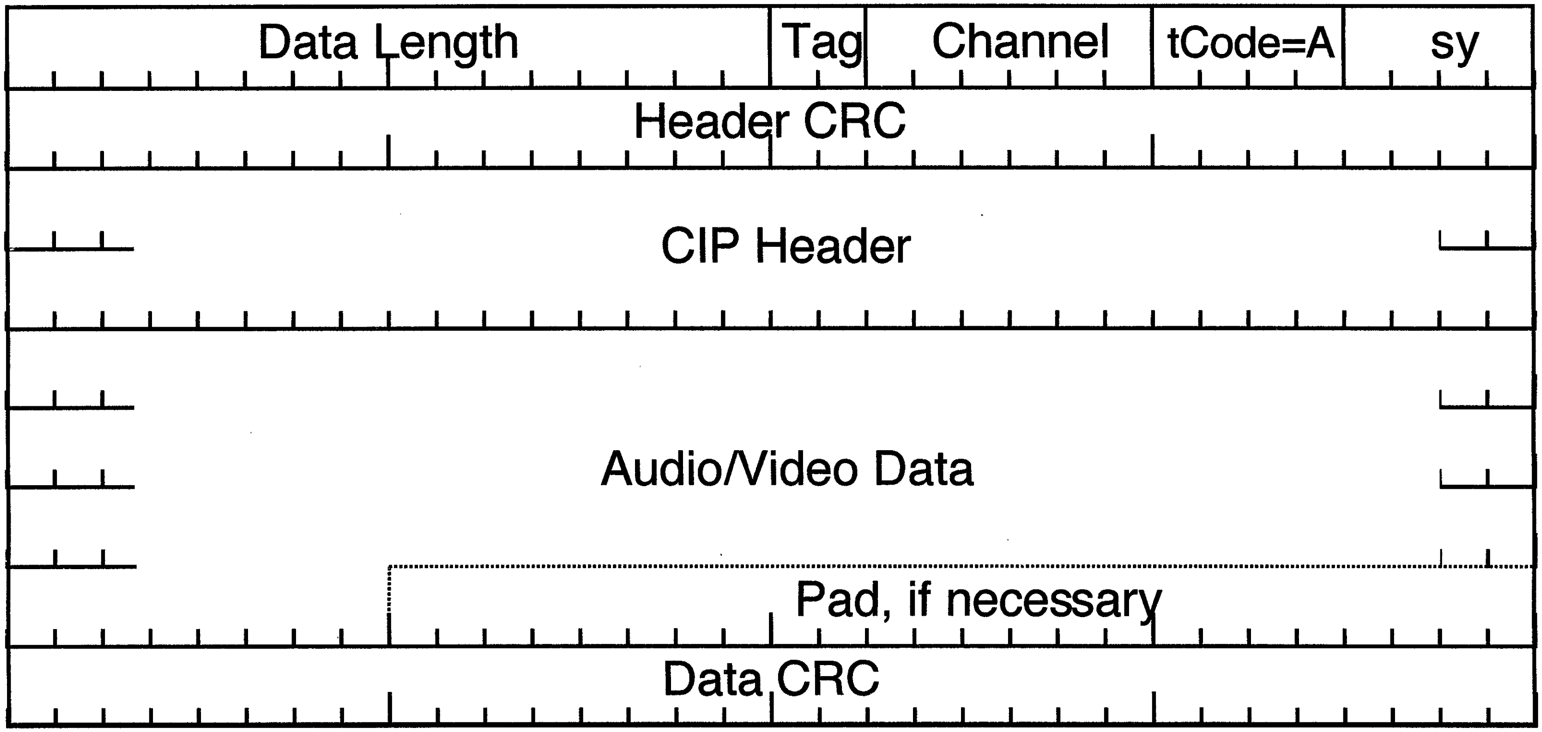
Disk Subunit Status Descriptor

Disk Subunit Status Descriptor	
Address	**Contents**
0000h	Descriptor length
0001h	
0002h	General disk subunit status Info block type 8800h
:	
:	
:	Destination plug status area Info block type 8801
:	
:	
:	Source plug status Info block type 8802
:	
:	
:	Synchro plug group status Info block type 8803
:	
:	



Isochronous Packets for Data Transfer

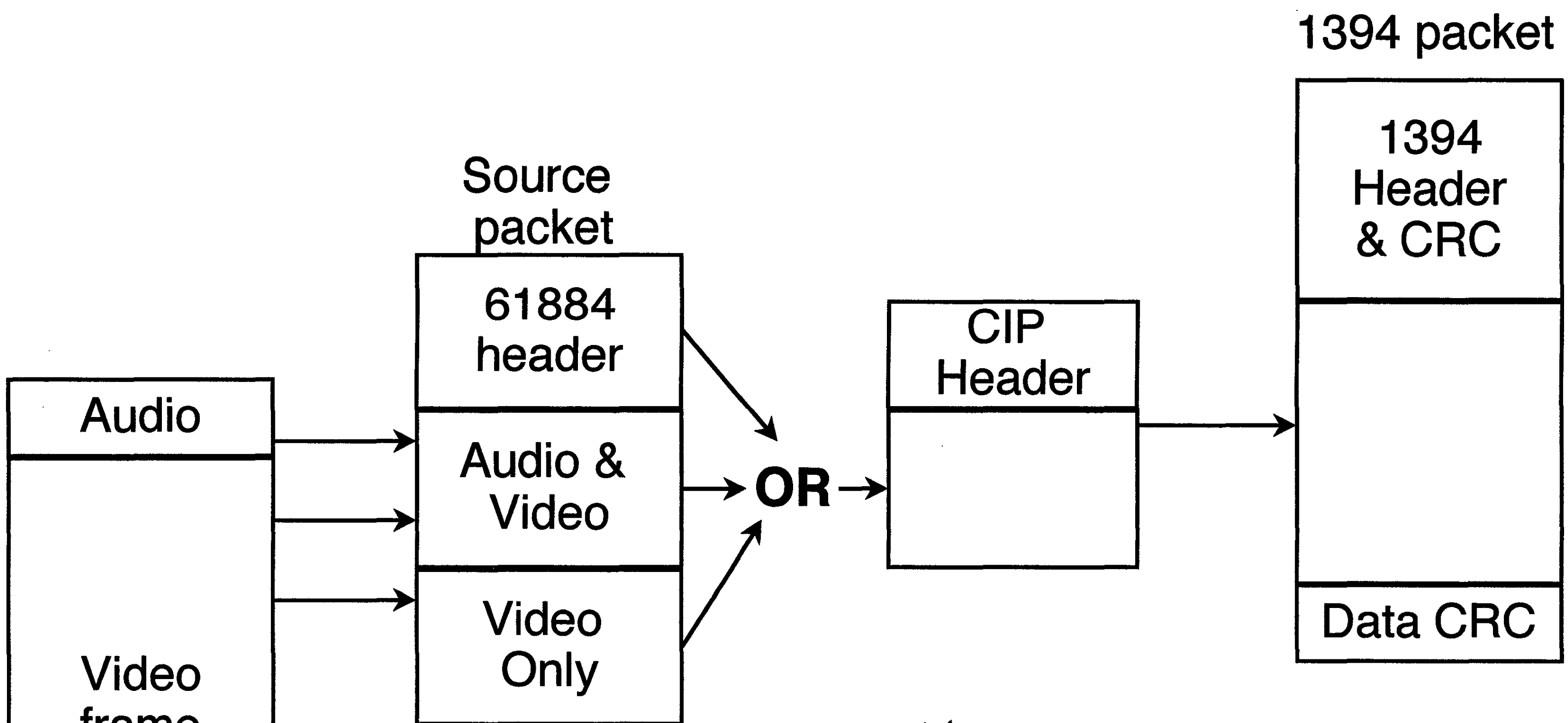
AV/C Document Structure



TAG

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

Packetizing Data



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

series of source packets until all data is covered

This page is intentionally blank

CIP Header

Common Isochronous Packet

2 FORMATS

eoh	0	SID	DBS	FN	QPC	S	R	DBC

eoh	0	0	FMT	50 - 60	SType	R	SYT

↑ frames/cycles per sec

eoh	0	SID	DBS	FN	QPC	S	R	DBC

eoh	0	1	FMT	FDF

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
S	Source packet header
DBC	Data Block Continuity counter
SType	See next page
SYT	Synchronization Timer (low order 16 bits of 1394 timer)
FDF	Format Dependent Field

Packetizing Data

scans fields

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

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) → to resynch: reconstruct the original blocks

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.

Example Source Packet Sequence for SDL Frame (60Hz)

Source Packet = 240 Bytes

80 Bytes

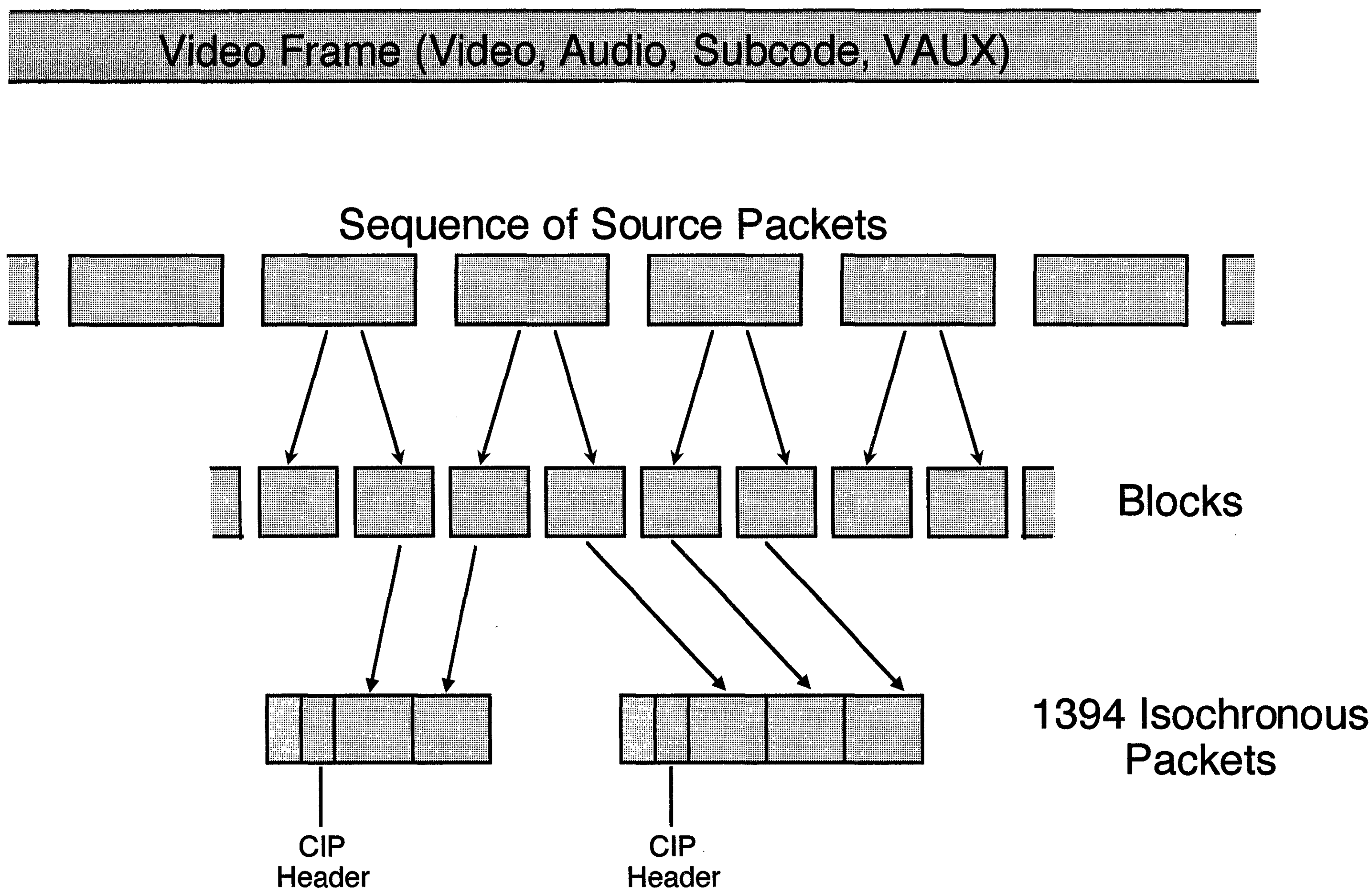
Source Packet #			
0	H	SC_0	SC_1
1	VA_0	VA_1	VA_2
2	A_0	V_0	V_1
3	V_2	V_3	V_4
49	V_{132}	V_{133}	V_{134}
50	H	SC_0	SC_1
51	VA_0	VA_1	VA_2
52	A_0	V_0	V_1
53	V_2	V_3	V_4
99	V_{132}	V_{133}	V_{134}
100	H	SC_0	SC_1

This sequence repeats 5 times per frame.

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

video
250 Source Packets per frame

Packaging the Data



Camera 1.20

Notes

Camera Specification 1.20 Works strictly w/ CSRs

- 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

Use CSRs to Control a Camera

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

OFFSETS

CSR Range	R/W	Function
000h	W	Initialize the Registers
100 - 1FFh	R	Inquiry formats and modes supported
200 - 3FFh	R	Inquiry frame rates supported
400 - 4FFh	R	Inquiry features supported
500 - 5FFh	R	Inquiry range of feature adjustment
600 - 6FFh	R	Inquiry Status for camera
	W	Set control for camera
700 - 7FFh		Reserved
800 - 8FFh	R	Inquiry Status for features
	W	Set control for feature

Locating Camera CSRs
Configuration ROM

	Offset	0-7	6-15	16-23	24-31
Bus Info Block	400h	04h	CRC length	ROM CRC value	
	404h	31h	33h	39h	34h
	408h	0 0 1 0 rsvd	FFh	max rec	rsvd
	40Ch	node vendor id			chp_id_hi
	410h	chip id lo			
Root Directory	414h	04h	CRC length	CRC	
	418h	03h	modular vendor id		
	41Ch	0Ch	rsvd	8380	
	420h	8Dh	indirect offset		
	424h	D1h	unit directory offset		

Root Directory

Configuration ROM (continued)

	Offset	0-7	6-15	16-23	24-31
Unit Directory	000h	0003h		CRC	
	004h	12h	unit spec ID (00A02Dh)		
	008h	13h	unit sw version (00010xh)		
	00Ch	D4h	unit dependent directory offset		

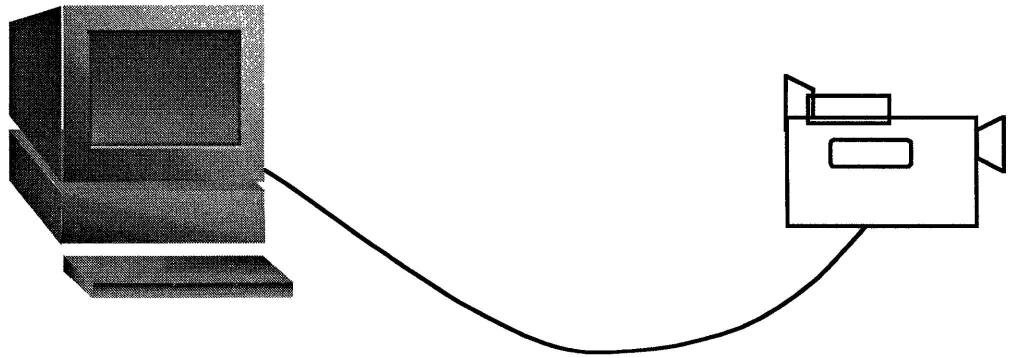
Unit Directory

X = 0 for Camera 1.04
1 for Camera 1.20

	Offset	0-7	6-15	16-23	24-31
Unit Dependent Info	000h	unit dep info length		CRC	
	004h	40h	command regs base		
	008h	81h	number of quadlets to vendor name leaf		
	00Ch	82h	number of quadlets to model name leaf		

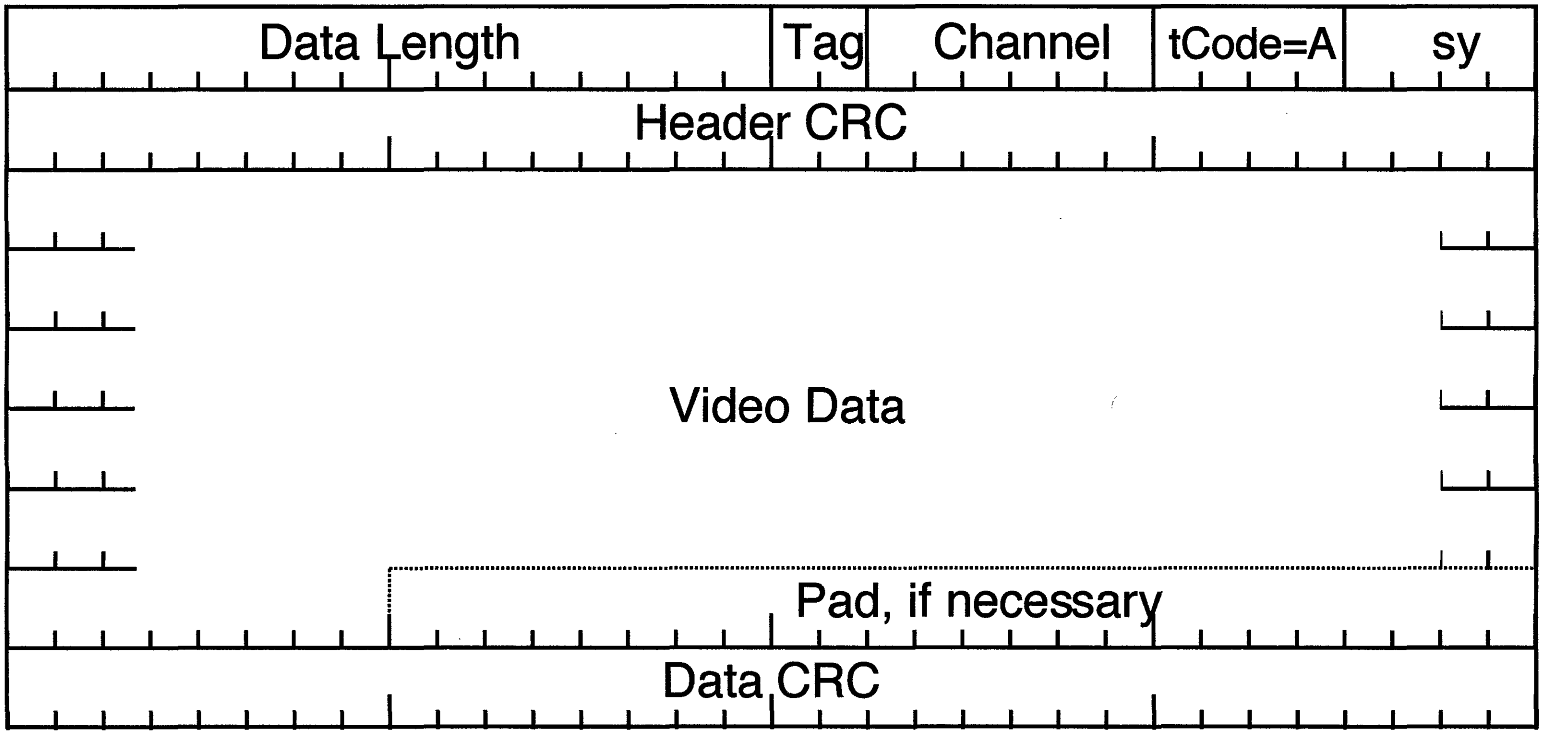
Unit Dependent Directory

Camera Operations



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



Digital Camera Initialize
For more information

Offset	Name	Field	Bit	Description
Offset	INITIALIZE	Initialize	[0]	If assert this bit, Camera will re-set to initial (factory setting value) state.
		-	[1..31]	Reserved (all zero)

Bits 0-7	Bits 8-15	Bits 16-23	Bits 24-31
	Reserved		

Digital Camera - Format Inquiry
For more information

Offset	Name	Field	Bit	Description
100h	V FORMAT INQ	Format x	[0..7]	Defined below
		-	[8..31]	Reserved (All zero)

Bits 0-7	Bits 8-15	Bits 16-23	Bits 24-31
Format	Reserved		

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

Offset	Name	Field	Bit	Description
180h	V MODE INQ 0 (format 0)	Mode 0	[0]	160 X 120 YUV(4:4:4) Mode (24 bit/pixel)
		Mode 1	[1]	320 X240 YUV(4:2:2) Mode (16 bit/pixel)
		Mode 2	[2]	640 X 480 YUV(4:1:1) Mode (12 bit/pixel)
		Mode 3	[3]	640 X 480 YUV(4:2:2) Mode (16 bit/pixel)
		Mode 4	[4]	640 X 480 RGB Mode (24 bit/pixel)
		Mode 5	[5]	640 X 480 Y (Mono) Mode (8 bit/pixel)
		Mode x	[6..7]	Reserved for another Mode
		-	[8..31]	Reserved (All zero)

Bits 0-7	Bits 8-15	Bits 16-23	Bits 24-31
V Mode	Reserved		

Digital Camera - Mode Inquiry

For more information

Offset	Name	Field	Bit	Description
184h	V MODE INQ 1 (format 1)	Mode 0	[0]	800 x 600 YUV(4:2:2) Mode (16 bit/pixel)
		Mode 1	[1]	800 x 600 RGB Mode (24 bit/pixel)
		Mode 2	[2]	800 x 600 Y (Mono) Mode (8 bit/pixel)
		Mode 3	[3]	1024 x 768 YUV(4:2:2) Mode (16 bit/pixel)
		Mode 4	[4]	1024 x 768 RGB Mode (24 bit/pixel)
		Mode 5	[5]	1024 x 768 Y (Mono) Mode (8 bit/pixel)
		Mode x	[6..7]	Reserved for another Mode
		-	[8..31]	Reserved (All zero)

Bits 0-7	Bits 8-15	Bits 16-23	Bits 24-31
V Mode	Reserved		

Digital Camera - Mode Inquiry
For more information

Offset	Name	Field	Bit	Description
188h	V MODE INQ 2 (format 2)	Mode 0	[0]	1280 x 960 YUV(4:2:2) Mode (16 bit/pixel)
		Mode 1	[1]	1280 x 960 RGB Mode (24 bit/pixel)
		Mode 2	[2]	1280 x 960 Y (Mono) Mode (8 bit/pixel)
		Mode 3	[3]	1600 x 1200 YUV(4:2:2) Mode (16 bit/pixel)
		Mode 4	[4]	1600 x 1200 RGB Mode (24 bit/pixel)
		Mode 5	[5]	1600 x 1200 Y (Mono) Mode (8 bit/pixel)
		Mode x	[6..7]	Reserved for another Mode
		-	[8..31]	Reserved (All zero)

Bits 0-7	Bits 8-15	Bits 16-23	Bits 24-31
V Mode	Reserved		

Digital Camera - Mode Inquiry
For more information

Offset	Name	Field	Bit	Description
198h	V MODE INQ 6 (format 6)	Mode 0	[0]	EXIF format
		Mode x	[6..7]	Reserved for another Mode
		-	[8..31]	Reserved (All zero)

Bits 0-7	Bits 8-15	Bits 16-23	Bits 24-31
V Mode	Reserved		

Digital Camera - Mode Inquiry
For more information

Offset	Name	Field	Bit	Description
19Ch	V MODE INQ 7 (format 7)	Mode 0	[0]	Format 7, Mode 0
		Mode 1	[1]	Format 7, Mode 1
		Mode 2	[2]	Format 7, Mode 2
		Mode 3	[3]	Format 7, Mode 3
		Mode 4	[4]	Format 7, Mode 4
		Mode 5	[5]	Format 7, Mode 5
		Mode 6	[6]	Format 7, Mode 6
		Mode 7	[7]	Format 7, Mode 7
		-	[8..31]	Reserved (All zero)

Bits 0-7	Bits 8-15	Bits 16-23	Bits 24-31
V Mode	Reserved		

Digital Camera - Frame Rate Inquiry Register

For more information

0	1	2	3	4	5	6	7	8 - 31 reserved

Bit	0	1.875 frames per second	
	1	3.75 fps	
	2	7.5 fps	
	3	15 fps	
	4	30 fps	
	5	60 fps	
Offset		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

0-7	8-15	16-23	24-31
a		c o m	mem

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

Offset	0-7	8-15	16-23	24-31
404h	b e s w h a m r	g l f d j	Reserved (all zero)	
408h	z p t k	n q	Reserved (all zero)	

Inquiry Register for Feature Presence - Definitions

For more information

B	Brightness	1 = Feature control is available
E	Exposure	0 = Feature control is not available
S	Sharpness	
W	Whiteness	
H	Hue	
A	Saturation	
M	Gamma	
R	Shutter	
G	Gain	
I	Iris	
F	Focus	
D	Temperature	
J	Trigger	
Z	Zoom	
P	Pan	
T	Tilt	
K	Optical filter	
N	Format 6 Size	
Q	Format 6 Quality	

Inquiry Register for Feature Elements

For more information

Offset	0-7						8-15	16-23	24-31
See below	p		r	o	a	m	min value		max value

FOR EACH REGISTER/FEATURE LISTED BELOW

Inquiry Register for Feature Elements - Offsets

For more information

Offset	Name
500h	BRIGHTNESS INQ
504h	EXPOSURE INQ
508h	SHARPNESS INQ
50Ch	WHITE BAL INQ
510h	HUE INQ
514h	SATURATION INQ
518h	GAMMA INQ
51Ch	SHUTTER INQ
520h	GAIN INQ
524h	IRIS INQ
528h	FOCUS INQ
52Ch : 57Ch	Reserved for other FEATURE HI INQ
580h	ZOOM INQ
584h	PAN INQ
588h	TILT INQ
58Ch : 5FCh	Reserved for other FEATURE LO INQ

Inquiry Register for Feature Elements - Definitions

For more information

P	Feature is Present
R	Capability of Reading this feature
O	Capability of turning feature On or Off
A	Capability of Automode
M	Capability of Manual mode
Min value	Minimum value of this feature
Max value	Maximum value of this feature

Control and Status Registers for Cameras
For more information

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

CSR for Camera(continued)
For more information

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

Status and Control Register for Feature
For more information

<table>
<tr><td>Offset</td><td colspan="4">0-7</td><td>8-15</td><td colspan="2">16-23</td><td>24-31</td></tr>
<tr><td>See below</td><td>p</td><td></td><td>o</td><td>a</td><td colspan="2">reserved/u value</td><td colspan="2">value/v value</td></tr>
</table>

Status and Control Register for Features - Offsets

For more information

Offset	Name
800h	BRIGHTNESS
804h	EXPOSURE
808h	SHARPNESS
80Ch	WHITE BAL
810h	HUE
814h	SATURATION
818h	GAMMA
81Ch	SHUTTER
820h	GAIN
824h	IRIS
828h	FOCUS
82Ch : 87Ch	Reserved for other FEATURE HI
880h	ZOOM
884h	PAN
888h	TILT
88Ch : 8FCh	Reserved for other FEATURE LO

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 as required of a 61883 node?
5. How is control done with camera 1.20 compliant nodes?

Notes

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

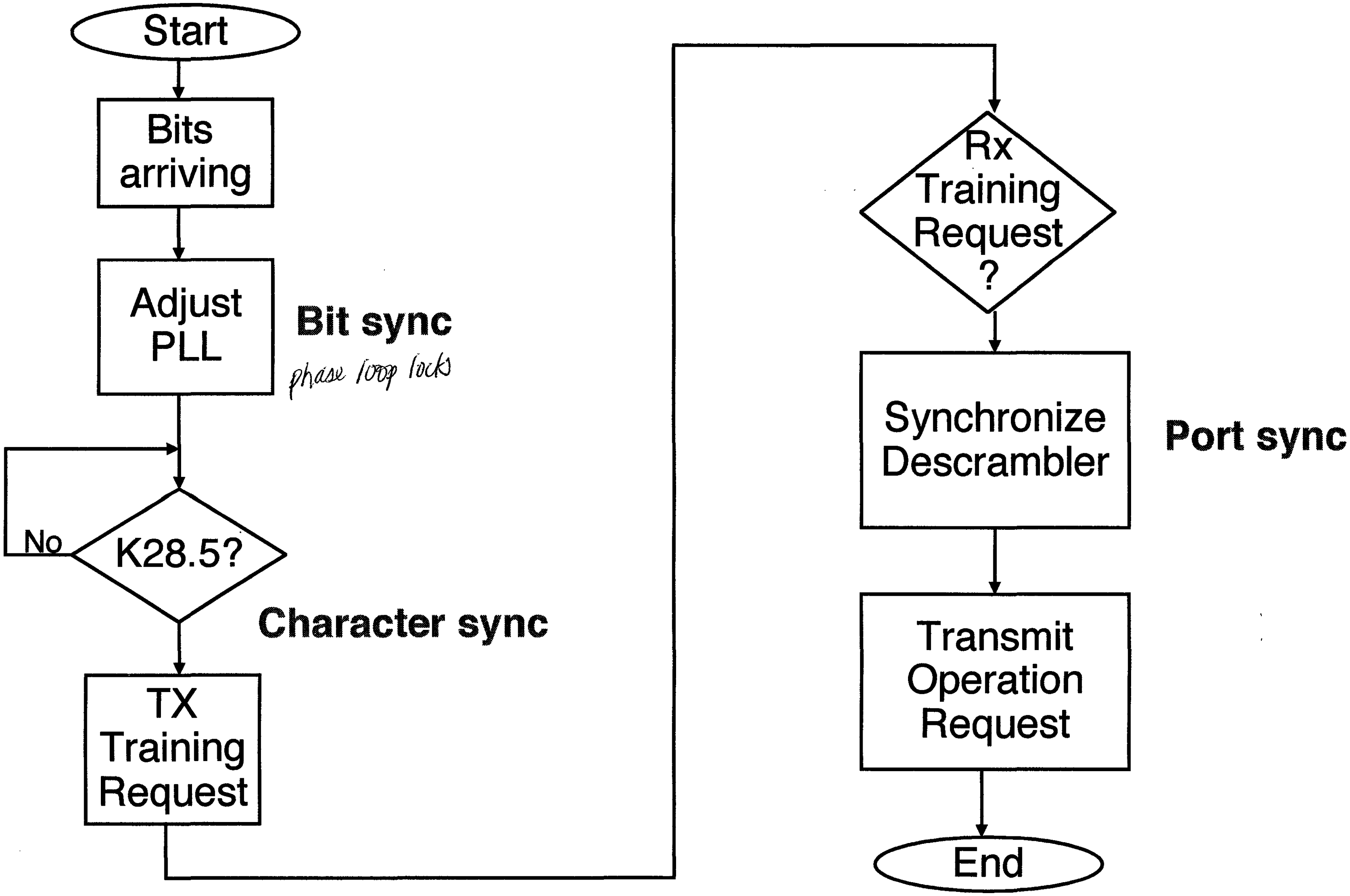
- DS encoding Data-Strobe
- DC coupling
- Maximum speed - 400 Mbps
- Gaps for fairness and priority

1394b

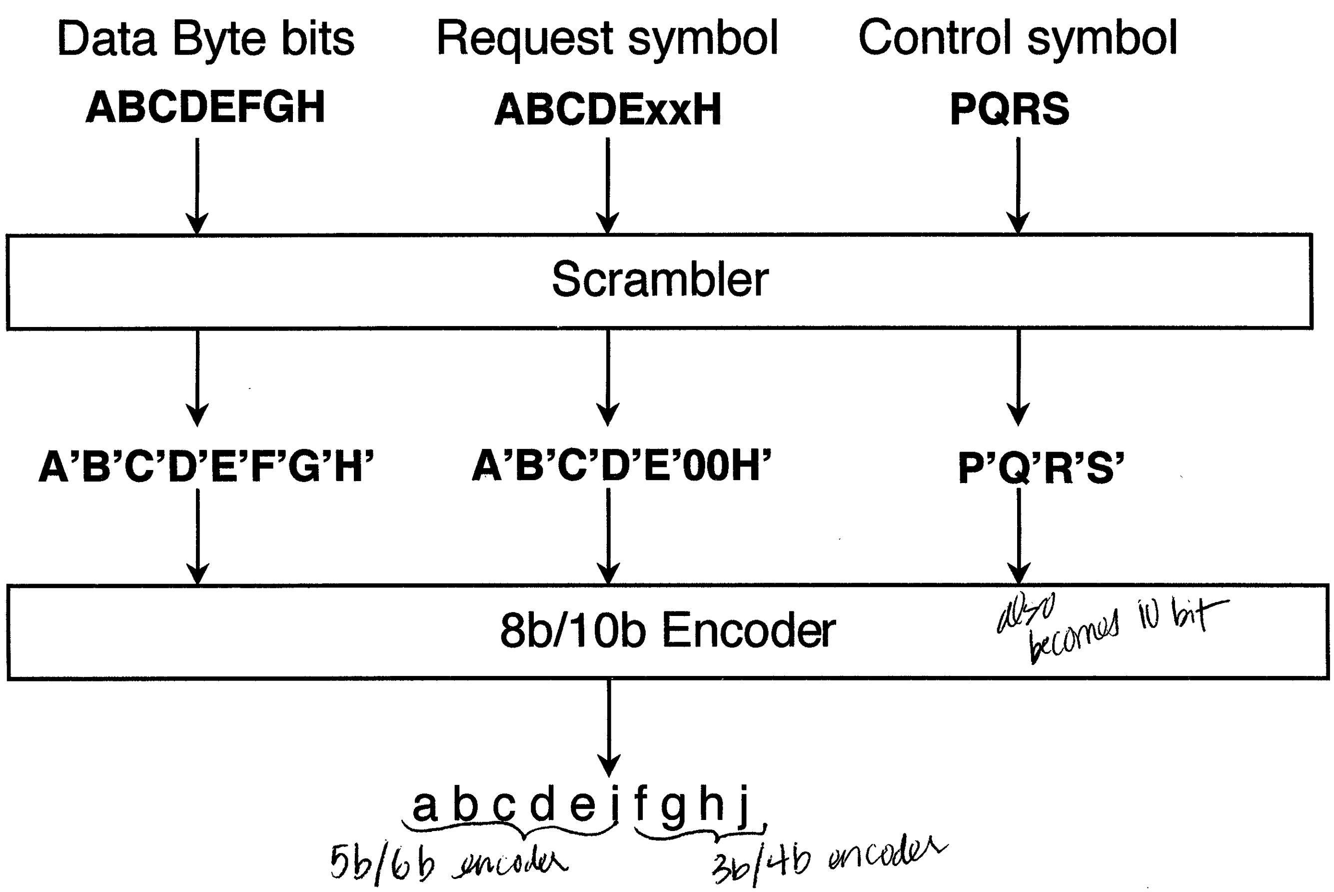
- 8b/10b encoding
- AC or DC coupled
- Up to 3200 Mbps
- May be Bilingual - compatible with 1394-1995
- Media defined
  - Category 5 UTP
  - Hard Polymer Clad Fiber
  - Plastic Optic Fiber
  - Glass Optic Fiber
  - 1394-1995 style cables
- Full Duplex
- No gaps

8b/10b Encoding

Synchronization



Data Encoding



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

Bits are transmitted "a" first

Request Signaling

can't use 1,0,Z
any may be
AC coupling -
can only use 1,0
see 17-11

Request	Symbol ABCDExxH
Training	0000 0xx0
Disable Notify	0010 0xx0
Child Notify, Ident done	0100 0xx0
Operation	0110 0xx0
Standby	1000 0xx0
Suspend	1010 0xx0
Parent Notify	1100 0xx0
Legacy request	1110 0xx0

Control Symbol Mapping

Control Token	Control Symbol PQRS	
	RD<0 Running Disparity	RD>0
Async Start	0000	
Cycle Start even	0001	
Cycle Start odd	0010	
Attach request/ Arb context	0011	
Speeda	0100	
Data end	0101	
Data null	0110	
Speedb	0111	
Grant	1000	
Data prefix	1010	1001
Reserved	1011	
Speedc	1100	
ARBRST even	1101	
ARBRST odd	1110	
Bus Reset	1111	

DC Balance
Disparity

What /Why Disparity

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

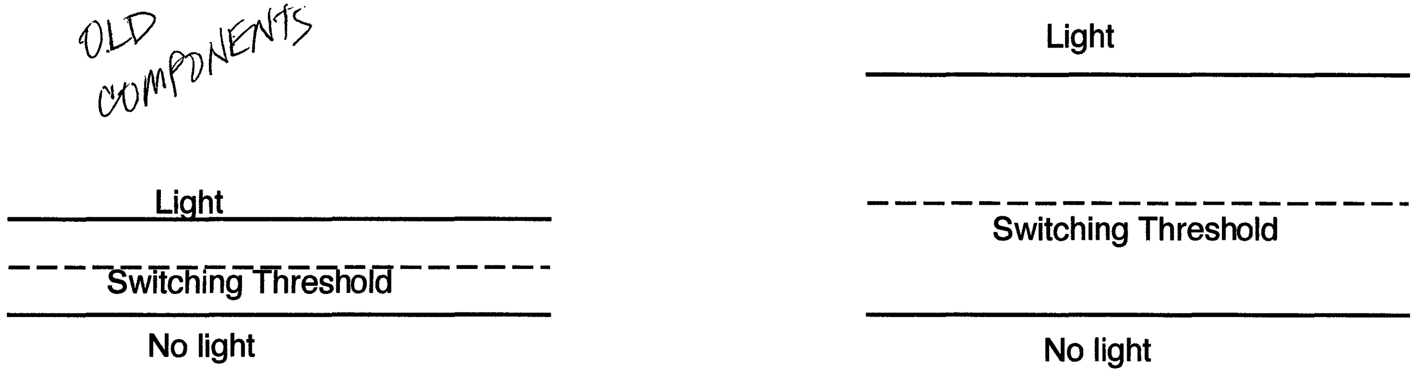
To maintain circuits in linear part of their operation, we must establish a bias or DC level

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.

NEW COMPONENTS

OLD COMPONENTS



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	Running Disparity -1
This character D9.6 Select either 100101 1101 (+2) or 100101 0010 (-2)	Select +2 since @ negative disparity
Now running disparity	+1

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

Speed Signaling

Method

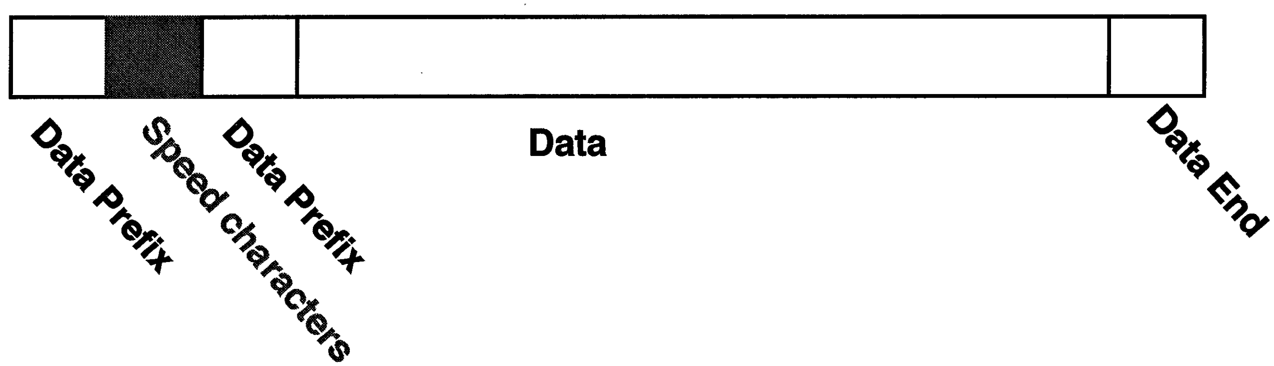
1394-1995 speed signaling not changed for DS ports

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

a or b depending on 1394a or 1394b

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

Packet Speed as a function of port speed

	Speedc	Speed(x)	Speedc
Same	0	1	0
1/2	1	1	0
1/4	2	1	1
1/8	3	1	4
1/16	4	1	11
1/32	5	1	26

Payload Speed Matching

Padding

Packet Speed as a function of Port Speed	Data	Control
Same	D	C*1
1/2	D + P	C*2
1/4	D + (P*3)	C*4
1/8	D + (P*7)	C*8
1/16	D + (P*15)	C*16
1/32	D + (P*31)	C*32

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

bilingual PHY (1394a & b)

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

Isochronous Priority

Request Name	Priority Level	Comment
Isochronous Odd	3 (highest) if last cycle start was odd, else 2	Used if last cycle start was odd and the packet is intended to transmit in the current cycle
Isochronous Even	2 if the last cycle start was odd, else 3	
Isochronous None	1	

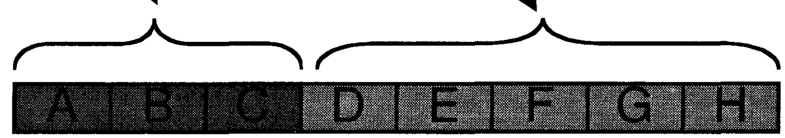
Arbitration Requests

Asynchronous

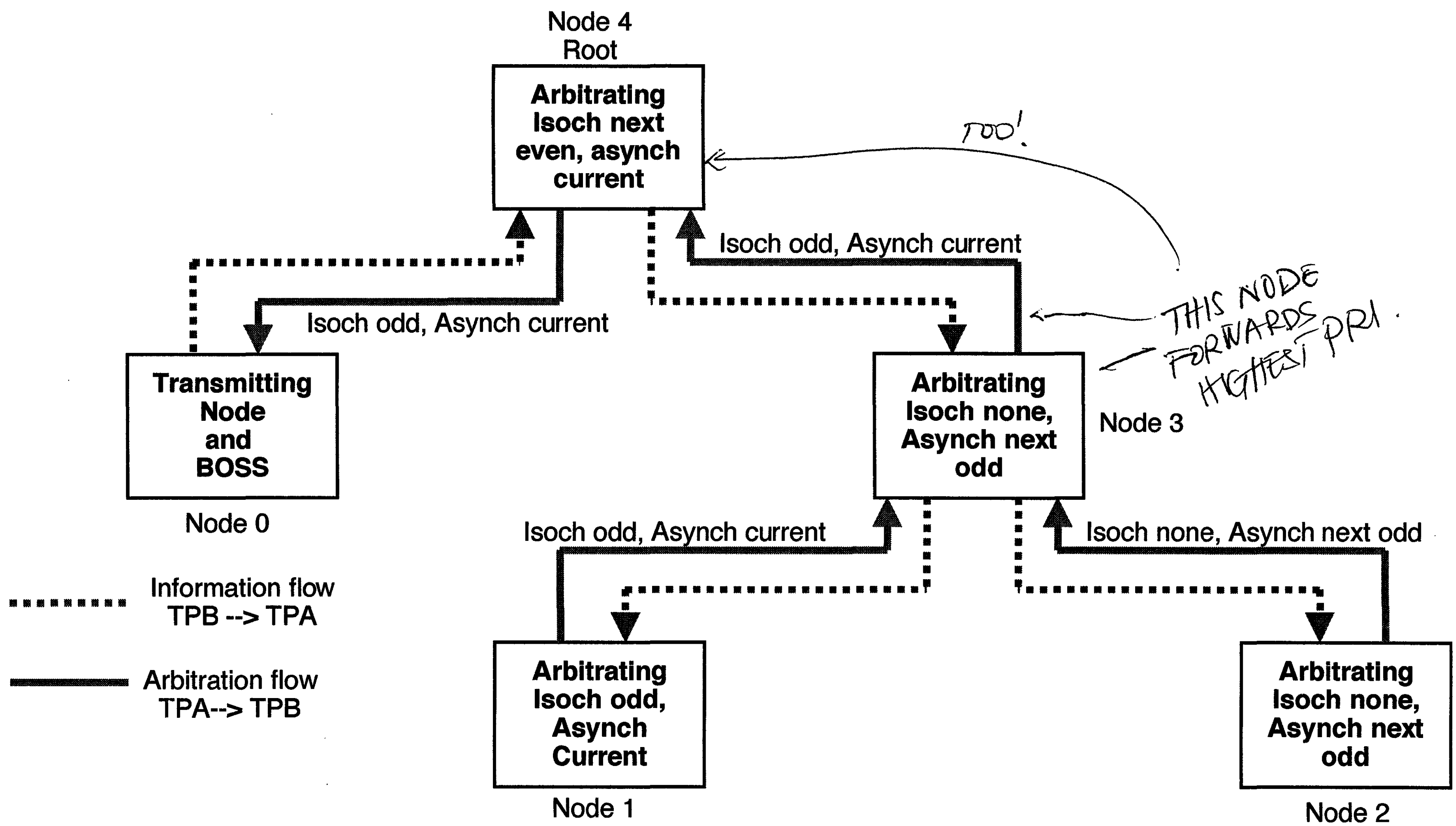
Request	Request symbol ABC
Reserved	000
Current	001
Next even	010
Cycle start request	011
None odd	100
Next odd	101
None even	110
Reserved	111

Isochronous

Request	Request symbol DEFGH
Not used	00xx0
Isochronous none	01xx0
Isochronous even	10xx0
Isochronous odd	11xx0
Reserved	00xx1
Reserved	01xx1
Reserved	10xx1
Not used	11xx1



BOSS Arbitration



Last ARBRST was even
Last Cycle Start was odd

Loop Free Build

Interconnects

Interconnects

Media	Distance	Max Speed	
STP	4.5 m	S1600	Shielded Twisted Pair
MMF	100 m	S1600	Glass Multi-mode Fiber, 50 micron
POF	50 m	S200	Plastic Optical Fiber
HPCF	100 m	S200	Hard Polymer Clad Fiber
UTP	100 m	S100	Unshielded Twisted Pair, Category 5

STP

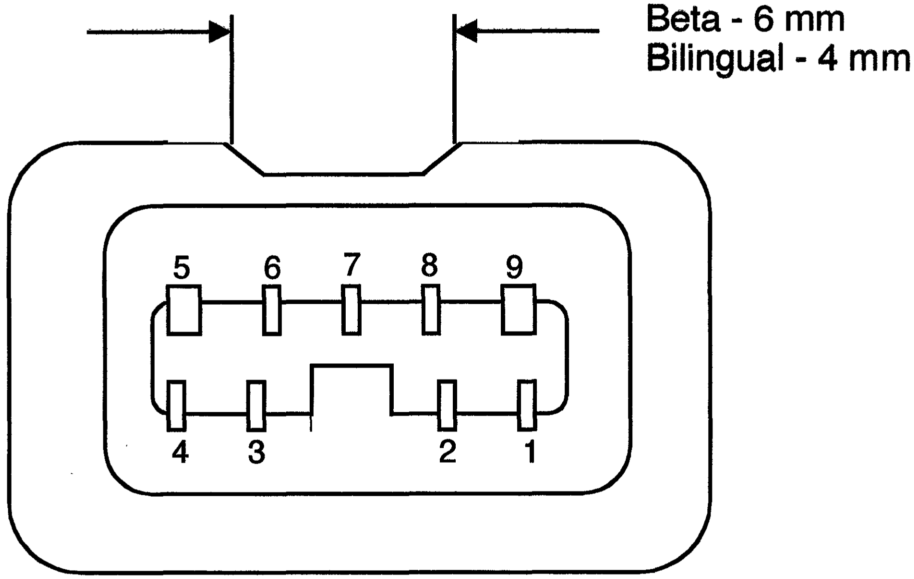
Similar cable to 1394a but with different connectors

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



Viewed from front of plug face

Pin	Connection
1	TPB*
2	TPB
3	TPA*
4	TPA
5	TPA return
6	Power
7	Reserved
8	Power Ground
9	TPB return

STP Cable Assemblies

Plug 1	Plug 2	Reference
Beta	Beta	1394b
1394-1995 6 pin	Bilingual	1394-1995
1394a 4 pin	Bilingual	1394a

4 4

6 6

46 6

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

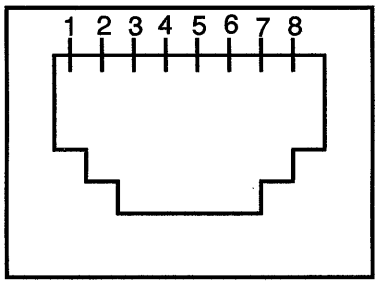
225 micron graded index multimode fiber
Distance is 100 meters

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



Pin	Signal
1	TPB
2	TPB*
3	
4	
5	
6	
7	TPA
8	TPA*

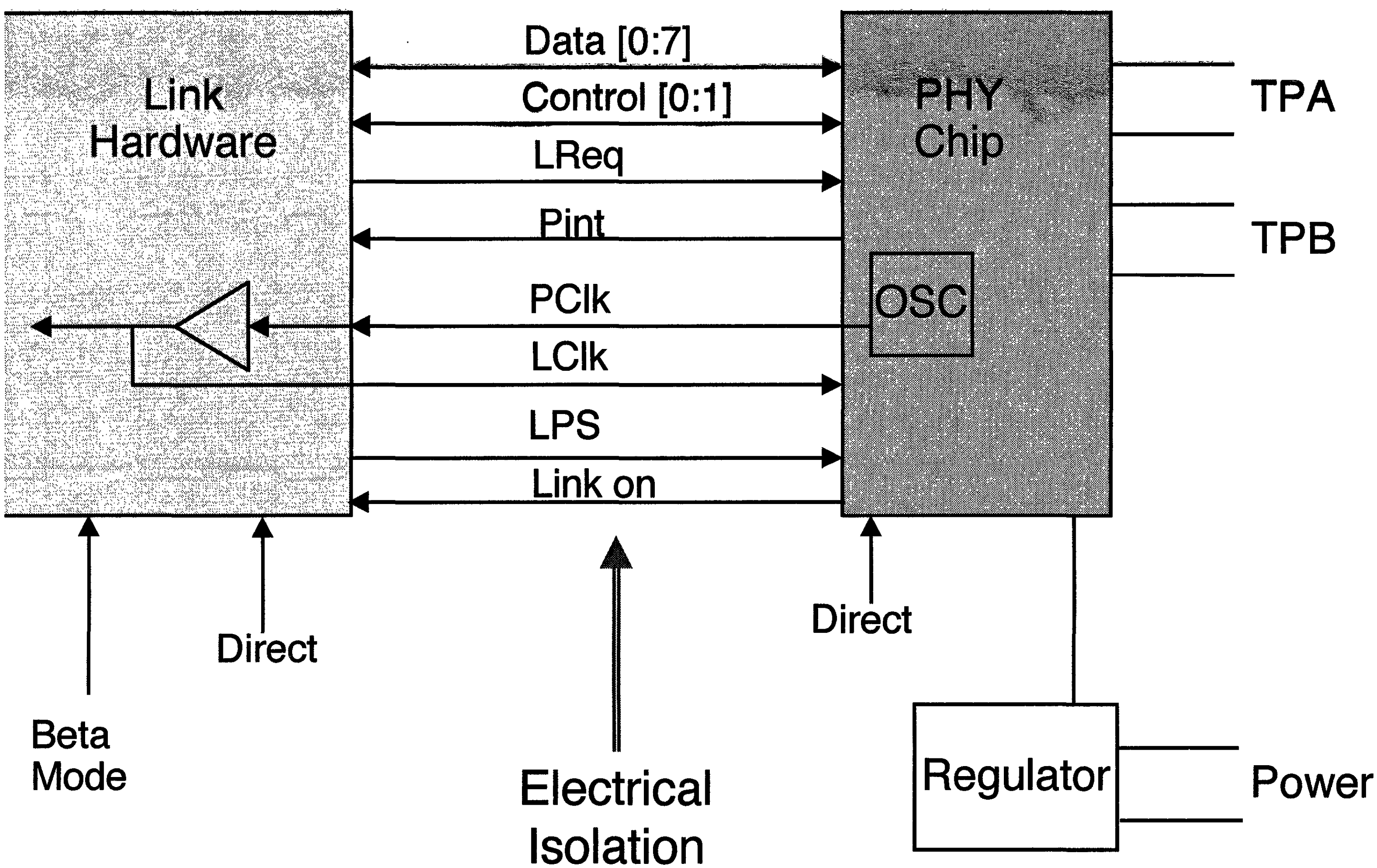
PHY-Link Interface

1394b Remote Command Packet

00	PHY ID	00	Type=8	e-cmd	port	0000 0000	cmd
Logical Inverse of first quadlet							

Field	Description
PHY ID	Node for this command
Type	Extend PHY packet type (8 indicates command packet)
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



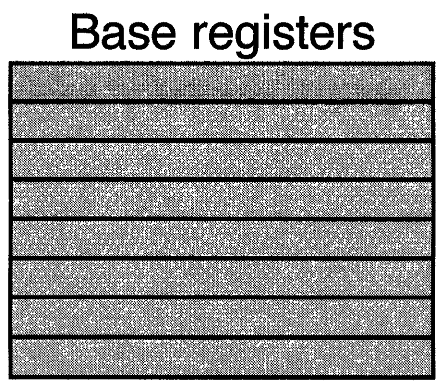
1394b Link/Phy Connection Definitions

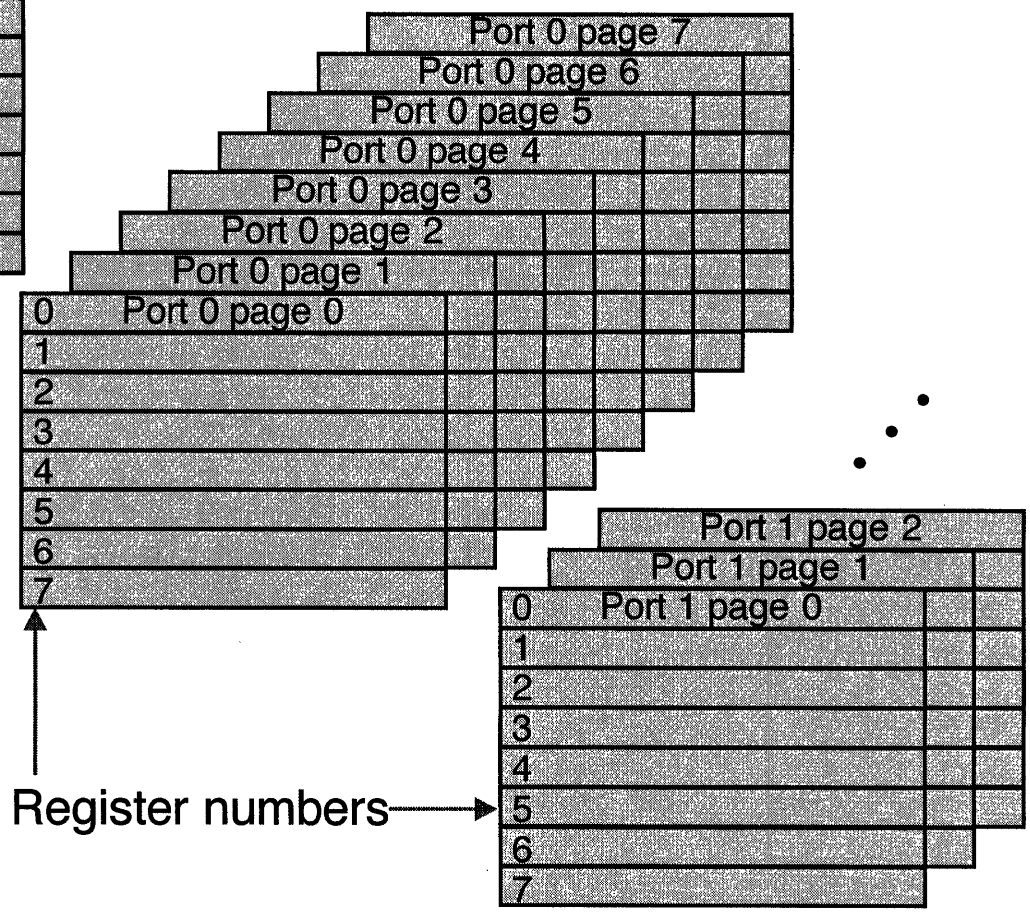
Data	[0:7] for S400
Control	What is the meaning of the data lines
LReq	Serial command to the Phy
PClk	98.304MHz clock, must be generated in PHY
LClk	PHY clock returned by link
LPS	Link power status
Link On	Commands Link to Power On
Direct	Indicates Link and Phy are directly connected
Pint	PHY interrupt to link

Extended PHY Registers

Bits	0	1	2	3	4	5	6	7
0000b	Physical ID						Root	CPS
0001b	RHB	IBR	Gap Count					
0010b	Extended = 7			Total Ports				
0011b	Max speed			Res	Delay			
0100b	L	C	Jitter			1394 Power class		
0101b	Watchdog	ISBR	Loop	Pwr fail	Timeout	Bias	enab acc	enab multi
0110b	Max legacy path speed			B-link	Reserved			Standby Reset
0111b	Page select			Res	Port select			
1000b	Register 0 (page select)							
1111b	Register 7 (page select)							

Extended Registers





Every port has 8 pages
Every page has 8 registers

Extended PHY registers - Page 0

	0	1	2	3	4	5	6	7
1000b	A Status		B Status		Ch	Con	RX OK	Dis
1001b	Negotiated speed			Int enbl	Fault	Standby fault	Disable Scramb	Beta only
1010b	DC con	Max port speed			LPP	Cable speed		
1011b	Conn Unreliable	Reserved			Beta Mode	Reserved		
1100b	Port Error							
1101b	Reserved					Loop Disable	Standby	Hard Disable
1110b	Reserved							
1111b	Reserved							

Extended Register - Page 0

AStat	0	invalid
	1	1
	2	0
	3	Z

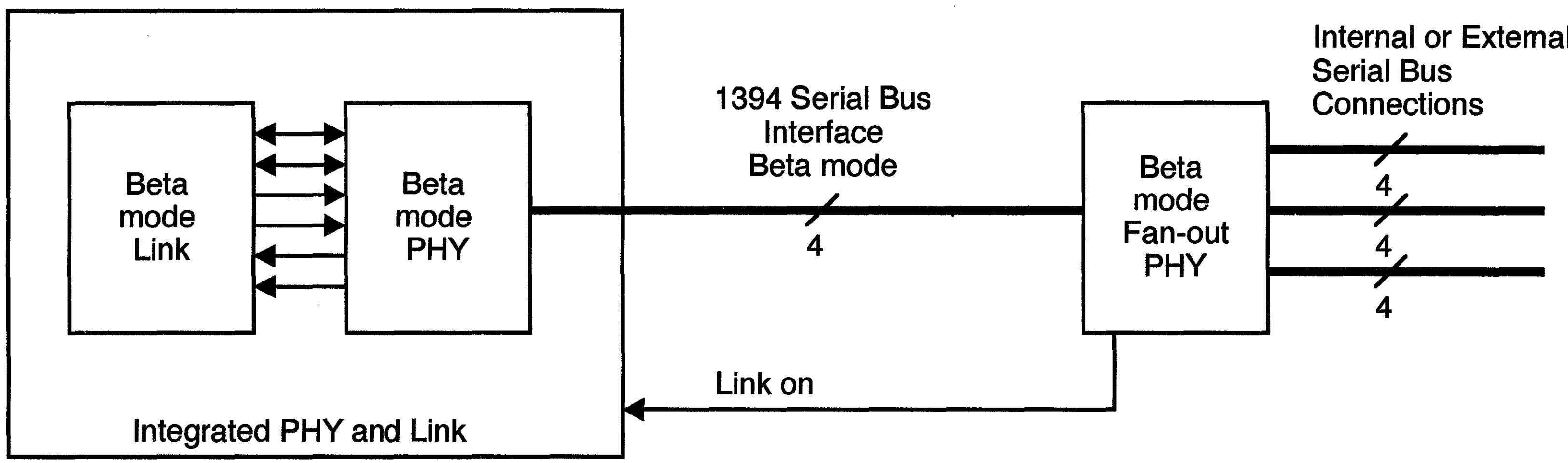
BStat	Valid only on DS port, Same encoding as AStat
Con	Connected and operating speed negotiation complete
RX OK	DS mode - Receiving a TPBias Beta mode - receiving a continuous electrically valid signal
Standby fault	Error was detected during transition to standby or active
Dis Scrambler	Used for test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

Extended PHY registers - Page 1

	0	1	2	3	4	5	6	7
1000b	Compliance level							
1001b	Reserved							
1010b	Vendor ID							
1011b								
1100b								
1101b	Product ID							
1110b								
1111b								

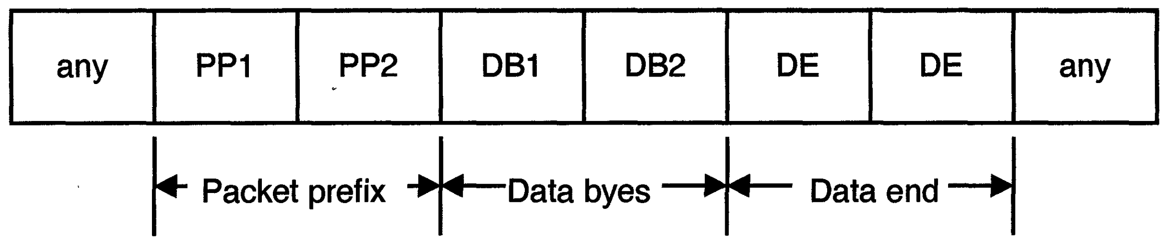
Compliance level		
	00h	Reserved
	01h	1394a
	02h	1394b
	all others	Reserved

PIL-FOP



Beta
mode
Link
Beta
mode
PHY
Integrated PHY and Link
1394 Serial Bus
Interface
Beta mode
4
Beta
mode
Fan-out
PHY
Internal or External
Serial Bus
Connections
4
4
4
Link on

PIL - FOP Interface packet



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.

Notes

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.

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) — **Video**

Streams (Asynchronous Commands to establish) — **SBP-2**

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)

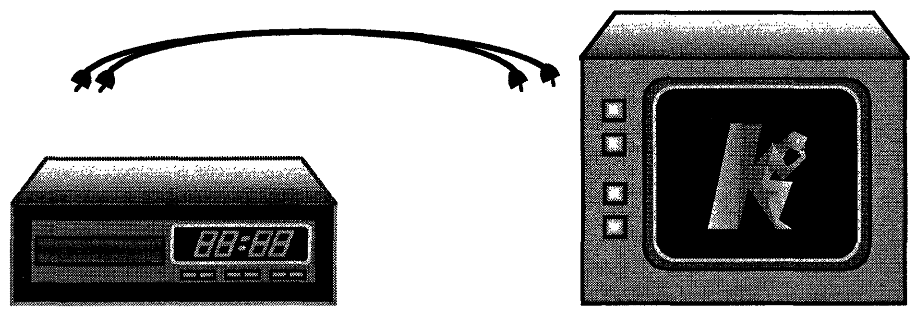
Listener

Input	Master	PCR (1)
Input	PCR	(1-32)

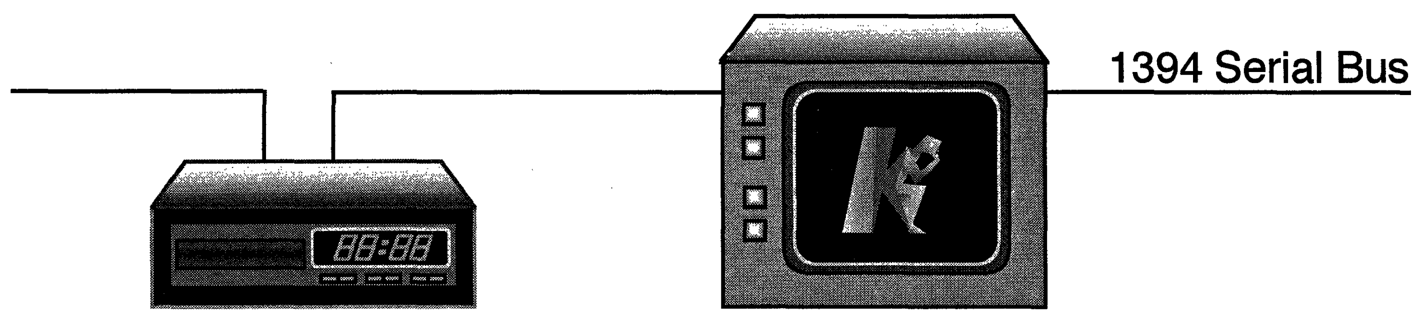
Isochronous Connection Management
Method 2

- 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

Output Master Plug

SPD	Broadcast Base	reserved	X Spd	Output Plugs

Spd

- 0 = S100
- 1 = S200
- 2 = S400
- 3 = S000 XSPD

X Spd

- 0 = S800
- 1 = S1600
- 2 = S3200
- 3 = Reserved

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

O	b	Point to Point	X Spd	Channel	Spd	Overhead	Payload

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

Spd	Reserved	Input Plugs

Input Plug

O	b	Point to Point	R	Channel	Reserved

All fields as defined in outplug plug registers

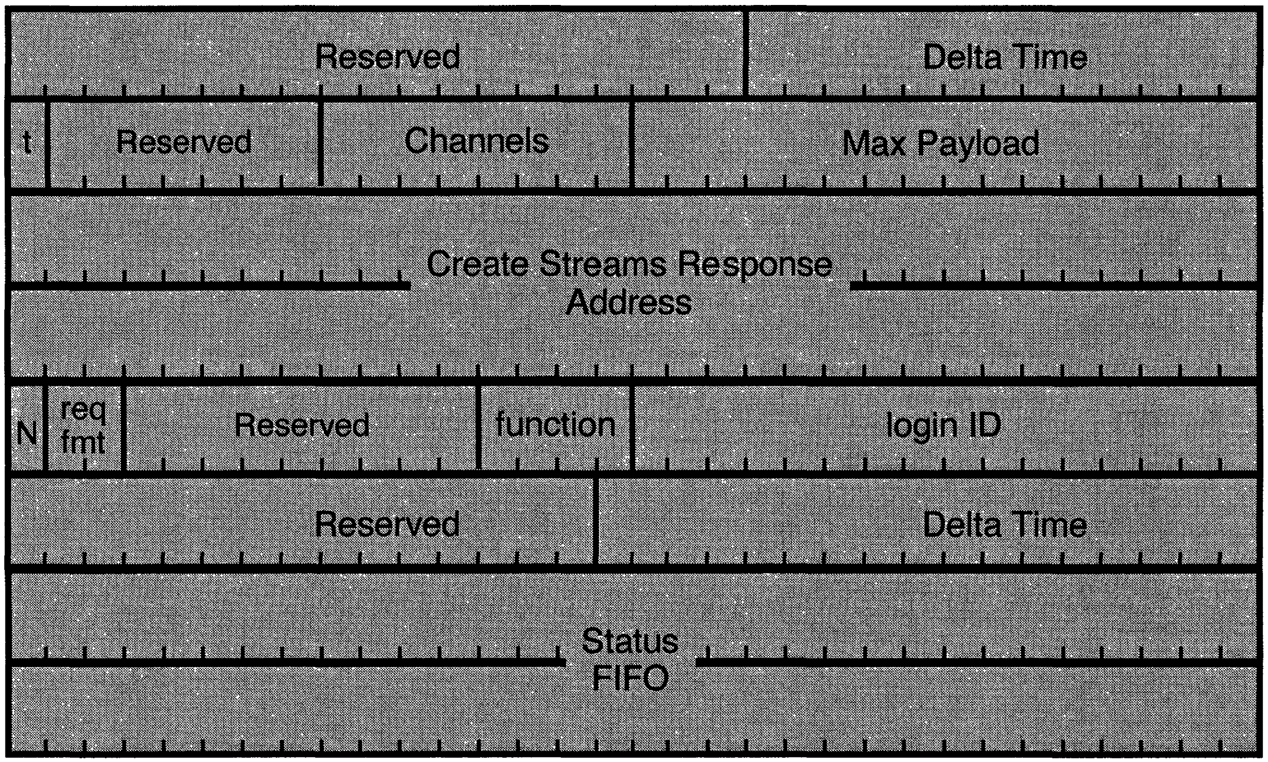
Isochronous Connection Management
Streams

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)



Isochronous Connection Management

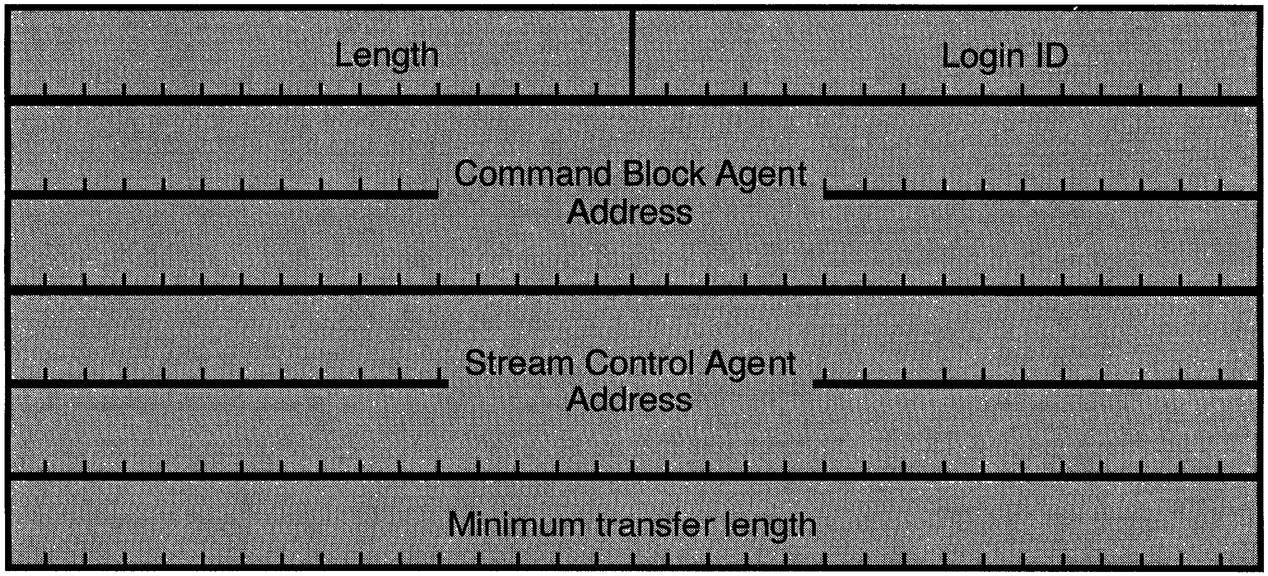
Delta Time	Range = 0 to 7999. Used to shift Isochronous cycles into the future up to 1 second
t	Talker
channels	Maximum number of Isochronous channels to be used
Max payload	Sum of data length of all channels for this talker per Isochronous cycle

v

This page intentionally left blank

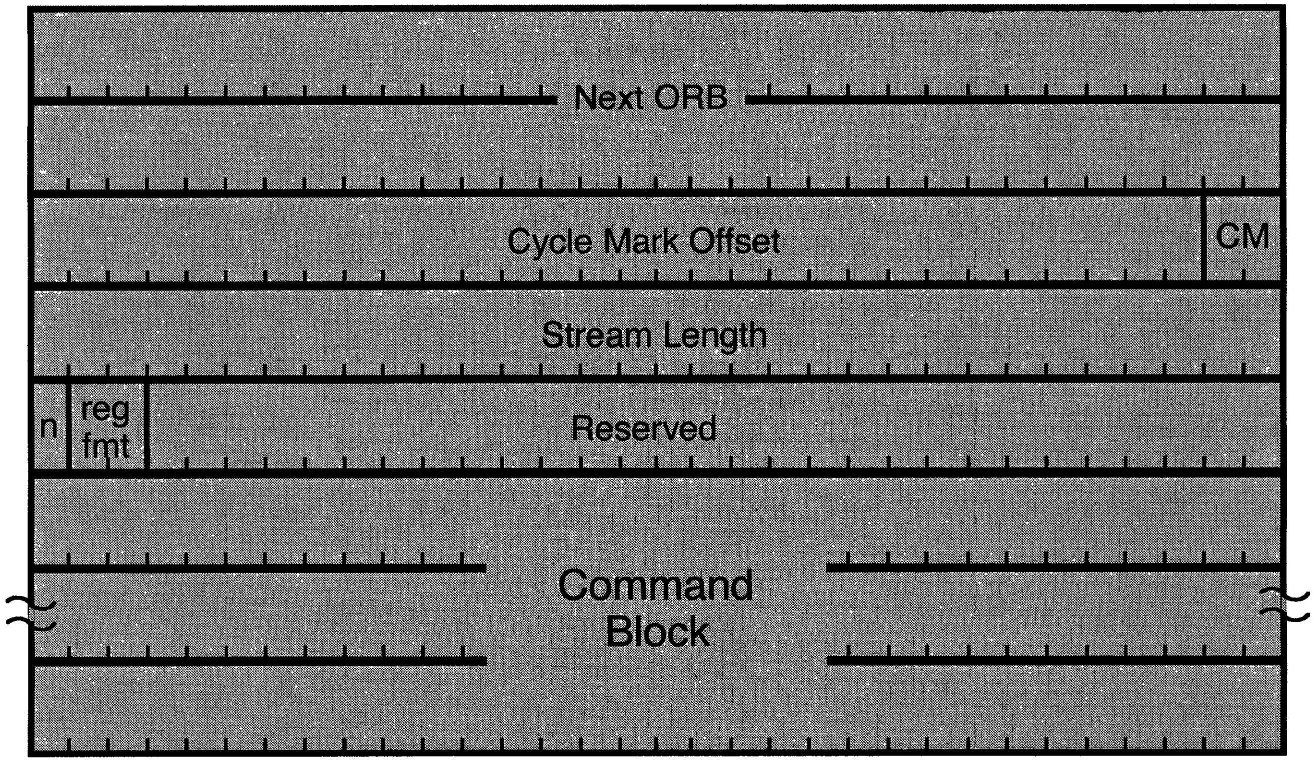
Isochronous Connection Management
Streams

Create Stream Response



Isochronous Connection Management
Streams

Stream Command Block ORB



^

Isochronous Connection Management

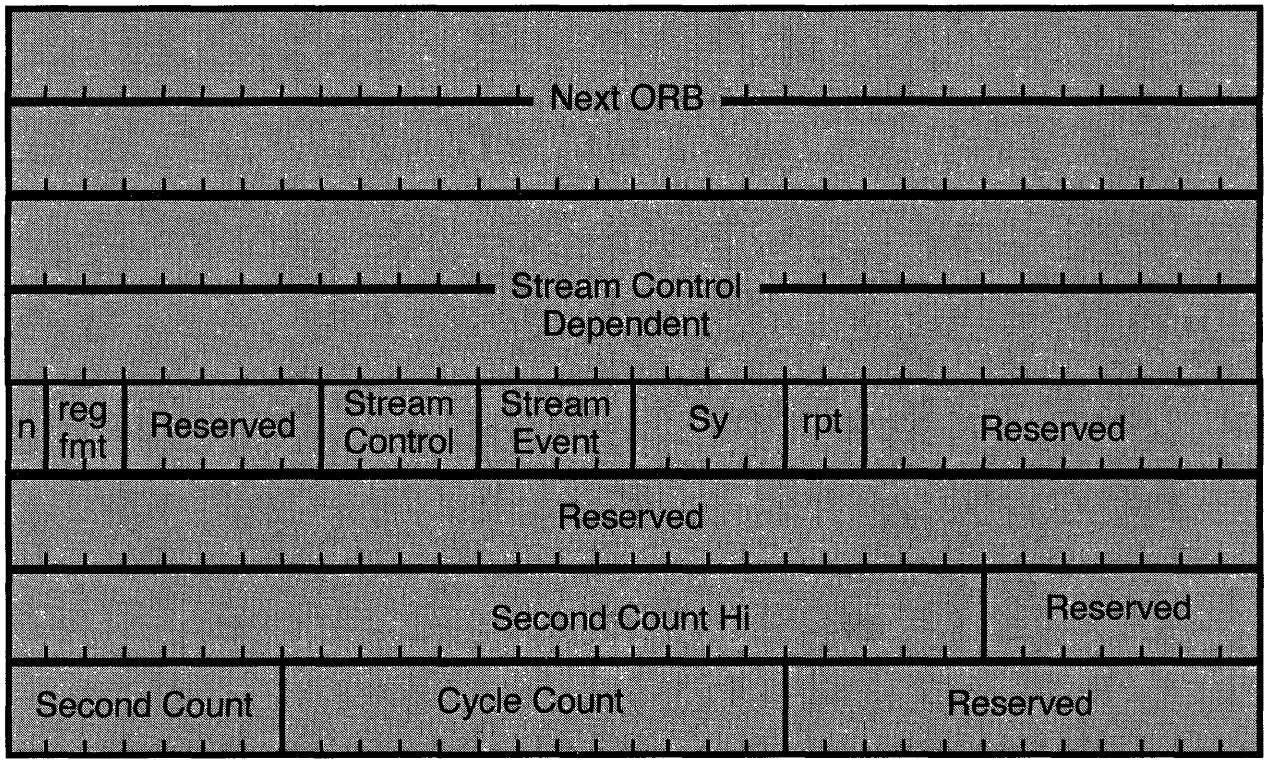
Cycle Mark offset and CM specify the location on the device of the first quadlet of ISO data

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

v

Isochronous Connection Management
Streams

Stream Control ORB



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

V

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

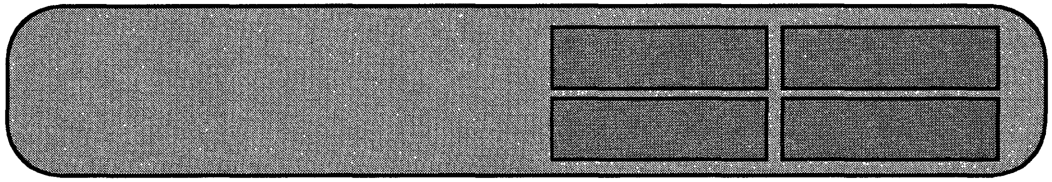
Device Bay Interface Specification

- 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

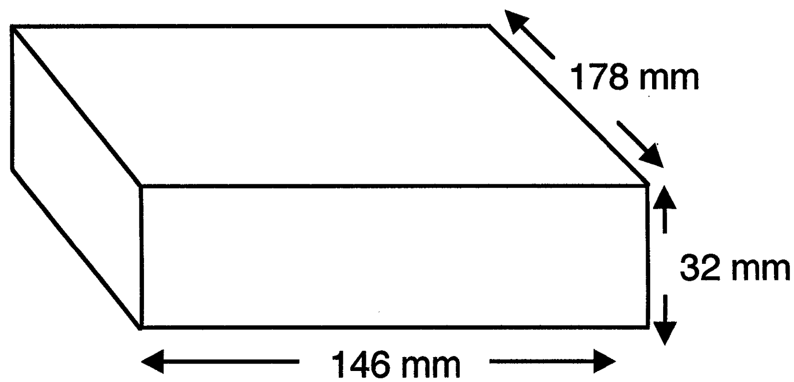
Non-Compliant Connector

- Devices that don't conform to USB/1394 or Power Requirements
- Batteries, etc.

Device Bay
Form Factors

<u>DB 32</u>

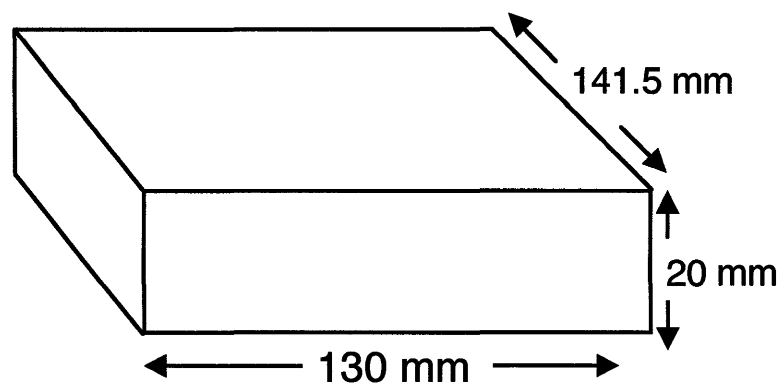
32.00 X 146.00 X 178.00 mm
1.260" high X 5.748" wide X 7.008" long



Device Bay
Form factors

DB 20

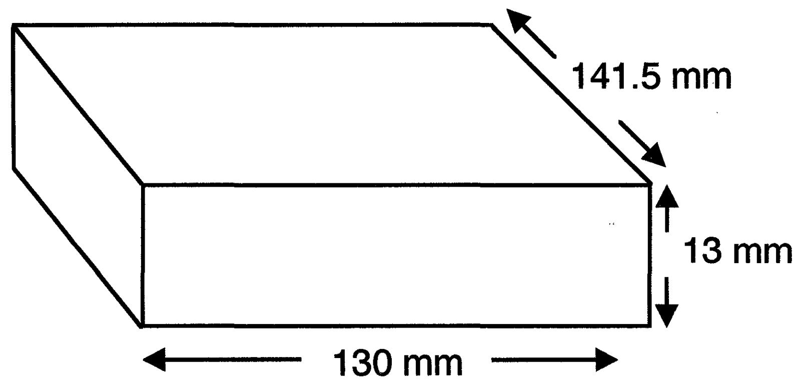
20.00 X 130.00 X 141.550 mm
.787" high X 5.118" wide X 5.571" long



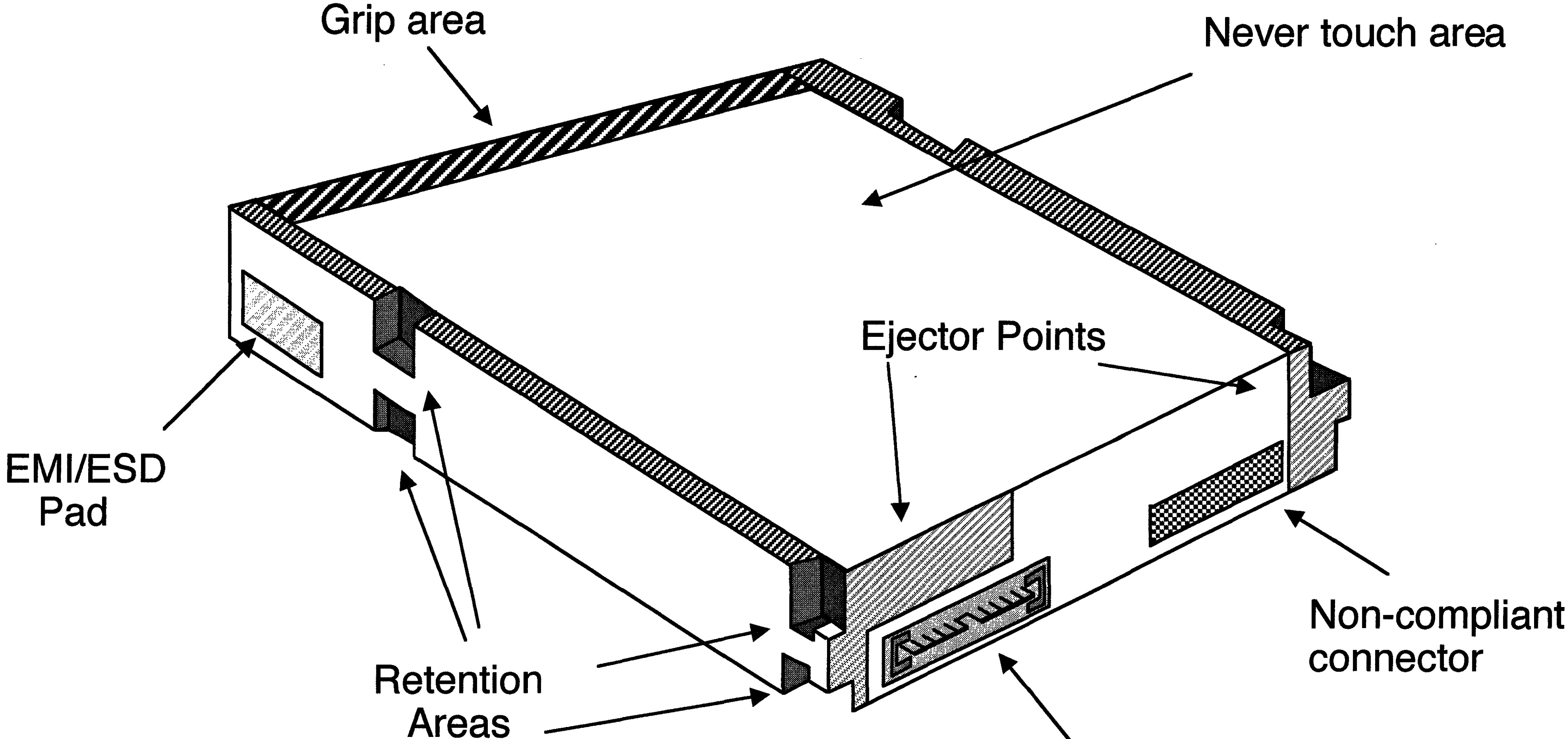
Device Bay
Form Factors

DB 13

13.00 X 130.00 X 141.50 mm
.512" high X 5.118" wide X 5.571" long

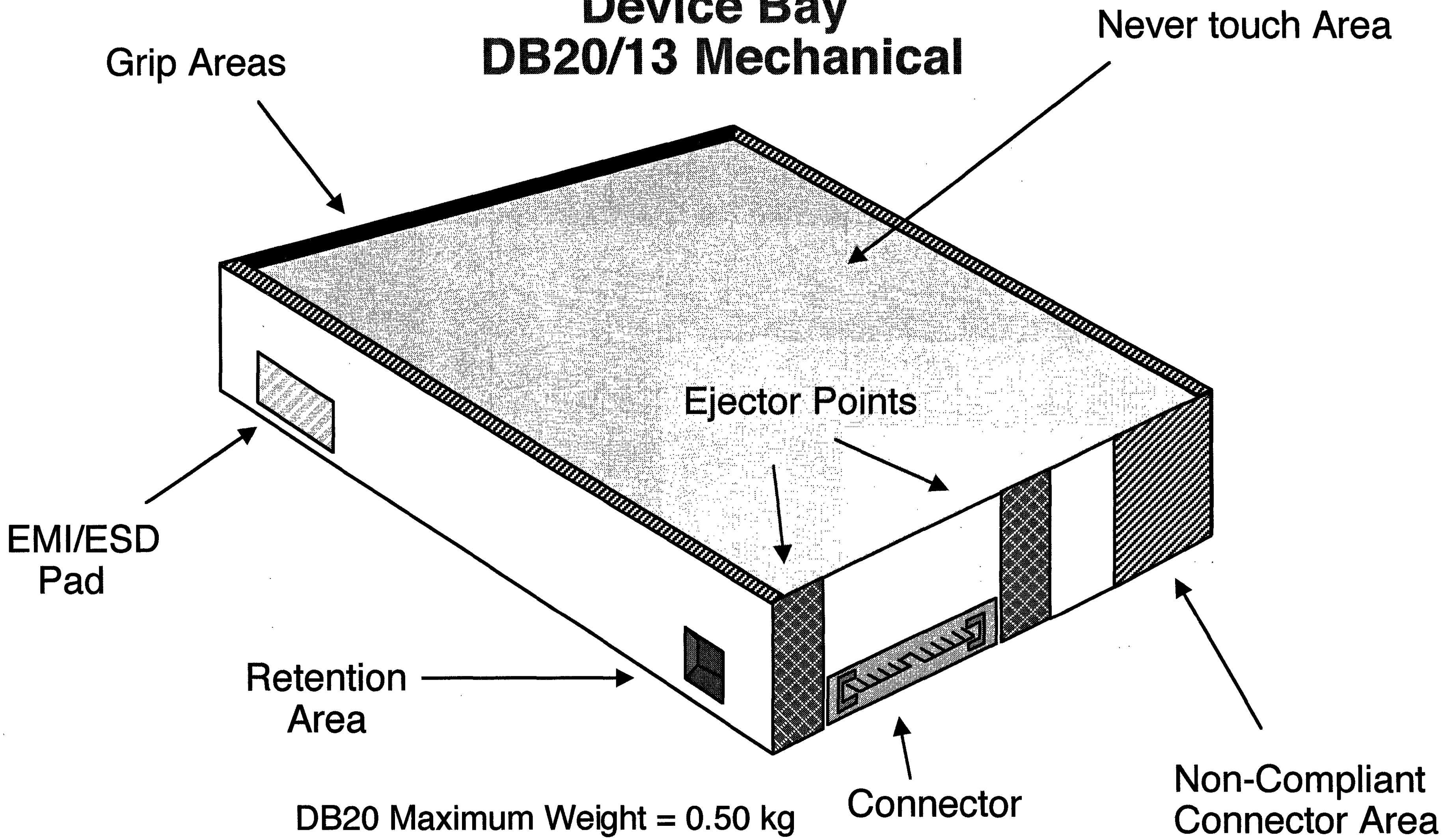


Device Bay
DB 32 Mechanical



Maximum Weight = 1.40 kg
3.08 pds

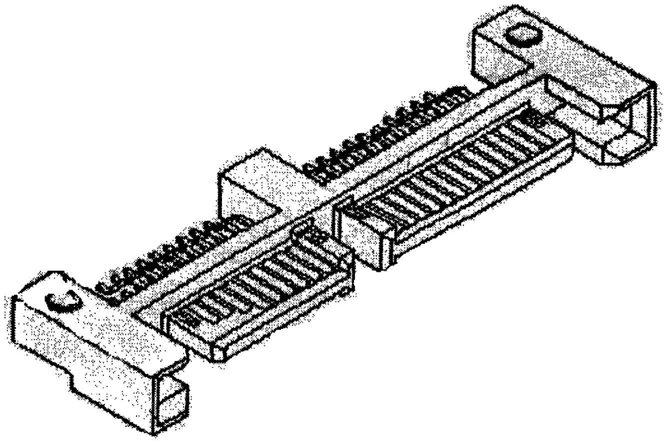
Device Bay
DB20/13 Mechanical



DB20 Maximum Weight = 0.50 kg
1.10 pds

DB13 Maximum Weight = 0.35 kg
0.77 pds

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

A14	Ground	Ground	A1
A15	Ground	(1394) TPA	A2
A16	Ground	(1394) TPA *	A3
A17	Ground	Ground	A4
A18	Ground	(1394) TPB	A5
A19	Ground	(1394) TPB *	A6
A20	Ground	Ground	A7
A21	Ground	(1394) PRSN#	A8
A22	Ground	DEV_ ACT#	A9
A23	Ground	(USB) PRSM #	A10
A24	Ground	(USB) D+	A11
A25	Ground	(USB) D-	A12
A26	Reserved	V id	A13

GAP

B10	Ground	V 3.3	B1
B11	Ground	V 3.3	B2
B12	Ground	V 3.3	B3
B13	Ground	V 3.3	B4
B14	Ground	V 3.3	B5
B15	Ground	V 3.3	B6
B16	V 5.0	V 12	B7
B17	V 5.0	V 12	B8
B18	V 5.0	V 12	B9

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

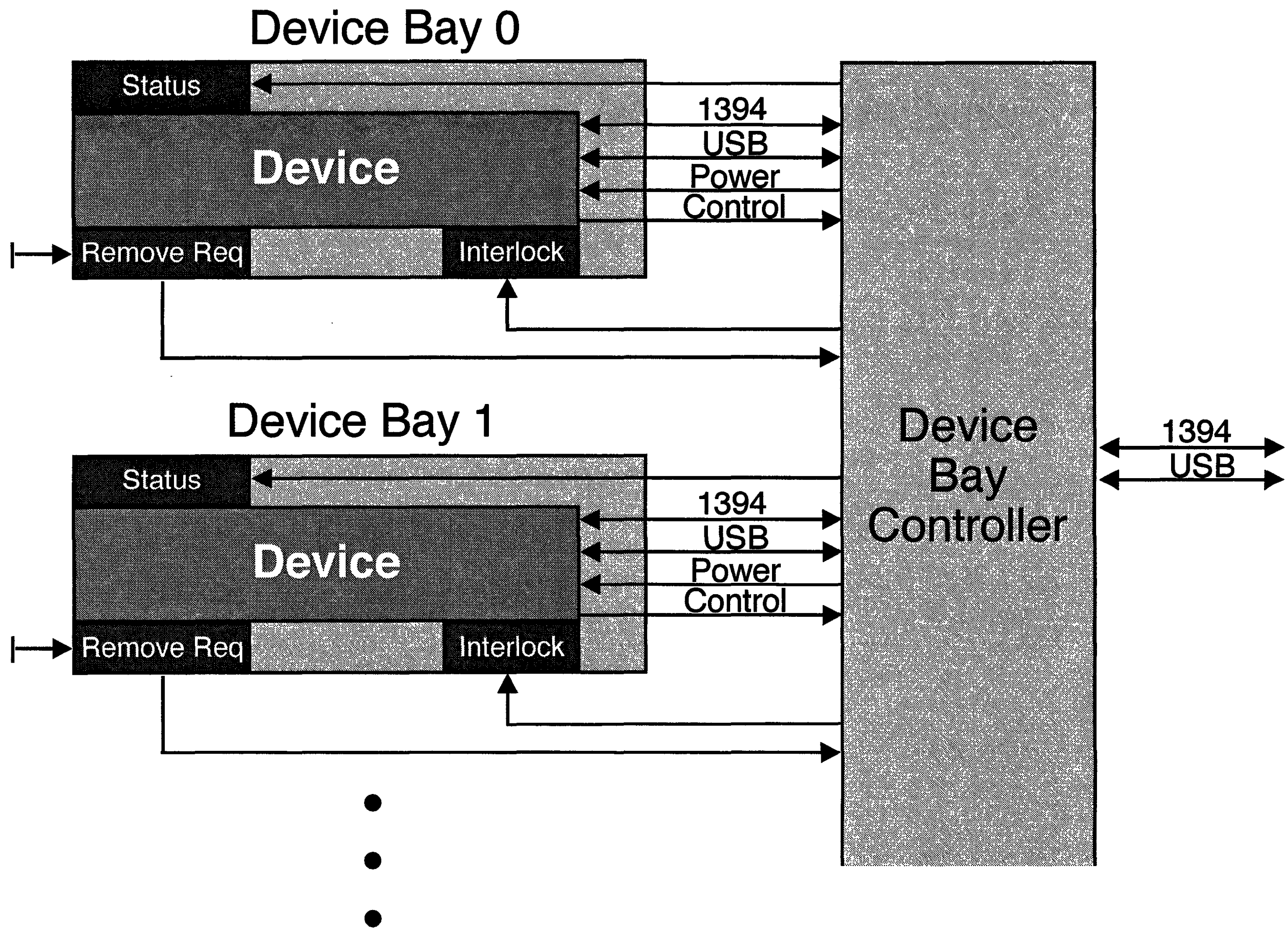
3 1394 PRSN# or USB PRSN# is asserted low

4 Bay supplies Vid

5 Bay supplies Vop

6 Device switches on

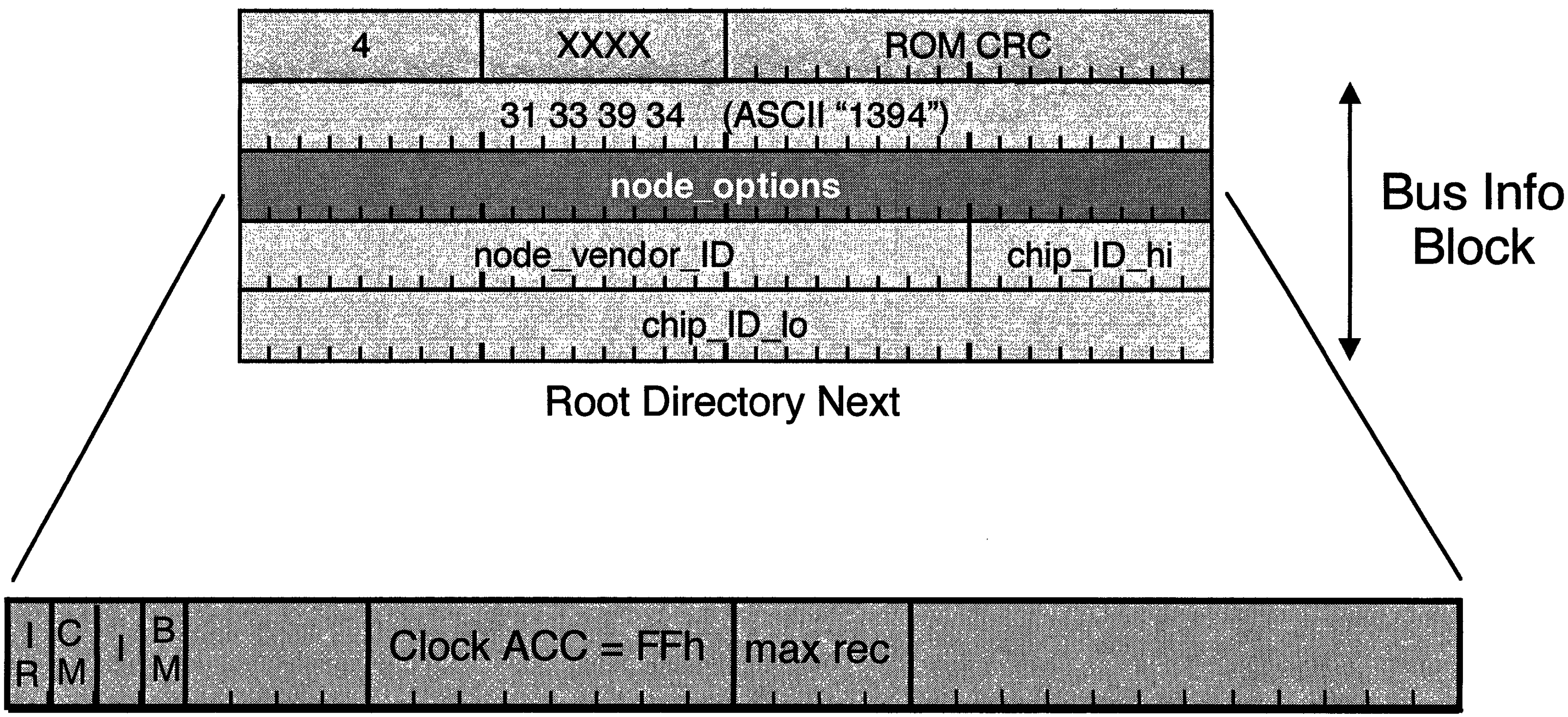
Device Bay Controller



Device Bay 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

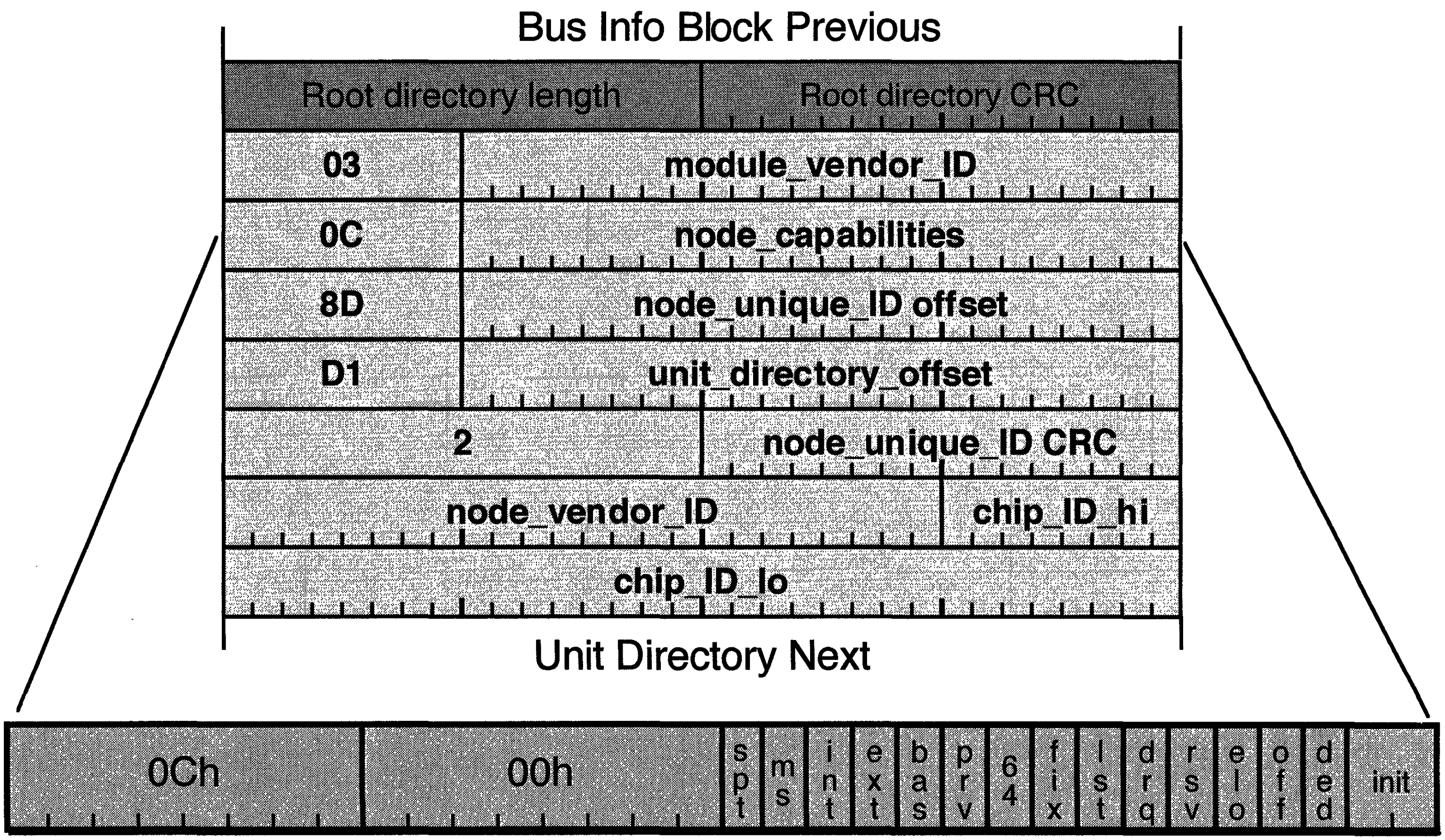
Device Bay Config ROM - Bus Info Block



IR, CM, I and BM must be 0

Note: This should be upgraded in spec to match 1394a.

Device Bay Config ROM - Required Root Directory Entries



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

Device Bay Config ROM - Required Unit Directory Entries

Root Directory Previous

Unit directory Length	Unit directory CRC
12_{16}	unit_spec_ID (00 80 $5F_{16}$)
13_{16}	unit_sw_version (01 00 00_{16})

Note: Different Standards Body than SCSI

Device Bay Controller Required CSR's

CSR Offset	Function
00h	State clear
04h	State set
08h	Node ID
0Ch	Reset Start
18h-1Ch	Split Timeout (only if DBC can be a requester)
210h	Busy Timeout (only if DBC supports retries)

DBC Control Registers

Register	Index Offset	Width	R/Lo	Description
Vendor ID	00h	16	RO	Vendor ID -same as PCI SIG Vendor ID
Revision ID	04h	8	RO	Vendor chosen revidionn number
Subsystem Vendor ID	08h	16	RO	Device Bay Vendor ID
Subsytem ID	0Ah	16	RO	Subsystem revision ID
DBCCR	0Ch	32	RO	Device Bay Controller Capabilities Register
BSTRO	10h	32	RO	Bay O Status Register
BCERO	14h	32	R/W	Bay O Control and Enable Reg
BSTRI	18h	32	RO	Bay 1 Status
BCERI	1ch	32	R/W	Bay 1 Control and Enable Reg
.	.	.	.	.
.	.	.	.	.
BSTR (N-1)	8(n-1)+10h	32	RO	Bay (n-1) Status
BCER (N-1)	8(n-1)+14h	32	R/W	Bay (n-1) Control and Enable Reg

DBC - Bay Status Register

Bit	Name	Access	Description
31-11	Reserved	R/O	Reserved
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

DBC - Bay Control and Enable Register

Bit	Name	Access	Description
31-8	Reserved	R/O	Reserved
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 interrrupt 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

Device Bay Notes

Appendix R:

Reduced Block Command Set (RBC)

Reduced Block Commands

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

* 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

	7	6	5	4	3	2	1	0
0	OP code = 12h							
1	Reserved						CMD DT	EUPD
2	Page or OP code							
3	Reserved							
4	Allocation Length							
5	Control							

Reduced Block Commands
Mode Select

	7	6	5	4	3	2	1	0
0	OP code = 55h							
1				PF=1				Sp=1
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB) Parameter List Length							
8								(LSB)
9	Reserved							

Reduced Block Commands
Mode Sense

Page Code

0 - Current-optional

1 - Changeable- not supported

2 - Default - Mandatory

3 - Saved - Mandatory

	7	6	5	4	3	2	1	0
0	OP code = 5Ah							
1	Reserved				DBD=1	Reserved		
2	Page Code		Page Code					
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB) Allocation Length							
8								(LSB)
9	Control							

RBC Mode page 3E
Device parameter page

	7	6	5	4	3	2	1	0
0	PS=1	R	Page code = 3Eh					
1	Page Length = 8							
2	Reserved							WCD
3	(MSB)		Logical Block Size					
4								(LSB)
5	(MSB)							
6			Number of Logical Blocks					
7								
8								
9								(LSB)

RBC Mode Page 3E
Device Parameters Continued

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

V

Reduced Block Commands
Read (10)

	7	6	5	4	3	2	1	0
0	OP code = 28h							
1	Reserved			DPO =0	FUA =0	Reserved		RA=0
2	(MSB)							
3	LBA							
4								
5								(LSB)
6	Reserved							
7	(MSB)							
8	Transfer Length							(LSB)
9	Control							

RBC
Start/Stop Unit

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
6 - Fh - Reserved

	7	6	5	4	3	2	1	0
0	OP code = 1Bh							
1	Reserved							1MM
2	Reserved							
3	Reserved							
4	Power Conditions				Reserved		Load Eject	Start
5	Control							

RBC
Synchronize Cache

	7	6	5	4	3	2	1	0
0	OP code = 35h							
1	Reserved					WCD	IMM=0	RA=0
2	(MSB)							
3	LBA=00							
4								
5								(LSB)
6	Reserved							
7	(MSB)							
8								(LSB)
9	Control							

Writes cache data to media
Applies to entire device only

WCD = Write cache disable

RBC
Test Unit Ready

	7	6	5	4	3	2	1	0
0	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-Illegal Request	25h, 00h - Logical Unit not ready
02	02 Not Ready	04h, 00h - 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)

	7	6	5	4	3	2	1	0
0	OP code = 2Ah							
1	Reserved			DPO =0	FUA	Reserved		RA=0
2	(MSB)							
3	LBA							
4								
5								(LSB)
6	Reserved							
7	(MSB)							
8	Transfer Length							(LSB)
9	Control							

Reduced Block Command
Write & Verify (10)

	7	6	5	4	3	2	1	0
0	OP code = 2Eh							
1	Reserved			DPO =0	Reserved		Byte Chk=0	RA=0
2	(MSB) LBA							
3								
4								
5								(LSB)
6								
7	(MSB) Transfer Length							
8								(LSB)
9	Control							

RBC Commands
Write Buffer

	7	6	5	4	3	2	1	0
0	OP code = 3Bh							
1	Reserved					Mode		
2	Buffer ID							
3	(MSB) Buffer Offset							
4								
5								(LSB)
6	(MSB) Parameter List Length							
7								
8								(LSB)
9	Control							

Reduced Block Commands
Write Buffer Mode

Mode	Description	Implementation
0h	Write combined header and data	Not Supported
1h	Vender specific	Vendor Specific
2h	Write Data	Not Supported
3h	Reserved	Reserved
4h	Download Microcode	Not Supported
5h	Download Microcode and Serve	Mandatory
6h	Download Microcode with Offset	Not Supported
7h	Download Microcode with Offset and Serve	Not Supported

V

This Page Is Intentionally Blank

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

Data Bytes

Reason For Request

Key Info Returned

All Numbers In Hex
Trace doesn't show Ack Packets

Here's What's Happening - Part 1

3	5	RReq	14	FFFF FFFF 0400	Get Bus Info Block
5	3	RRsp		04 XX XX XX	
				31 33 39 34	
				F8 01 00 08	
				12 34 56 78	
				9A BC DE F0	Got Length - Skip Around
3	5	RReq	4	FFFF FFFF 0414	Get ROM Root Length
5	3	RRsp		00 04 XX XX	ROM Root Length = 4 Quads
3	5	RReq	10	FFFF FFFF 0418	Read ROM Root
5	3	RRsp		03 12 34 56	
				0C 00 83 80	
				8D 00 00 02	
				D1 00 00 04	Offset to Unit Dir = 4 Quads
3	5	RReq	4	FFFF FFFF 0434	Get Unit Dir Length
5	3	RRsp		00 07 XX XX	Unit Dir Length = 7 Quads

Tell Us What's Happening - Part 2

3	5	RReq	1C	FFFF FFFF 0438	Get Unit Dir
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	Management Agent at 4000 Quads
3	5	WReq	08	FFFF F010 0000	Write Management Agent
				FF C3 00 00	ORB at Address 0000 1000 0000
				10 00 00 00	
5	3	RReq	20	0000 1000 0000	Read Login Request
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	Login Response at 0000 1010 0000
				FF C3 00 00	Status at 0000 1020 0000
				10 20 00 00	

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
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
```

Read Command ORB

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
```

Read Command ORB

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

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

Inquiry Data Returned

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

Post Status

For ORB from 0000 1000 0000
Resp = Completed

What Condition Is The Target In ?