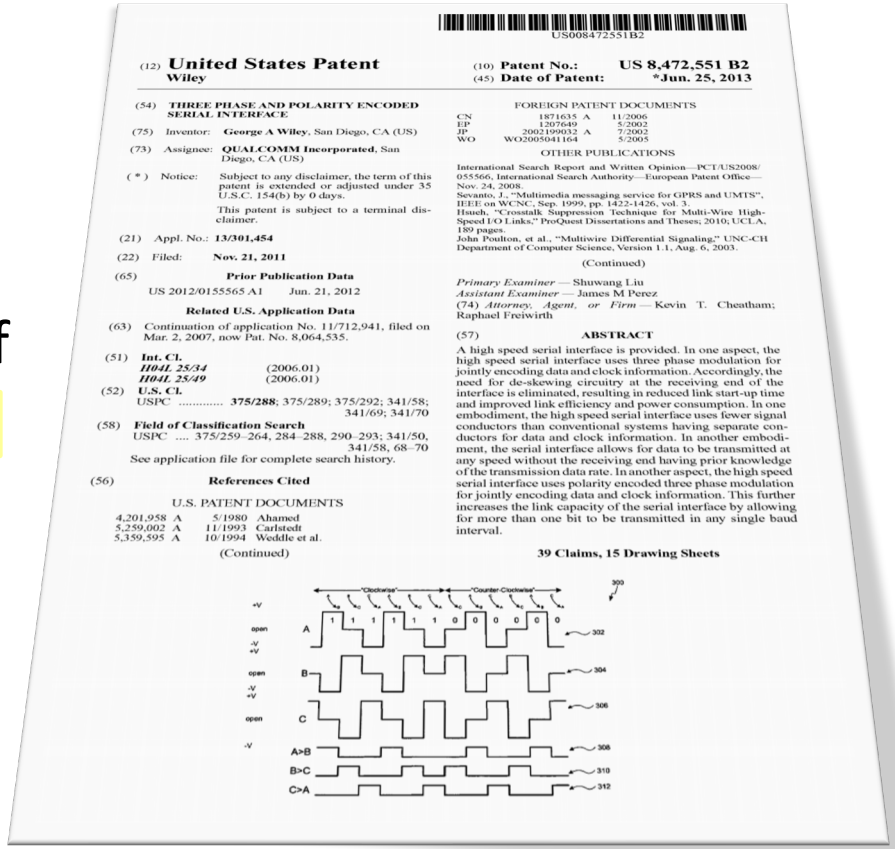
MIPI C-PHY℠: Introduction

From Basic Theory to Practical Implementation

Mohamed Hafed
Introspect Technology

Original Spark: Three Phase Encoding!

1 Unit Interval of Data ⟷ 2.285 Bits of Information



US008472551B2

(12) **United States Patent**
Wiley

(10) **Patent No.: US 8,472,551 B2**
(45) **Date of Patent: *Jun. 25, 2013**

(54) **THREE PHASE AND POLARITY ENCODED SERIAL INTERFACE**

(75) Inventor: **George A Wiley**, San Diego, CA (US)

(73) Assignee: **QUALCOMM Incorporated**, San Diego, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **13/301,454**

(22) Filed: **Nov. 21, 2011**

(65) **Prior Publication Data**

US 2012/0155565 A1 Jun. 21, 2012

Related U.S. Application Data

(63) Continuation of application No. 11/712,941, filed on Mar. 2, 2007, now Pat. No. 8,064,535.

(51) **Int. Cl.**
H04L 25/34 (2006.01)
H04L 25/49 (2006.01)

(52) **U.S. Cl.**
USPC **375/288**; 375/289; 375/292; 341/58; 341/69; 341/70

(58) **Field of Classification Search**
USPC 375/259–264, 284–288, 290–293; 341/50, 341/58, 68–70
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,201,958 A 5/1980 Ahamed
5,259,002 A 11/1993 Carlstedt
5,359,595 A 10/1994 Weddle et al.
(Continued)

FOREIGN PATENT DOCUMENTS

CN 1871635 A 11/2006
EP 1207649 5/2002
JP 2002199032 A 7/2002
WO WO2005041164 5/2005

OTHER PUBLICATIONS

International Search Report and Written Opinion—PCT/US2008/055566, International Search Authority—European Patent Office—Nov. 24, 2008.
Sevanto, J., "Multimedia messaging service for GPRS and UMTS", IEEE on WCNC, Sep. 1999, pp. 1422-1426, vol. 3.
Hauch, "Crosstalk Suppression Technique for Multi-Wire High-Speed I/O Links," ProQuest Dissertations and Theses; 2010; UCLA, 189 pages.
John Poulton, et al., "Multiwire Differential Signaling," UNC-CH Department of Computer Science, Version 1.1, Aug. 6, 2003.
(Continued)

Primary Examiner — Shuwang Liu
Assistant Examiner — James M Perez
(74) *Attorney, Agent, or Firm* — Kevin T. Cheatham; Raphael Freiwirth

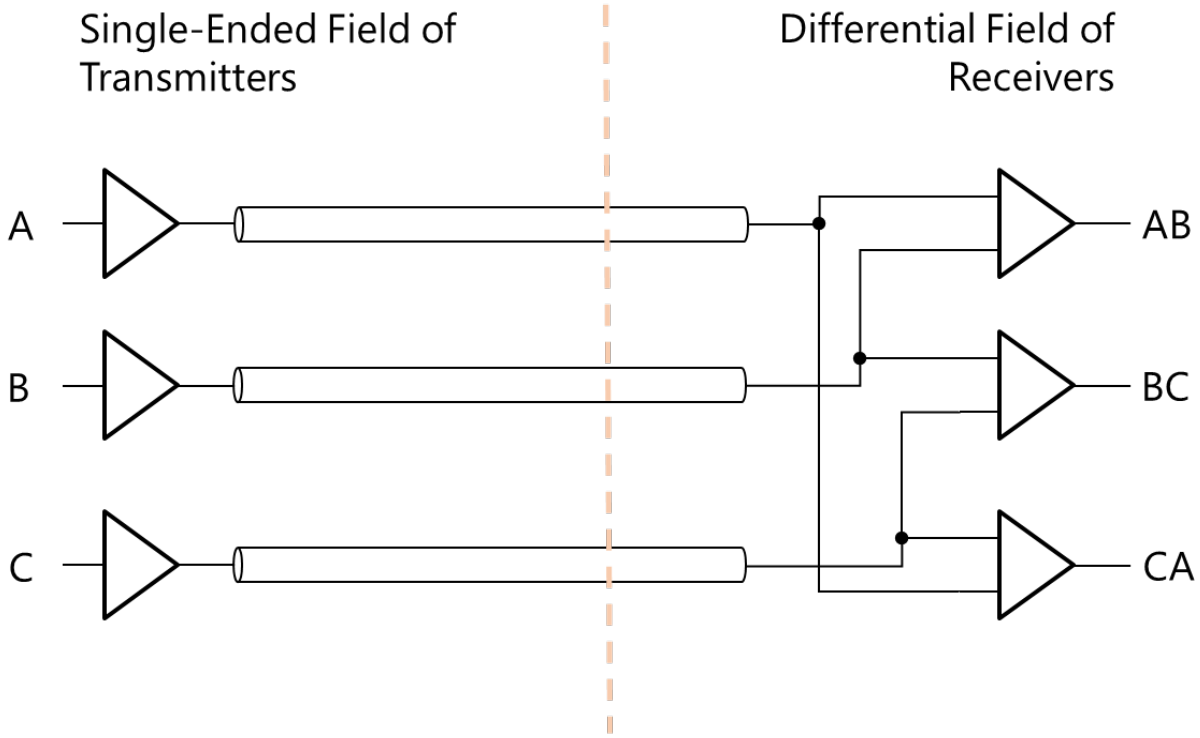
(57) **ABSTRACT**

A high speed serial interface is provided. In one aspect, the high speed serial interface uses three phase modulation for jointly encoding data and clock information. Accordingly, the need for de-skewing circuitry at the receiving end of the interface is eliminated, resulting in reduced link start-up time and improved link efficiency and power consumption. In one embodiment, the high speed serial interface uses fewer signal conductors than conventional systems having separate conductors for data and clock information. In another embodiment, the serial interface allows for data to be transmitted at any speed without the receiving end having prior knowledge of the transmission data rate. In another aspect, the high speed serial interface uses polarity encoded three phase modulation for jointly encoding data and clock information. This further increases the link capacity of the serial interface by allowing for more than one bit to be transmitted in any single baud interval.

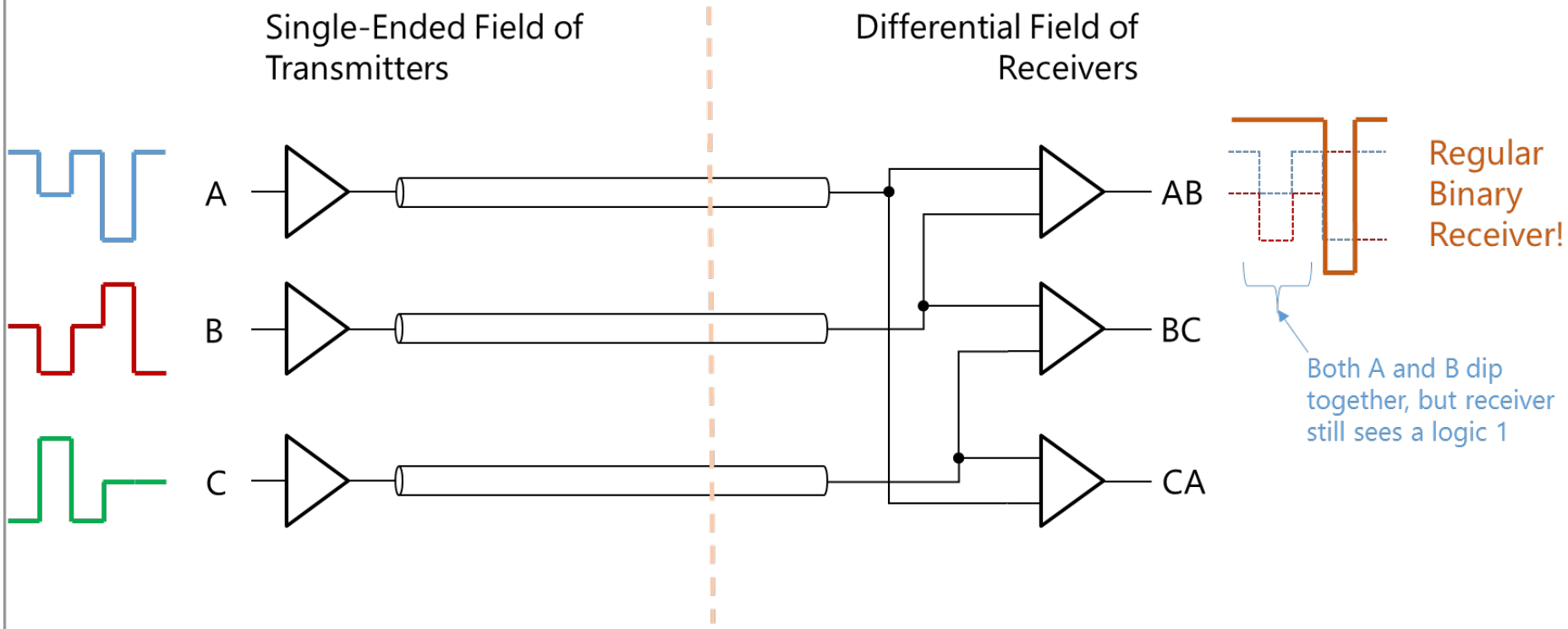
39 Claims, 15 Drawing Sheets

George Wiley, Qualcomm

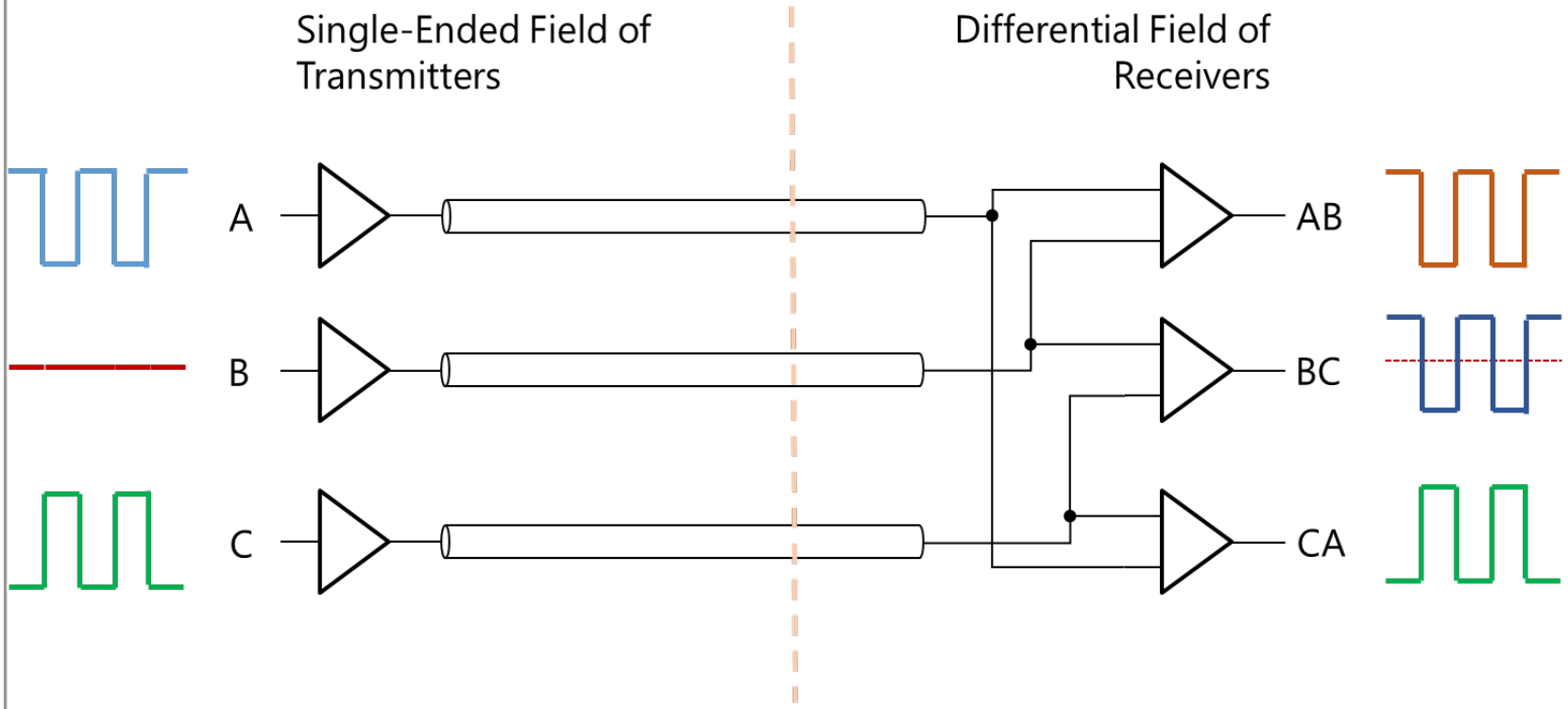
Basic Concept of Three Phase Encoding



Three Voltage Levels Per Wire Ensure Proper Differential Reception



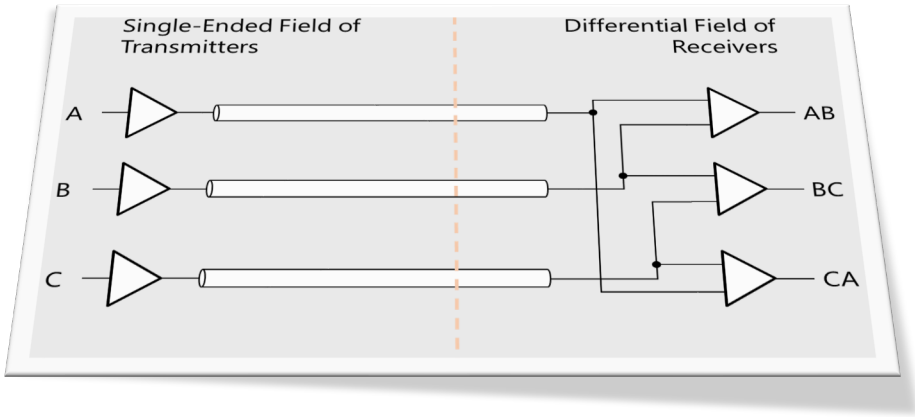
Always-Toggle Design Allows for Simple Clock Recovery (100% Transition Density)



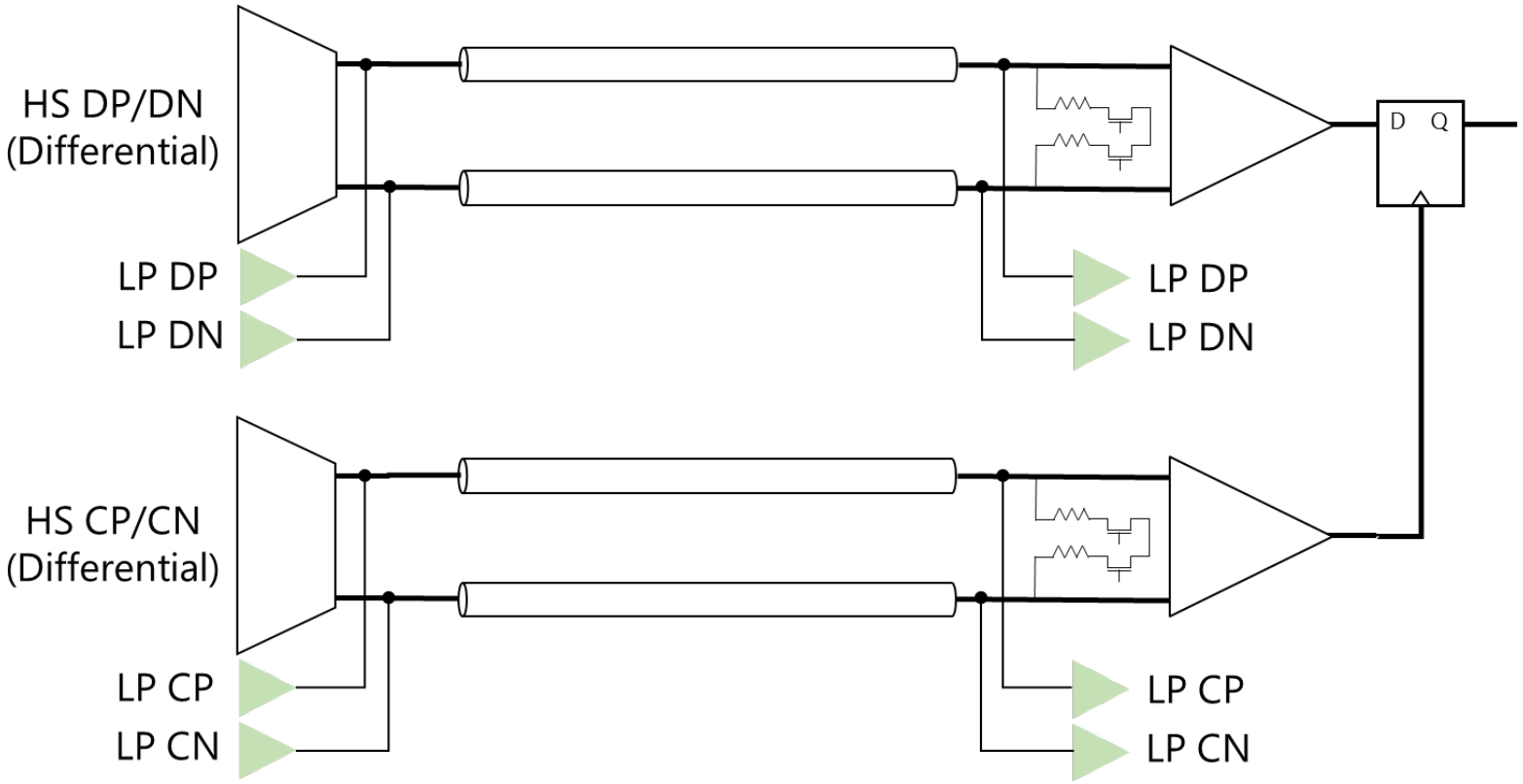
Key Takeaways

Three-level single-ended signaling

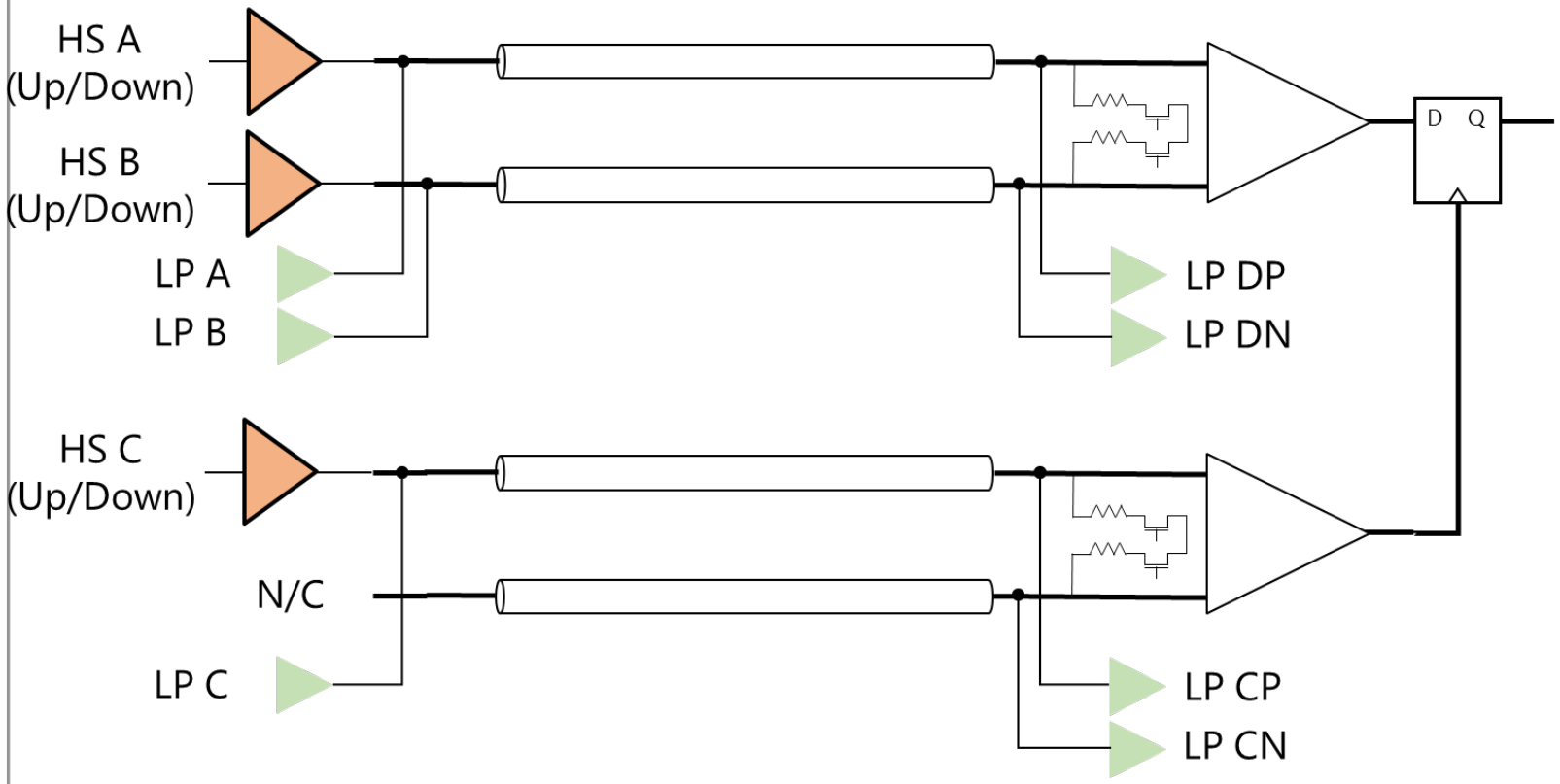
Non-deterministic transitions based on self-clocked mapping and encoding algorithm



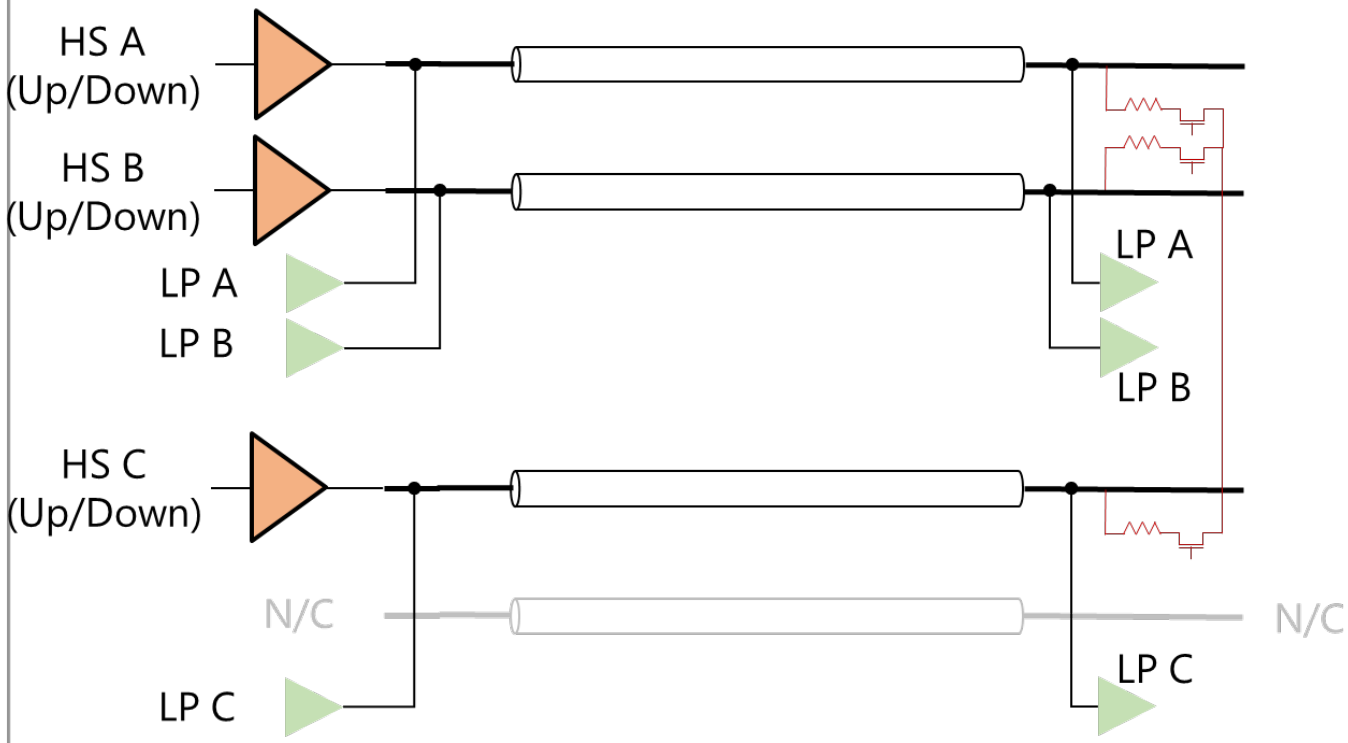
Evolution from D-PHY (1 Lane, 4 Wires)



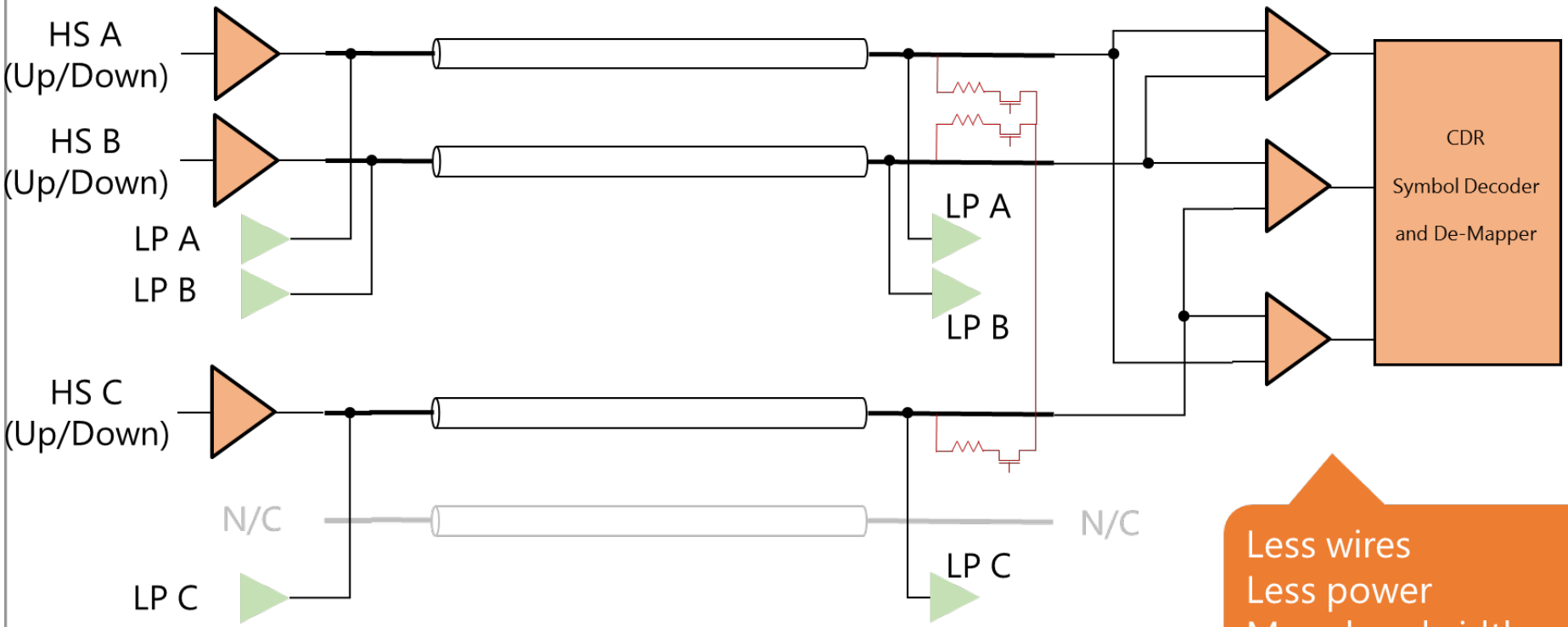
Evolution from D-PHY (1 Lane, 4 Wires)



Evolution from D-PHY (1 Lane, 4 Wires)

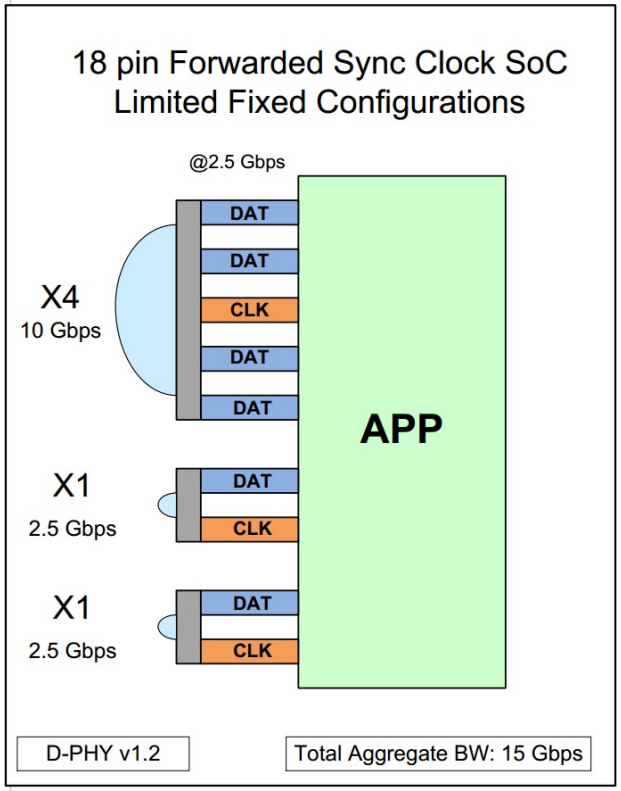


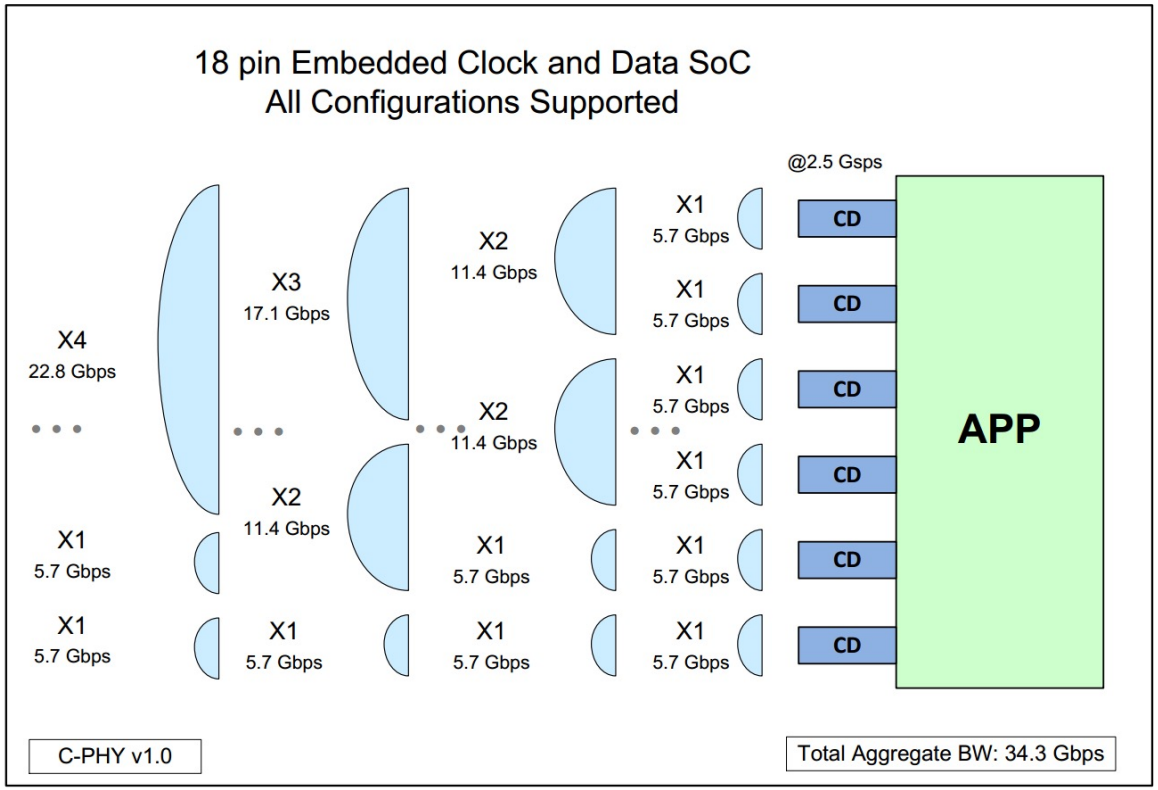
Evolution from D-PHY (1 Lane, 4 Wires)



Less wires
Less power
More bandwidth

Architecturally Flexible





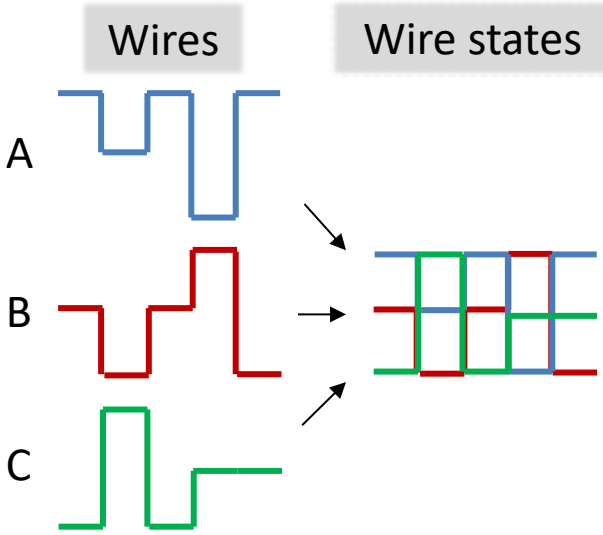
Source: MIPI Alliance

Mapping and Encoding

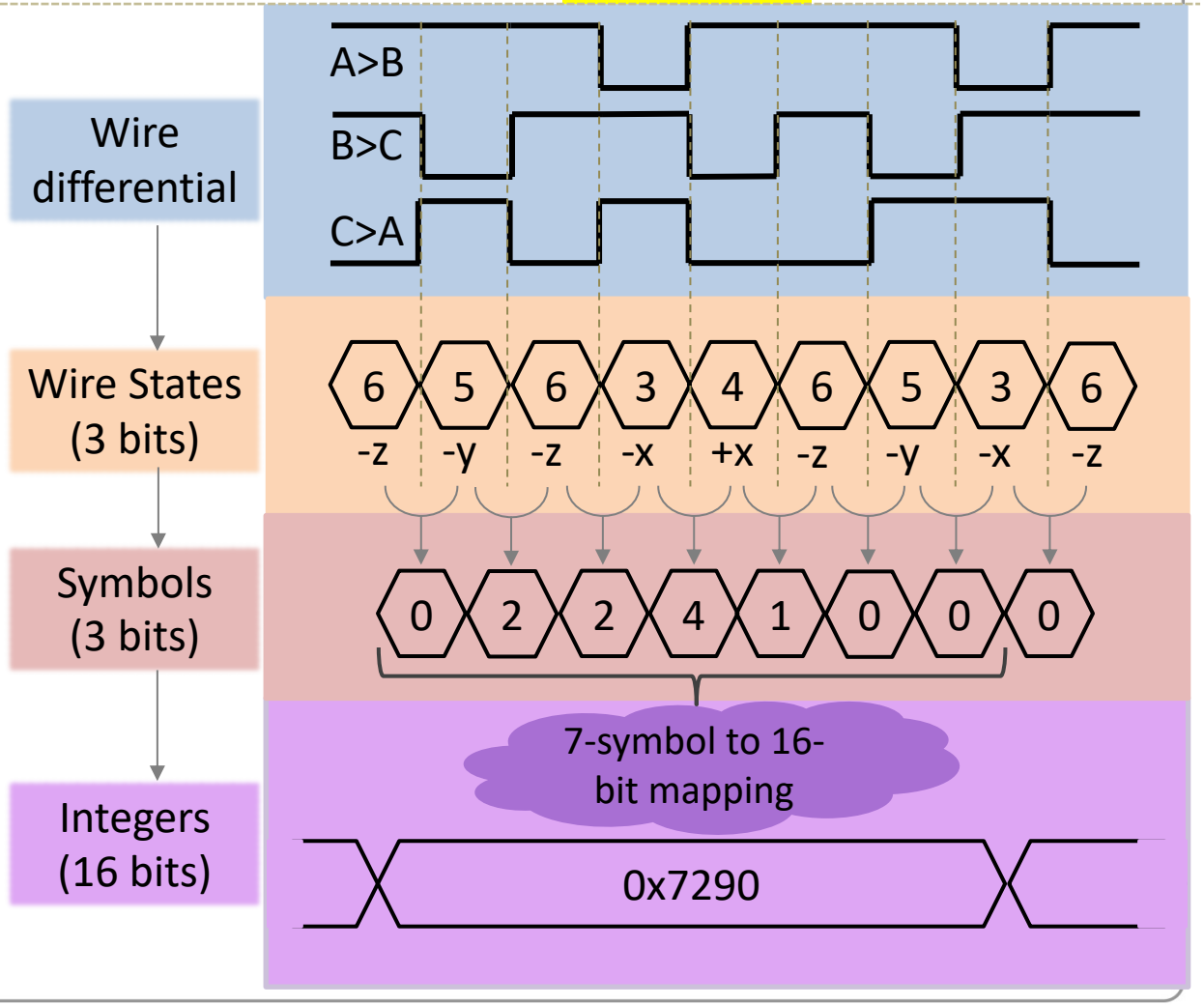
C-PHY Data Types

ANALOG

DIGITAL



- 3 wires per lane
- 3-level wires (LOW, MID, HIGH)
- Every unit interval must contain LOW, MID, and HIGH wires
- No two consecutive identical wire states



Wire States

- A wire state is the collection of A, B, and C
- 6 possible wire states

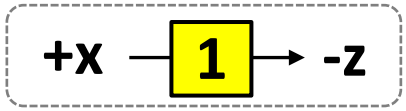
ANALOG			DIGITAL (3 bits)			
A	**B**	**C**	**A>B**	**B>C**	**C>A**	**Wire state name**
HIGH	LOW	MID	1	0	0	**+x**
LOW	HIGH	MID	0	1	1	**-x**
MID	HIGH	LOW	0	1	0	**+y**
MID	LOW	HIGH	1	0	1	**-y**
LOW	MID	HIGH	0	0	1	**+z**
HIGH	MID	LOW	1	1	0	**-z**

Symbols: Now We're Transmitting!

- A symbol represents a transition between two wire states
- 5 possible symbols

	Symbol (3 bits)		
	Flip	**Rotate**	**Polarity**
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	DC	DC

Example:



Flip	
0	-
1	Same letter, toggle sign.

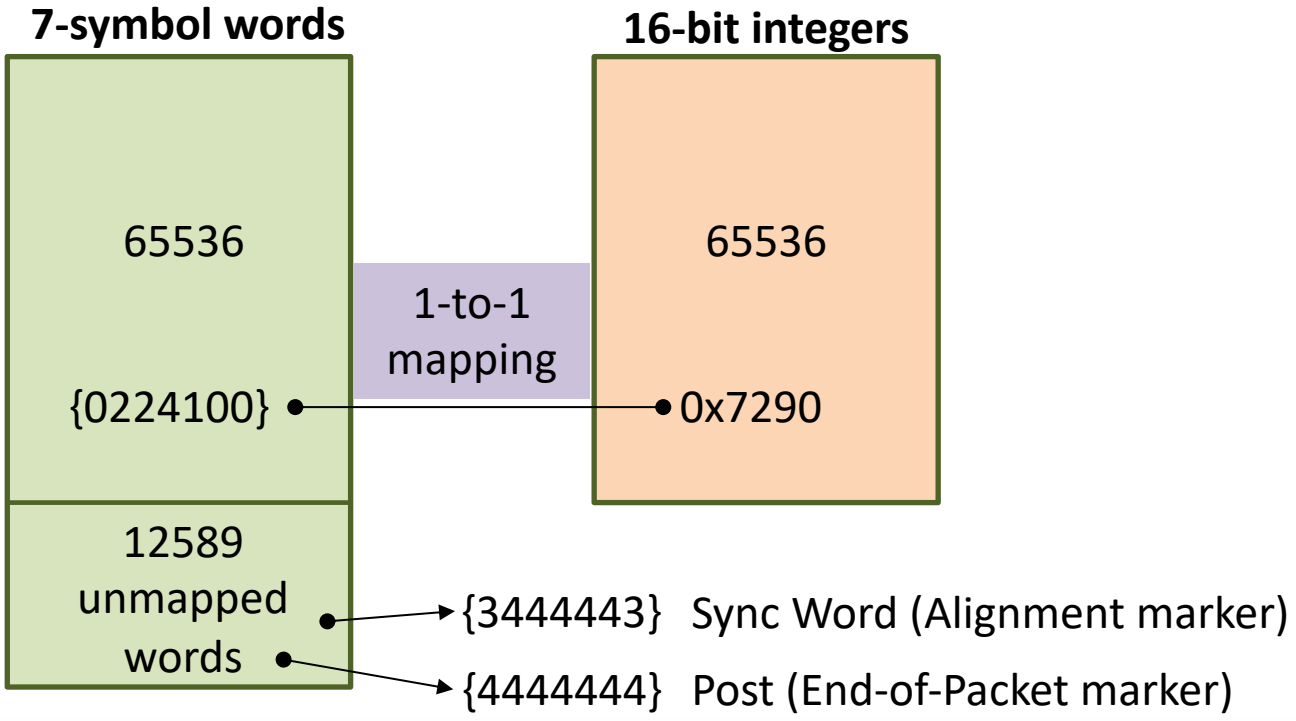
Rotate	
0	Decr. letter
1	Incr. letter

Polarity	
0	-
1	Toggle sign

Mapping 7 Symbols ⟷ 16-bit Integers

- C-PHY defines a mapping between 7-symbol words and 16-bit integers

Number of **7-symbol words**: $5^7 = 78125$ > Number of **16-bit integers**: $2^{16} = 65536$



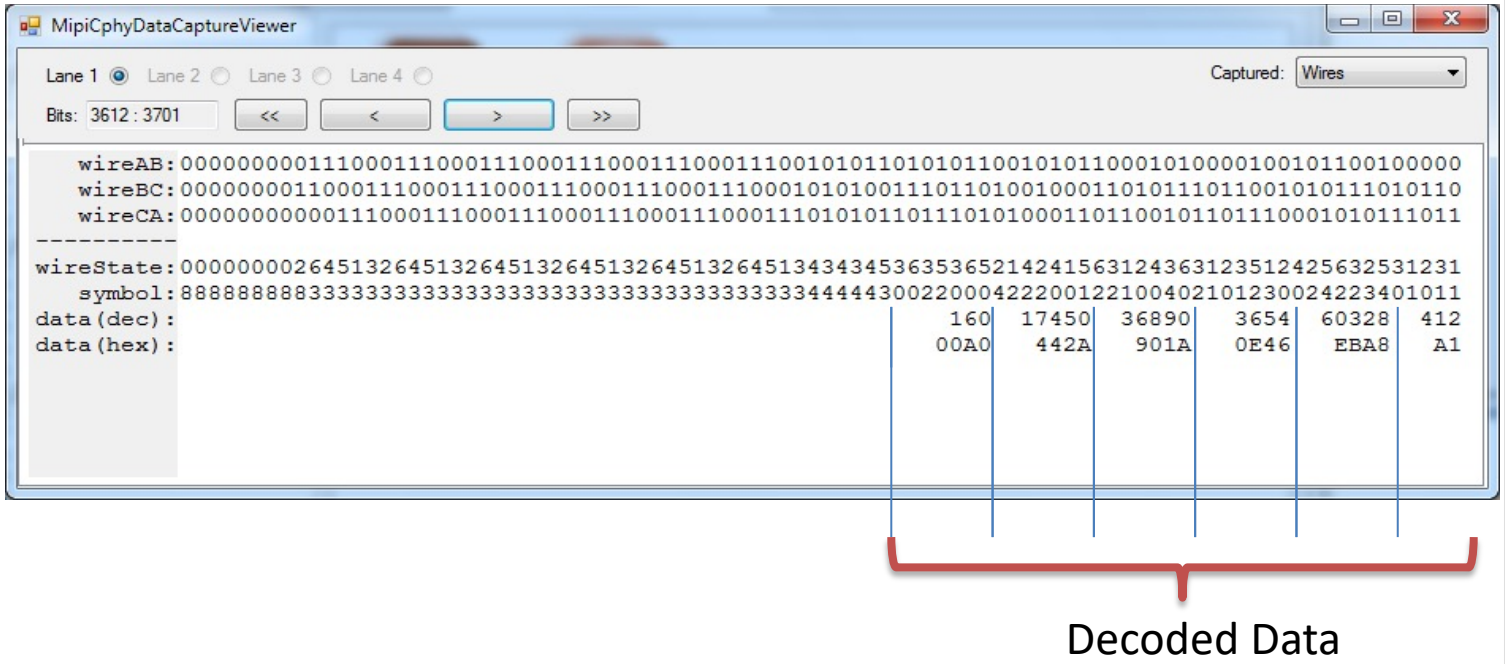
Well Defined Algorithms from MIPI Alliance

Integer range	Flip bits	Integer	Rotate and Polarity bits
0xFC00-0xFFFF:	Flip=[1,0,1,0,0,0,0]	Integer=[1, 1, 1, 1, 1,	1,ro5,po5,ro3,po3,ro2,po2,ro1,po1,ro0,po0]
0xF800-0xFBFF:	Flip=[0,1,1,0,0,0,0]	Integer=[1, 1, 1, 1, 1,	0,ro6,po6,ro3,po3,ro2,po2,ro1,po1,ro0,po0]
0xF400-0xF7FF:	Flip=[1,0,0,1,0,0,0]	Integer=[1, 1, 1, 1, 0,	1,ro5,po5,ro4,po4,ro2,po2,ro1,po1,ro0,po0]
0xF000-0xF3FF:	Flip=[0,1,0,1,0,0,0]	Integer=[1, 1, 1, 1, 0,	0,ro6,po6,ro4,po4,ro2,po2,ro1,po1,ro0,po0]
0xEC00-0xEFFF:	Flip=[0,0,1,1,0,0,0]	Integer=[1, 1, 1, 0, 1,	1,ro6,po6,ro5,po5,ro2,po2,ro1,po1,ro0,po0]
0xE800-0xEBFF:	Flip=[1,0,0,0,1,0,0]	Integer=[1, 1, 1, 0, 1,	0,ro5,po5,ro4,po4,ro3,po3,ro1,po1,ro0,po0]
0xE400-0xE7FF:	Flip=[0,1,0,0,1,0,0]	Integer=[1, 1, 1, 0, 0,	1,ro6,po6,ro4,po4,ro3,po3,ro1,po1,ro0,po0]
0xE000-0xE3FF:	Flip=[0,0,1,0,1,0,0]	Integer=[1, 1, 1, 0, 0,	0,ro6,po6,ro5,po5,ro3,po3,ro1,po1,ro0,po0]
0xDC00-0xDFFF:	Flip=[0,0,0,1,1,0,0]	Integer=[1, 1, 0, 1, 1,	1,ro6,po6,ro5,po5,ro4,po4,ro1,po1,ro0,po0]
0xD800-0xDBFF:	Flip=[1,0,0,0,0,1,0]	Integer=[1, 1, 0, 1, 1,	0,ro5,po5,ro4,po4,ro3,po3,ro2,po2,ro0,po0]
0xD400-0xD7FF:	Flip=[0,1,0,0,0,1,0]	Integer=[1, 1, 0, 1, 0,	1,ro6,po6,ro4,po4,ro3,po3,ro2,po2,ro0,po0]
0xD000-0xD3FF:	Flip=[0,0,1,0,0,1,0]	Integer=[1, 1, 0, 1, 0,	0,ro6,po6,ro5,po5,ro3,po3,ro2,po2,ro0,po0]
0xCC00-0xCFFF:	Flip=[0,0,0,1,0,1,0]	Integer=[1, 1, 0, 0, 1,	1,ro6,po6,ro5,po5,ro4,po4,ro2,po2,ro0,po0]
0xC800-0xCBFF:	Flip=[0,0,0,0,1,1,0]	Integer=[1, 1, 0, 0, 1,	0,ro6,po6,ro5,po5,ro4,po4,ro3,po3,ro0,po0]
0xC400-0xC7FF:	Flip=[1,0,0,0,0,0,1]	Integer=[1, 1, 0, 0, 0,	1,ro5,po5,ro4,po4,ro3,po3,ro2,po2,ro1,po1]
0xC000-0xC3FF:	Flip=[0,1,0,0,0,0,1]	Integer=[1, 1, 0, 0, 0,	0,ro6,po6,ro4,po4,ro3,po3,ro2,po2,ro1,po1]
0xBC00-0xBFFF:	Flip=[0,0,1,0,0,0,1]	Integer=[1, 0, 1, 1, 1,	1,ro6,po6,ro5,po5,ro3,po3,ro2,po2,ro1,po1]
0xB800-0xBBFF:	Flip=[0,0,0,1,0,0,1]	Integer=[1, 0, 1, 1, 1,	0,ro6,po6,ro5,po5,ro4,po4,ro2,po2,ro1,po1]
0xB400-0xB7FF:	Flip=[0,0,0,0,1,0,1]	Integer=[1, 0, 1, 1, 0,	1,ro6,po6,ro5,po5,ro4,po4,ro3,po3,ro1,po1]
0xB000-0xB3FF:	Flip=[0,0,0,0,0,1,1]	Integer=[1, 0, 1, 1, 0,	0,ro6,po6,ro5,po5,ro4,po4,ro3,po3,ro2,po2]
0xA000-0xAFFF:	Flip=[1,0,0,0,0,0,0]	Integer=[1, 0, 1, 0,	ro5,po5,ro4,po4,ro3,po3,ro2,po2,ro1,po1,ro0,po0]
0x9000-0x9FFF:	Flip=[0,1,0,0,0,0,0]	Integer=[1, 0, 0, 1,	ro6,po6,ro4,po4,ro3,po3,ro2,po2,ro1,po1,ro0,po0]
0x8000-0x8FFF:	Flip=[0,0,1,0,0,0,0]	Integer=[1, 0, 0, 0,	ro6,po6,ro5,po5,ro3,po3,ro2,po2,ro1,po1,ro0,po0]
0x7000-0x7FFF:	Flip=[0,0,0,1,0,0,0]	Integer=[0, 1, 1, 1,	ro6,po6,ro5,po5,ro4,po4,ro2,po2,ro1,po1,ro0,po0]
0x6000-0x6FFF:	Flip=[0,0,0,0,1,0,0]	Integer=[0, 1, 1, 0,	ro6,po6,ro5,po5,ro4,po4,ro3,po3,ro1,po1,ro0,po0]
0x5000-0x5FFF:	Flip=[0,0,0,0,0,1,0]	Integer=[0, 1, 0, 1,	ro6,po6,ro5,po5,ro4,po4,ro3,po3,ro2,po2,ro0,po0]
0x4000-0x4FFF:	Flip=[0,0,0,0,0,0,1]	Integer=[0, 1, 0, 0,	ro6,po6,ro5,po5,ro4,po4,ro3,po3,ro2,po2,ro1,po1]
0x0000-0x3FFF:	Flip=[0,0,0,0,0,0,0]	Integer=[0, 0,	ro6,po6,ro5,po5,ro4,po4,ro3,po3,ro2,po2,ro1,po1,ro0,po0]

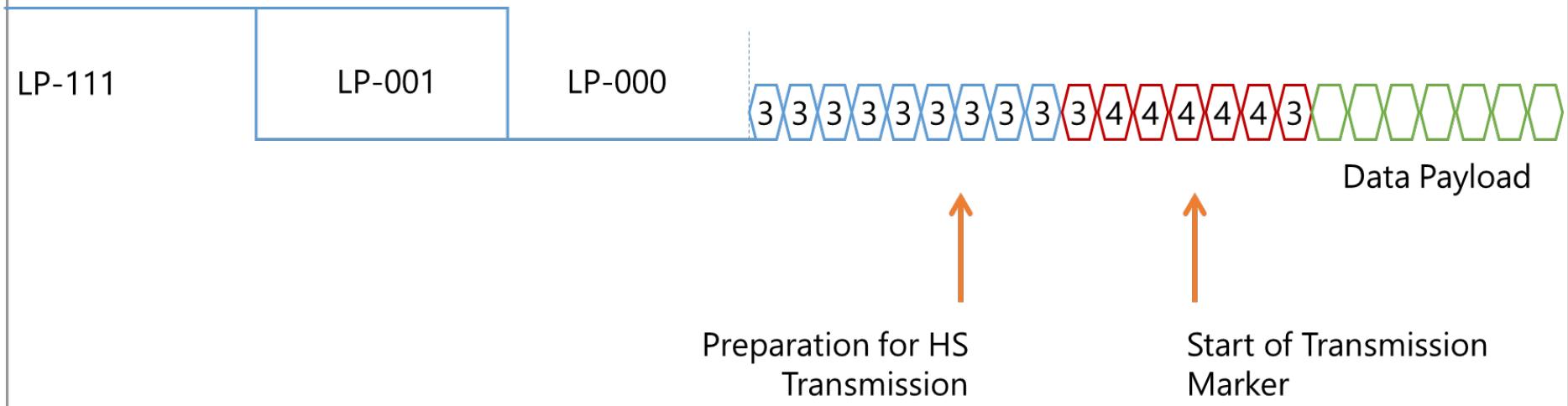
“Don’t Even Worry About It”

Eco-System Is Developed for Tools

Three-Phase Signals

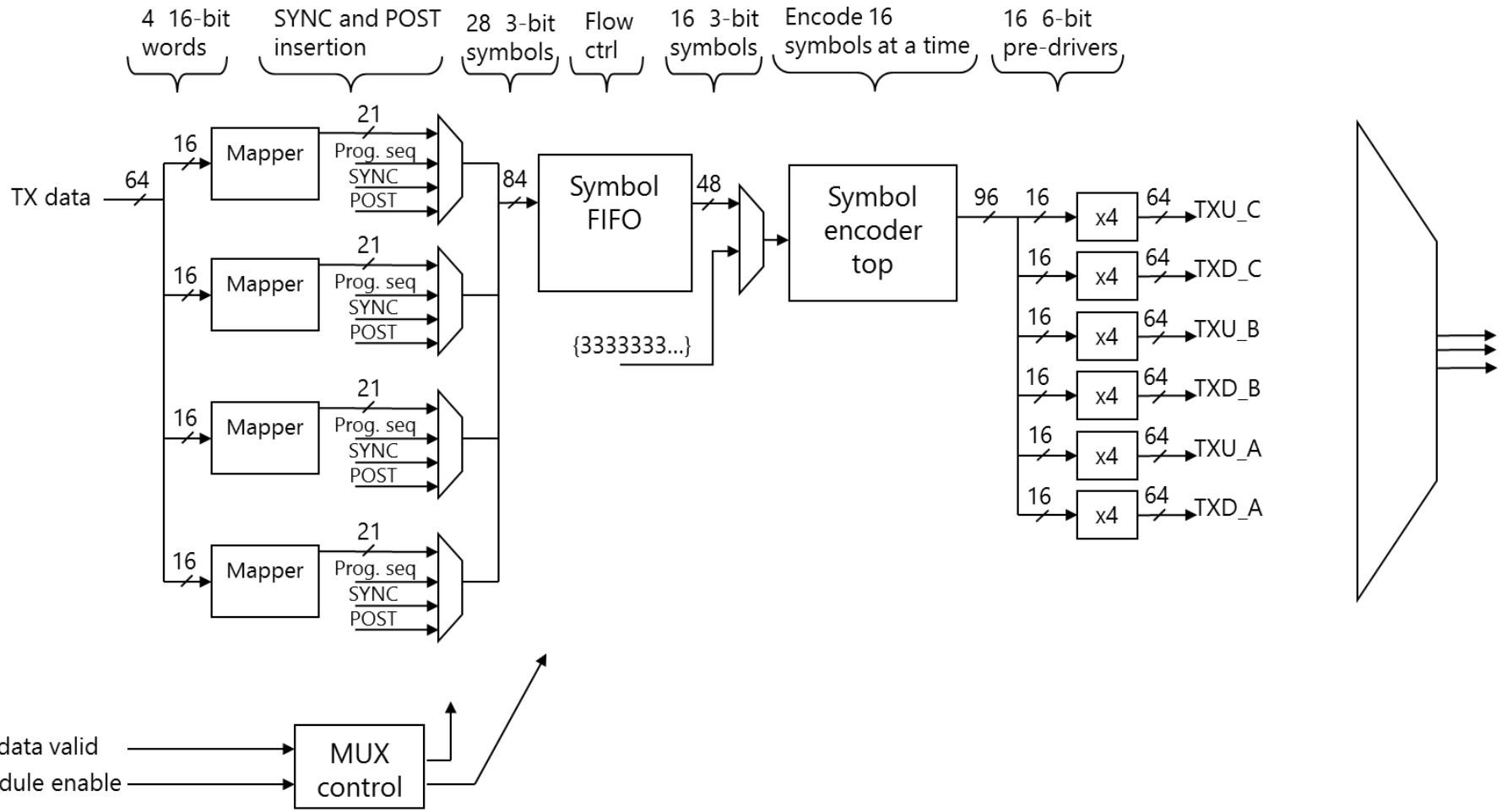


Anatomy of a Packet Transmission

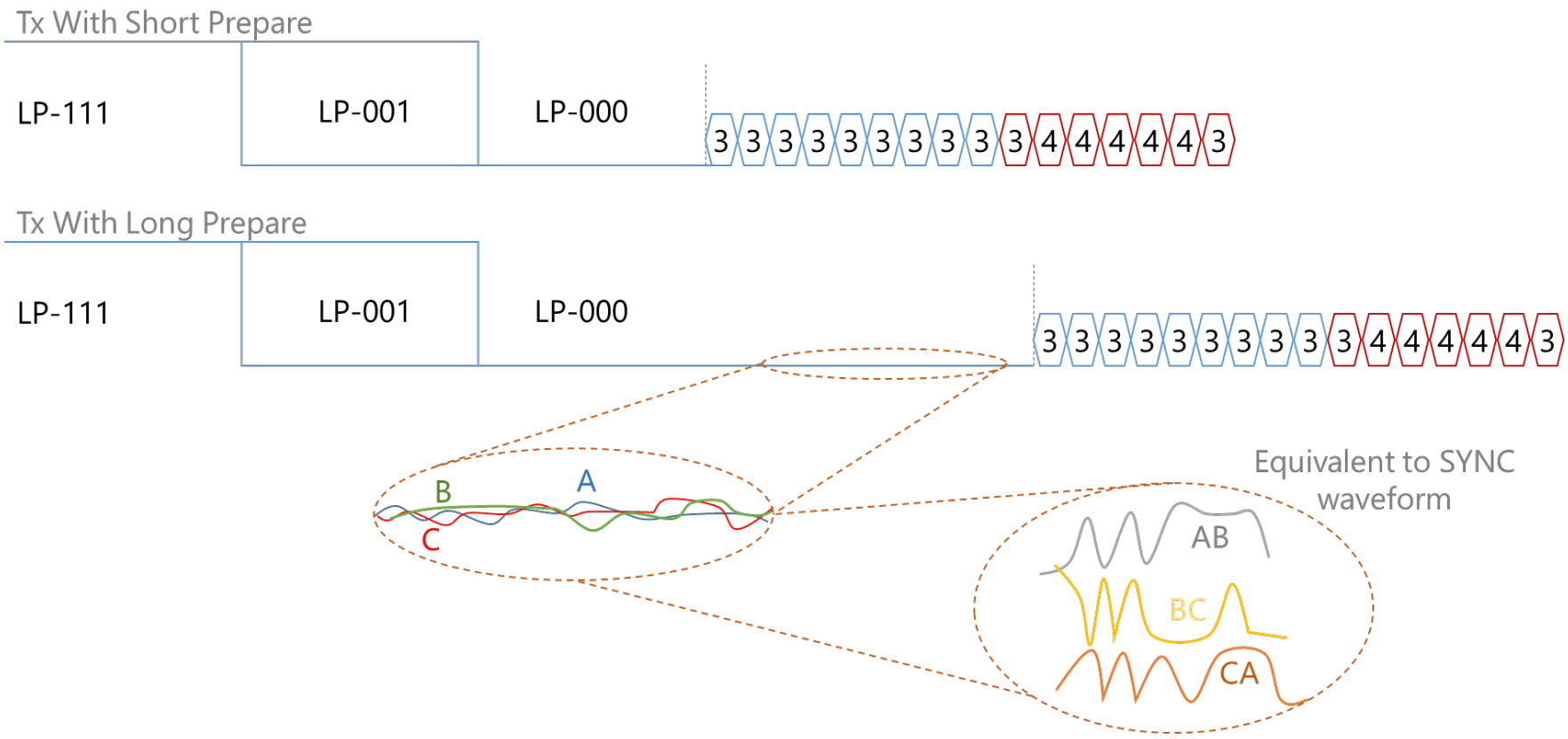


Practical Experiences

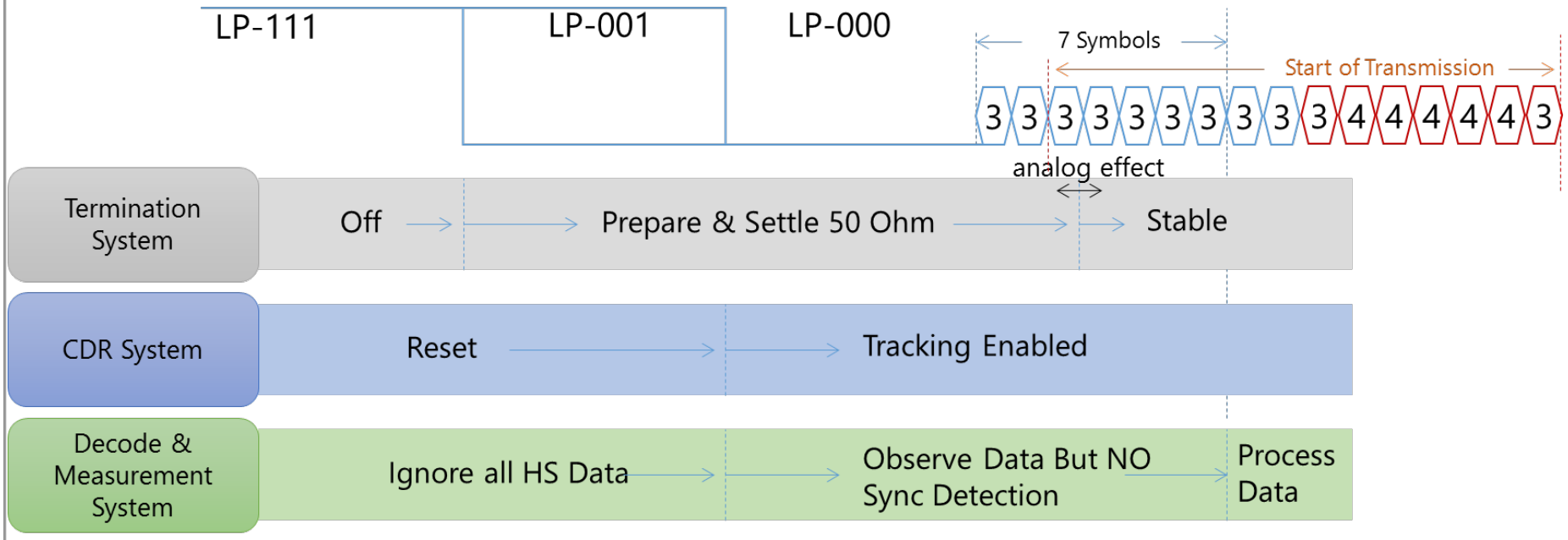
Tx: Both Mapping and Encoding Before Serializer



Rx: Avoiding False Sync Detection (Problem Statement)

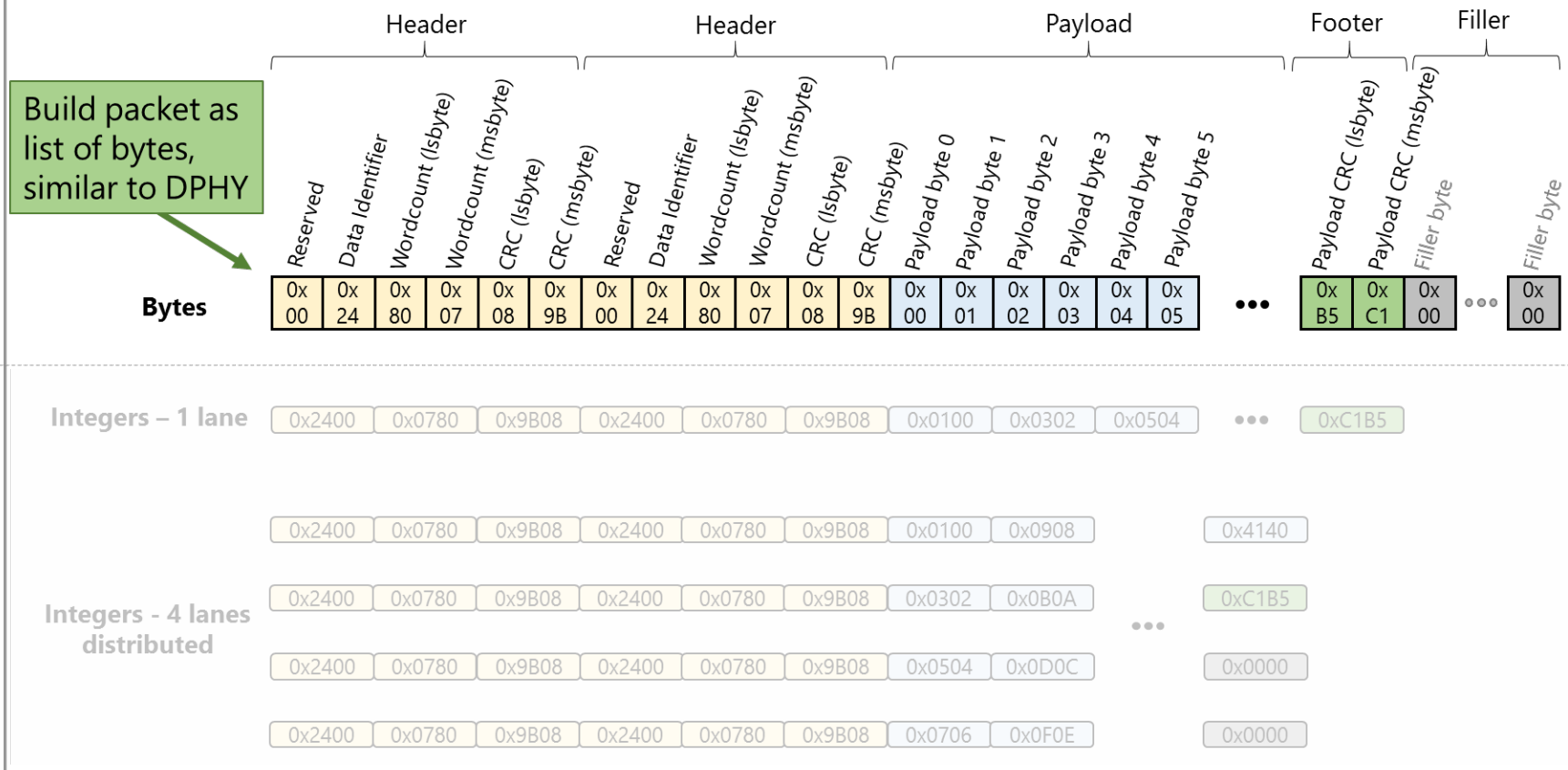


Rx: Avoiding False Sync Detection (Solution)

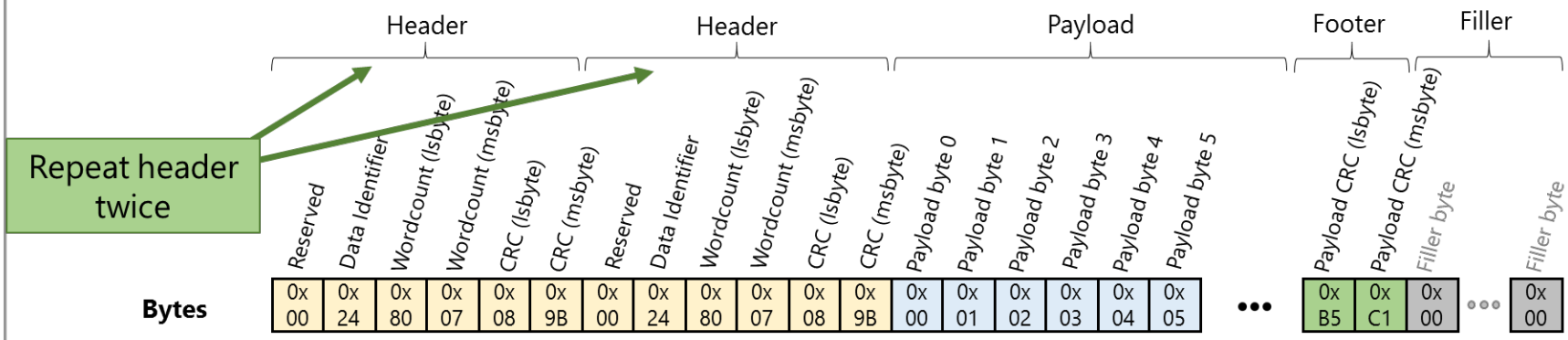


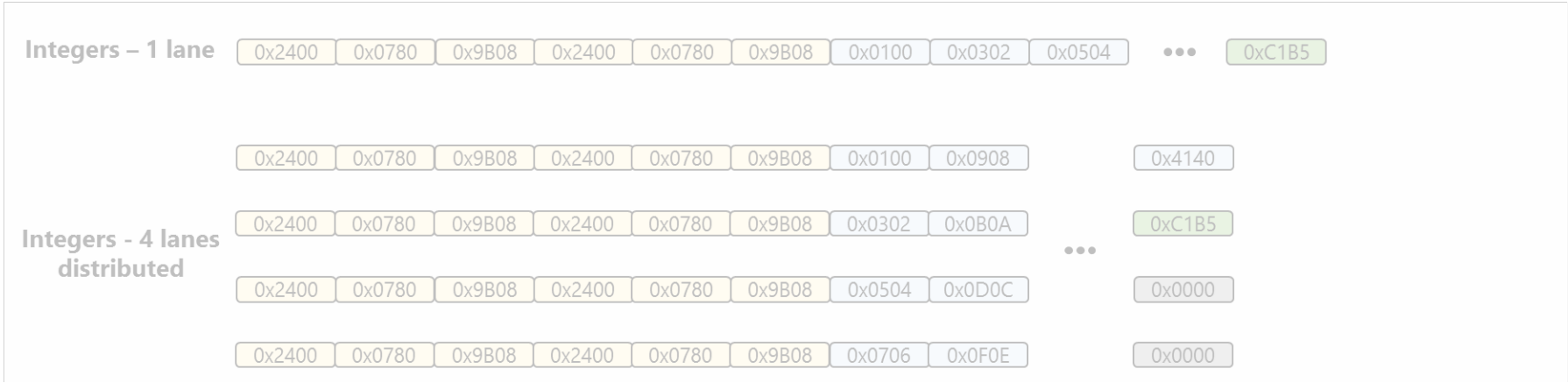
Detect SYNC with Pre-End as Marker for Start of Transmission

CSI-2 Long Packets in C-PHY

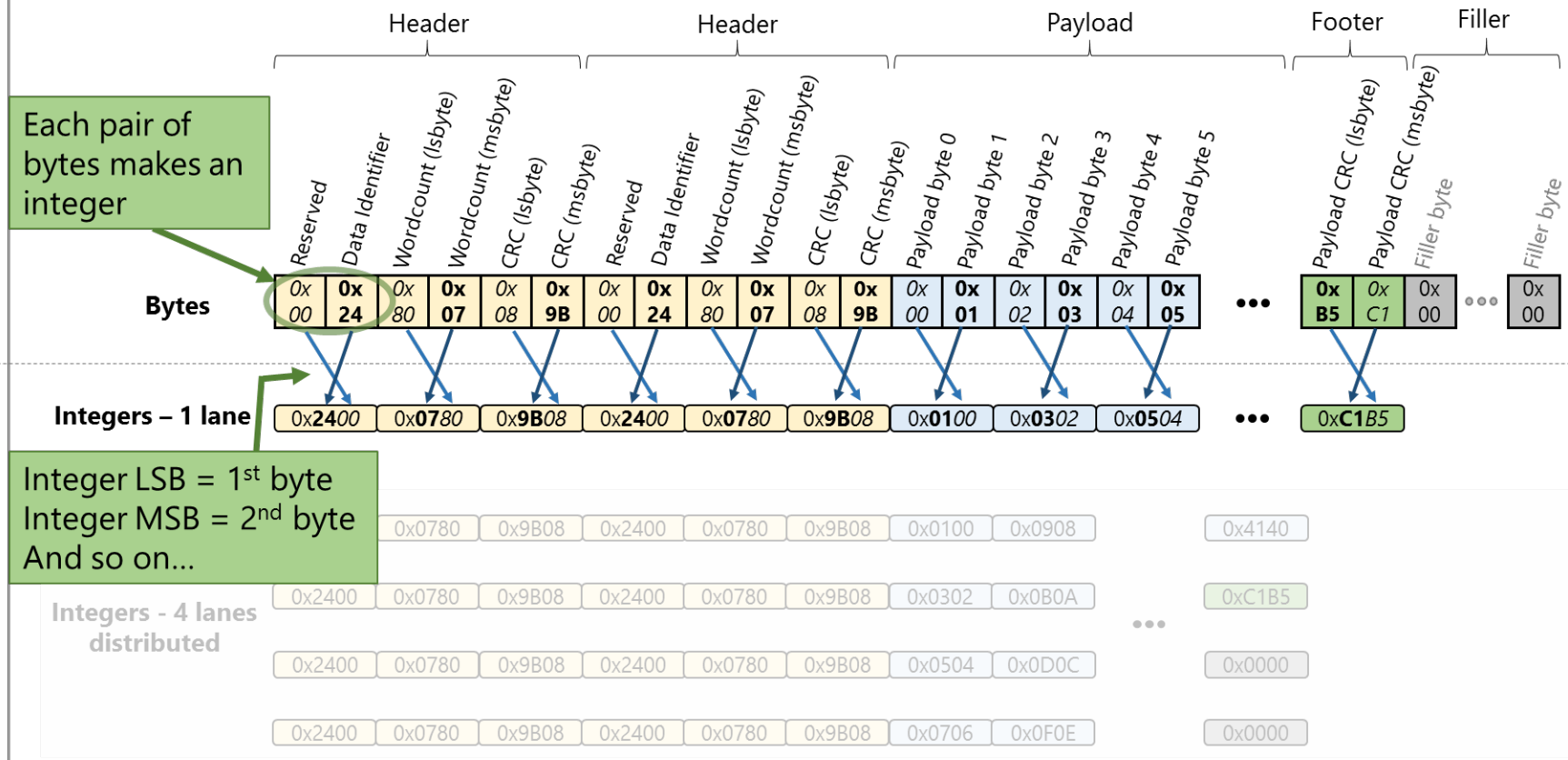


CSI-2 Long Packets in C-PHY

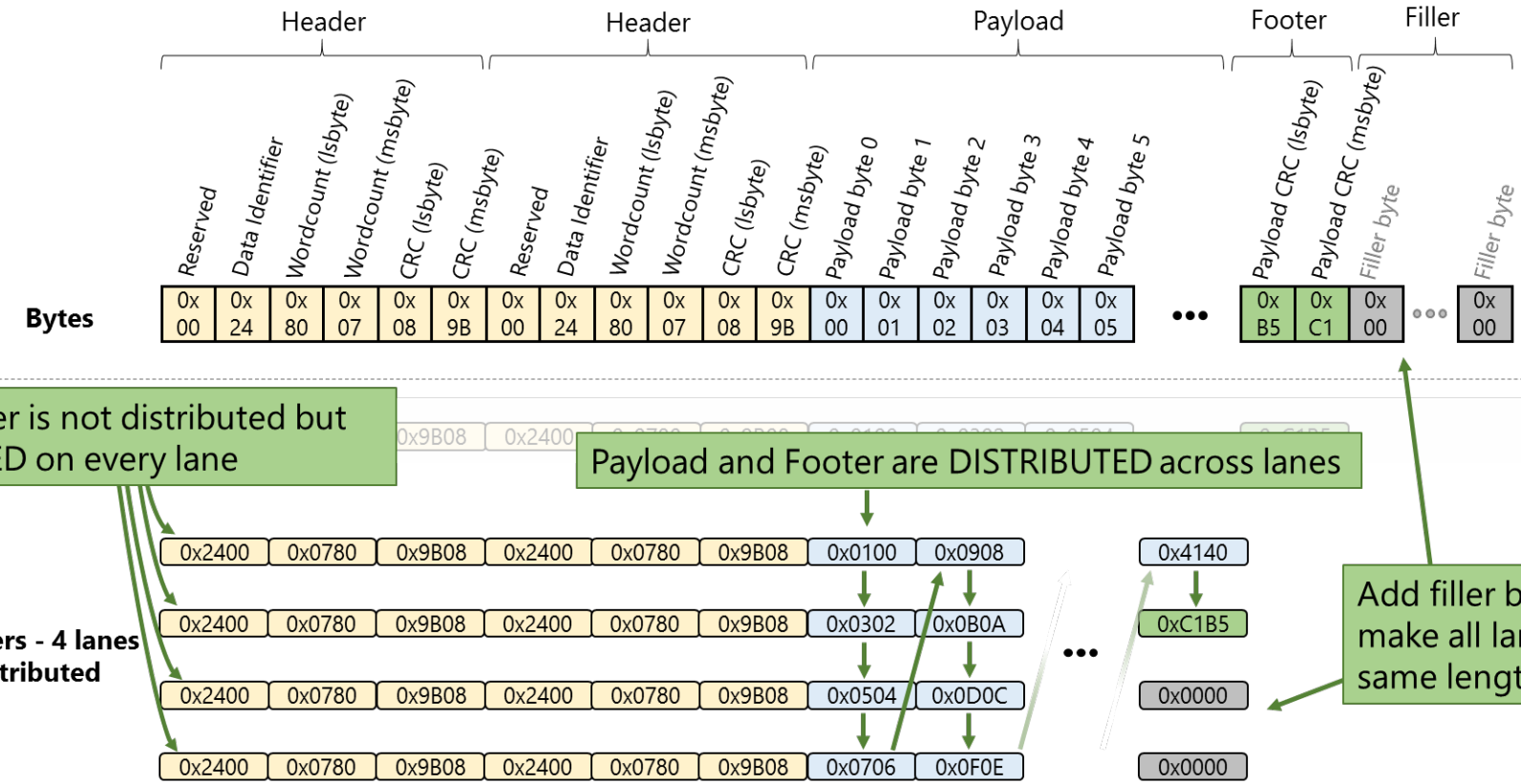




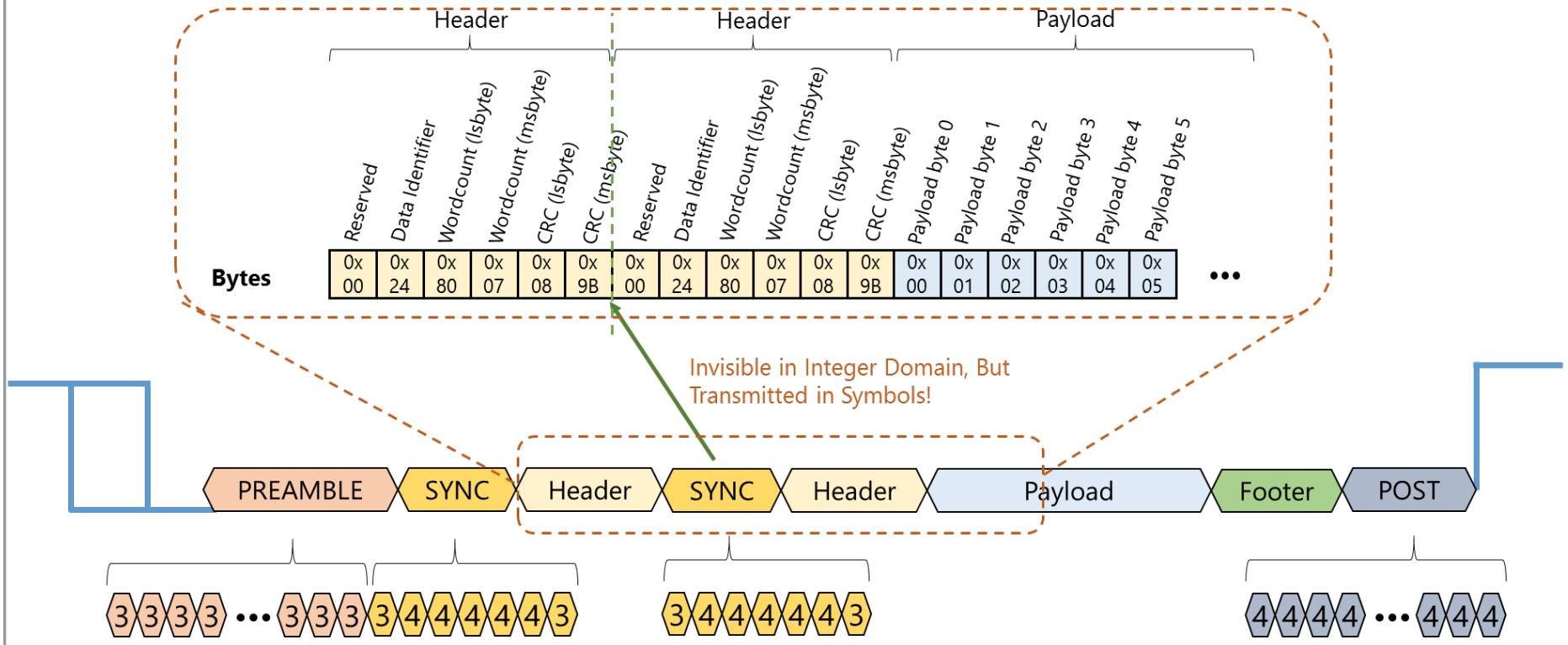
CSI-2 Long Packets in C-PHY



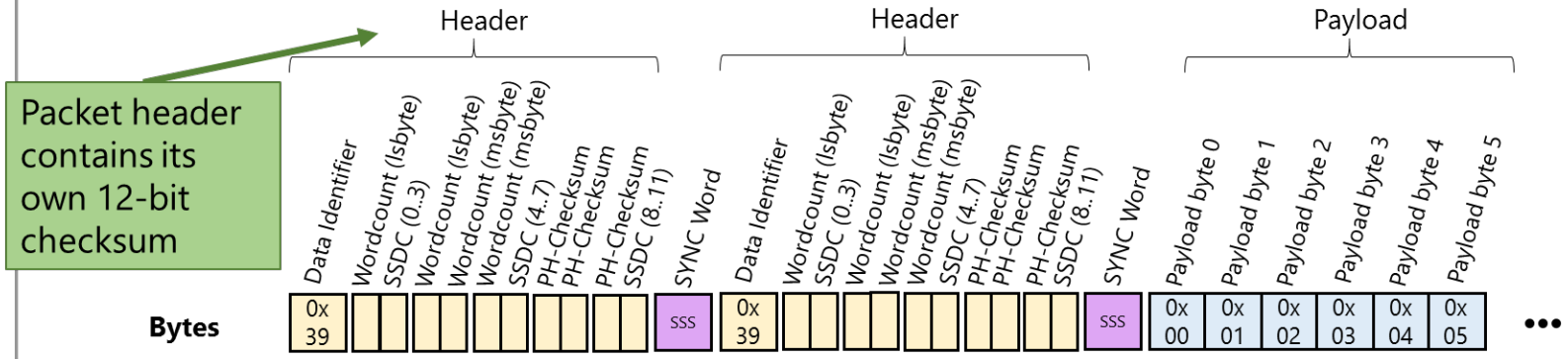
CSI-2 Long Packets in C-PHY



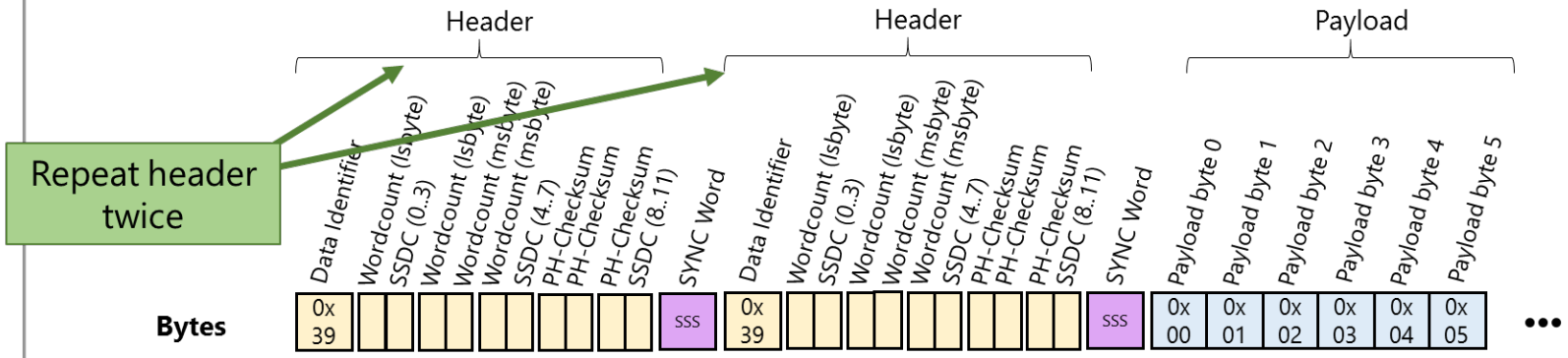
CSI-2 Long Packets in C-PHY: The Invisible SYNC



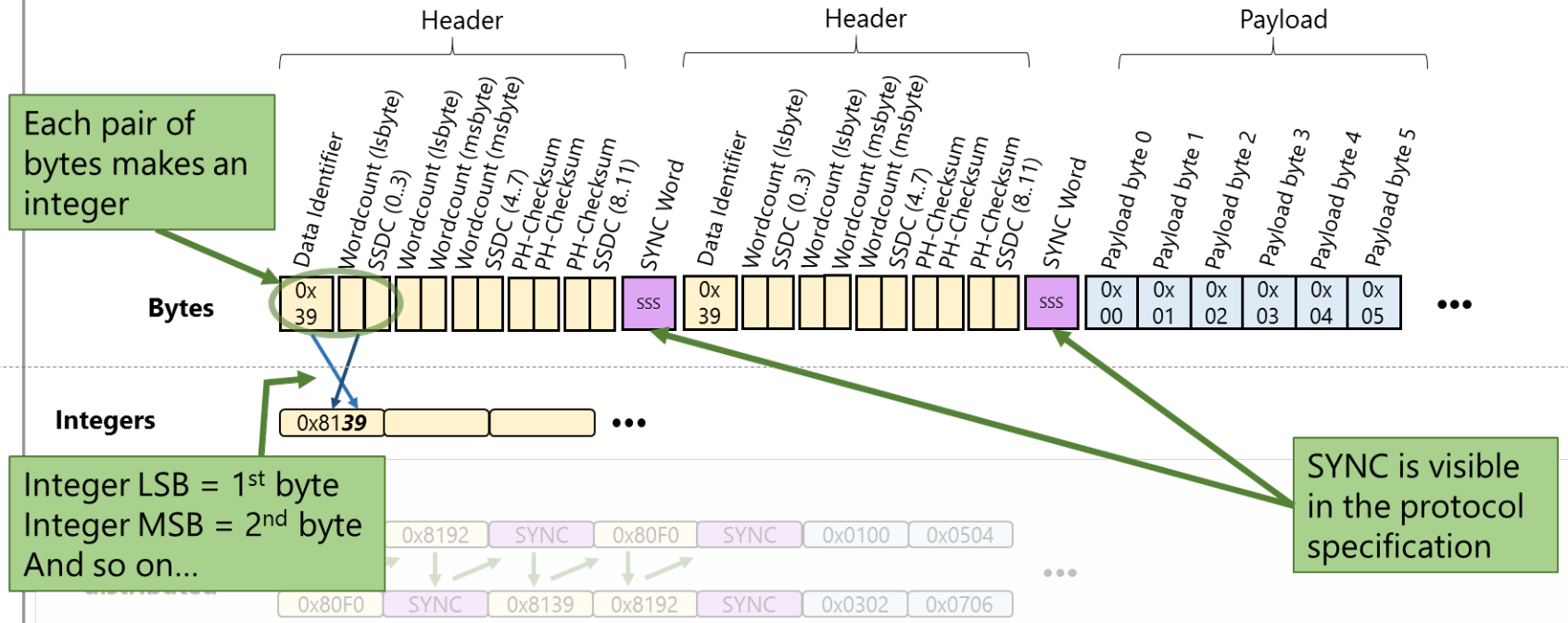
DSI-2 Long Packets in C-PHY



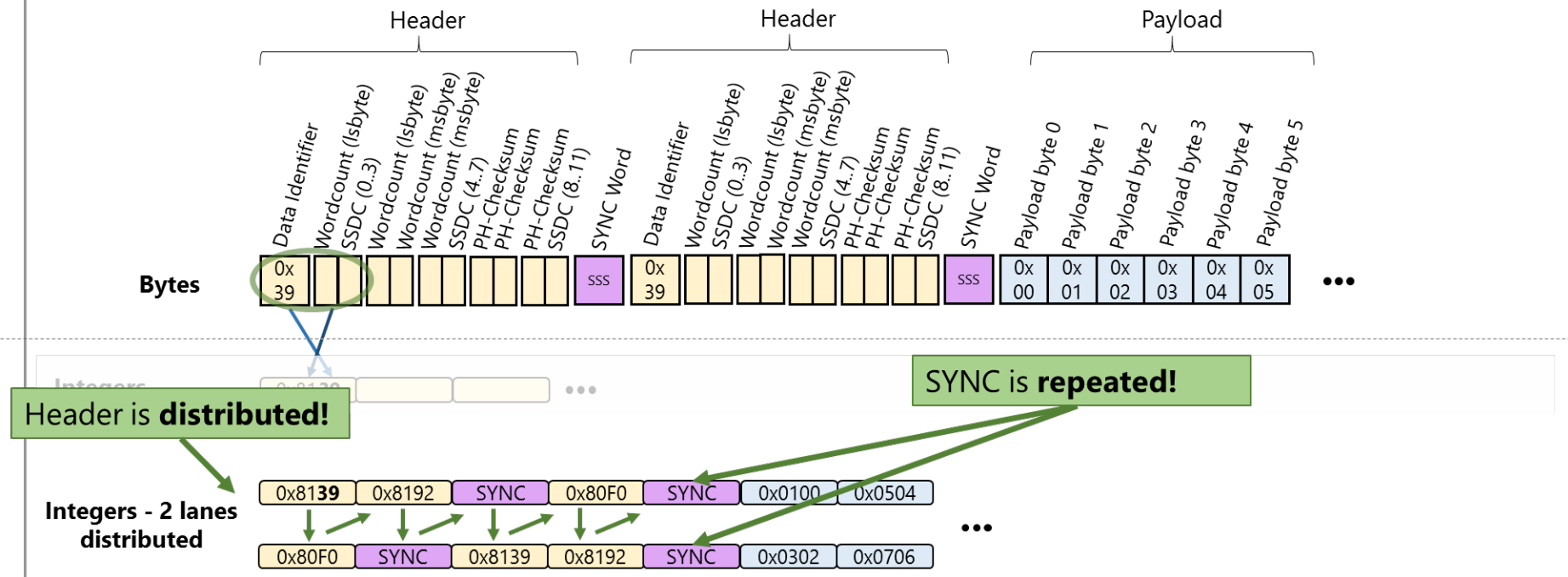
DSI-2 Long Packets in C-PHY



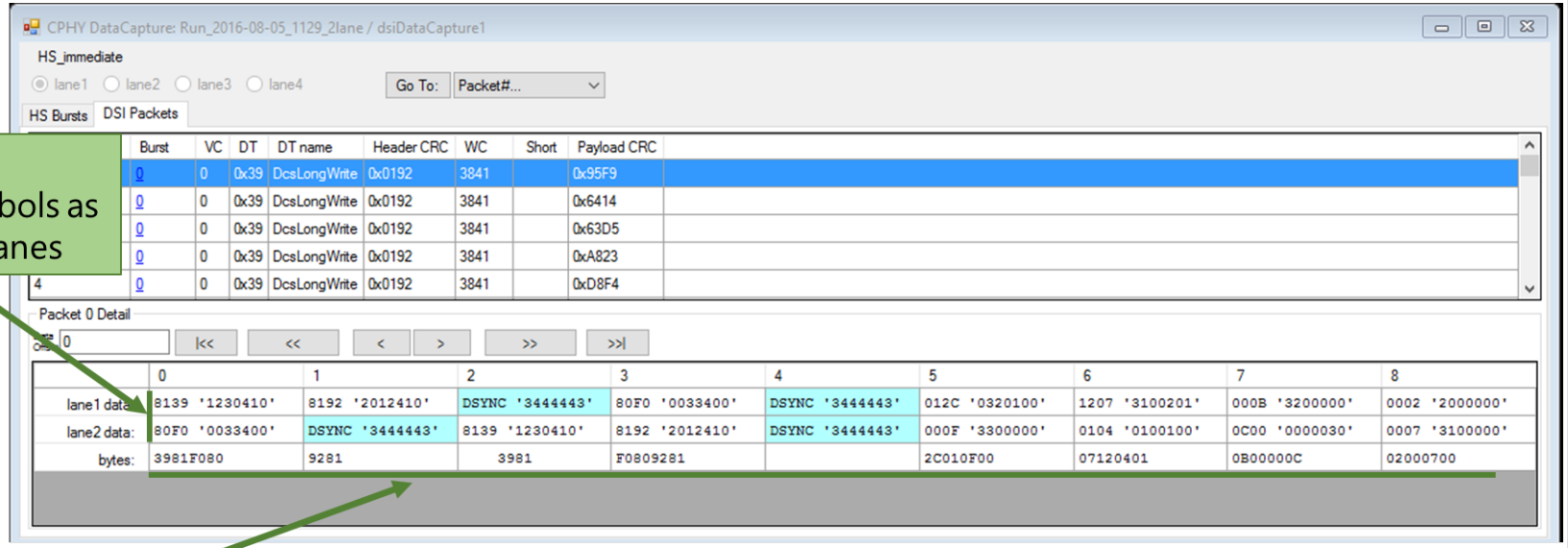
DSI-2 Long Packets in C-PHY



DSI-2 Long Packets in C-PHY



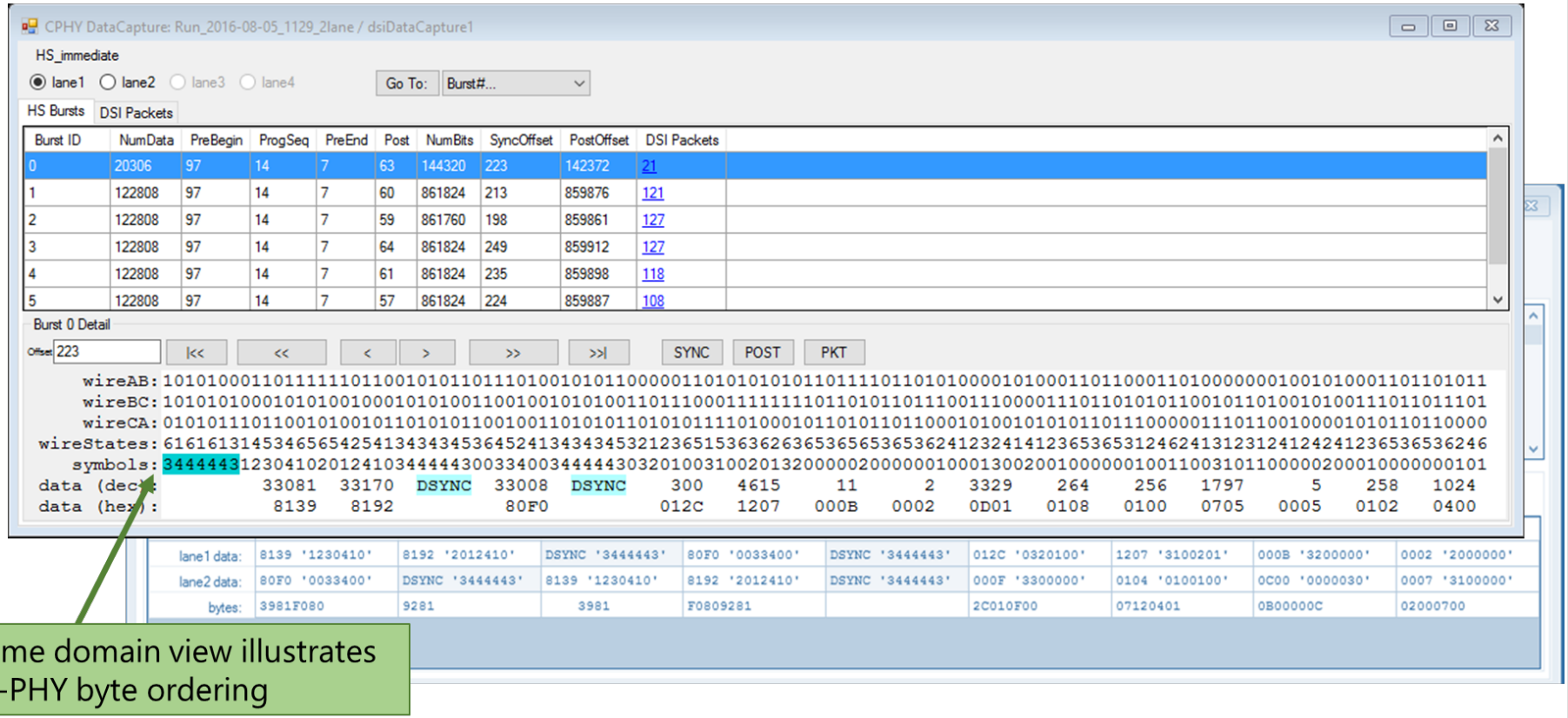
DSI-2 Sample Protocol Analyzer Trace



Distributed integers/symbols as seen on the lanes

Transmitted bytes

DSI-2 Sample Protocol Analyzer Trace



Time domain view illustrates C-PHY byte ordering

Key Takeaways

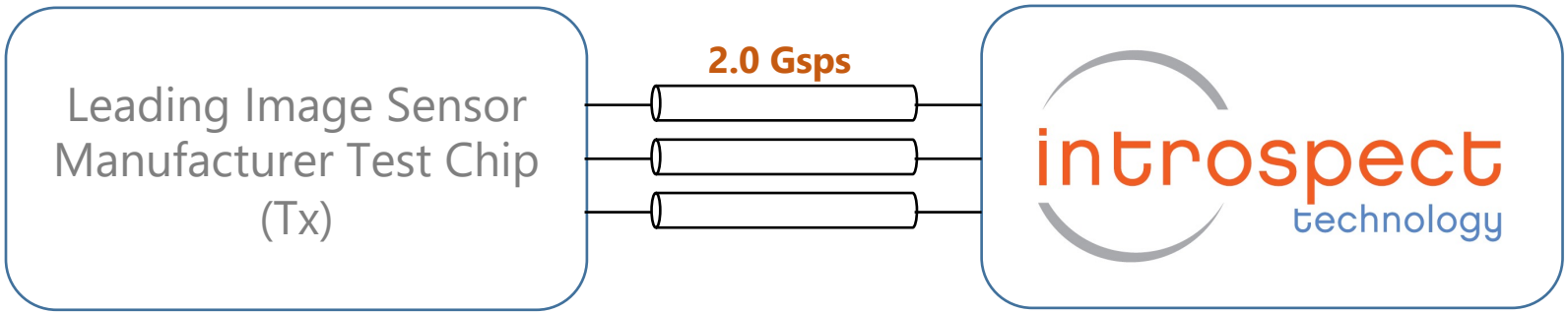
Tx mapping and encoding in parallel domain

Rx false sync avoidance required pre-begin monitoring

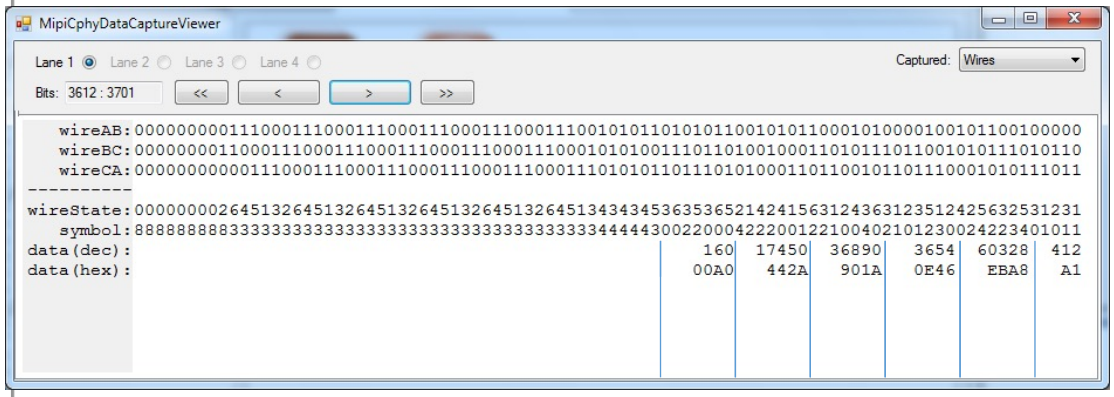
Packet header definition required careful design of SYNC manipulation (both Tx and Rx)

CSI-2 & DSI-2 treat SYNC insertion differently

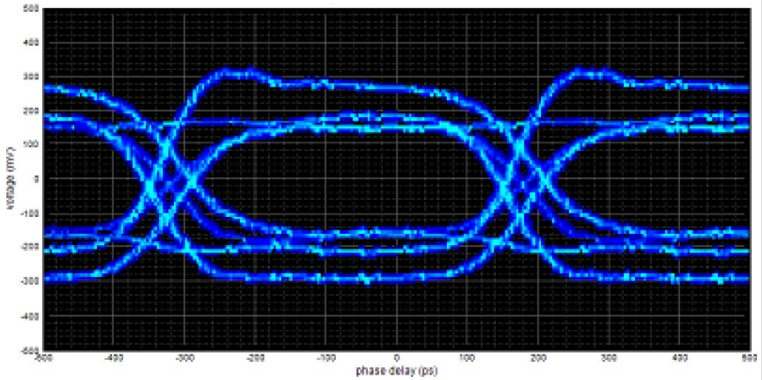
April 10, 2014: World's First C-PHY Interoperability!



First Packet Received



First Eye Diagram



Thank You!