S800base-T: the Best of Both Worlds

Michael Johas Teener

Plumbing Architect, Apple Computer Inc.

teener@apple.com

Agenda

- Goals, justification & requirements
- Architecture
  - Reconciliation layer
  - Auto-negotiation
- Status

Goals

- Take advantage of Gigabit Ethernet technology to use Category 5 unshielded twisted pair cabling for S800 transport 1394b links
- Allow appropriate negotiation to be done so that the endpoints can select which of 5 protocols to be used:
  - 10baseT Ethernet
  - 100baseTX Ethernet
  - S100 1394b
  - 1000baseT Ethernet
  - S800 1394r

Sidebar: what is 1394r?

- New PAR (project action request) from IEEE to update/revise 1394-1995.
- Combine 1394-1995, 1394a-2000, 1394b-2002, 1394b errata, and S3200 improvements into one document.
- S800base-T study group is defining S800 over Cat5 as a new section for 1394r

Goals (continued)

- Allow a simple hub-like-thing to be built that:
  - Connects all endpoints that negotiate to Ethernet using standard hub or switch technology
  - Connects all endpoints that negotiate to 1394 using standard PHY or 1394.1 technology
  - Bridges IP data between the two network domains
- For the end user, the objective is to have a single RJ-45 socket that is labeled “network”, and works for any kind of connection.

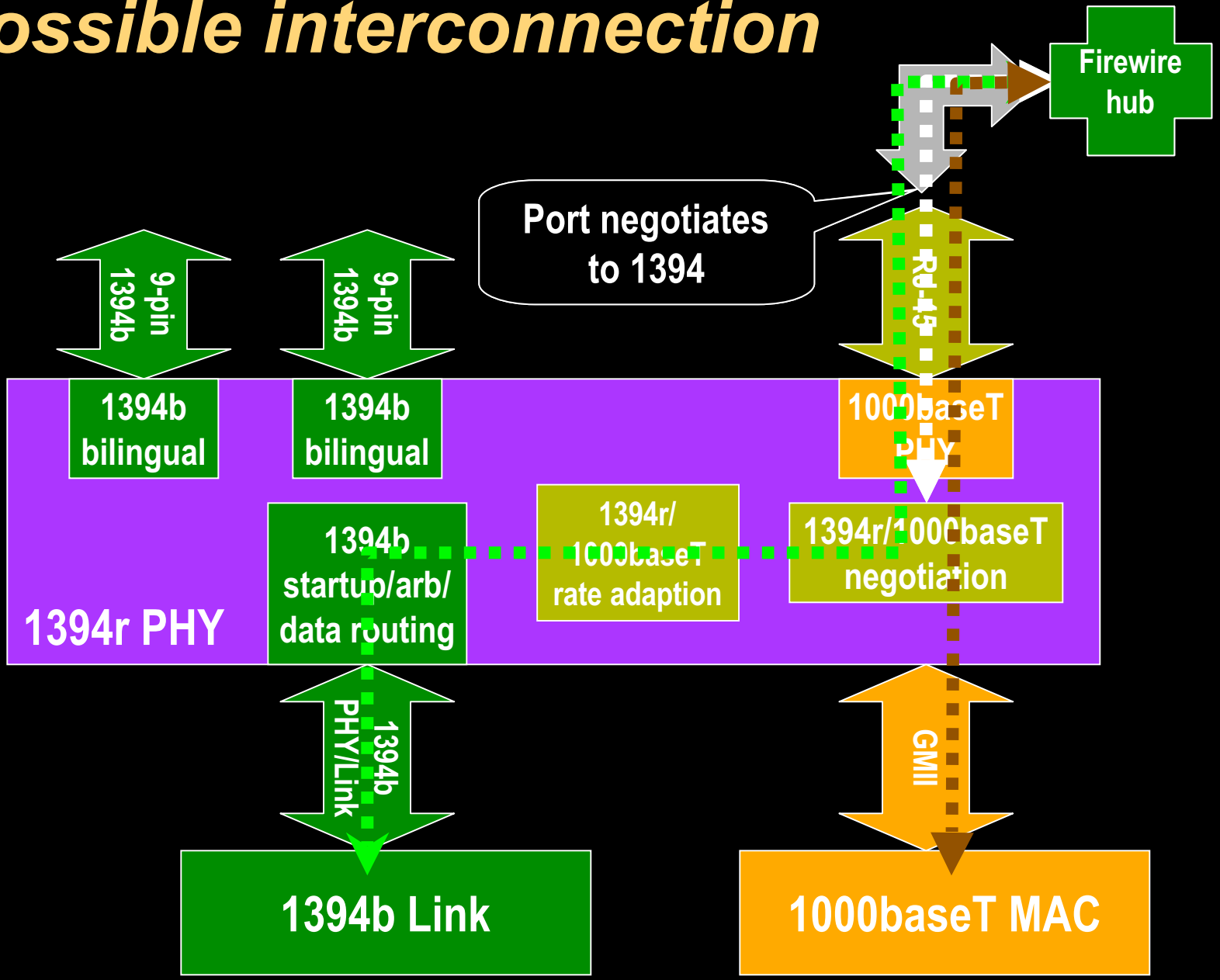
Technical justification

- 1000baseT links are full duplex 1000 Mbit/sec transports at the PHY (4x250Mbit/sec at the cable)

  -100ppm tolerance = 999.9 Mbit/sec
- 1394b S800 links are full duplex (10/8)*8*98.304 Mbit/sec at the cable (983.04 Mbit/sec)

  +100ppm tolerance ≈ 983.1 Mbit/sec
- There is clearly enough bandwidth at the 1000baseT PHY to accept a fully encoded 1394b S800 stream

Possible interconnection



Requirements

- At PHY/Link interface must appear to be standard 1394b PHY
- At GMII must appear to be standard 1000baseT PHY
- When network port negotiates to be 1394, must appear to be standard 1394b port connection to 1394 management software
  - Looks like network is unconnected to Ethernet driver
- When a network port negotiates to be Ethernet, must appear to be standard Ethernet connection to Ethernet management software
  - Looks like unconnected port to 1394 driver

More requirements

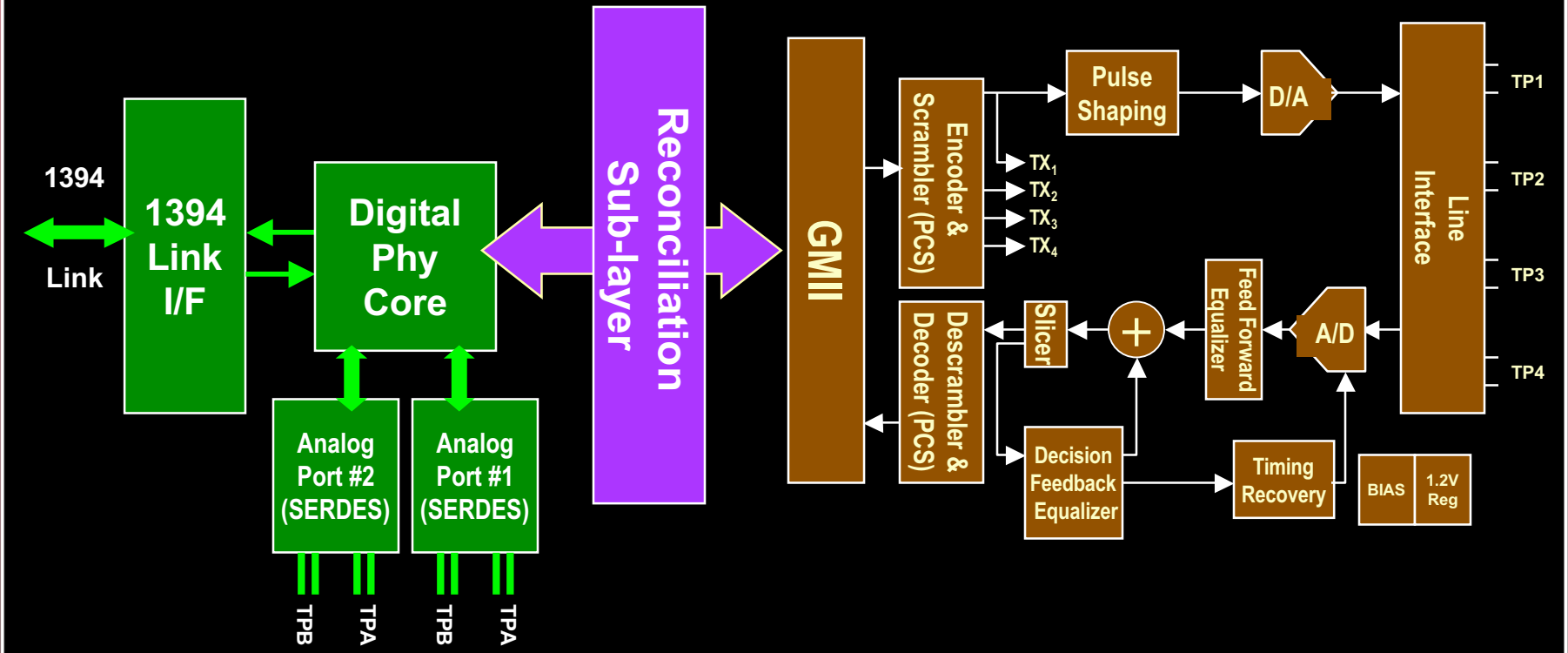
- Must support 1394b S100 as defined in IEEE Std 1394b-2002, and S800 using 1000baseT modulation
  - S100 uses 100BaseT Ethernet pairs (1/2, 3/6)
- Must support 10baseT, 100baseT, 1000baseT (full and half duplex) Ethernet as defined in IEEE Std 802.3
- Negotiation preference set at device endpoint (NOT at hub/switch/bridge) … e.g., Apple would prefer FireWire for Mac OS X, others may prefer alternate connections.
  - Or do we always prefer 1394?

Study group charter

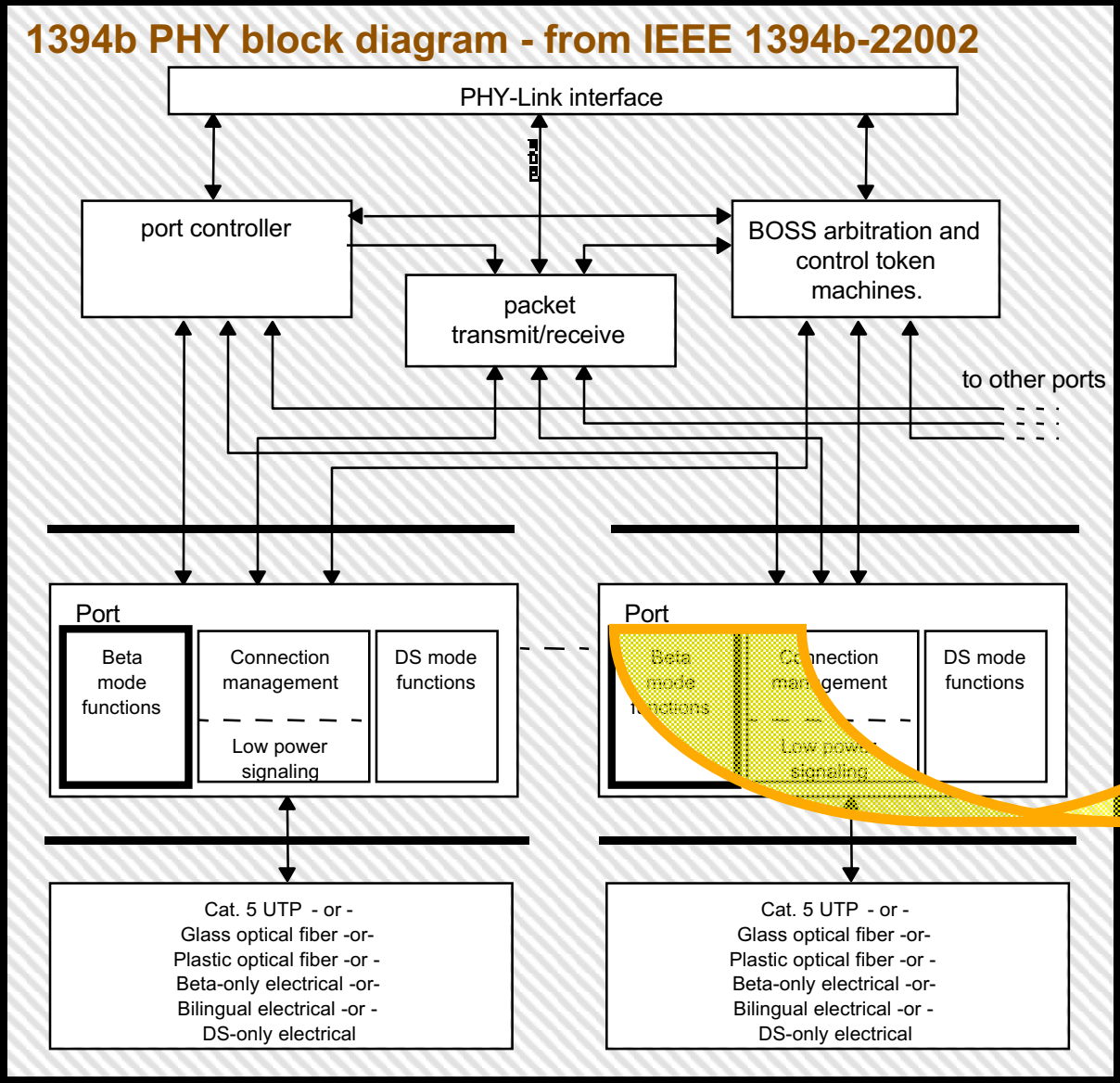
Bob Davis, Chair of the Microcomputer Standards Committee of the IEEE approved the formation of the study group on March 5, 2003, with the following statement:

"The group is chartered to investigate methods of running IEEE 1394 over up to 100 meters of UTP-5 by leveraging existing gigabit Ethernet PHY technology."

S800BASE-T Architecture

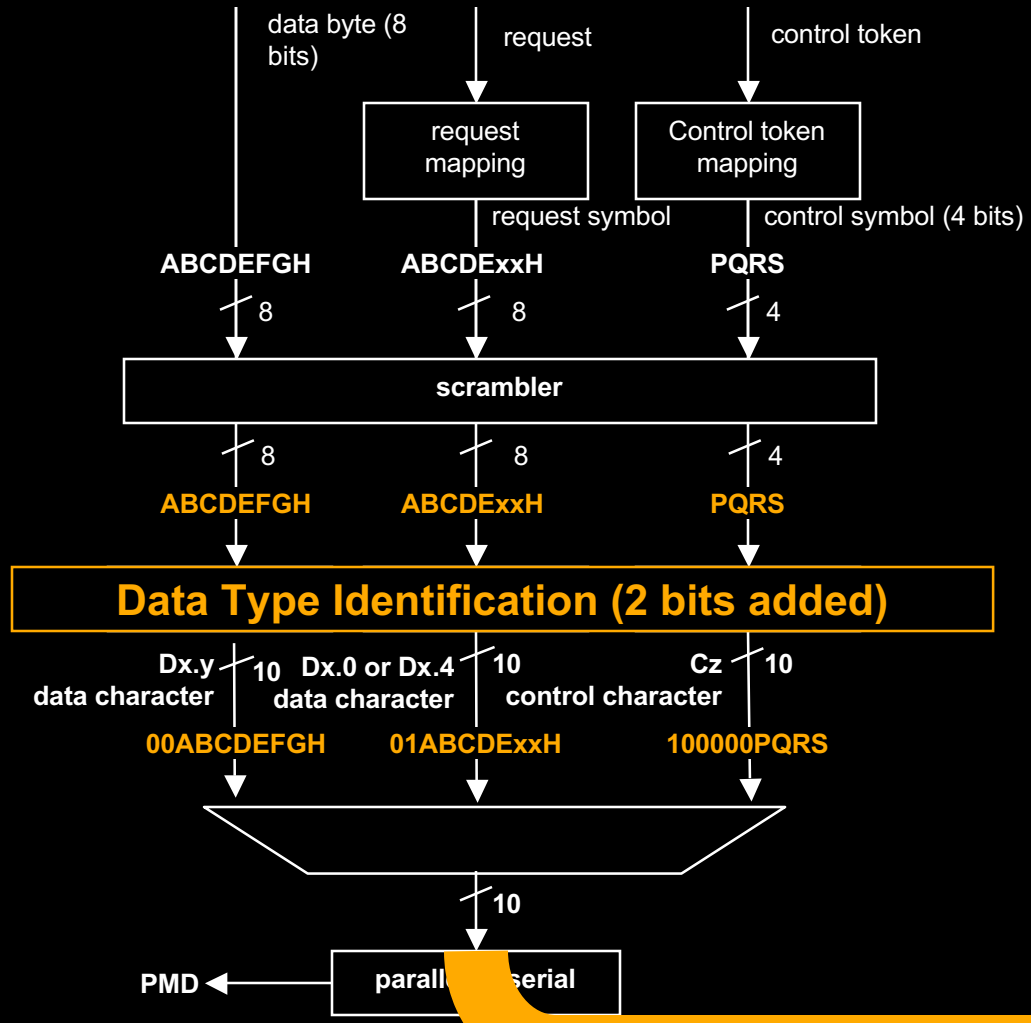


Where S800base-T fits



Interface to 100BASE-T PHY

S800base-T modifications to 1394b PHY



Scrambler is bypassed

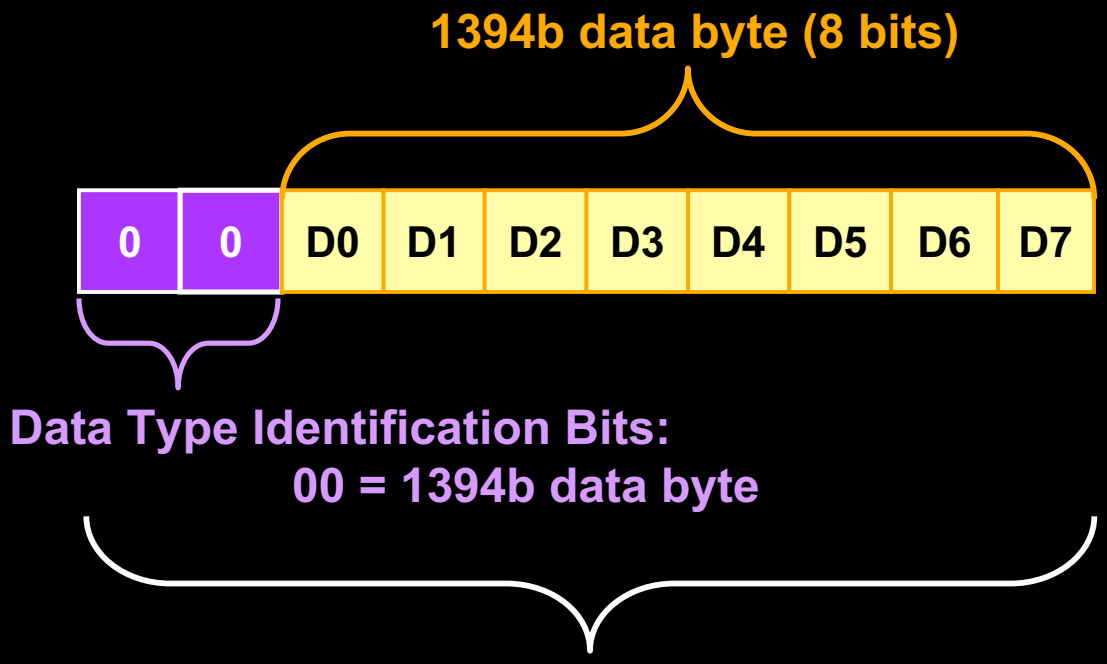
Encoder is replaced by Data Type ID

Serializer is bypassed

Interface to Ethernet PHY is 10 bits parallel data

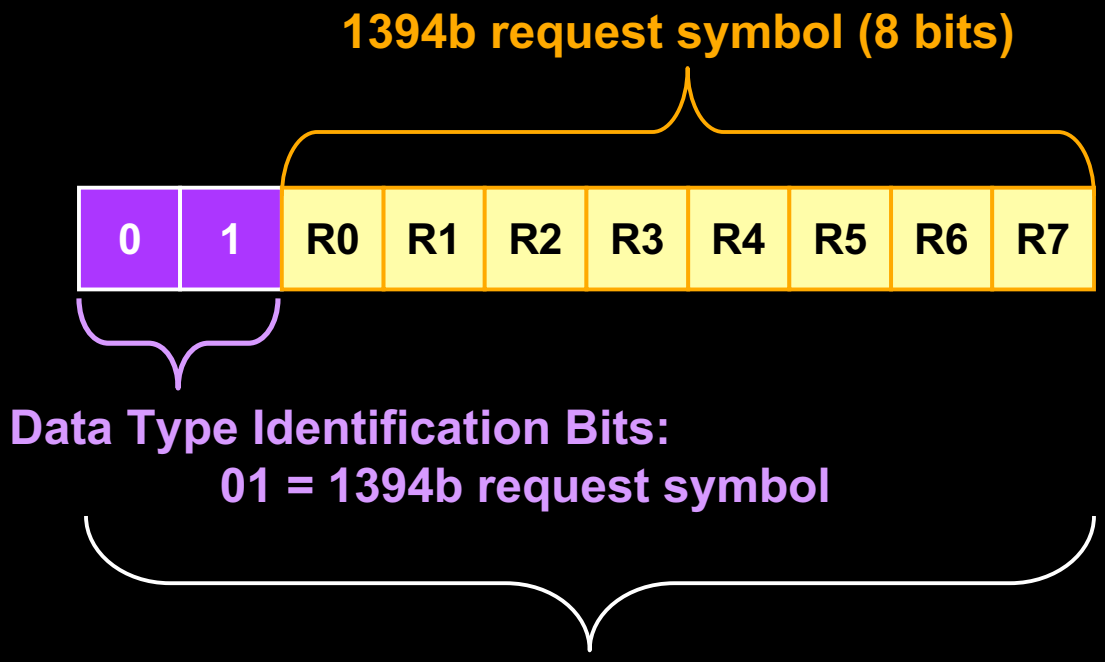
Data Type Identification

Data byte



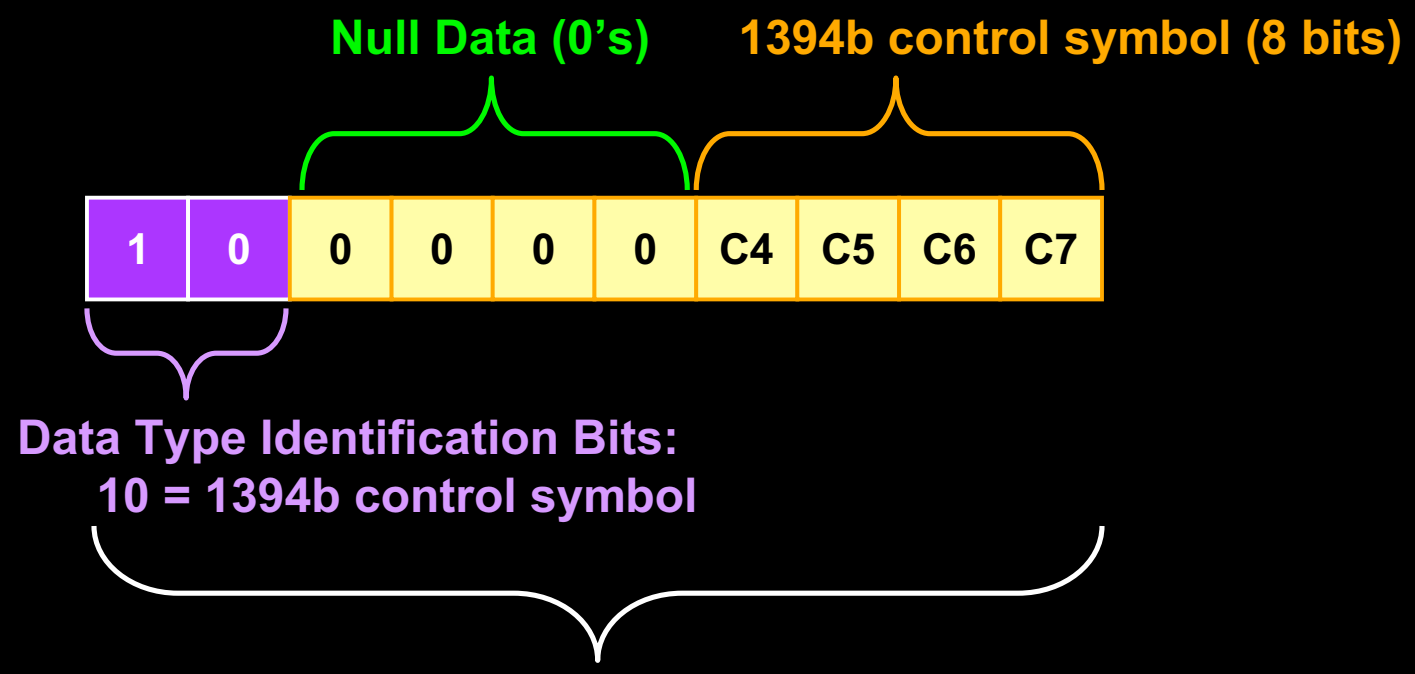
Data Type Identification

Request symbol

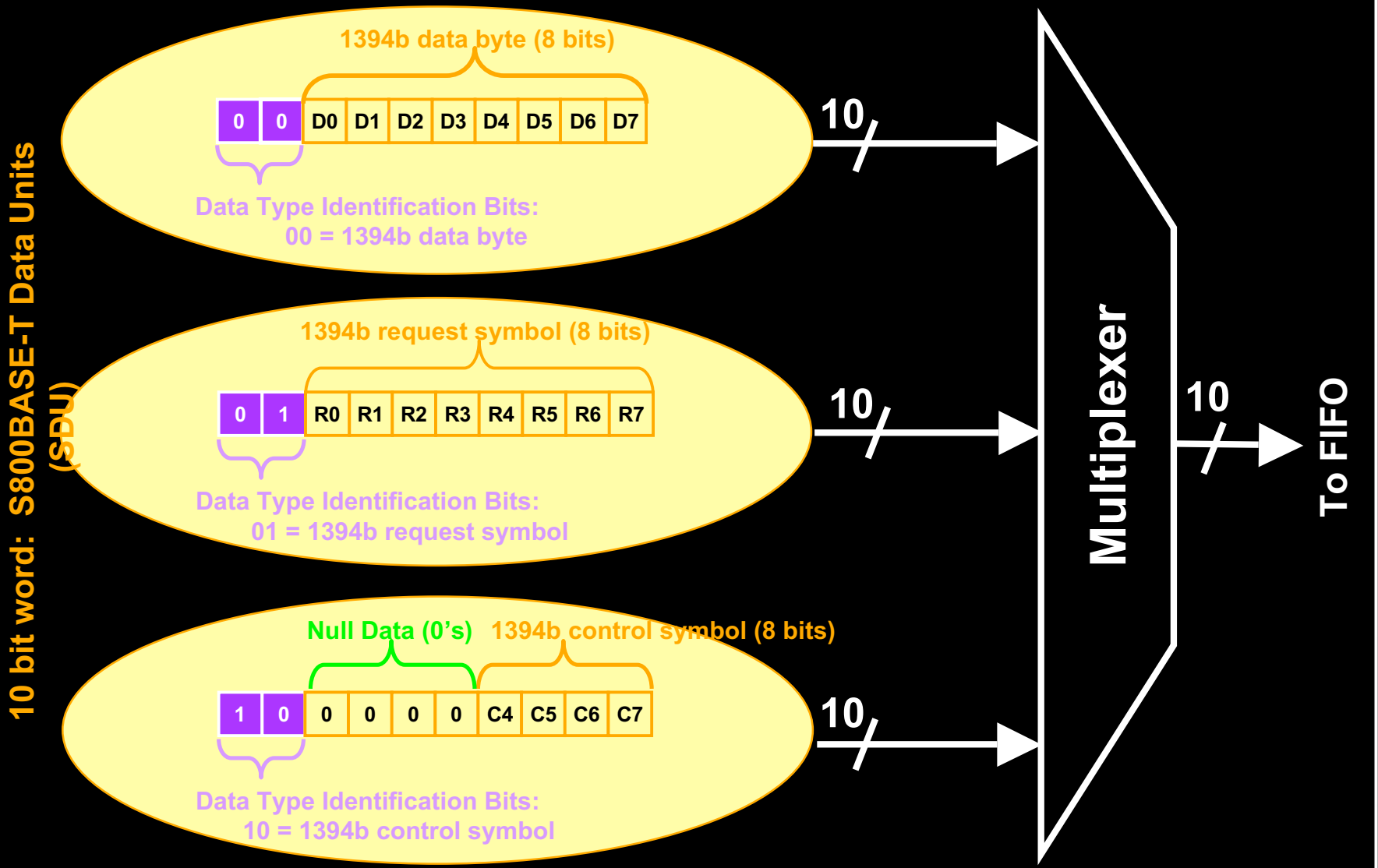


Data Type Identification

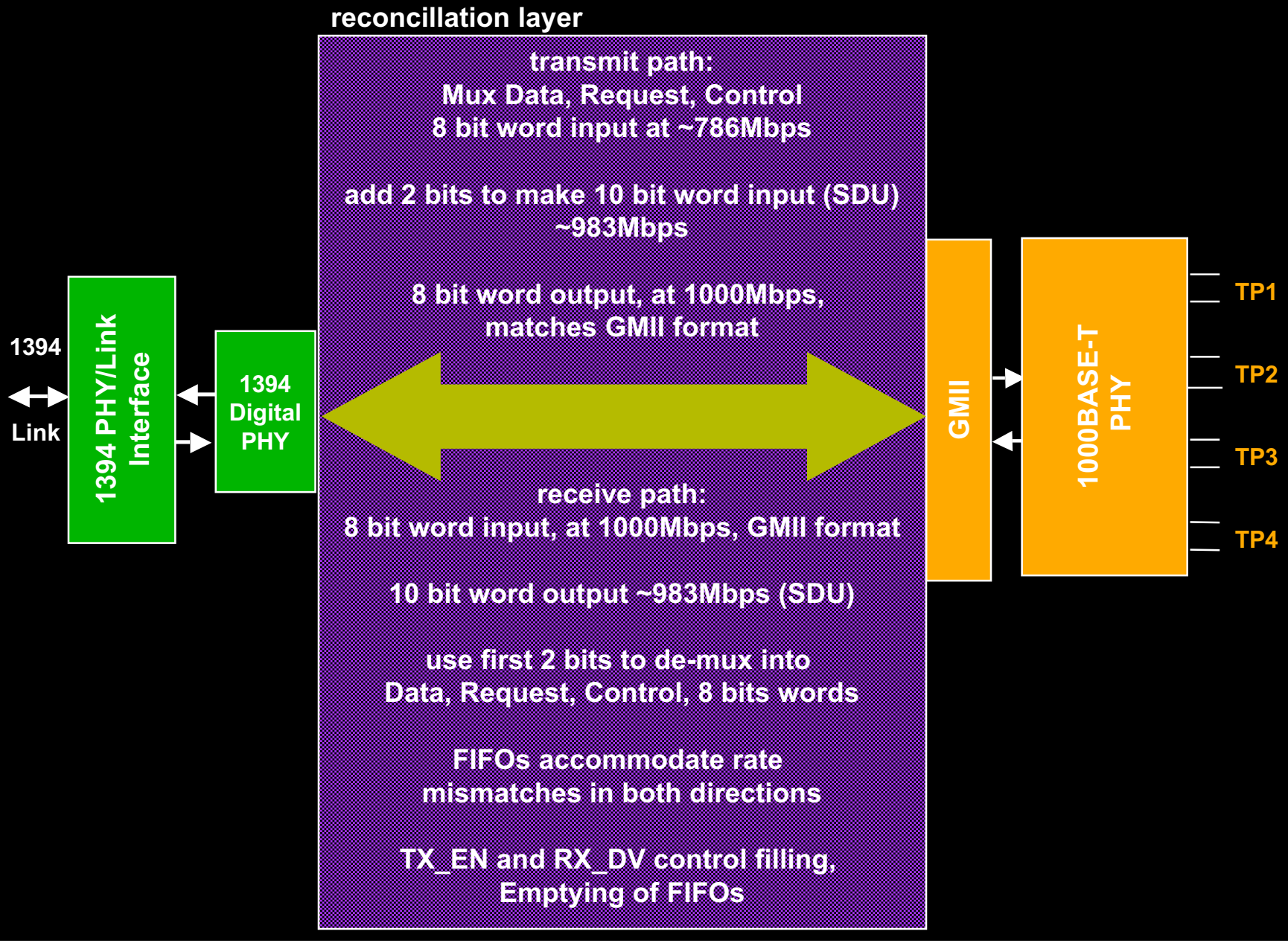
Control symbol



Mux SDUs into single stream



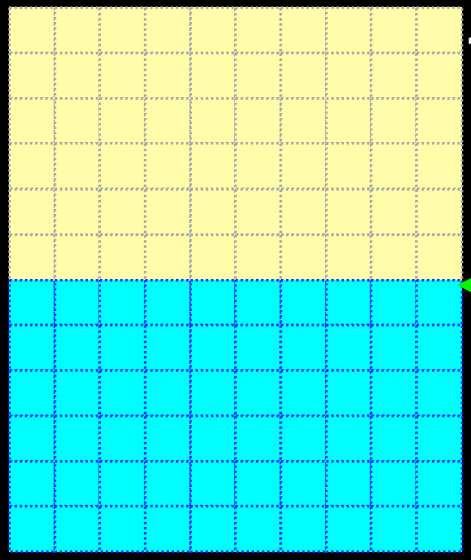
S800BASE-T Block Diagram



Reconciliation sublayer: transmitter

Encoded S800 data stream:
10 bit word SDU (S800BASE-T Data Unit)

Shift 10 bit word at 98.3MHz into FIFO



120 bit deep FIFO

Pointer

Transmit Enable

When pointer indicates FIFO reach FULL state, TX_EN goes HIGH

When pointer indicates FIFO reached EMPTY state, TX_EN goes LOW

TX_EN

8 bit bytes go directly to 1000BASE-T GMII

Reconciliation Sublayer: Transmitter

Transmit Sequence:

1. 1394b PHY begins transmitting to FIFO at 983Mbps
2. FIFO takes 80-88ns to reach FULL state
3. TX_EN goes HIGH, 1000BASE-T PHY begins transmitting data
4. FIFO empties while 1000BASE-T PHY transmits at 1000Mbps (faster than the incoming data)
5. When FIFO reaches EMPTY state, TX_EN goes LOW
6. 1000BASE-T PHY sends IDLE while FIFO is re-filling
7. After 80-88ns, FIFO is FULL, TX_EN goes HIGH, data transmission resumes

1394b PHY sends data + control symbols continuously at 983Mbps

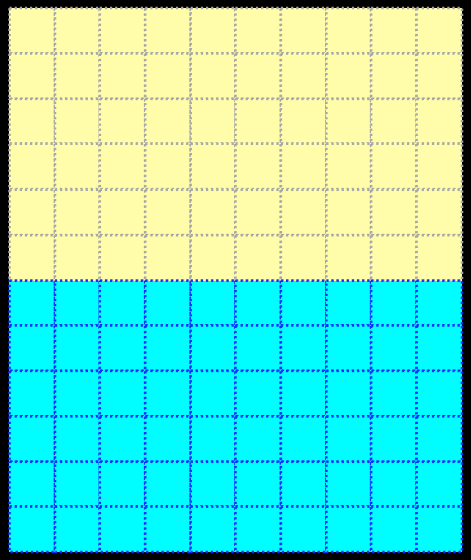
1000BASE-T PHY alternates between bursts of data and IDLE symbols

Reconciliation Sublayer: Receiver

8 bit bytes come directly to 1000BASE-T GMII

RX_DV

120 bit deep FIFO



Receive Data Valid

When RX_DV is LOW (IDLE) no data is loaded into FIFO

When RX_DV is HIGH, valid data is loaded into FIFO

Encoded S800 data stream:
10 bit word SDU (S800BASE-T Data Unit)

Shift 10 bit word at 98.3MHz out of FIFO

Reconciliation Sublayer: Receiver

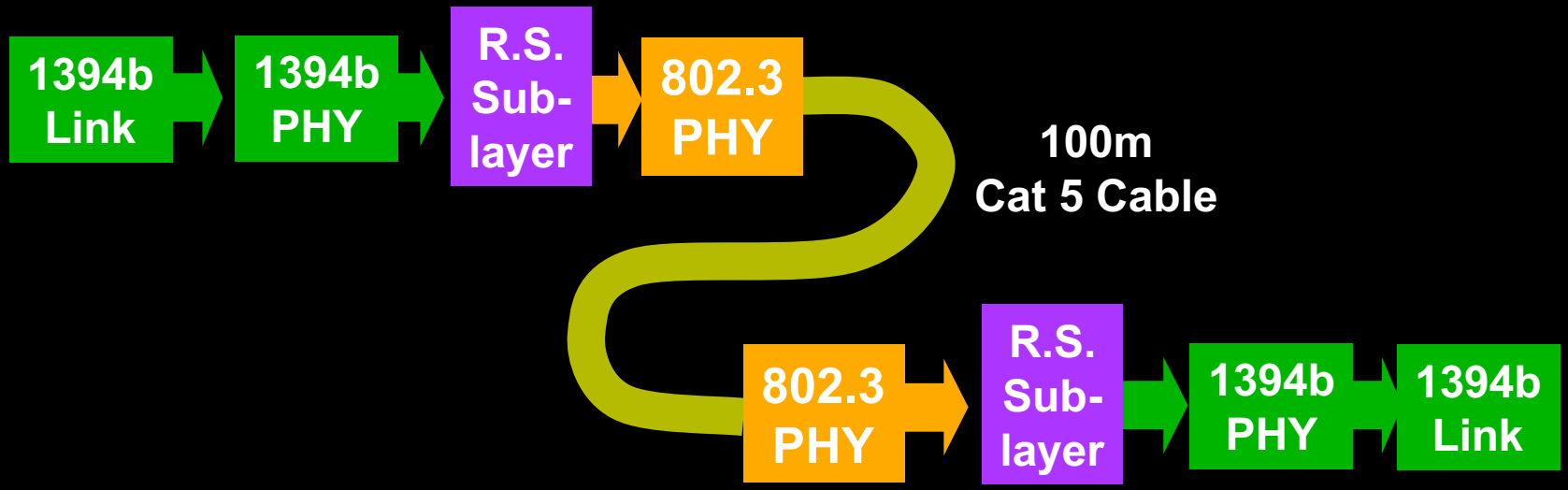
Receive Sequence:

1. 1000BASE-T receives IDLE, no data is loaded into FIFO
2. When 1000BASE-T receives data, RX_DV goes high
3. FIFO fills with data from 1000BASE-T at 1000Mbps
4. FIFO empties data to 1394b PHY at 983Mbps
5. Periodic IDLE patterns allow FIFO to empty, in order to prevent overflowing

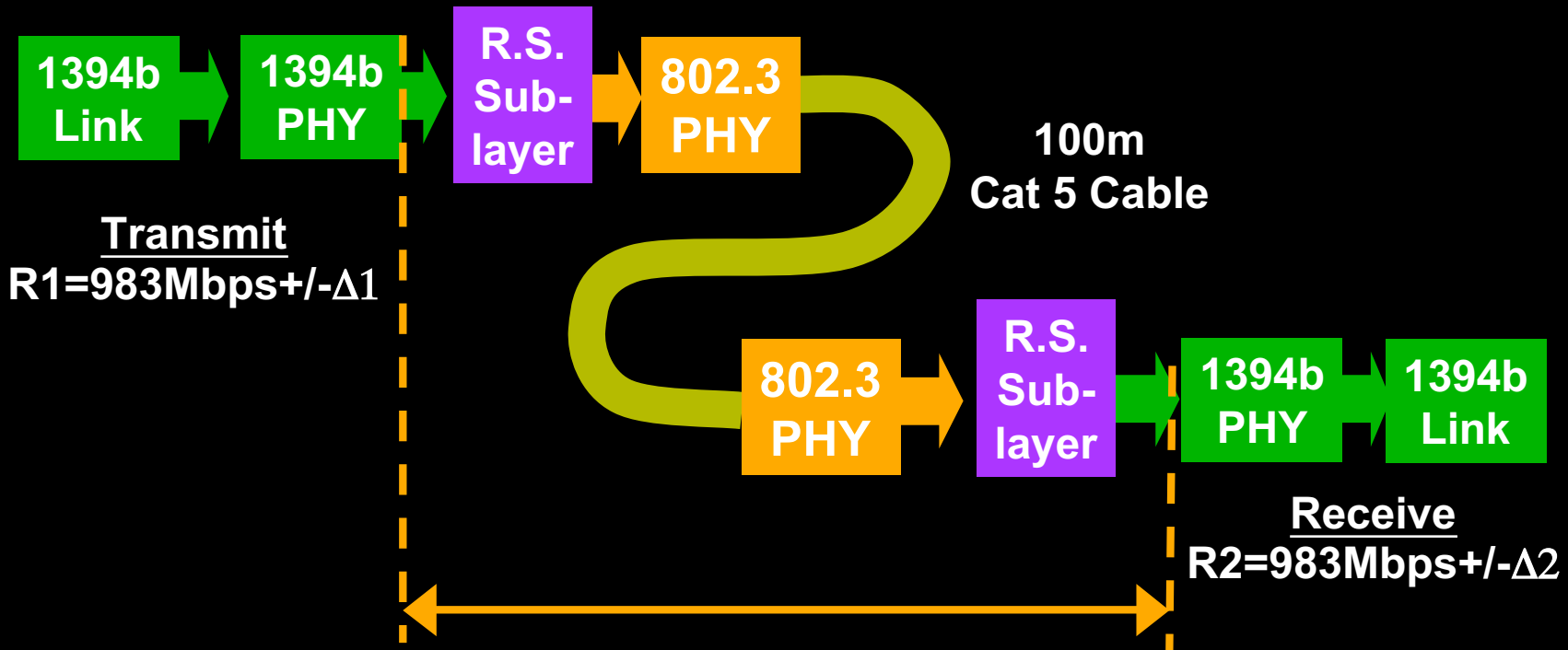
1000BASE-T PHY alternates between bursts of data and no data (no input to FIFO)

1394b PHY receives data + control symbols continuously at 983Mbps

Link Equivalence



Link Equivalence



Recovered clock not sent to 1394b receive PHY 2
Data rate does NOT match exactly
1394b receive PHY 2 handles rate difference delta, after each 1394 packet

Reconciliation layer summary

- Straight-forward design
- No exotic technology
- Actually simpler than standard 1394b beta port
  - No scrambler/descrambler
  - Minimal data encode/decode
- Some extra latency due to 1000base-T PHY encoding/decoding

Auto-Negotiation for S800base-T

- Purpose of Auto-Negotiation
- How does Auto-Negotiation Work
- S800Base-T Auto-Negotiation Scenarios
- Relevant Ethernet Standards that need to be Modified
- Technical Proposal
- Summary

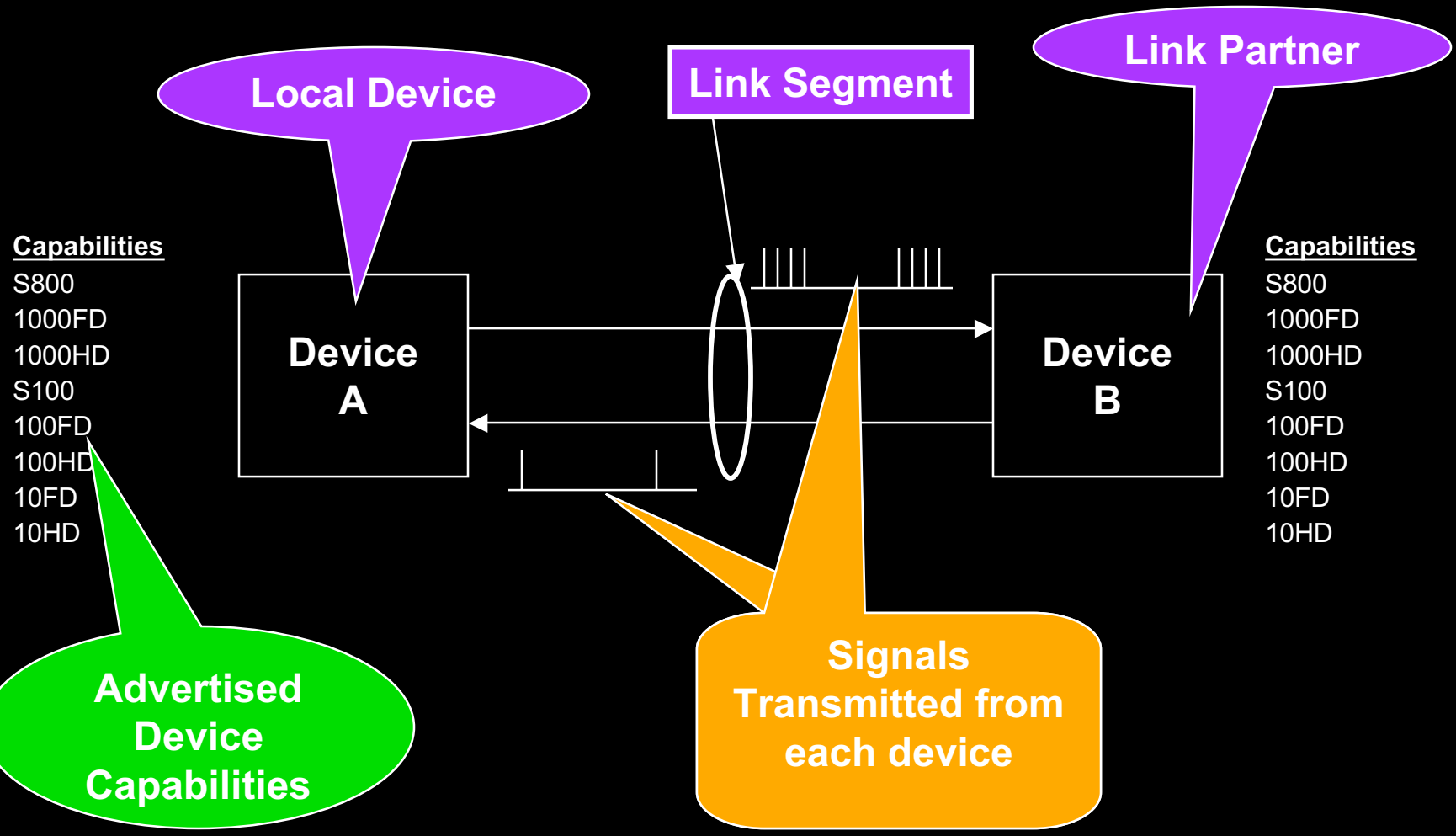
Auto-Negotiation Goals

- Enable automatic connection between two auto-negotiating devices at the best possible speed and duplex that they both posses
- Automatically configure technology and speed to match a legacy link partner's capabilities even though it may not support auto-negotiation
- Allow communication of additional link level information like flow control support

Auto-Negotiation in a nutshell

- Works between two devices on a link segment
- Exchanges and Acknowledges 16-bit data words using variation of 10Base-T Link Pulse signaling
- Data words contain information about a device's supported capabilities
- The best common technology is automatically selected and enabled
- A-N ends once the chosen technology is enabled and stays out of the way until the link status changes

S800Base-T Auto-Negotiation Scenarios: Key to Diagrams



Terminology

- *Auto-Negotiation* (A-N) is a process that occurs prior to enabling a specific communication technology that determines the *Highest Common Denominator* (HCD) technology between two devices on a *Link Segment*. Auto-Negotiation hands off to the HCD technology when it is finished and stays out of the way until the *[Receive] Link Status* goes down.
- A-N Advertises the *Capabilities* of the *Local Device* (ex. Device A) it is running on to it's *Link Partner* (ex. Device B). The Link Partner does the same thing.
- A-N uses a sequence of 10Base-T *Link Pulses* called *Fast Link Pulses* (FLPs) to communicate a device's capabilities.
- 17 to 33 FLPs are sent in a *FLP Burst* to convey 16 bits of encoded data.

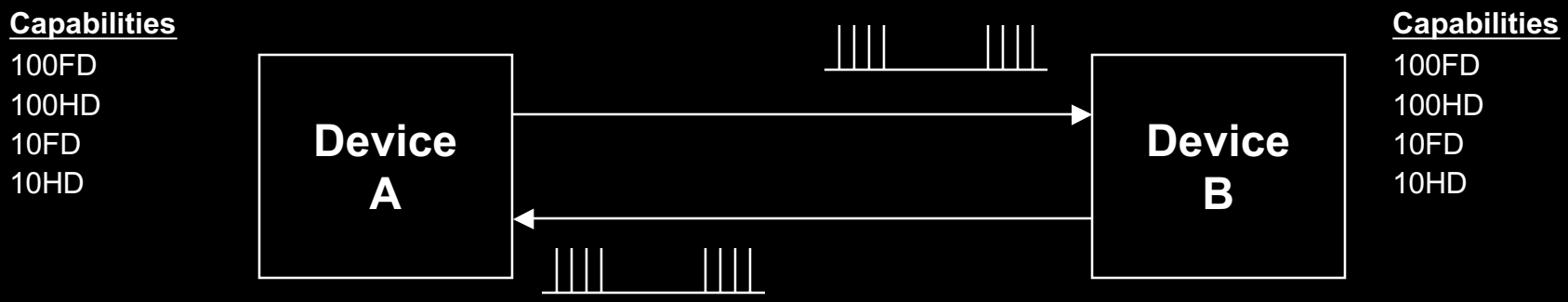
Terminology pt. 2

- One *Base Page* of data is always sent. Additional *Next Pages* conveying additional device capabilities may also be exchanged.
- *Legacy Devices* do not implement A-N. They must be configured into a single mode of operation with a jumper or software.
- A-N uses *Parallel Detection* to attempt to identify Legacy Devices by examining the default signals sent out at link startup.

Scenarios Overview

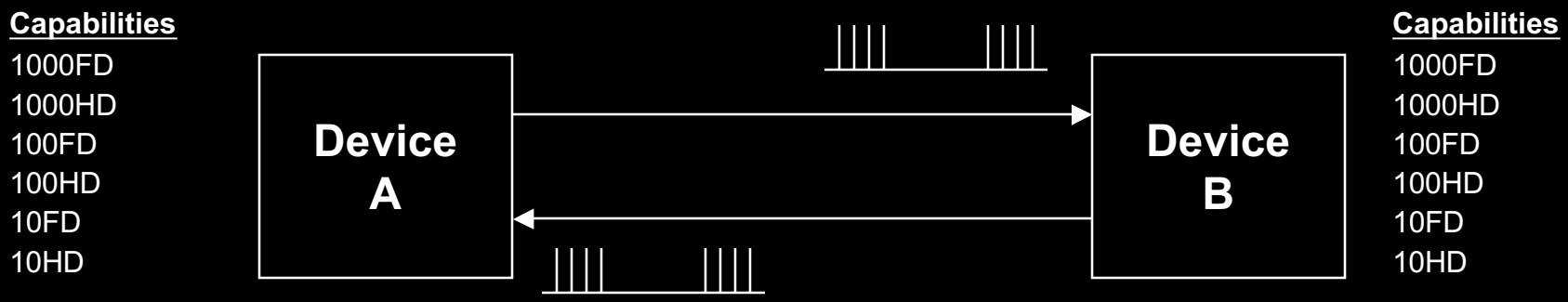
- The following Auto-Negotiation connection scenarios have all been evaluated:
  - 100Base-T A-N to 100Base-T A-N
  - 1000Base-T A-N to 1000Base-T A-N
  - S800Base-T Aware (GE) to S800Base-T Aware (GE)
  - S800Base-T Aware (1394) to S800Base-T Aware (1394)
  - S800Base-T Aware (All) to S800Base-T Aware (All)
  - S800Base-T Aware to Clause 40 Auto-Negotiation
  - S800Base-T Aware to Clause 28 Auto-Negotiation
  - S800Base-T Aware to Legacy 10Mb or 100Mb
  - S800Base-T Aware to Legacy 1394

100Base-T A-N to 100Base-T A-N



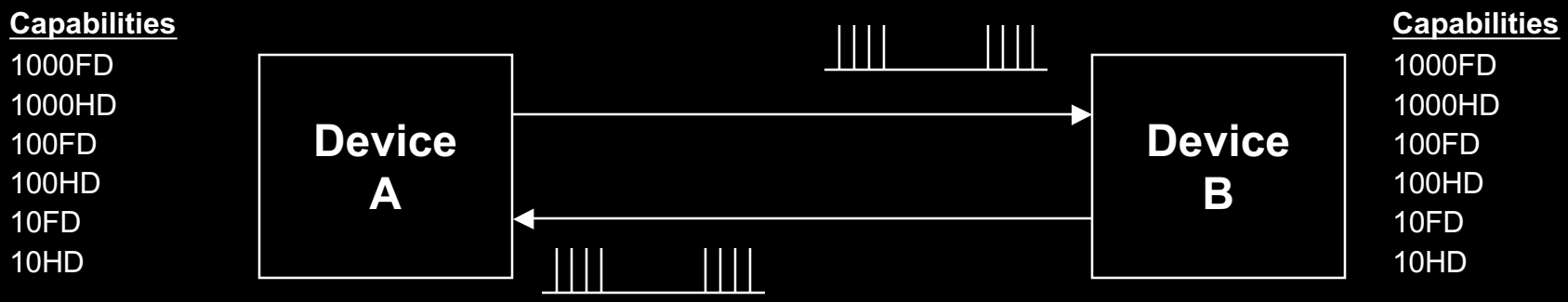
- Here is an easy Auto-Negotiation scenario to warm up on!
- Both devices send out FLP Bursts advertising their capabilities
- Auto-Negotiation completes successfully and enables the HCD technology – 100Base-T Full Duplex

1000Base-T A-N to 1000Base-T A-N



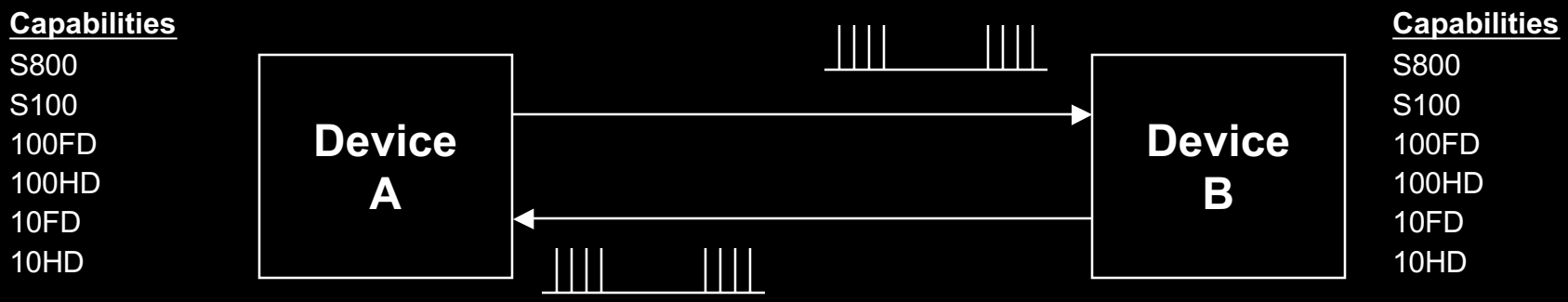
- Same behavior as 100 A-N to 100 A-N except
  - 1000Base-T Full Duplex is the HCD
  - Negotiation requires multiple Next Pages to be exchanged in addition to the Base Page

S800Base-T Aware (no 1394) to S800Base-T Aware (no 1394)



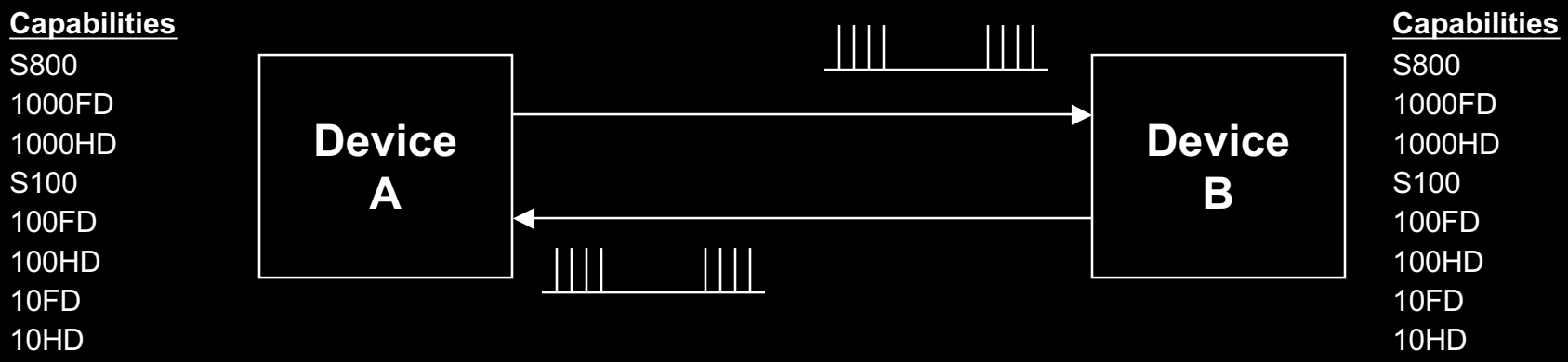
- Both devices send out FLP Bursts advertising their capabilities
- Auto-Negotiation completes successfully and enables the HCD technology – 1000Base-T Full Duplex

S800Base-T Aware (no GE) to S800Base-T Aware (no GE)



- Both devices send out FLP Bursts advertising their capabilities
- Auto-Negotiation completes successfully and enables the HCD technology – S800Base-T

S800Base-T Aware (All) to S800Base-T Aware (All)



- Both devices send out FLP Bursts advertising their capabilities
- Auto-Negotiation completes successfully and enables the HCD technology – S800Base-T

S800Base-T Aware to Legacy 10Mb or 100Mb

Capabilities
S800
1000FD
1000HD
100FD
100HD
10FD
10HD

Capabilities
100HD **FORCED**

- The S800Base-T device sends out FLP Bursts advertising it's capabilities
- The Legacy device sends out it's native signaling – Scrambled Idle Line State
- The Auto-Negotiating device Parallel Detects the Scrambled ILS and enables 100Base-T Half Duplex, completing successfully
  - NOTE: The addition of S800Base-T does not change the one weakness of A-N, in that it still can not Parallel Detect a Full Duplex legacy device.

S800Base-T Aware to Legacy S100 1394b

- The S800Base-T device sends out FLP Bursts advertising it's capabilities
- The Legacy S100 1394b sends out it's native signaling: 48 - 64MHz tone at 1.5% duty cycle
- The Auto-Negotiating device Parallel Detects the toning and enables S100 1394b, completing successfully
  - NOTE: The S100 1394b toning is sufficiently different from Ethernet FLP that parallel detection will work correctly, and S100-only devices will not confuse Ethernet-only devices into making a connection.

Relevant Ethernet Standards

- 802.3
  - Clause 28 – Basic Auto-Negotiation
  - Annex 28A – Selector Field Definitions
  - Annex 28B – 802.3 Selector Base Page Definition
    - Also Priority Resolution
  - Annex 28C – Next Page Message Code Field definitions
    - 1000Base-T Next Pages
      - 1xMC(=8) + 2xUP
  - Annex 28D – Description of Extensions to Clause 28 and associated annexes
    - Clause 40 Extensions
  - Clause 40.5
  - Annex 40C – Add-on interface for additional Next Pages

Possible Approaches

- Bits in 802.3 Base Page
  - Only 1 bit left
- 1394 Selector Field
  - Harder to do 1394 to Ethernet interoperability
    - Existing auto-negotiating devices will ignore these pages
- Add to Gigabit Ethernet Next Page (MC=8)
  - 6 bits leftover in 1st Unformatted Page
- Generic Next Page mechanism (MC=9)
  - Same way Gigabit Ethernet was done

Technical Proposal

- Use the Next Page Mechanism in Auto-Negotiation
  - MC = 9
  - UP = 1 or 2 pages
- This gives us an approach that is completely separate from existing Auto-Negotiation standardization of other technologies
  - Achieve interoperability
  - Probably easier to work through IEEE committee

Base Page

- NO CHANGE

- D15 = 1 to indicate that Next Pages Follow
- D14:D1 = As specified in 28.2.1.2
  - These bits cover 10Base-T and 100Base-TX capabilities and provide the mechanisms needed for base page exchange

Next Page 1: Message Code

- NEW MESSAGE CODE
- M10:M0 = 9
  - Means S800Base-T 1394 over Gigabit Ethernet negotiation
  - Specifies how many next pages in this sequence
    - 1xMC + 2xUP

Next Page 2: First Unformatted Page – New Capabilities

- U10:U4 = Reserved for future use – Transmit as 0
- U5 = S800Base-T Capable
- U4 = 1000Base-T Half Duplex
- U3 = 1000Base-T Full Duplex
- U2 = 1000Base-T Port Type
  - 1=multi-port, 0=single-port device
- U1 = 1000Base-T Master-Slave Manual Configuration value
  - 1=Master, 2=Slave
- U0 = 1000Base-T Master-Slave Manual Configuration enable
  - 1=Manual Configuration Enable

Next Page 3:
Second Unformatted Page – Seed Value

- Keep if use GE pages
- Otherwise we can eliminate

Possible problem, and solution

- Message code 9 may have problems being interpreted/generated by current silicon
- If so, may be required to use existing MC 8 message, and ask to add S800 field to GE page.
  - Must validate both approaches with all major vendors.

Priority Resolution Table 28B.3

- Insert 1394 S800 at top of table due to Isochronous capabilities at nearly the same speed
- New Table
  - S800Base-T
  - 1000Base-T full duplex
  - 1000Base-T half duplex
  - (S100Base-T?)
  - 100Base-T2 full duplex
  - 100Base-TX full duplex
  - 100Base-T2 half duplex
  - 100Base-T4 half duplex
  - 100Base-TX half duplex
  - 10Base-T full duplex
  - 10Base-T half duplex

S800Base-T A-N summary

- Implementing Auto-Negotiation for S800Base-T will allow easy interoperability with 1000Base-T and slower Ethernet devices
- There are no technical hurdles to implementing Auto-Negotiation for S800Base-T
- The IEEE standards possibilities are well understood
- All that remains is to prepare a new draft standard and work with the IEEE committee to get it approved

S800Base-T Status

- No technical problems remain
  - Compatibility validation still required, particularly with actual implementations
- Study group meets every 6 weeks or so …
  - Chair is Michael Johas Teener, Apple
    - teener@apple.com
  - Secretary is Burke Henehan, TI
    - bhenehan@ti.com
  - Major contributions by Broadcom, Apple, Avaya
- study group website is http://grouper.ieee.org/groups/1394/S800BASE-T
- Study group uses main 1394 email list:
  - Send “subscribe stds-1394” to “majordomo@ieee.org”

Thank you!

Thanks particularly for slides on the reconciliation layer to Kevin Brown of Broadcom
and slides on auto-negotiation to Walter K. Hurwitz, also of Broadcom