

Universal Serial Bus 4 (USB4™) Re-Timer Specification

Apple Inc.
HP Inc.
Intel Corporation
Microsoft Corporation
Renesas Corporation
STMicroelectronics
Texas Instruments

Version 0.96
June 2020

Release History

Version	Comments	Issue Date
0.96	First release	June 2020

NOTE: Adopters may only use this USB specification to implement USB or third party functionality as expressly described in this Specification; all other uses are prohibited.

LIMITED COPYRIGHT LICENSE: The Promoters grant a conditional copyright license under the copyrights embodied in this USB Specification to use and reproduce the Specification for the sole purpose of, and solely to the extent necessary for, evaluating whether to implement the Specification in products that would comply with the specification. Without limiting the foregoing, use of the Specification for the purpose of filing or modifying any patent application to target the Specification or USB compliant products is not authorized. Except for this express copyright license, no other rights or licenses are granted, including without limitation any patent licenses. In order to obtain any additional intellectual property licenses or licensing commitments associated with the Specification a party must execute the USB Adopters Agreement. **NOTE:** By using the Specification, you accept these license terms on your own behalf and, in the case where you are doing this as an employee, on behalf of your employer.

INTELLECTUAL PROPERTY DISCLAIMER

THIS SPECIFICATION IS PROVIDED TO YOU “AS IS” WITH NO WARRANTIES WHATSOEVER INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE. THE AUTHORS OF THIS SPECIFICATION DISCLAIM ALL LIABILITY FOR INFRINGEMENT OF ANY PROPRIETARY RIGHTS, RELATING TO THE USE OR IMPLEMENTATION OF INFORMATION IN THIS SPECIFICATION. THE PROVISION OF THIS SPECIFICATION TO YOU DOES NOT PROVIDE YOU WITH ANY LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS.

Please send comments to techsup@usb.org.

For industry information, refer to the USB Implementers Forum web page at <http://www.usb.org>.

USB Type-C®, USB-C™, USB 2.0 Type-C™, and USB4™ are trademarks of the Universal Serial Bus Implementers Forum (USB-IF). DisplayPort™ is a trademark of VESA. All product names are trademarks, registered trademarks, or service marks of their respective owners.

Thunderbolt™ is a trademark of Intel Corporation. You may only use the Thunderbolt™ trademark or logo in conjunction with products designed to this specification that complete proper certification and executing a Thunderbolt™ trademark license – see usb.org/compliance for further information

Copyright © 2020, USB Promoter Group (Apple Inc., HP Inc., Intel Corporation, Microsoft Corporation, Renesas Corporation, STMicroelectronics, and Texas Instruments).

Acknowledgement of Technical Contribution

The authors of this specification would like to recognize the following people who participated in the USB4 Specification technical work group.

Apple Inc. – Promoter Company Employees			
Majd Abu Tayeh	Nimrod Agmon	Lior Aloni	Brian Baek
Omer Bar-Lev	Moshe Benyamini	Gopu Bhaskar	Carlos Calderon
David Conroy	Bill Cornelius	Scott Deandrea	William Ferry
Amit Flanter	Itay Franko	Radia Gantous	Alex Gerber
Mark Goikhman	Nir Guetta	Yair HersHKovitz	Scott Jackson
Husam Khashiboun	Alan Kobayashi	Alexei Kosut	Christine Krause
Dmitri Krichevsky	Alex Lozovik	Rachel Menes	Shlomi Mor
Shlomi Museri	Mona Omari	Tal Ostro	Arie Peled
Collin Pieper	Idan Reller	Reese Schreiber	Oleg Schtofenmaher
Etan Shirron	Ori Moshe Stern	Jose Tierno	Maxim Tsudik
Anatoly Uskach	Jeff Wilcox	Dan Wilson	Ohad Zalcman

HP Inc. – Promoter Company Employees			
Roger Benson	Marcus Benzel	Alan Berkema	Kenneth Chan
Frank Chen	Phil Chen	Hosup Chung	Glen Dower
Mark Lessman	Nam Nguyen	Roger Pearson	Kenneth Smith
Chris Tabarez			

Intel Corporation – Promoter Company Employees			
Nausheen Ansari	Noam Arzy	Alexandre Audier	Binata Bhattacharyya
Huimin Chen	Hengju Cheng	Salauddin Choudhury	John Crouter
Maxim Dan	Jhuda Dayan	Yoni Dishon	Eran Galil
Saranya Gopal	Venkataramani Gopalakrishnan	Raul Gutierrez	Michael Gouzenfeld
Mickey Gutman	Benjamin Hacker	Yaniv Hayat	Uri Hermoni
Alon Horn	Abdul Ismail	Abhilash K V	Ziv Kabiry
Vijaykumar Kadgi	Vijay Kasturi	Sergey Khaykin	Lev Kolomiets
Vladislav Kopzon	Efraim Kugman	Edmond Lau	Uriel Lemberger
Yun Ling	Guobin Liu	Balaji Manoharan	Liran Manor
Uma Medepalli	Assaf Mevorach	Hezi Naaman	Ohad Navon

Naod Negussie	CheeLim Nge	Leonid Plaks	Duane Quiet
Rajaram Regupathy	Reuven Rozic	Oren Salomon	Zeeshan Sarwar
Brad Saunders	Leonid Shaposhnik	Ehud Shoor	Ari Sharon
Uri Soloveychik	Einat Surijan	Aviel Uzan	Karthi Vadivelu
Alex Vekker	Chen Vrubel	Stephanie Wallick	Tzewen Wang
Sarel Wechsler	Ady Weiss	Vitaly Zhivov	Gal Yedidia
Vladimir Yudovich			

Microsoft Corporation – Promoter Company Employees

Randy Aull	Jim Belesiu	Martin Borve	Anthony Chen
Jesse Chen	Matt Chung	Aacer Daken	Rajib Dutta
Mark Friend	Philip Froese	David Hargrove	Robbie Harris
Kit Hui	Toby Nixon	Rahul Ramadas	Andrea Severson
Kiran Shastry	Nathan Sherman	Ji Sun	Shyamal Varma

Renesas Corporation – Promoter Company Employees

Tam Do	Robert Dunstan	Philip Leung	Kiichi Muto
Ziba Nami	Hajime Nozaki	Raman Sargis	Yoshiyuki Tomoda
Starry Tsai	Jia Wei	Toshifumi Yamaoka	

STMicroelectronics – Promoter Company Employees

Nathalie Ballot	Joel Huloux	Gerard Mas	
-----------------	-------------	------------	--

Texas Instruments – Promoter Company Employees

Mike Campbell	Anant Gole	Craig Greenburg	Michael Koltun IV
Sai Karthik Rajaraman	Anwar Sadat	Cory Stewart	Sue Vining
Deric Waters	Gregory Watkins		

Contributor Company Employees

ACON, Advanced-Connectek, Inc.	Victory Chen	Conrad Choy	Vicky Chuang
	Jessica Feng	Sharon Hsiao	Wayne Wang
Advanced Micro Devices	Dennis Au	Nat Barbiero	Jason Chang
	Michael Comai	Walter Fry	Will Harris
	Jason Hawken	Jim Hunkins	Ling Kong
	Scott Ogle	Victor Salim	Joseph Scanlon

	Peter Teng		
Allion Labs, Inc.	Howard Chang	Casper Lee	Brian Shih
Analogix Semiconductor, Inc.	Greg Stewart	Haijian Sui	Yueke Tang
	Ning Zhu		
Anritsu Corporation	Wataru Aoba	John Jerico Custodio	Kazuhiro Fujinuma
	Hiroshi Goto	Alessandro Messina	Tadanori Nishikobara
	Ryo Sunayama	Toshihiro Suzuki	Mitsuhiro Usuba
	Takeshi Wada		
ASMedia Technology Inc.	Chang Chinyu	Chang Weiyun	Chen Chiahsin
	Chen Chunhung	Chuang Weber	Kuo Han Sung
	Lin Curtis	Lin ShuYu	Tseng PS
	Tseng YD	Wei Daniel	Wu ShengChung
Avery Design Systems, Inc.	Chris Browy	Chilai Huang	Zhihong Zeng
BitifEye Digital Test Solutions GmbH	Sebastian Muschala	Hermann Stehling	
Bizlink Technology, Inc.	Alex Chou	Morphy Hsieh	Kevin Tsai
Cadence Design Systems, Inc.	Marcin Behrendt	Jacek Duda	Shikha Gupta
	Gaurav Jain	Poonam Khatri	Yash Kothari
	Vinod Lakshman	Shivaji Magadum	Andy Mauffet-Smith
	Rohit Mishra	Uyen Nguyen	Raja Pounraj
	Thirumal Reddy	Anand RK	Anshul Shah
	Neelabh Singh	Ofer Steinberg	Mark Summers
	Claire Ying	Wasiq Zia	
Corigine, Inc.	Kevin Fan	Ali Khan	Xiao Xiao
Corning Optical Communications LLC	Mark Bradley	Wojciech Giziewicz	Ian McKay
	Jamie Silva		
Cypress Semiconductor	Mark Fu	Naman Jain	Savan Javia
	Palani Subbiah		
Dell Inc.	Mohammed Hijazi	Tom Lanzoni	Ken Nicholas
	Marcin Nowak	Scott Ogle	Adie Tan

	Lee Zaretsky		
Diodes Incorporated	Qun Song		
DisplayLink (UK) Ltd.	Pete Burgers	Dan Ellis	
DJI Technology Co., Ltd.	Steve Huang		
Electronics Testing Center, Taiwan	Sophia Liu		
Elka International Ltd.	Alvin Cheng	Chloe Hsieh	Roy Ting
	Jui-Ming Yang		
Ellisys	Abel Astley	Mario Pasquali	Chuck Trefts
	Tim Wei		
Etron Technology, Inc.	Andy Chen	Shihmin Hsu	Bryan Huang
	Chien-Cheng Kuo	Jen Hong Larn	
Foxconn / Hon Hai	Patrick Casher	Joe Chen	Jason Chou
	Fred Fons	Bob Hall	Terry Little
	Christine Tran	A.J. Yang	Jie Zheng
Fresco Logic Inc.	Tim Barilovits	Bob McVay	Christopher Meyers
	Jie Ni	Jeffrey Yang	
Genesys Logic, Inc.	Sean Chen	Gerry Chou	Thomas Hsieh
	Jerry Hu	Perlman Hu	Roy Huang
	ChunYen Kuo	Weddell Lee	Jimmy Lin
	Miller Lin	D.C. Lu	Greg Tu
	Han Wu	Yihsun Wu	
Google Inc.	Mark Hayter	Benson Leung	Raj Mojumder
	David Schneider		
Granite River Labs	Nikhil Acharya	Yun Han Ang	Sandy Chang
	Allen Chen	Cyan Chen	Swee Guan Chua
	Alan Chuang	Steven Lee	Caspar Lin
	Tim Lin	Krishna Murthy	Johnson Tan
	Rajaraman V	Chin Hun Yaep	
Hotron Precision Electronic Ind. Corp.	Rosa Chen	Patrick Yeh	YF Zhang

I-PEX (Dai-ichi Seiko)	Alan Kinningham	Ro Richard	
Japan Aviation Electronics Industry Ltd.	Mark Saubert	Junichi Takeuchi	
JMicon Technology Corp.	Charon Chen	Mika Cheng	Kevin Liu
Kandou Bus SA	Brian Holden	Hitaish Sharma	David Stauffer
	Andrew Stewart	Mark Vennebarger	
Keysight Technologies Inc.	Atsushi Imaoka	Biing Lin Lem	Jit Lim
	Francis Liu	Roland Scherzinger	
L&T Technology Services	Binu Chinna Thankam	Sunil Kumar	Siddharth Pethe
	Badrinath Ramachandra	Arunkumar Selvam	Gayathri SN
LeCroy Corporation	Alan Blankman	Patrick Connally	Carl Damn
	David Fraticelli	Daniel H Jacobs	Farnoosh Jafary
	Tyler Joe	Carlo Mazzetti	Mike Micheletti
	Kathryn Morales	Jeff Sabuda	Joseph Schachner
	Chris Webb		
Lenovo	Toshikazu Horino	Shinji Matsushima	Yuuki Matsuura
	Nozomu Nagata	Munefumi Nakata	Kazuya Shibayama
	Shunki Sugai	Chikara Takahashi	Masahiro Tokuno
	Kayanagi Tsuneo		
LG Electronics Inc.	Do Kyun Kim	Yoon Jong Lee	Seung Yoo
Lintes Technology Co., Ltd.	Tammy Huang	Charles Kaun	RD Lintes
	Max Lo	CT Pien	Jin Yi Tu
	Jason Yang		
Lotes Co., Ltd.	Regina Liu-Hwang	John Lynch	
Luxshare-ICT	Josue Casillo	CY Hsu	Antony Lin
	John Lin	Stone Lin	Scott Shuey
	Eric Wen	Pat Young	
Maxio Technology (Hangzhou) Ltd.	George Fang		
MediaTek Inc.	Henry Chen	Alexyc Lin	Pochou Lin

	Chiachun Wang		
MegaChips Corporation	Rahul Agarwal	Ramesh Dandapani	Satoru Kumashiro
	Ryuichi Mariizumi	Sireesha Vemulapalli	Nobu Yanagisawa
Mercedes-Benz Research & Development, North America, Inc.	Hans Wickler		
Microchip Technology Inc.	Mark Bohm	Atish Ghosh	Fernando Gonzalez
	Mark Gordon	Richard Petrie	Brigham Steele
	Anthony Tarascio	Robert Zakowicz	
Molex LLC	Alan MacDougall		
MQP Electronics Ltd.	Sten Carlsen	Pat Crowe	
Newnex Technology Corp.	Sam Liu		
NVIDIA	Jamie Aitken	Mark Overby	
NXP Semiconductors	Mahmoud El Sabbagh	Ken Jaramillo	Abhijeet Kulkarni
	Vijendra Kuroodi	Krishnan TN	
Oculus VR LLC	Marty Evans	Joaquin Fierro	Chao Hu
ON Semiconductor	Eduardo De Reza	Oscar Freitas	Christian Klein
	Amir Lahooti		
Parade Technologies, Inc.	Jian Chen	Jimmy Chiu	Mark Qu
	Craig Wiley	Paul Xu	Kevin Yuan
	Alan Yuen		
Phison Electronics Corp.	Jimmy Chen	Ko Hong Lipp	Sebastien Jean
	Stark Kuan	Thomas Lee	Anton Lin
	Winnie Lu	Wei Sui-Ning	James Tsai
	Michael Wu	Fu-Hua Yang	Chang Yuan-Cheng
Qualcomm, Inc	Tomer Ben Chen	Yiftach Benjamini	Richard Burrows
	Amit Gil	James Goel	Philip Hardy
	Raja Jagadeesan	Lalan Mishra	Dmitrii Vasilchenko
	Chris Wiesner		
	Chung-Chun Chen	Jen Wen Chen	Shen Chen

Realtek Semiconductor Corp.	Jonathan Chou	Chang Ding	Yao Feng
	Bokai Huang	An-Ming Lee	Ray Lee
	Ryan Lin	Terry Lin	Luobin Wang
	Kay Yin	Chris Zeng	
Rohde & Schwarz GmbH & Co. KG	Johannes Ganzert	Randy White	
Samsung Electronics Co., Ltd.	Jaedeok Cha	KangSeok Cho	CheolYoon Chung
	Sangju Kim	Termi Kwon	Cheolho Lee
	Edward Lee	Jun Bum Lee	Chahoon Park
	Sunggeun Yoon		
Seagate Technology LLC	Alvin Cox	Paul McParland	Michael Morgan
	Cuong Tran		
Silicon Line GmbH	Ian Jackson		
SiliConch Systems Private Limited	Kaustubh Kumar	Rakesh Polasa	Satish Anand Verkila
Softnautics LLP	Bhavesh Desai	Hetal Jariwala	Dipakkumar Modi
	Ishita Shah	Ujjwal Talati	
Spectra7 Microsystems Corp.	Alex Chow	James McGrath	
Specwerkz	Sydney Fernandes	Amanda Hosler	Diane Lenox
	Soren Petersen		
STMicroelectronics	Nathalie Ballot	Joel Huloux	Gerard Mas
Sumitomo Electric Ind., Ltd., Optical Comm. R&D Lab	Sainer Siagian	Mitsuaki Tamura	
Synaptics Inc.	Jeff Lukanc	Mark Miller	Prashant Shamarao
Synopsys, Inc.	Prishkrit Abrol	Subramaniam Aravindhnan	Jeanne Cai
	Jun Cao	Morten Christiansen	Scott Guo
	Eric Huang	Joseph Juan	Venkataraghavan Krishnan
	Jitendra Kushwaha	Behram Minwalla	Saleem Mohammad
	Rick Schmidt	Jasjeet Singh	Mahendra Singh

	John Stonick	Zongyao Wen	Fred Yu
Tektronix, Inc.	Madhusudan Acharya	Sourabh Das	Keyur Diwan
	Mark Guenther	Abhijeet Shinde	Gary Simontom
Thine Electronics, Inc.	Shuhei Yamamoto		
Tyco Electronics Corp., a TE Connectivity Ltd. company	Simon Li	Jeff Mason	Jacky Mo
	Tommy Yu	Yuanbo Zhang	Tony Zhu
Varjo Technologies	Kai Inha		
VIA Labs, Inc.	Wayne Tseng		
VIA Technologies, Inc.	Benjamin Pan	Terrance Shih	Jay Tseng
	Fong-Jim Wang		
Weltrend Semiconductor	Chao-Chee Ku	Jeng Cheng Liu	Wayne Lo
	Ho Wen Tsai	Eric Wu	Randolph Wu
	Simon Yeh		
Western Digital	David Landsman	Larry McMillan	Rob Ryan
Wilder Technologies	Steve Bright	Zach Moore	Joe O'Brien
	Majid Shayegh		

Contents

1	Preface	15
1.1	Scope of Document.....	15
1.2	Document Organization.....	15
1.3	Related Documents	15
1.4	Terms and Abbreviations	15
1.5	Documentation Conventions.....	15
1.5.1	Capitalization	15
1.5.2	Italic Text.....	15
1.5.3	Numbers and Number Bases	16
1.5.4	Bit, Byte, DW, and Symbol Conventions	16
1.5.5	Implementation Notes	16
1.5.6	Word Usage.....	16
1.5.7	FourCC.....	16
1.5.8	Reserved Values and Fields.....	16
2	Overview.....	18
2.1	External Terms and Definitions	18
2.2	Internal Terms and Definitions	19
3	Electrical Layer.....	21
3.1	On-Board Re-timers	21
3.2	Cable Re-timers	21
3.3	Active Electrical Cables	21
4	Logical Layer.....	22
4.1	Sideband Channel.....	22
4.1.1	Transactions.....	22
4.1.2	Lane Initialization	29
4.2	Re-timer Channel State Machine.....	32
4.2.1	CLd.....	33
4.2.2	Bit Lock	35
4.2.3	Forwarding.....	35
4.2.4	Low Power (CL0s, CL1, and CL2).....	35
4.3	Lane Decoding.....	40
4.3.1	Error Cases	41
4.4	Timing Parameters.....	41
5	Port Operations	43
5.1	Enumerating Port Operations.....	44
5.1.1	QUERY_LAST_RE-TIMER.....	45
5.1.2	QUERY_CABLE_RE-TIMER	46
5.2	NVM Port Operations.....	46

5.2.1	SET_INBOUND_SBTX	48
5.2.2	UNSET_INBOUND_SBTX	48
5.2.3	GET_NVM_SECTOR_SIZE	48
5.2.4	NVM_SET_OFFSET	49
5.2.5	NVM_BLOCK_WRITE	49
5.2.6	NVM_AUTH_WRITE	50
5.2.7	NVM_READ	51
5.3	Receiver Lane Margining Port Operations	52
6	Interoperability with Thunderbolt™ 3 (TBT3) Systems	53
6.1	Electrical Layer	53
6.2	Logical Layer	53
6.2.1	Sideband Channel	53
6.2.2	Re-timer Channel Layer State Machine	62
A.	Examples of Entry to Low Power State	63
A.1	Successful Entry to CL2 state	63
A.2	Successful entry to CL0s state	64
A.3	Error in CL2_REQ Ordered Sets	65
A.4	Error in CL2_ACK Ordered Sets	65
A.5	Error in CL_OFF Ordered Sets	66

Figures

Figure 2-1.	USB4 Link with Re-timers	18
Figure 2-2.	External Re-timer Relationships	19
Figure 2-3.	Internal Re-timer Relationships	20
Figure 4-1.	Propagation of Broadcast RT Transactions.....	23
Figure 4-2.	Example of Lane Reversal.....	30
Figure 4-3.	Progression of Clock Switch Done Bit.....	32
Figure 4-4.	The Re-timer Channel State Machine	33
Figure 6-1.	Propagation of Broadcast RT Transactions.....	58
Figure A-1.	Successful Entry to CL2 State	63
Figure A-2.	Successful Entry to CL0s State.....	64
Figure A-3.	Error in CL2_REQ Ordered Sets	65
Figure A-4.	Error in CL2_ACK Ordered Sets	66
Figure A-5.	Error in CL_OFF Ordered Sets.....	67

Tables

Table 1-1.	Rsvd Value and Field Handling.....	17
Table 4-1.	Re-timer SB Registers	24
Table 4-2.	SB Register Fields Access Types	25
Table 4-3.	SB Register Fields.....	25
Table 4-4.	Required Error Cases.....	41
Table 4-5.	Optional Error Cases.....	41
Table 4-6.	Re-timer Timing Parameters	41
Table 5-1.	Port Operation Supported by a Re-timer (Required).....	43
Table 5-2.	QUERY_LAST_RE-TIMER Completion Metadata	46
Table 5-3.	QUERY_CABLE_RE-TIMER Completion Metadata	46
Table 5-4.	GET_NVM_SECTOR_SIZE Completion Metadata	49
Table 5-5.	NVM_SET_OFFSET Operation Metadata	49
Table 5-6.	NVM_BLOCK_WRITE Operation Data	50
Table 5-7.	NVM_AUTH_WRITE Completion Metadata.....	51
Table 5-8.	NVM_READ Operation Metadata	51
Table 5-9.	NVM_READ Completion Data	52
Table 6-1.	Re-timer SB Registers.....	55
Table 6-2.	SB Register Fields.....	55
Table 6-3.	TxFE Negotiation flows	58

1 Preface

1.1 Scope of Document

This specification is primarily targeted to developers of USB4™ Re-timers and system OEMs. This specification can be used for developing Cable Re-timers and On-Board Re-timers.

1.2 Document Organization

Chapters 1 and 2 provide an overview for all readers. Chapters 3 through 6 contain detailed technical information defining USB4 Re-timers. The following summarizes each chapter:

- Chapter 1 (Preface) – This chapter defines document conventions and terms.
- Chapter 2 (Overview) – This chapter gives an overview of the Re-timer architecture.
- Chapter 3 (Electrical Layer) – This chapter defines the requirements for electrical compliance.
- Chapter 4 (Logical Layer) – This chapter defines Lane initialization and operation.
- Chapter 5 (Port Operation) – This chapter defines the software interface to a Re-timer.
- Chapter 6 (Interoperability with Thunderbolt™ 3 Systems) – This chapter defines the requirements for a Re-timer to operate in a Link that includes Thunderbolt 3 Routers or a Thunderbolt 3 Cable.

1.3 Related Documents

Universal Serial Bus (USB4™) Specification, Revision 1.0, August 2019 (USB4 Specification)

USB Type-C® Cable and Connector Specification, Release 2.0 (USB Type-C Specification)

Universal Serial Bus Power Delivery Specification, Release 3.0, Version 2.0, August 2019 (USB PD Specification)

1.4 Terms and Abbreviations

This specification uses the same terms and abbreviations as defined in the USB4 Specification.

1.5 Documentation Conventions

1.5.1 Capitalization

Some terms are capitalized to distinguish their definition in the context of this document from their common English meaning. Words not capitalized have their common English meaning.

1.5.2 Italic Text

Italic text is used to identify variable names, register field and packet field names, or reference document titles.

1.5.3 Numbers and Number Bases

Hexadecimal numbers are written with a lower case “h” suffix, e.g. FFFFh and 01h. Hexadecimal numbers larger than four digits are represented with a space dividing each group of four digits, e.g. 1EFF FFFF FFFFh. Binary numbers are written with a lower case “b” suffix, e.g. 1001b and 01b. Binary numbers larger than four digits are written with a space dividing each group of four digits, e.g. 1010 1000 1100b.

All other numbers are decimal.

1.5.4 Bit, Byte, DW, and Symbol Conventions

A bit, byte, DW, or Symbol residing in location n within an array is denoted as bit[n], byte[n], DW[n], or Symbol[n].

A sequence of bits, bytes, DWs, or Symbols residing in locations n to m (inclusive) within an array is denoted as bit[m:n], byte[m:n], DW[m:n], or Symbol[m:n].

1.5.5 Implementation Notes

Implementation Notes are not a normative part of this specification. They are included for clarification and illustration only. Implementation notes within this document are enclosed in a box and set apart from other text.

1.5.6 Word Usage

The word “shall” is used to indicate mandatory requirements. Mandatory requirements are strictly to be followed in order to conform to this specification and no deviation is permitted.

The phrase “it is recommended” is used to convey that, among several possibilities, one is preferred but not necessarily required.

The word “may” is used to indicate a course of action permissible within the limits of the specification. The word “can” is used only for statements of possibility or capability (i.e. “can” equals “is able to”).

1.5.7 FourCC

A FourCC is a sequence of four bytes used to represent ASCII strings. It is limited to ASCII printable characters (one byte per character), with space characters reserved for padding shorter sequences.

For example, the FourCC string “ABC ” is represented by a hexadecimal value of 20434241h, where the rightmost byte (41h) represents the first ASCII character (“A”).

1.5.8 Reserved Values and Fields

Unless otherwise specified, fields and values marked “Rsvd” shall be handled as described in Table 1-1.

Table 1-1. Rsvd Value and Field Handling

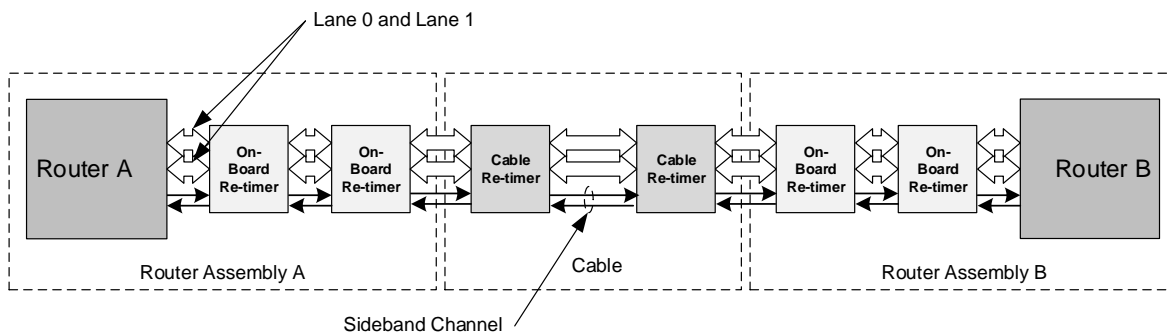
Type	Handling
Transaction Values	A transmitter shall not use a value in this specification that is marked as “Rsvd”. The target of a Transaction shall ignore a Transaction that has any of its defined fields set to a Rsvd value and proceed as if the Transaction was never received.
Ordered Set Values	A transmitter shall not use a value in this specification that is marked as “Rsvd”. The target of an Ordered Set shall ignore an Ordered Set that has any of its defined fields set to a Rsvd value and proceed as if the Ordered Set was never received.

2 Overview

A USB4™ Link supports up to six Re-timers. Of those six Re-timers, four may be On-Board Re-timers (two On-Board Re-timers at each end of the Link).

Figure 2-1 depicts a USB4 Link with six Re-timers between the two connected Routers (referred to as “Router A” and “Router B”). Router Assembly A and Router Assembly B each include two On-Board Re-timers. The cable connecting the ends of the Link is an Active Cable with a Cable Re-timer at each end.

Figure 2-1. USB4 Link with Re-timers



Electrical compliance for an On-Board Re-timer is measured at the USB Type-C connector of the Router Assembly that contains the Re-timer. The electrical requirements for a Router Assembly are defined in the USB4 Specification. Additional requirements for an On-Board Re-timer are defined in Section 3.1 of this specification.

Electrical compliance for a Cable Re-timer is measured at the USB Type-C connector of the Active Cable that contains the Re-timer. The electrical requirements for an Active Cable are defined in Section 3.3 of this specification. Additional Requirements for Cable Re-timers are defined in Section 3.2 of this specification.

The Re-timer interfaces that are inside a Router Assembly or Active Cable are outside the scope of this specification.

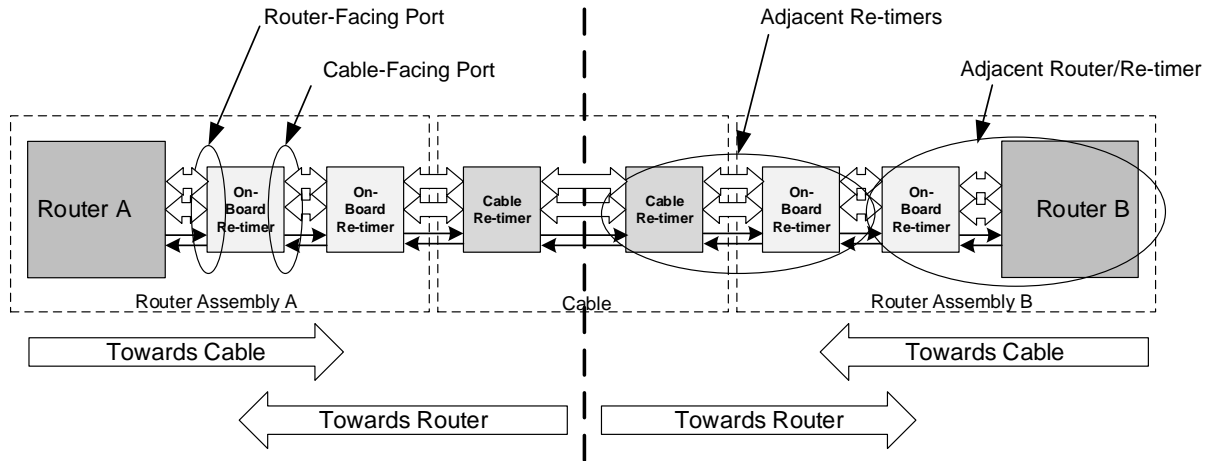
2.1 External Terms and Definitions

This specification uses the following terms to identify the relationships between the entities that are external to a Re-timer:

- **Router-Facing** – Describes a USB4 Port, Lane Adapter, transmitter, or receiver that faces away from the center of the Cable.
- **Cable-Facing** – Describes a USB4 Port, Lane Adapter, transmitter, or receiver that faces towards the center of the Cable.
- **Adjacent** – Describes a Re-timer, USB4 Port, Lane Adapter, transmitter, or receiver that is directly connected to another Re-timer, USB4 Port, Lane Adapter, transmitter, or receiver with no other Re-timers or Routers in between.

Figure 2-2 shows an example of how these terms are used.

Figure 2-2. External Re-timer Relationships



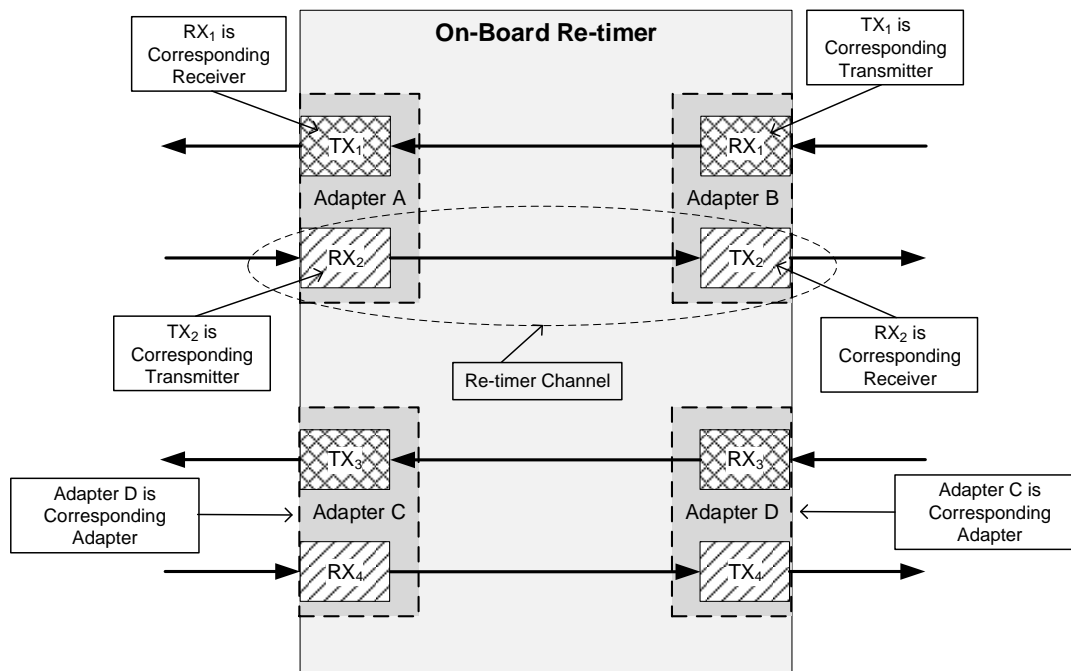
2.2 Internal Terms and Definitions

This specification uses the following terms to identify the relationships between the entities within a Re-timer.

- Corresponding Receiver – The receiver that feeds the transmitter with USB4 traffic.
- Corresponding Transmitter – The transmitter that accepts USB4 traffic from the receiver.
- Corresponding Adapter – The Lane Adapter that contains the Corresponding Receiver and Corresponding Transmitter.
- Re-timer Channel – a transmitter and its Corresponding Receiver, or a receiver and its Corresponding Transmitter.

Each transmitter has one Corresponding Receiver and each receiver has one Corresponding Transmitter.

Figure 2-3. Internal Re-timer Relationships



3 Electrical Layer

3.1 On-Board Re-timers

An On-Board Re-timer shall implement an Electrical Layer as defined in the USB4 Specification with the following changes:

- An On-Board Re-timer shall support Gen 2 speed of 10Gbps. Support for other speeds is optional.
- An On-Board Re-timer shall support two Lanes.
- An On-Board Re-timer operating at Gen 2 speed shall transmit a bit within t_{Latency2} after receiving the bit.
- An On-Board Re-timer operating at Gen 3 speed shall transmit a bit within t_{Latency3} after receiving the bit.
- For a pair of Corresponding Adapters, the propagation latency of one Re-timer Channel shall be within t_{Skew} of the Re-timer Channel in the opposite direction. Propagation latency is measured from the time a bit is received by a receiver until the time it is transmitted by the Corresponding Transmitter.
- An On-Board Re-timer shall add no more than t_{Skew} amount of skew on its transmitting Lanes relative to the skew measured at the receiving Lanes.

3.2 Cable Re-timers

A Cable Re-timer shall meet the requirements in the USB4 Specification with the following changes:

- A Cable Re-timer shall support Gen 2 speed of 10Gbps and Gen 3 speed of 20Gbps.
- A Cable Re-timer shall support two Lanes.
- A Cable Re-timer operating at Gen 2 speed shall transmit a bit within t_{Latency2} after receiving the bit.
- A Cable Re-timer operating at Gen 3 speed shall transmit a bit within t_{Latency3} after receiving the bit.
- For a pair of Corresponding Adapters, the propagation latency of one Re-timer Channel shall be within t_{Skew} of the other Re-timer Channel. Propagation latency is measured from the time a bit is received by a receiver until the time it is transmitted by the Corresponding Transmitter.
- A Cable Re-timer shall add no more than t_{Skew} amount of skew on its transmitting lanes relative to the skew measured at the receiving Lanes.

3.3 Active Electrical Cables

See the USB Type-C Specification for the electrical specifications of an Active Cable.

4 Logical Layer

A Re-timer shall have two USB4™ Ports. USB4 Ports are defined in the USB4 Specification. The USB4 Ports on a Re-timer shall have the same number of Lane Adapters and shall support the same capabilities.



IMPLEMENTATION NOTE

An On-Board Re-timer needs to be aware of which USB4 Port faces towards the Router and which USB4 Port faces towards the USB Type-C connector. An On-Board Re-timer also needs to know if a USB4 Port directly interfaces a USB Type-C connector. How this knowledge is acquired is implementation specific.

4.1 Sideband Channel

Sideband Channel transactions are sent and received over the SBTX and SBRX wires as defined in the USB4 Specification.

4.1.1 Transactions

When forwarding Transactions from an SBRX input to an SBTX output, a Re-timer shall maintain the order of Transactions as received on the SBRX input.

4.1.1.1 LT Transactions

A Re-timer shall support LT Transactions as defined in the USB4 Specification with the following changes:

- When a Re-timer receives an LT Transaction, it shall forward the Transaction to its other USB4 Port.

4.1.1.2 AT Transactions

A Re-timer shall support AT Transactions as defined in the USB4 Specification with the following changes:

- A Cable Re-timer shall forward an AT Transaction, regardless of the value of the *Recipient* bit.
- An On-Board Re-timer shall forward an AT Transaction, regardless of the value of the *Recipient* bit.
- A Re-timer shall not initiate AT Commands.

4.1.1.3 RT Transactions

A Re-timer shall support RT Transactions as defined in the USB4 Specification with the changes defined in this section.

4.1.1.3.1 Broadcast RT Transactions

A Router uses Broadcast RT Transactions to enumerate the Re-timers in a Link and set Re-timer Index values. The Re-timer Index is used to identify and address the Re-timers between two connected Routers. A Re-timer has two Re-timer Indexes. The Re-timer Index from a Broadcast RT Transaction received on the Router-Facing USB4 Port is referred to as the “Router-Facing Index” and the Re-timer Index from a Broadcast RT Transaction received on the Cable-Facing USB4 Port is referred to as the “Cable-Facing Index”.

Note: The Router-Facing Index and the Cable-Facing Index for a Re-timer are not related and can have different values.

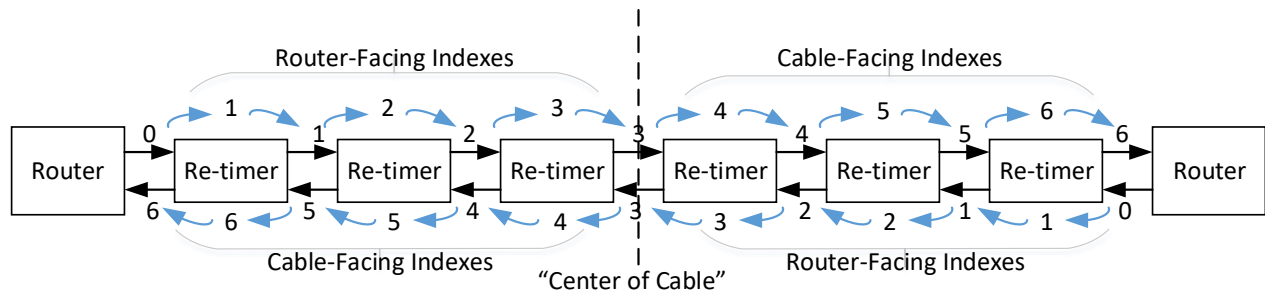
When a Re-timer receives a Broadcast RT Transaction on its Router-Facing USB4 Port, it shall increment the value in the *Index* field of the Transaction by one and shall store the resulting Re-timer Index locally as its Router-Facing Index. The Re-timer shall then forward the Broadcast RT Transaction.

When a Re-timer receives a Broadcast RT Transaction on its Cable-Facing USB4 Port, it shall increment the value in the *Index* field of the Transaction by one and shall store the resulting Re-timer Index locally as its Cable-Facing Index. The Re-timer shall then forward the Broadcast RT Transaction.

When a Re-timer forwards a Broadcast Transaction, it shall set the *SSCalways* bit to 0b to indicate that the Re-timer supports exiting CLx state with SSC turned off.

Figure 4-1 shows an example of how a Broadcast RT Transaction is used to assign Re-timer Indexes to a Link with 6 Re-timers.

Figure 4-1. Propagation of Broadcast RT Transactions



4.1.1.3.2 Addressed RT Transactions

An On-Board Re-timer uses Addressed RT Commands to access the SB Register Space of an adjacent Router or adjacent Re-timer. When sending an Addressed RT Command, a Re-timer shall set the *Index* field to 0.

System Software also uses Addressed RT Commands to access the SB Register Space of a Re-timer.

Section 4.1.2.5 defines how Addressed RT Commands are used for Lane equalization.

The rules below define how a Re-timer handles Addressed RT Commands:

- When a Re-timer receives an Addressed RT Command with the *Index* field set to 0, it shall process the Command and send a response as described in the USB4 Specification. The Re-timer shall not forward the Transaction.
- When a Re-timer receives an Addressed RT Command on its Router-Facing USB4 Port with an *Index* field that matches its Router-Facing Index, it shall process the Command and send a response as described in the USB4 Specification. The Re-timer shall not forward the Transaction.

- When an On-Board Re-timer receives an Addressed RT Command on its Cable-Facing USB4 Port with an *Index* field that matches its Cable-Facing Index, it may process the Command and send a response as described in Section 4.1.2.5.1 of the USB4 Specification, but is not required to do so. It shall not forward the Command.
- When a Cable Re-timer receives an Addressed RT Command on its Cable-Facing USB4 Port with an *Index* field that matches its Cable-Facing Index, it shall process the Command and send a response as described in the USB4 Specification. The Re-timer shall not forward the Transaction.
- When a Re-timer receives an Addressed RT Transaction on its Router-Facing USB4 Port with an *Index* field that does not match its Router-Facing Index, it shall forward the RT Transaction without sending a response.
- When a Re-timer receives an Addressed RT Transaction on its Cable-Facing USB4 Port with an *Index* field that does not match its Cable-Facing Index, it shall forward the RT Transaction without sending a response.

The rules below define how a Re-timer handles Addressed RT Responses:

- If the *Index* field in the Addressed RT Transaction is 0, then the Re-timer shall consume the Addressed RT Response.
- If the *Index* field in the Addressed RT Transaction is not 0, the Re-timer shall forward the Addressed RT Response.

4.1.1.4 SB Register Space

A Re-timer shall maintain the SB Register Space defined in Table 4-1.

Table 4-1. Re-timer SB Registers

Register	Size (Bytes)	Name	Description
0	4	Vendor ID	Identifies the manufacturer of the Re-timer silicon.
1	4	Product ID	Identifies the type of the Re-timer.
2 to 7	N/A	Rsvd	Reserved.
8	4	Instruction Opcode	A Port Operation Opcode in FourCC format.
9	16	Metadata	Metadata written or read with a Port Operation.
10 to 12	N/A	Rsvd	Reserved.
13	4	TxFE	Used to set the TxFFE parameters of transmitters.
14	N/A	Rsvd	Reserved.
15	4	Sideband Channel Version	A vendor defined version of the Sideband Channel implementation.

16 to 17	Vendor specific	Vendor specific	Vendor specific register.
18	64	Data	Data written or read with a Port Operation.
19 to 127	Vendor specific	Vendor specific	Vendor specific register.
128 to 256	N/A	Rsvd	Reserved.

Table 4-2 defines the access types for SB Register fields.

Table 4-2. SB Register Fields Access Types

Access Type	Description
RO	Read Only. An AT Write Command or an RT Write Command to a field with this access type shall have no effect. An AT Read Command or an RT Read Command shall return a meaningful value.
RW	Read/Write. A field with this access type shall be capable of both Read Commands and Write Commands. The value read from this field shall reflect the last value written to it unless the field was reset in the interim.
Rsvd	Reserved. Reserved for future implementation. A Write Command to this field shall have no effect.
RsvdV	Reserved with Non-Zero Value. Reserved for future implementation. A Write Command to this field shall have no effect. A read shall return the specified value.

The SB Register Space registers shall have the structure and fields described in Table 4-3.

Table 4-3. SB Register Fields

Register	Register Name	Byte	Bits	Field Name and Description	Type	Default Value
0	Vendor ID	0	7:0	Vendor ID Low Identifies the manufacturer of the Re-timer silicon.	RO	Vendor Defined
		1	7:0	Vendor ID High Identifies the manufacturer of the Re-timer silicon.	RO	Vendor Defined
		2	7:0	Reserved	Rsvd	0
		3	7:0	Reserved	Rsvd	0
1	Product ID	0	7:0	Product ID Low Assigned by the manufacturer to identify the type of the Re-timer.	RO	Vendor Defined

		1	7:0	Product ID High Assigned by the manufacturer to identify the type of the Re-timer.	RO	Vendor Defined
		2	7:0	Reserved	Rsvd	0
		3	7:0	Reserved	Rsvd	0
2	Version	0	7:0	Reserved Shall be set to 00h or 01h. It is recommended that this field be set to 01h.	RsvdV	00h or 01h
		1	7:0	Reserved	Rsvd	0
		2	7:0	Reserved	Rsvd	0
		3	7:0	Reserved	Rsvd	0
8	Opcode	0	7:0	Opcode 0 Contains the first character of the Opcode.	RW	0
		1	7:0	Opcode 1 Contains the second character of the Opcode.	RW	0
		2	7:0	Opcode 2 Contains the third character of the Opcode.	RW	0
		3	7:0	Opcode 3 Contains the fourth character of the Opcode.	RW	0
9	Metadata	16	7:0	Metadata	RW	0
13	TxFFE	0 ¹	3:0	TxFFE Request (Lane 0) Identifies one of 16 predefined TxFFE configurations requested by the receiver.	RO	Vendor Defined
			4	Rx Locked (Lane 0) Indicates that the receiver completed TxFFE negotiation: 0b – TxFFE negotiation is not done 1b – TxFFE negotiation is done	RO	0b

¹This byte is the *Local Rx Status & TxFFE Request* byte for Lane 0

			5	Rx Active (Lane 0) Indicates whether or not the receiver is active: 0b – Receiver is inactive 1b – Receiver is active	RO	0b
			6	Clock Switch Done (Lane 0) Indicates whether or not the Corresponding Transmitter uses the receiver clock: 0b – Transmitter uses local clock 1b – Transmitter uses receiver clock	RO	0b
			7	New Request (Lane 0) Indicates whether or not the receiver is providing a new index in the <i>TxFFE Request</i> field: 0b – Receiver is still processing a previous TxFFE configuration 1b – Receiver is providing a new index in the <i>TxFFE Request</i> field	RO	0b
		1 ²	3:0	TxFFE Request (Lane 1) Identifies one of 16 predefined TxFFE configurations requested by the receiver.	RO	Vendor Defined
			4	Rx Locked (Lane 1) Indicates that the receiver completed TxFFE negotiation: 0b – TxFFE negotiation is not done 1b – TxFFE negotiation is done	RO	0b
			5	Rx Active (Lane 1) Indicates whether or not the receiver is active: 0b – Receiver is inactive 1b – Receiver is active	RO	0b
			6	Clock Switch Done (Lane 1) Indicates whether or not the Corresponding Transmitter uses the receiver clock: 0b – Transmitter uses local clock 1b – Transmitter uses receiver clock	RO	0b

²This byte is the *Local Rx Status & TxFFE Request* byte for Lane 1

			7	New Request (Lane 1) Indicates whether or not the receiver is providing a new index in the <i>TxFFE Request</i> field: 0b – Receiver is still processing a previous TxFFE configuration 1b – Receiver is providing a new index in the <i>TxFFE Request</i> field	RO	0b
		2 ³	3:0	TxFFE Setting (Lane 0) Index of the TxFFE configuration loaded the transmitter.	RO	Vendor Defined
			5:4	Rsvd	Rsvd	0
			6	Request Done (Lane 0) Indicates whether or not the transmitter loaded the recent requested index of TxFFE configuration: 0b – Transmitter has not yet loaded the recent requested index of TxFFE configuration 1b – Transmitter has loaded the recent requested index of TxFFE configuration	RO	0b
			7	Tx Active (Lane 0) Indicates whether or not the transmitter is transmitting a valid signal: 0b – Transmitter is not transmitting a valid signal 1b – Transmitter is transmitting a valid signal	RO	0b
		3 ⁴	3:0	TxFFE setting (Lane 1) Index of the TxFFE configuration loaded the transmitter.	RO	Vendor Defined
			5:4	Rsvd	Rsvd	0
			6	Request Done (Lane 1) Indicates whether or not the transmitter loaded the recent requested index of TxFFE configuration: 0b – Transmitter has not yet loaded the recent requested index of TxFFE configuration 1b – Transmitter has loaded the recent requested index of TxFFE configuration	RO	0b

³This byte is the *Local Tx Status* byte for Lane 0

⁴This byte is the *Local Tx Status* byte for Lane 1

			7	Tx Active (Lane 1) Indicates whether or not the transmitter is transmitting a valid signal: 0b – Transmitter is not transmitting a valid signal 1b – Transmitter is transmitting a valid signal	RO	0b
15	Sideband Channel Version	0	7:0	Sub Version The binary coded decimal digit of the sub version number of the Sideband Channel implementation. Contains a vendor defined value.	RO	Vendor Defined
		1	7:0	Minor Version The binary coded decimal digit of the minor version number of the Sideband Channel implementation. Contains a vendor defined value.	RO	Vendor Defined
		2	7:0	Major Version LSB The least-significant binary coded decimal digit of the major version number of the Sideband Channel implementation. Contains a vendor defined value.	RO	Vendor Defined
		3	7:0	Major Version MSB The most-significant binary coded decimal digit of the major version number. Contains a vendor defined value.	RO	Vendor Defined
18	Data	16	7:0	Data	RW	0

4.1.2 Lane Initialization

A Re-timer shall perform Lane Initialization as described in the USB4 Specification with the modifications described in this section.

4.1.2.1 Phase 1 - Determination of Initial Conditions

During phase 1, a Re-timer discovers whether or not USB4 Mode is established on the Link.

An On-Board Re-timer shall also determine whether or not there is a reverse insertion at the Type-C connector.

A Re-timer shall not continue on to Phase 2 until it has obtained the connection information listed above. The mechanism for discovering the connection information is defined in the USB Type-C Specification and the USB PD Specification.

A Re-timer shall not proceed to phase 2 unless USB4 Mode is established on the Link.

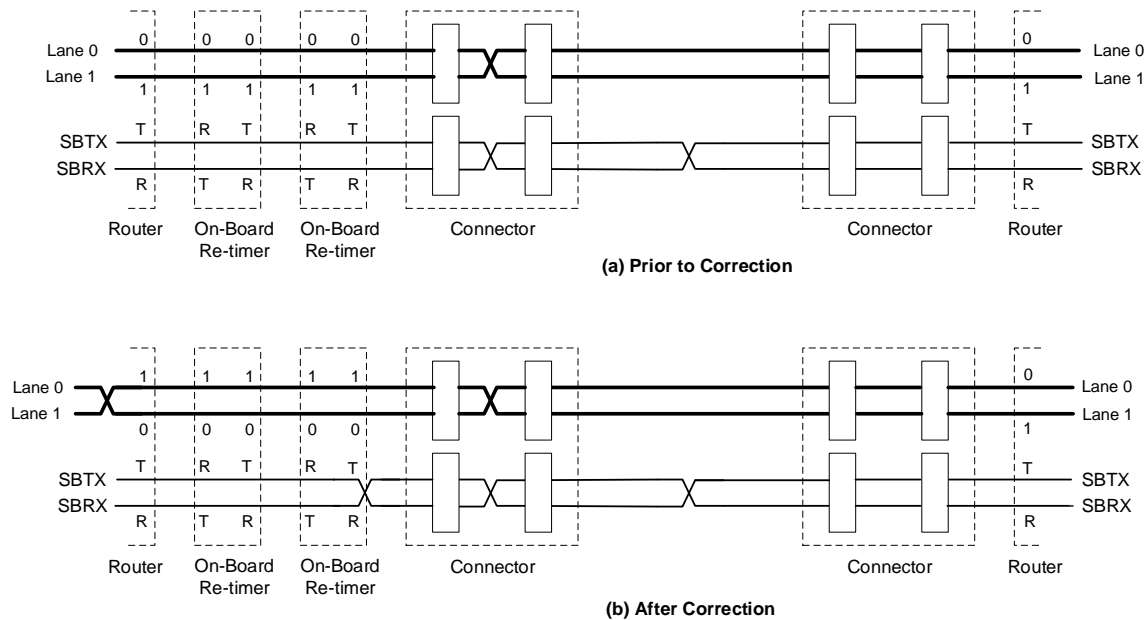
4.1.2.1.1 Lane Reversal

Lane mismatch is caused by reverse insertion of a USB Type-C connector. Lane mismatch results in Lane 0 at one end being connected to Lane 1 at the other end and the Sideband Channel SBTX (or SBRX) wires connected to each other across the Link.

When necessary to correct for Lane mismatch, Lane reversal shall take place in phase 1.

- An On-Board Re-timer that is adjacent to the USB Type-C connector shall swap its SBTX and SBRX lines facing the connector.
- An On-Board Re-timer shall swap its designation of Lane 0 and Lane 1 in both USB4 Ports.

Figure 4-2. Example of Lane Reversal



4.1.2.2 Phase 2 - Router Detection

During this phase, a Re-timer determines if there is a connection on either USB4 Port.

When a Re-timer detects a logic high on SBRX of one USB4 Port for $t_{ConnectRx}$ time, it shall drive SBTX on the other USB4 Port to logic high. The Re-timer shall then begin forwarding Transactions on the Sideband Channel in this direction.

After both USB4 Ports detect a logic high on SBRX and drive SBTX high, the Re-timer shall transition to phase 4 of Lane Initialization.

Note: If a USB4 Port receives a Transaction while it is still in Phase 2, it may ignore or drop the Transaction.

4.1.2.3 Phase 3 - Determination of USB4 Port Characteristics

Re-timers do not take an active role in phase 3.

4.1.2.4 Phase 4 - Lane Parameters Synchronization

During phase 4, a Re-timer determines the operating characteristics of its USB4 Ports as set by the Routers. When a Re-timer receives a Broadcast RT Transaction it shall update its Link parameters to match the Link parameters in the Transaction.

When a Re-timer detects an LT_Resume Transaction on any USB4 Port, it shall transition to phase 5.

4.1.2.5 Phase 5 - Lane Equalization

Upon entry to phase 5, a receiver shall perform the receiver flow for symmetric TxFFE negotiation as defined in the USB4 Specification. The Re-timer shall use RT Transactions (with the *Index* field set to 0b) to access the SB Register Space of the adjacent USB4 Port.

When the *Rx Active* bit for a receiver is set to 1b, the Re-timer shall turn on the Corresponding Transmitter and shall start transmitting CL_WAKE1.X Symbols, where X is the Re-timer Index assigned by the Router that is the target of the CL_WAKE1.X Symbols. The transmitter shall use a locally generated, non-SSC clock to transmit the CL_WAKE1.X Symbols. The Corresponding Transmitter shall then perform the transmitter flow for symmetric TxFFE negotiation as defined in the USB4 Specification. The Re-timer shall use Addressed RT Transactions (with the *Index* field set to 0b) to access the SB Register Space of the adjacent USB4 Port.

The equalization flow between two Cable Re-timers is implementation specific.

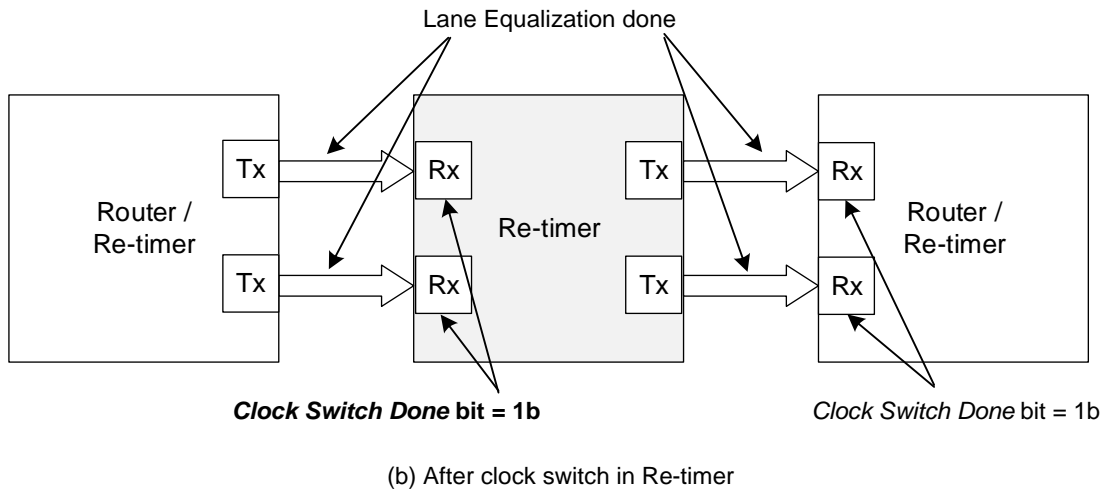
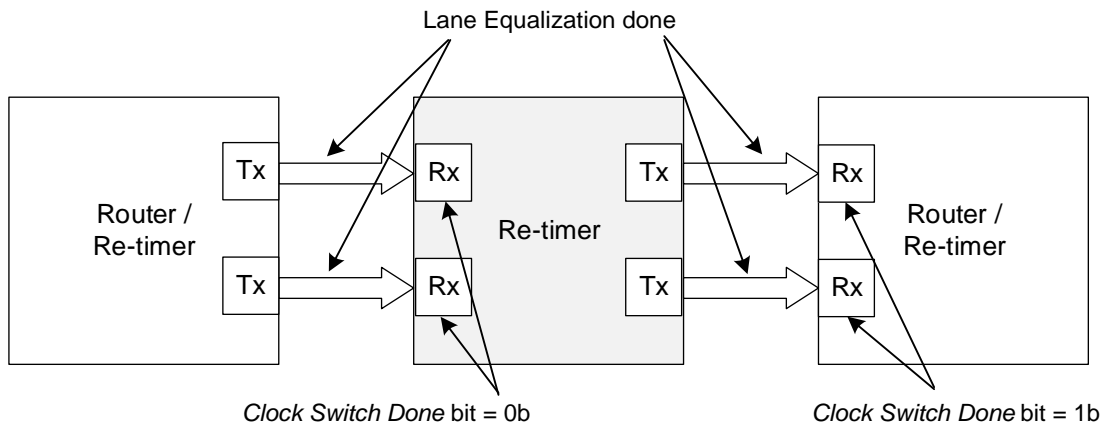
A transmitter shall stop using the local clock and shall start using the recovered clock from the Corresponding Receiver when all of the following are true:

- The Re-timer has completed TxFFE negotiation for all transmitters in that USB4 Port and all their Corresponding Receivers.
- The *Clock Switch Done* bit for Lane 0 of the adjacent USB4 Port is 1b.

The transition may or may not take place on Symbol boundary.

After a transmitter switches to using the receiver clock, it shall forward the bit stream it receives from the Corresponding Receiver instead of transmitting its locally-generated CL_WAKE1.X Symbols. The Re-timer shall set the *Clock Switch Done* bit in the Corresponding Receiver to 1b. Figure 4-3 shows how the *Clock Switch Done* bits are set.

Figure 4-3. Progression of Clock Switch Done Bit



During the transition from local clock to receiver clock, the Re-timer shall meet the SSC_SLEW_RATE requirement as specified in the USB4 Specification. The transition may or may not take place on Symbol boundary.

4.2 Re-timer Channel State Machine

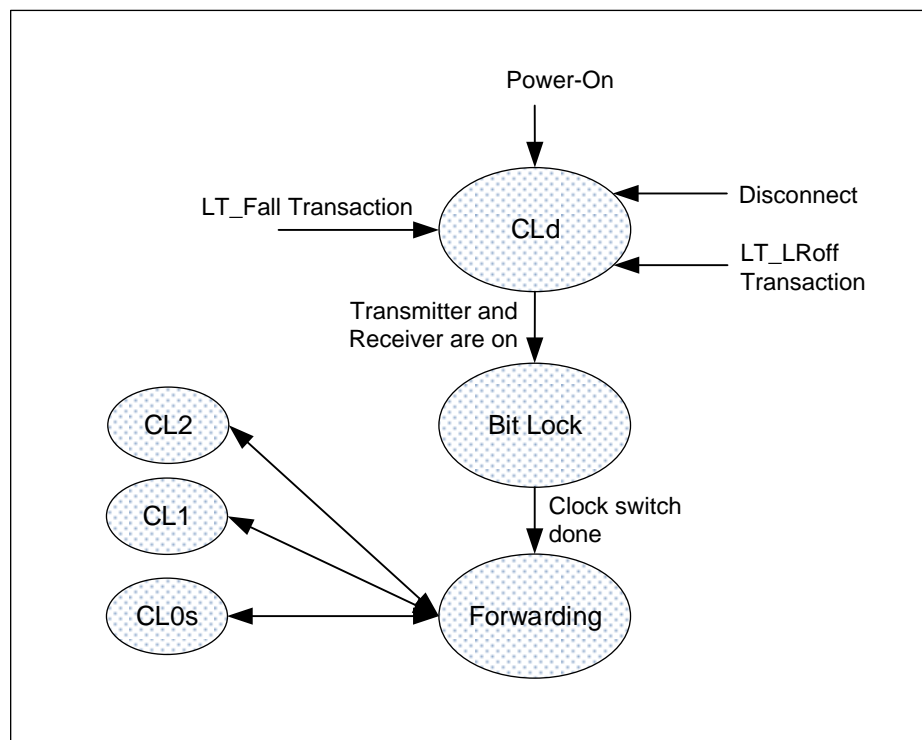
This section defines the state machine of a Re-timer Channel. Unlike the USB4 Specification, this specification describes the progression of a Re-timer Channel rather than of a Lane Adapter.

A Re-timer Channel shall support the following states:

- CLd state - The Re-timer Channel starts Lane Initialization as defined in Section 4.1.2
- Bit Lock state - The Re-timer Channel is performing bit synchronization
- Forwarding state - The Re-timer Channel forwards traffic from its receiver to its transmitter
- CL0s, CL1, CL2 states - The Re-timer Channel is in a low-power state

The state machine in Figure 4-4 describes the behavior of a Re-timer Channel. A detailed description of the states and transitions between states follows.

Figure 4-4. The Re-timer Channel State Machine



4.2.1 CLd

4.2.1.1 Entry to State

A Re-timer Channel shall transition to the CLd state when any of the following occur:

- The Re-timer is first powered on.
- The Re-timer detects a disconnect event (i.e. SBRX transitions to logical low on any USB4 Port for more than tDisconnectRx time).

- The SBRX of both Re-timer USB4 Ports are at logical high and both USB4 Ports receive an LT_LRoff Transaction within tLROff of each other.

Note: This is the case where a Domain enters Sleep state with wake enabled on the Link. It is recommended that the Re-timer store the last set of TxFFE parameters used prior to entry to CLd state in order to shorten exit time from this state.

- The Re-timer Channels for Lane 0 shall also transition to CLd state when the Re-timer receives an LT_Fall Transaction on any USB4 Port with *LSELane* field set to 0b.
- The Re-timer Channels for Lane 1 shall also transition to CLd state when the Re-timer receives an LT_Fall Transaction on any USB4 Port with *LSELane* field set to 1b.

A Re-timer Channel that transitions to CLd state due to an LT_Fall Transaction shall maintain any Lane state acquired in phase 1 and phase 2 of the previous Lane Initialization.

4.2.1.2 Behavior in State

A Re-timer Channel does not need to maintain Lane common mode voltages while in CLd state.

When Lane Initialization begins:

- If the Re-timer Channel entered this state after power-on, then Lane Initialization starts unconditionally and in phase 1.
- If the Re-timer Channel entered this state after detecting a disconnect event, or if a disconnect event occurred while in CLd state, then a connect event starts Lane Initialization in phase 1.
- If the Re-timer Channel entered this state after detecting LT_LRoff Transactions, then detection of a Broadcast RT Transaction by any USB4 Port starts Lane Initialization in phase 4.
 - During Lane Initialization, it is recommended that the Re-timer starts phase 5 with the last set of TxFFE parameters used prior to entry to CLd state.
- If the Re-timer Channel entered this state after detecting an LT_Fall Transaction, then detection of a Broadcast RT Transaction starts Lane Initialization in phase 4.
 - The Re-timer Channels for Lane 0 shall start Lane Initialization when the Re-timer receives a Broadcast RT Transaction on any USB4 Port with the *Lane0 Enabled* bit set to 1b.
 - The Re-timer Channels for Lane 1 shall start Lane Initialization when the Re-timer receives a Broadcast RT Transaction on any USB4 Port with the *Lane1 Enabled* bit set to 1b.

4.2.1.3 Exit from State

A Re-timer Channel shall exit this state when its transmitter is transmitting and its receiver is enabled.

After exiting the CLd state, a Re-timer Channel shall transition to the Bit Lock state.

4.2.2 Bit Lock

4.2.2.1 Entry to State

A Re-timer Channel shall enter this state when it exits the CLd state.

4.2.2.2 Behavior in State

A Re-timer Channel's receiver achieves bit synchronization, receiver equalization, and clock switch to recovered clock during this state.

The transmitter and receiver are on while in this state.

4.2.2.3 Exit from State

A Re-timer Channel shall exit this state after its receiver achieves bit lock and its transmitter is transmitting the bit stream received by its receiver.

After exiting the Bit Lock state, a Re-timer Channel shall transition to the Forwarding state.

4.2.3 Forwarding

4.2.3.1 Entry to State

A Re-timer Channel shall enter this state upon successful completion of receiver lock.

4.2.3.2 Behavior in State

When a Re-timer Channel is in Forwarding state, it shall forward traffic from its receiver to its transmitter. A Re-timer shall forward traffic regardless of whether or not it receives Logical Layer Symbols. For example, during compliance testing, a Re-timer forwards PRBS31 pattern instead of Logical Layer Symbols.

Note: A Re-timer does not support entry to Low Power states when it is forwarding traffic that is not Logical Layer Symbols.

A Re-timer Channel shall not modify the logical level of a bit. A Re-timer Channel shall neither add nor discard any bits.

4.2.3.3 Exit from State

A Re-timer Channel shall only exit this state after one of the following occurs:

- Transition to CLd state (see Section 4.2.1.1).
- Transition to CL0s, CL1, or CL2 states.

4.2.4 Low Power (CL0s, CL1, and CL2)

When a Router Lane Adapter transitions to CL0s state, any Re-timer Lane Adapters on the Link transition one Re-timer Channel to CL0s state and leave the other Re-timer Channel in CL0.

When a Router Lane Adapter enters CL1 state, any Re-timer Lane Adapters on the Link transition both their Re-timer Channels to CL1 state.

When a Router Lane Adapter enters CL2 state, any Re-timer Lane Adapters on the Link transition both their Re-timer Channels to CL2 state.

4.2.4.1 Entry to State

A Lane Adapter follows the rules in this section to transition its Re-timer Channels to CL2, CL1, or CL0s state. See the USB4 Specification for the definitions of the Ordered Sets used in this section. See Appendix 0 for several examples of Re-timer behavior during entry to low power state.

Rules for low power state entry:

- On detection of 3 back-to-back CL2_ACK Ordered Sets, the Re-timer Lane Adapter shall start counting time in Symbol Time units. This counter is called the “CL2_ACK Counter”. The initial value of CL2_ACK Counter shall be the number of CL2_ACK Ordered Set Symbols that were already forwarded.
- On detection of 3 back-to-back CL1_ACK Ordered Sets, the Re-timer Lane Adapter shall start counting time in Symbol Time units. This counter is called the “CL1_ACK Counter”. The initial value of CL1_ACK Counter shall be the number of CL1_ACK Ordered Set Symbols that were already forwarded.
- When the CL2_ACK Counter reaches a count of tRxShut it may shut down the Channel’s receiver. While the receiver is shut down and the Channel still in Forwarding state, the transmitter shall transmit a DC balanced signal and may use a local clock without SSC. When the CL2_ACK Counter reaches a count of tEnterCLx, the Re-timer Lane Adapter shall:
 - Transition the Re-timer Channel in the direction forwarding the CL2_ACK Ordered Sets to a CL2 state.
 - Reset the CL2_ACK Counter.
 - Transition the Re-timer Channel transmitter to electrical idle within tTxOff time from expiration of the CL2_ACK Counter.
 - Wait tEnterLFPS1, then enable detection of Low Frequency Periodic Signaling (LFPS).
- • When the CL1_ACK Counter reaches a count of tRxShut it may shut down the Channel’s receiver. While the receiver is shut down and the Channel still in Forwarding state, the transmitter shall transmit a DC balanced signal and may use local clock without SSC. When the CL1_ACK Counter reaches a count of tEnterCLx, the Re-timer Lane Adapter shall:
 - Transition the Re-timer Channel in the direction forwarding the CL1_ACK Ordered Sets to a CL1 state.
 - Reset the CL1_ACK Counter.
 - Transition the Re-timer Channel transmitter to electrical idle within tTxOff time from expiration of the CL1_ACK Counter.

- Wait $t_{\text{EnterLFPS1}}$, then enable detection of Low Frequency Periodic Signaling (LFPS).
- On detection of 3 back-to-back CL_OFF Ordered Sets, the Re-timer Lane Adapter shall start counting time in Symbol Time units. This counter is called the “CL_OFF Counter”. The initial value of CL_OFF Counter shall be the number of CL_OFF Ordered Set Symbols that were already forwarded.
- When the CL_OFF Counter reaches a count of t_{RxShut} it may shut down the Channel’s receiver. While the receiver is shut down and the Channel still in Forwarding state, the transmitter shall transmit a DC balanced signal and may use local clock without SSC. When the CL_OFF Counter reaches a count of t_{EnterCLx} , the Re-timer Lane Adapter shall:
 - Transition the Re-timer Channel in the direction forwarding the CL_OFF Ordered Sets to a low power state as follows:
 - If CL2_ACK Ordered Sets were detected during the entry flow, transition to CL2 state.
 - If CL1_ACK Ordered Sets were detected during the entry flow, transition to CL1 state.
 - If CL0s_ACK Ordered Sets were detected during the entry flow, transition to CL0s state.
 - Reset the CL_OFF Counter.
 - Transition the Re-timer Channel transmitter to electrical idle within t_{TxOff} time from expiration of the CL_OFF Counter.
 - Wait $t_{\text{EnterLFPS1}}$, then enable detection of Low Frequency Periodic Signaling (LFPS).

A Re-timer Lane Adapter shall respond to Logical Layer Errors as defined in Section 4.3.1.

Note: The detection latency might be greater than the forwarding latency, hence the initial value of the Counters may be greater than 3.

If a Re-timer Lane Adapter detects 15 back-to-back SLOS Symbols, it shall abort the entry flow. The Re-timer Lane Adapter shall also reset the CL2_ACK Counter and the CL1_ACK Counter to zero.

Note: Error scenarios during the CLx entry flow are handled by the Routers and may cause Link disconnect.



IMPLEMENTATION NOTE

When the transmitter at the other end of a Lane stops forwarding after t_{RxShut} or is transitioning to electrical idle during entry to CL2, CL1, or CL0s states, the received signal may be invalid. It is recommended that a Re-timer not update its PHY parameters during that time to avoid capturing stale values.

4.2.4.2 Behavior in State

While a Re-timer Channel is in CL2 state, its transmitter shall be in electrical idle. Lane common mode voltages shall be maintained.

While a Re-timer Channel is in CL1 state, its transmitter shall be in electrical idle. Lane common mode voltages shall be maintained.

While a Re-timer Channel is in CL0s state, its transmitter shall be in electrical idle. Lane common mode voltages shall be maintained.

4.2.4.3 Exit from State

A Re-timer follows the rules in this section when a Re-timer Channel exits from CL2, CL1, or CL0s state. See the USB4 Specification for examples of end-to-end flows describing the behavior of Re-timers during CL2, CL1, or CL0s exit.

4.2.4.3.1 CL0s Exit

This section describes Re-timer behavior during CL0s exit.

When a Re-timer detects an LFPS burst on one of its receivers, the Re-timer shall:

1. Send a Low Frequency Periodic Signaling (LFPS) burst from the Corresponding Transmitter. The duration of the LFPS burst shall be at least 16 LFPS cycles and no more than $t_{LFPSDuration}$.
2. Return the Corresponding Transmitter to Electrical Idle for $t_{PreData}$.
3. Enable the receiver to start calibration. The Re-timer shall not enable the receiver until at least t_{IdleRx} after the last LFPS cycle was received.
4. Start sending CL_WAKE1.X Symbols from the Corresponding Transmitter, where X is the Re-timer Index assigned by the Router that is the target of the CL_WAKE1.X Symbols. The Re-timer shall transmit the Symbols using a locally generated, non-SSC, clock.

After a Re-timer receives 3 back-to-back CL_WAKE2.X Symbols (where X is the same value as in step 4) on at least one Lane Adapter, the Re-timer shall transition each Re-timer Channel that is in CL0s state to transmit on the clock recovered from the received Symbols rather than on its local clock.

- The transition shall happen only after bit lock is achieved by all Re-timer Channels that are in CL0s state.
- The transition may or may not take place on Symbol boundary.
- Each Re-timer Channel in CL0s state shall transition to Forwarding state. From this point on, a Re-timer Channel shall forward the bit stream it receives from the Lane and stop generating CL_WAKE1.X Symbols.
- During the transition from local clock to receiver clock, the Re-timer shall meet the SSC_SLEW_RATE requirement as specified in the USB4 Specification.

4.2.4.3.2 CL1/CL2 Exit

There are three phases that a Re-timer Channel can through when exiting the CL1 or CL2 state:

- Phase 1 – Re-timer Channel is transmitting local copies of the CL_WAKE1 Symbol using its local clock.
- Phase 2 – Re-timer Channel toggles between transmitting local copies of the CL_WAKE1 Symbol and the copies of the received CL_WAKE2 Symbol. The Re-timer transmits Symbols using its local clock.
- Phase 3 – Re-timer Channel is transmitting received data using the recovered clock.

A Re-timer Channel will always go through Phases 1 and Phases 3 when exiting the CL1 or CL2 state. A Re-timer Channel may or may not go through Phase 2.

4.2.4.3.2.1 Phase 1

When a Re-timer detects an LFPS burst of 2 cycles on one of its Lane Adapters, it shall do the following:

1. Send LFPS as follows:
 - The Lane Adapter that detected the LFPS shall send LFPS for at least 5 LFPS cycles and no more than tLFPSDuration.
 - The Corresponding Adapter shall send LFPS until it detects LFPS.
2. Enable the receivers for the Lane Adapter and its Corresponding Adapter.
 - The Re-timer shall wait at least tIdleRx after a Lane Adapter stops detecting LFPS before enabling the receiver for that Lane Adapter.
3. For each Adapter, after the last LFPS is transmitted, transition the transmitter to Electrical Idle for tPreData. Then, start transmitting CL_WAKE1.X Symbols, where X is the index of the Re-timer provided by the Router that is the target of the CL_WAKE1.X Symbols. The Re-timer shall transmit the Symbols using a locally generated, non-SSC, clock.

4.2.4.3.2.2 Phase 2

When a Re-timer receives 3 CL_WAKE2.(X+1) Symbols (where X is the Re-timer Index programmed by the Router that is the source of the CL_WAKE2.(X+1) Symbols), and if the Re-timer is still transmitting on its local clock in this Re-timer Channel, then the Re-timer shall transition this Re-timer Channel to toggle between transmitting two locally-generated CL_WAKE1.X Symbols and transmitting the last two CL_WAKE2.Y Symbols received by the Re-timer Channel. CL_WAKE1 Symbols received by the Port shall not be transmitted while the Port is in toggling mode. Only CL_WAKE2 Symbols received by the Port are transmitted.

The transition may or may not take place on Symbol boundary.

4.2.4.3.2.3 Phase 3

After a Re-timer receives 3 back-to-back CL_WAKE2.X Symbols (where X is the Re-timer Index programmed by the Router that is the source of the CL_WAKE2.X Symbols) on Lane 0 of a Re-timer Channel in a given direction, the Re-timer shall transition all Re-timer Channels in the opposite direction to transmit on the clock recovered from the received traffic rather than on its local clock. A Re-timer Channel shall ignore any received CL_WAKE1.Y (where Y is any value) Symbols interleaved with CL_WAKE2.X Symbols when it determines the reception of back-to-back CL_WAKE2.X Symbols.

- The transition shall happen only after bit lock is achieved by both active receivers in the Re-timer Channel performing the transition.
- The transition may or may not take place on Symbol boundary.
- Each Re-timer Channel performing the transition shall transition to Forwarding state. From this point on, the Re-timer Channel shall forward the bit stream it receives from the Lane and stop generating CL_WAKE1.X Symbols.
- During the transition from local clock to receiver clock, the Re-timer Channel shall meet the SSC_SLEW_RATE requirement as specified in the USB4 Specification.

4.2.4.3.3 Timing Requirements

A Re-timer shall meet the following timing requirements during exit from CL0, CL1, or CL2 states:

- A receiver shall complete Symbol lock within tCLxLock of receiving SLOS or CL_WAKE1.X Symbols.
- A transmitter shall complete the transition to a recovered clock within tSwitchNoSSC if the received clock is a non-SSC clock.
- A transmitter shall complete the transition to a recovered clock within tSwitchSSC if the received clock is an SSC clock.

4.3 Lane Decoding

To track entry into CLx states and to participate in exit from CLx states, a Re-timer shall decode Ordered Sets received on its Lane Adapters as defined in the USB4 Specification.



IMPLEMENTATION NOTE

It is recommended that a Re-timer implement RS-FEC error detection. Not implementing RS-FEC error detection may cause the Re-timer to lose track on the RS-FEC activation on the Link and prevent it from entering CLx states.



IMPLEMENTATION NOTE

The path for decoding Ordered Sets is independent of the forwarding path (see section 4.2.3.2). Decoding Ordered sets does not affect the Ordered Set that is forwarded by the Re-timer.

4.3.1 Error Cases

Table 4-4 lists the error cases that a Re-timer shall support along with how that error shall be handled.

Table 4-4. Required Error Cases

Error	Event	Response
SLOS detection	Lane Adapter detects 3 SLOS in Forwarding state with RS-FEC on or Lane Adapter detects 15 SLOS in Forwarding state with RS-FEC off.	If RS-FEC decoding is on, turn-off RS-FEC decoding on this Lane in both USB4 Ports and perform Symbol lock on received SLOS.

Table 4-5 lists the additional error cases that a Re-timer may optionally support along with how that error shall be handled.

Table 4-5. Optional Error Cases

Error	Event	Response
RS-FEC Decoder Error	The RS-FEC decoder identifies an uncorrectable error.	Turn off RS-FEC decoding on this Lane. Perform Symbol lock on received SLOS.

4.4 Timing Parameters

Table 4-6 lists the timing parameters for a Re-timer.

Table 4-6. Re-timer Timing Parameters

Parameter	Description	Min	Max	Units
tSkew	Additional skew allowed for a transmitter between Lane 0 and Lane 1. Also, the difference in propagation delay between Re-timer Channels in a pair of Corresponding Adapters.	--	8	ns
tConnectRx	The time that SBRX will stay at logical high to signal a connect event.	25	--	μs
tDisconnectRx	The time that SBRX will stay at logical low to signal a disconnect event.	14	1000	μs
tLatency2	At Gen 2 speed, the time from when a bit is received to the time it is transmitted by the Corresponding Transmitter.	--	50	ns
tLatency3	At Gen 3 speed, the time from when a bit is received to the time it is transmitted by the Corresponding Transmitter.	--	30	ns
tPollTxFFE	The rate of polling during the TxFFE flows.	1	5	ms

tTxOff	Time between sending the last CL_OFF Ordered Set and shutting off the transmitters of a requesting USB4 Port when entering a CLx state.	--	100	ns
tIdleRx	The time for a receiver to wait after the last LFPS cycle received before starting calibration.	130	--	ns
tCLxLock	The time to achieve Symbol lock during exit from a CLx state	--	60	μs
tSwitchNoSSC	The time to perform clock switch during exit from a CLx state, with a non-SSC receive clock.	--	10	μs
tSwitchSSC	The time to perform clock switch during exit from a CLx state, with an SSC receive clock.	--	60	μs
tRxShut	Number of Symbols to forward before receiver may shut down.	300	--	Symbol
tEnterCLx	The time the transmitter transmits DC balanced signal before Channel enters to CLx.	372	410	Symbol time
tLROff	Timeout to detect that the Link enters sleep state.	30	50	ms

Note: tLFPSDuration and tEnterLFPS1 are defined in the USB4 Specification.

5 Port Operations

When the Opcode register in SB Register Space is written, a Re-timer Port shall execute the Port Operation associated with the Opcode register using the information in the Metadata and Data registers.

After executing the Port Operation, the Re-timer Port updates the Opcode, Metadata, and Data register as follows:

- If the Re-timer Port successfully completed the Port Operation, it shall set the Opcode register to 0. The Re-timer Port shall update the Metadata register with completion metadata (if the Port Operation is defined to return metadata), and the Data register with completion data (if the Port Operation is defined to return data).
- Else, if the Port Operation is not supported, the Re-timer Port shall set the Opcode register to a FourCC value of “!CMD” (444D4321h). The USB4 Port may update the Metadata and Data registers. However, any updates will be ignored by the Connection Manager.
- Else, the Re-timer Port shall set the Opcode register to a FourCC value of “ERR ” (20525245h) to indicate that the Port Operation is supported, but could not be completed. The USB4 Port may update the Metadata and Data registers. However, any updates will be ignored by the Connection Manager.

A Re-timer Port shall support the Port Operations listed in Table 5-1.

Table 5-1. Port Operation Supported by a Re-timer (Required)

Port Operation	Opcode ⁵	Operation		Completion		Reference
		Metadata DW	Data DWs	Metadata DW	Data DWs	
QUERY_LAST_RE-TIMER	LAST (5453414Ch)	0	0	1	0	Section 5.1.1
QUERY_CABLE_RE-TIMER	CBLR (524C4243h)	0	0	1	0	Section 5.1.2
SET_TX_COMPLIANCE	TXCM (4D435854h)	1	0	0	0	USB4 Specification
SET_RX_COMPLIANCE	RXCM (4D435852h)	1	0	0	0	USB4 Specification
GET_RX_TXFFE_PRESET	GRTP (50545247h)	1	0	0 or 1	0	USB4 Specification
ACK_RX_TXFFE_PRESET	ARTP (50545241h)	1	0	0	0	USB4 Specification
START_BER_TEST	SBER (52454253h)	1	0	0	0	USB4 Specification

⁵ Byte 0 of the Opcode is the rightmost byte of the hexadecimal representation.

END_BER_TEST	EBER (52454245h)	1	0	0	2	USB4 Specification
END_BURST_TEST	BBER (52454242h)	1	0	0	3	USB4 Specification
READ_BURST_TEST	RBER (52454252h)	1	0	0	3	USB4 Specification
SET_INBOUND_SBTX	LSUP (5055534Ch)	0	0	0	0	Section 5.2.1
UNSET_INBOUND_SBTX	USUP (50555355h)	0	0	0	0	Section 5.2.2
GET_NVM_SECTOR_SIZE	GNSS (53534E47h)	0	0	1	0	Section 5.2.3
NVM_SET_OFFSET	BOPS (53504F42h)	1	0	0	0	Section 5.2.4
NVM_BLOCK_WRITE	BLKW (574B4C42h)	0	16	0	0	Section 5.2.5
NVM_AUTH_WRITE	AUTH (48545541h)	0	0	1	0	Section 5.2.6
NVM_READ	AFRR (52524641h)	1	0	0	0 to 16	Section 5.2.7
READ_LANE_MARGIN_CAP	RDCP (50434452h)	0	0	0	2	USB4 Specification
RUN_HW_LANE_MARGINING	RHMG (474D4852h)	1	0	0	2	USB4 Specification
RUN_SW_LANE_MARGINING	RSMG (474D5352h)	1	0	0	0	USB4 Specification
READ_SW_MARGIN_ERR	RDSW (57534452h)	0	0	1	0	USB4 Specification

5.1 Enumerating Port Operations

The Port Operations defined in the subsections below are used to identify and enumerate the Re-timers in a Link. The following is an example Port Operation sequence used for the purpose:

1. Set the *Index* field to 1 to access a Re-timer with a Re-timer Index that equals 1.
2. Issue an Addresses RT Command that reads register 0 in the SB Register Space of the Re-timer.
3. Poll the *Pending* bit in the USB4 Port Capability.
4. Read the result of the Command from the *No Response* bit in the USB4 Port Capability.

5. If the *No Response* bit is set to 0b, repeat from step 2, incrementing the *Index* field value. If the *No Response* bit is set to 1b, the number of Re-timers is the last *Index* field value sent minus 1. If no Re-timer is detected, exit. Else, go to next step.
6. Set the *Index* field to 1 to access a Re-timer with a Re-timer Index that equals 1.
7. Issue a QUERY_CABLE_RE-TIMER Operation to the Re-timer. Issue consecutive QUERY_CABLE_RE-TIMER Operations, each time incrementing the *Index* field, as long as the response contains a *Cable Re-timer* bit set to 1b and as long as the last Re-timer has not been queried. Capture the number of Cable Re-timers identified. If all Re-timers have been identified, exit. Else, continue to the next step.
8. Issue a QUERY_LAST_RE-TIMER Operation to the Re-timer with the *Index* field value. Issue consecutive QUERY_LAST_RE-TIMER Operations, each time incrementing the *Index* field, as long as the response contains a *Last* bit set to 0b and as long as the last Re-timer has not been queried. Capture the number of On-Board Re-timers identified. If all Re-timers have been identified, exit. Else, continue to the next step.
 - If Cable Re-timers have been identified in step 7, then the Re-timers identified in step 8 are far-end Re-timers.
 - If Cable Re-timers have not been identified in step 7, and the last Re-timer accessed in step 8 returned a *Last* bit set equal to 0b, then the Re-timers identified in step 8 are far-end Re-timers.
 - Else, the Re-timers identified in step 8 are near-end On-Board Re-timers.

Increment the *Index* field. Issue a QUERY_CABLE_RE-TIMER Operation to the Re-timer. Issue consecutive QUERY_CABLE_RE-TIMER Operations, each time incrementing the *Index* field, as long as the response contains a *Cable Re-timer* bit set to 1b and as long as the last Re-timer has not been queried. Capture the number of Cable Re-timers identified. If all Re-timers have been identified, exit. Else, continue to the next step.

Any Re-timer with a Re-timer Index equal or larger than the *Index* field value is a far-end On-Board Re-timer.

5.1.1 QUERY_LAST_RE-TIMER

The QUERY_LAST_RE-TIMER Port Operation checks if an On-Board Re-timer is directly connected to a USB Type-C connector.

This Port Operation does not have Operation Metadata.

This Port Operation does not have Operation Data.

The Completion for this Operation does not have Completion Data.

Table 5-2 describes the Completion Metadata that a Re-timer shall return.

Table 5-2. QUERY_LAST_RE-TIMER Completion Metadata

DW	Bits	Field Name and Description
0	0	Last 0b – The target of the Operation is not an On-Board Re-timer directly connected to a USB Type-C connector. 1b – The target of the Operation is an On-Board Re-timer directly connected to a USB Type-C connector.
0	31:1	Reserved

5.1.2 QUERY_CABLE_RE-TIMER

The QUERY_CABLE_RE-TIMER Port Operation checks if a Re-timer is a Cable Re-timer.

This Port Operation does not have Operation Metadata.

This Port Operation does not have Operation Data.

The Completion for this Operation does not have Completion Data.

Table 5-3 describes the Completion Metadata that a Re-timer shall return.

Table 5-3. QUERY_CABLE_RE-TIMER Completion Metadata

DW	Bits	Field Name and Description
0	0	Cable Re-timer 0b – The target of the Operation is not a Cable Re-timer 1b – The target of the Operation is a Cable Re-timer
0	31:1	Reserved

5.2 NVM Port Operations

The Port Operations defined in the subsections below are used to update the NVM of a Re-timer through the Sideband Channel. This section applies to On-Board Re-timers and Cable Re-timers.

The following is an example of a Port Operation sequence used by software to enable Sideband Channel Transactions with On-Board Re-timers before updating the NVM of an On-Board Re-timer:

1. Software issues a ROUTER_OFFLINE_MODE Port Operation to a Router with the *Enter Offline Mode* bit set to 0b. After receiving such a Port Operation, the Router ignores connect and disconnect events until it receives a ROUTER_OFFLINE_MODE Port Operation with the *Enter Offline Mode* bit set to 1b (see step 2 in sequence below to exit this mode).

2. Software issues an ENUMERATE_RE-TIMERS Port Operation, which causes the Router to send an RT Broadcast Transaction.
3. Software issues a SET_INBOUND_SBTX Port Operation to each On-Board Re-timer, starting with the Re-timer with Re-timer Index =1. The Port Operation enables Sideband Channel Transactions on the SBTX line from the Re-timer towards the Router.

The following is an example of a Port Operation sequence used by software to update the NVM of a Re-timer:

1. Software issues a GET_NVM_SECTOR_SIZE Operation to read the sector size of the NVM.
2. Software issues a NVM_SET_OFFSET Port Operation, which sets the first location in NVM to be written by the following NVM_BLOCK_WRITE Port Operation.
3. Software issues a sequence of NVM_BLOCK_WRITE Port Operation, each writing a 64B block of data to NVM.
 - Following a NVM_BLOCK_WRITE Port Operation, if the value of the *Opcode* register is “ERR”, software repeats all previous NVM_BLOCK_WRITE Port Operation. The first Port Operation to be issued is a NVM_SET_OFFSET Port Operation that sets the location in NVM of the first block to be rewritten.
4. Software issues a NVM_AUTH_WRITE Port Operation to indicate to the target that all data was sent to the target. After receiving the NVM_AUTH_WRITE Port Operation, the target performs an authentication check over the data written.

The following Port Operation sequence is used by software to validate the results of the NVM update:

1. Software then waits for at least 5 seconds from the completion of the NVM_AUTH_WRITE Port Operation. It then issues another ENUMERATE_RE-TIMERS Port Operation, which causes the Router to send an RT Broadcast Transaction.
2. Software reads the results of the NVM_AUTH_WRITE Port Operation. If the *Status* field in the Completion Metadata is 0h, the update completed successfully. Otherwise, software repeats all previous NVM_BLOCK_WRITE Port Operation. The first Port Operation to be issued is a NVM_SET_OFFSET Port Operation that sets the location in NVM of the first block to be rewritten, followed by issuing one or more NVM_AUTH_WRITE Port Operation again.

If this is the case of updating the NVM of an On-Board Re-timer, then the following Port Operation sequence is used by software to exit this mode:

1. Software issues a UNSET_INBOUND_SBTX Port Operation to each On-Board Re-timer, starting with the Re-timer closest to the USB Type-C connector. The Port Operation sets the SBTX line from the Re-timer towards the Router to the logical level of the SBRX line from the direction of the USB Type-C connector.
2. Software issues an ROUTER_OFFLINE_MODE Port Operation to the Router with the *Enter Offline Mode* bit set to 1b. After receiving such a Port Operation, the Router is ready to process connect events.

5.2.1 SET_INBOUND_SBTX

Software uses the SET_INBOUND_SBTX Port Operation to enable Sideband Channel Transactions on the SBTX of a Re-timer. The SBTX that faces towards the Router is enabled.

A Re-timer that is the target of the Port Operation drives its SBTX line in the direction of the Router to logical high and enables the transmission of RT Responses towards the Router. The Re-timer that is the target of the Port Operation is not required to send an RT Response for the first SET_INBOUND_SBTX Port Operation.

Note: This Port Operation does not cause bi-directional communication on SBTX. Communication over SBTX remains unidirectional.

This Port Operation does not have Operation Metadata.

This Port Operation does not have Operation Data.

The Completion for this Operation does not have Completion Metadata.

The Completion for this Operation does not have Completion Data.

5.2.2 UNSET_INBOUND_SBTX

A Re-timer that is the target of the UNSET_INBOUND_SBTX Port Operation drives its SBTX line in the direction of the Router to the logical level of the SBRX line from the direction of the USB Type-C connector.

This Port Operation does not have Operation Metadata.

This Port Operation does not have Operation Data.

The Completion for this Operation does not have Completion Metadata.

The Completion for this Operation does not have Completion Data.

5.2.3 GET_NVM_SECTOR_SIZE

The GET_NVM_SECTOR_SIZE Port Operation returns the size of an NVM sector.

This Port Operation does not have Operation Metadata.

This Port Operation does not have Operation Data.

The Completion for this Operation does not have Completion Data.

Table 5-4 describes the Completion Metadata that a Re-timer shall return.

Table 5-4. GET_NVM_SECTOR_SIZE Completion Metadata

DW	Bits	Field Name and Description
0	23:0	Sector Size Equal to the sector size of the NVM in bytes. For example, a value of 1000h indicates a sector size of 4KB.
0	31:24	Reserved

5.2.4 NVM_SET_OFFSET

The NVM_SET_OFFSET Port Operation sets the first location in NVM to be written by the following NVM_BLOCK_WRITE Port Operation.

This Port Operation does not have Operation Data. Table 5-5 describes the Operation Metadata for this Port Operation.

Table 5-5. NVM_SET_OFFSET Operation Metadata

DW	Bits	Field Name and Description
0	1:0	Reserved
0	23:2	NVM Offset This field contains the first address to be written relative to the base address of the region been written. NVM Offset is incremented by 16 after each NVM_BLOCK_WRITE (see Section NVM_BLOCK_WRITE). Address is specified in DWs.
0	31:24	Reserved

The Completion for this Operation does not have Completion Metadata.

The Completion for this Operation does not have Completion Data.

5.2.5 NVM_BLOCK_WRITE

The NVM_BLOCK_WRITE Port Operation writes 64B NVM, starting at the address equal to the NVM Offset. A Re-timer shall increment its NVM Offset value by 16 after executing this Port Operation (even if the write operation fails).

If the write operation fails, the Port Operation shall complete with an “ERR ” status.

This Port Operation does not have Operation Metadata. Table 5-6 describes the Operation Data for this Port Operation.

Table 5-6. NVM_BLOCK_WRITE Operation Data

DW	Bits	Field Name and Description
0	31:0	DW0 The first Doubleword, to be written into NVM at NVM Offset address.
1	31:0	DW1 The second Doubleword, to be written into NVM at the next NVM Offset.
...	31:0	...
15	31:0	DW15 The last Doubleword of this Port Operation, to be written into NVM at the next NVM Offset.

The Completion for this Operation does not have Completion Metadata.

The Completion for this Operation does not have Completion Data.

5.2.6 NVM_AUTH_WRITE

The NVM_AUTH_WRITE Port Operation indicates to the Re-timer that all data was sent to it. The Re-timer performs an authentication check over the data written. The authentication check is implementation specific and outside the scope of this specification.

Following the execution of an NVM_AUTH_WRITE Port Operation, the Re-timer shall maintain the SBTX line of its Router-Facing USB4 Port with the same state as before the execution of the Operation. The Re-timer may maintain its Re-timer Index value but is not required to do so.

This Port Operation does not have Operation Metadata.

This Port Operation does not have Operation Data.

The Completion for this Operation does not have Completion Data.

Table 5-7 describes the Completion Metadata that a Re-timer shall return.

Table 5-7. NVM_AUTH_WRITE Completion Metadata

DW	Bits	Field Name and Description
0	5:0	Status 0h - Authentication completed successfully 1h - Authentication Failed 2h - Retry NVM write Else - Reserved
0	31:6	Reserved

If the *Status* field in the Completion Metadata is 2h, then software repeats all previous NVM_BLOCK_WRITE Port Operation. The first Port Operation to be issued is a NVM_SET_OFFSET Port Operation that sets the location in NVM of the first block to be rewritten, followed by issuing a NVM_AUTH_WRITE Port Operation again.

5.2.7 NVM_READ

The NVM_READ Port Operation reads up to 64 bytes from NVM.

This Port Operation does not have Operation Data. Table 5-8 describes the Operation Metadata.

Table 5-8. NVM_READ Operation Metadata

DW	Bits	Field Name and Description
0	1:0	Reserved
0	23:2	NVM Offset This field contains the first address to be read relative to the base address of the region being read. NVM Offset is specified in DWs.
0	27:24	Length Number of Doublewords that shall be read starting from the <i>NVM Offset</i> field value. If this field is zero, then 16 DWs are read.
0	31:28	Reserved

The Completion for this Operation does not have Completion Metadata.

If completed successfully, the Port Operation shall return *Length* number of Doublewords of Completion Data as described in Table 5-9.

Table 5-9. NVM_READ Completion Data

DW	Bits	Field Name and Description
0	31:0	DW0 The first Doubleword read from NVM at <i>NVM Offset</i> address.
1	31:0	DW1 The second Doubleword read from NVM at the next NVM Offset.
...	31:0	...
15	31:0	DW15 The last Doubleword read from NVM at the next NVM Offset.

5.3 Receiver Lane Margining Port Operations

A USB4 Port shall perform the Receiver Lane Margining Port Operations as defined in the USB4 Specification. A Re-timer shall execute a Receiver Lane Margining Port Operation on the Port on which the operation is received.

6 Interoperability with Thunderbolt™ 3 (TBT3) Systems

This section defines the requirements for an On-Board Re-timer to operate as part of a Link that includes a Thunderbolt 3 cable and/or one or more Thunderbolt 3 Routers. It also defines the requirements for a Cable Re-timer to operate as part of a Link that includes a Thunderbolt 3 Router.

A Cable Re-timer shall support the requirements defined in this chapter.

An On-Board Re-timer may optionally support the requirements defined in this chapter.

6.1 Electrical Layer

A Cable Re-timer shall support TBT3-Compatible Gen 2 speed (10.3125 Gbps) and TBT3-Compatible Gen 3 speed (20.625 Gbps).

An On-Board Re-timer shall support TBT3-Compatible Gen 2 speed (10.3125 Gbps).

6.2 Logical Layer

6.2.1 Sideband Channel

When operating in a TBT3-Compatible Link, a Re-timer uses either a TBT3-Compatible Sideband Channel or a USB4-Compatible Sideband Channel, depending on what the Routers on each side of the Link support (see Section 13.2.1.4.2 in the USB4 Specification for more detail).

This section defines the additional Sideband Channel behavior that a Re-timer shall implement when operating in a TBT3-Compatible that uses a TBT3-Compatible Sideband Channel.

Note: A Re-timer on a TBT3-Compatible Link with a TBT3-Compatible Sideband will always have RS-FEC disabled since there is no Broadcast RT Transaction to tell the Re-timer about the RS-FEC enabling decision. A TBT3-Compatible Link with a TBT3-Compatible Sideband does not support CLx states.

6.2.1.1 Bidirectional Re-timers

A bidirectional Re-timer is a Re-timer that sends Transactions on its SBRX wire. A unidirectional Re-timer does not send Transaction on its SBRX wire. The Cable Re-timers in a Thunderbolt active cable are bidirectional. The Router and On-Board Re-timers in a Thunderbolt host or device are unidirectional.

This section defines when a USB4 Re-timer does and does not behave as a bidirectional Re-timer in order to maintain TBT3-Compatability.

6.2.1.1.1 Cable Re-Timers

A Cable Re-timer shall implement a unidirectional behavior when the Sideband Channel operates in TBT3-Compatible mode. A Cable Re-timer does not need to implement bi-directional behavior.

6.2.1.1.2 On-Board Re-Timers

An On-Board Re-timer that is adjacent to a USB Type-C connector shall implement both unidirectional and bidirectional behavior.

- When the Re-timer is adjacent to a Thunderbolt Cable Re-timer, it shall operate with a bidirectional Sideband Channel on its Cable-Facing USB4 Port. The Router-Facing USB4 Port shall operate with a unidirectional Sideband Channel.
- When the Re-timer is not adjacent to a Thunderbolt Cable Re-timer, it shall operate with a unidirectional Sideband Channel on both USB4 Ports.

An On-Board Re-timer that is not adjacent to a USB Type-C connector shall implement unidirectional behavior on both USB4 Ports.

When a USB4 Port on an On-Board Re-timer is operating in bidirectional mode:

- The USB4 Port shall support concurrent reception of Transactions on SBTX and on SBRX.
- The USB4 Port shall drive its SBTX for up to 2 bit times after the last Stop bit of an AT Command.
- The USB4 Port shall not forward a received Transaction if it is still waiting for a Response to a RT Command it sent.

6.2.1.2 Transactions

6.2.1.2.1 LT Transactions

A Re-timer shall forward an LT_Resume Transaction received on one USB4 Port to its other USB4 Port.

A Cable Re-timer shall forward an LT_Resume2 Transaction received on its Router-Facing USB4 Port to its Cable-Facing USB4 Port.

A Re-timer shall forward a received LT_Gen_2 Transaction to its other USB4 Port.

A Re-timer shall forward a received LT_Gen_3 Transaction to its other USB4 Port.

6.2.1.2.2 AT Transactions

6.2.1.2.2.1 Cable Re-timers

A Cable Re-timer shall support the Bounce mechanism as defined in the USB4 Specification.

A Cable Re-timer shall not initiate AT Commands.

A Cable Re-timer shall respond to a received AT Command that has the *Recipient* bit set to 0b and the *Bounce* bit set to 0b. Else, it shall forward the received AT Command to its other USB4 Port.

6.2.1.2.2.2 On-Board Re-timers

An On-Board Re-timer may generate AT Commands (see Section 6.2.1.3.3).

An On-Board Re-timer shall respond to a received AT Command with the *Recipient* bit set to 1b that access the TxFFE Register in the SB Register Space. It shall forward other received AT Commands to its other USB4 Port.

An On-Board Re-timer shall process to a received AT Response with the *Recipient* bit set to 1b that access the TxFFE Register in the SB Register Space. It shall not forward the AT Response to its other USB4 Port. It shall forward other received AT Responses to its other USB4 Port.

6.2.1.2.3 RT Transactions

An On-Board Re-timer that is adjacent to a USB Type-C connector shall not forward Broadcast RT Transactions towards the cable.

It is recommended that an On-Board Re-timer not forward Addressed RT Transactions towards the cable.

6.2.1.2.4 SB Register Space

The SB Register Space of a Re-timer shall have the additional fields described in Table 6-1 and Table 6-2.

Table 6-1. Re-timer SB Registers

Register	Size (Bytes)	Name	Description
13	8	TxFFE	Used to set the TxFFE parameters of transmitters. Bytes 4-7 in this register are Rsvd for On-Board Re-timers.

Table 6-2. SB Register Fields

Register	Register Name	Byte	Bits	Field Name and Description	Type	Default Value
13	TxFFE	2	5	Forward Switch Done (Lane 0) Indicates that the transmitter transmits using the Router's clock: 0b – Transmitter uses a Re-timer's clock 1b – Transmitter uses Router's clock	RO	0b
		3	5	Forward Switch Done (Lane 1) Indicates that the transmitter transmits using the Router's clock: 0b – Transmitter uses a Re-timer's clock 1b – Transmitter uses Router's clock	RO	0b
		4 ⁶	3:0	TxFFE Request (Lane 0) Identifies one of 16 predefined TxFFE configurations requested by the receiver.	RW	0
			4	Rsvd	Rsvd	0b

⁶ This byte is the *Partner Rx Status & TxFFE Request* byte for Lane 0

			5	Rx Active (Lane 0) Indicates whether or not the receiver is active: 0b – Receiver is inactive 1b – Receiver is active	RW	0b
			6	Rsvd	Rsvd	0
			7	New Request (Lane 0) Indicates whether or not the receiver is providing a new index in the <i>TxFFE Request</i> field: 0b – Receiver is still processing a previous TxFFE configuration 1b – Receiver is providing a new index in the <i>TxFFE Request</i> field	RW	0b
		5 ⁷	3:0	TxFFE Request (Lane 1) Identifies one of 16 predefined TxFFE configurations requested by the receiver.	RW	0
			4	Rsvd	Rsvd	0b
			5	Rx Active (Lane 1) Indicates whether or not the receiver is active: 0b – Receiver is inactive 1b – Receiver is active	RW	0b
			6	Rsvd	Rsvd	0
			7	New Request (Lane 1) Indicates whether or not the receiver is providing a new index in the <i>TxFFE Request</i> field: 0b – Receiver is still processing a previous TxFFE configuration 1b – Receiver is providing a new index in the <i>TxFFE Request</i> field	RW	0b
		6 ⁸	3:0	TxFFE setting (Lane 0) Index of the TxFFE configuration loaded the transmitter.	RW	0
			6:4	Rsvd	Rsvd	0

⁷ This byte is the *Partner Rx Status & TxFFE Request* byte for Lane 1

⁸ This byte is the *Partner Tx Status* byte for Lane 0

			7	Tx Active (Lane 0) Indicates whether or not the transmitter is transmitting a valid signal: 0b – Transmitter is not transmitting a valid signal 1b – Transmitter is transmitting a valid signal	RW	0b
		7 ⁹	3:0	TxFFE setting (Lane 1) Index of the TxFFE configuration loaded the transmitter.	RW	0
			6:4	Rsvd	Rsvd	0
			7	Tx Active (Lane 1) Indicates whether or not the transmitter is transmitting a valid signal: 0b – Transmitter is not transmitting a valid signal 1b – Transmitter is transmitting a valid signal	RW	0b

6.2.1.3 Lane Initialization

6.2.1.3.1 Phase 1 – Determination of Initial Conditions

During phase 1, an On-Board Re-timer uses the mechanism defined in the USB Type-C Specification and the USB PD Specification to obtain the following additional information for each of its USB4 Ports:

- Cable incorporates Thunderbolt 3 bidirectional Re-timers (True/False)
- Passive or Active Cable (Passive/Active)

6.2.1.3.2 Phase 4 – Lane Parameters Synchronization

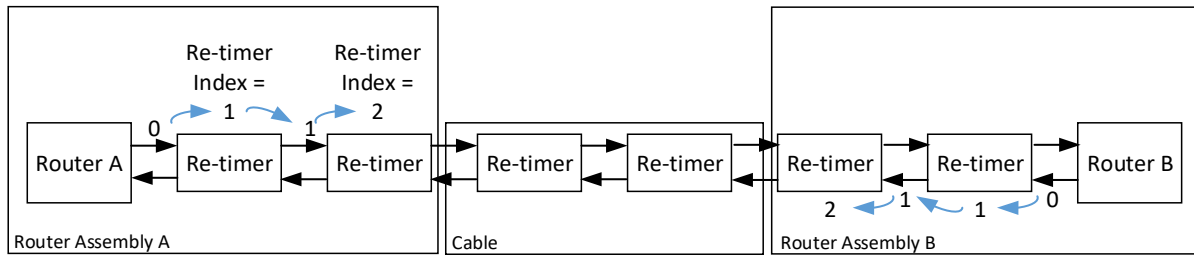
This section replaces Section 4.1.2.4.

An On-Board Re-timer shall decode Broadcast RT Transactions, LT_Gen_2 Transactions, and LT_Gen_3 Transactions, and shall use the Link parameter from the most recently received Transaction.

Figure 6-1 depicts an example of how a Broadcast RT Transaction is used to assign indexes in a Thunderbolt 3-compatible system to a Link with 6 Re-timers. Each Router enumerates the On-Board Re-timers in its Router Assembly.

⁹ This byte is the Partner Tx Status byte for Lane 1

Figure 6-1. Propagation of Broadcast RT Transactions



A Cable Re-timer shall decode LT_Gen_2 Transactions and LT_Gen_3 Transactions, and shall use the Link parameter from the most recently received LT Transaction.

If, on any USB4 Port, a Cable Re-timer detects LT_Resume for Lane 0 before it detects at least one LT_Gen_2 Transaction or LT_Gen_3 Transaction for Lane 1, then the Re-timer shall keep the Lane 1 Adapters in both USB4 Ports in the CLd state and shall not proceed with Lane Initialization.

When a Re-timer detects LT_Resume on any Lane of any USB4 Port, it shall transition to phase 5.

6.2.1.3.3 Phase 5 – Lane Equalization

This section replaces Section 4.1.2.5.

Re-timer type and Lane Adapter position determine which TxFFE negotiation flow a transmitter or receiver performs. Table 6-3 lists which TxFFE negotiation flows a transmitter or receiver shall perform in phase 5.

Table 6-3. TxFFE Negotiation flows

Re-timer Type	Lane Adapter Position	Rx/Tx	TxFFE Negotiation Flow	Reference	Transactions
On-Board Re-timer	Adjacent to a Router or an On-Board Re-timer in the same Router Assembly	Receiver or Transmitter	Symmetric TxFFE Negotiation flow	USB4 Specification	Addressed RT Transactions, with the <i>Index</i> field set to 0
On Board Re-timer	Adjacent to a Cable Re-timer	Transmitter only	Asymmetric TxFFE Parameter Negotiation with a Transmitting Primary Partner flow	USB4 Specification	AT Transactions, with the <i>Recipient</i> bit set to 0
On-Board Re-timer	Adjacent to a Cable Re-timer	Receiver only	Asymmetric TxFFE Parameter Negotiation with a Receiving Primary Partner	USB4 Specification	AT Transactions, with the <i>Recipient</i> bit set to 0
On-Board Re-timer	Adjacent to a Passive Cable	Receiver or Transmitter	Symmetric TxFFE Negotiation flow	USB4 Specification	AT Transactions, with the <i>Recipient</i> bit set to 1

Cable Re-timer	Adjacent to a USB Type-C connector	Transmitter only	Asymmetric TxFFE Parameter Negotiation with a Receiving Subordinate Partner	Section 6.2.1.3.3.1	AT Transactions, with the <i>Recipient</i> bit set to 0
Cable Re-timer	Adjacent to a USB Type-C connector	Receiver only	Asymmetric TxFFE Parameter Negotiation with a Transmitting Subordinate Partner	Section 6.2.1.3.3.2	AT Transactions, with the <i>Recipient</i> bit set to 0

The following rules determine the behavior of an On-Board Re-timer in phase 5:

- An On-Board Re-timer Lane Adapter that is not adjacent to a USB Type-C connector shall turn on its transmitter when the *Rx Active* bit of its Corresponding Receiver is set to 1b. It shall start transmitting CL_WAKE1.X Symbols, where X is the Re-timer Index assigned by the Router that is the target of the CL_WAKE1.X Symbols. The transmitter shall use a locally generated, non-SSC clock to transmit the CL_WAKE1.X Symbols. It shall then set the *Tx Active* bit for the Lane Adapter to 1b.
- A Re-timer adjacent to a USB Type-C connector shall set the *Clock Switch Done* bit to 1b in a Router-Facing Adapter when all receivers in Router-Facing Adapters complete TxFFE negotiation. The Re-timer may turn on transmitters adjacent to the USB Type-C connector. Once the *Clock Switch Done* bit is set to 1b, a transmitter adjacent to a USB Type-C connector that is turned on shall forward the bit stream it receives from the Corresponding Receiver using a recovered clock.
- When all of the following are true, a transmitter that is not adjacent to a USB Type-C connector shall stop using the local clock, shall start using the recovered clock from the Corresponding Receiver, and shall forward the bit stream it receives from the Corresponding Receiver:
 - In the transmitter's USB4 Port, TxFFE negotiation is complete between all transmitters and their Adjacent Receivers.
 - In the USB4 Port opposite the transmitter's USB4 Port, TxFFE negotiation is complete between all receivers and their Adjacent Transmitter.
 - The *Clock Switch Done* bit for Lane 0 of the transmitter's adjacent USB4 Port is 1b.
- A Re-timer adjacent to a Router shall set the *Forward Switch Done* bit to 1b in a Lane Adapter of a Cable-Facing USB4 Port when transmitter of the Lane Adapter is using the receiver clock from the Corresponding Receiver.
- A Re-timer shall set the *Forward Switch Done* bit to 1b in a Lane Adapter if the *Forward Switch Done* bit is set to 1b in the adjacent USB4 Port of the Corresponding Receiver.
- When the *Forward Switch Done* bit is set to 1b in an adapter adjacent to a USB Type-C connector, the Lane Adapter shall turn on its transmitter, if it has not done so already. It shall then set the *Tx Active* bit to 1b and perform TxFFE Parameter Negotiation.
- When an On-Board Re-timer transmitter completes TxFFE negotiation with a Cable Re-timer receiver, the transmitter shall send an LT_Resume2 Transaction. The *LSELane* field in the LT_Resume2 Transaction shall equal the Lane number associated with the transmitter.

The following rules determine the behavior of a Cable Re-timer in phase 5.

- A Cable Re-timer shall set the *Tx Active* bit to 1b in a Router-Facing Adapter when the Lane Adapter is transmitting the bit stream received by a Lane Adapter at the other end of the Cable, and transmission uses the recovered clock from the Lane Adapter at the other end of the Cable.
- The flow between two Cable Re-timers is implementation specific.

6.2.1.3.3.1 Asymmetric TxFFE Parameter Negotiation with a Receiving Subordinate Partner

During TxFFE negotiation, the transmitter negotiates TxFFE parameters with the receiver at the other end. The transmitting end is defined as the Primary Partner. The receiving end is defined as the Subordinate Partner. The Primary Partner sends AT Command to the Subordinate Partner and the Subordinate Partner sends AT Responses to the Primary Partner.

Receiver flow:

The steps that the receiver shall perform to complete negotiation are listed below:

1. The receiver shall set the following initial values on entry to Phase 5:
 - Local Rx Status & TxFFE Request byte of the TxFFE register
 - *Rx Locked* bit = 0b
 - *TxFFE Request* field = index of an initial set of TXFFE parameters
 - *Rx Active* bit = 0b
 - *New Request* bit = 0b
 - Partner Tx Status byte of the TxFFE register
 - *Tx Active* bit = 0b
 - Request Done bit = 0b
2. The receiver shall evaluate the value of the *Tx Active* bit in the *Partner Tx Status* byte of the *TxFFE* register:
 - If *Tx Active* = 1b (i.e. the transmitter is transmitting), then the receiver shall enable the receiver, set *Rx Active* bit to 1b, and continue to Step 3.
 - Else, repeat Step 2 within tPollTxFFE.
3. The receiver shall evaluate its behavior. If equalization is complete, the receiver shall set the *Rx Locked* field in the *Local Rx Status & TxFFE Request* byte to 1b.

4. The receiver shall do the following:
 - If the *Rx Locked* bit is set to 1b, negotiation is complete and no further TxFFE negotiation steps shall be taken.
 - Else:
 - *TxFFE Request* field shall be set to the index of a selected set of TxFFE parameters.
 - *New Request* bit shall be set to 1b to indicate the receiver is providing a new TxFFE index.
5. Continue to Step 6 only after sending a read Response with the updated values of its TxFFE Register.
6. On reception of an AT Command with a write Command targeting its *Partner Tx Status* byte of the *TxFFE* register, the receiver shall:
 - If (*Tx Active* = 1b) AND (*TxFFE setting* = value of *TxFFE Request* in the *Local Rx Status & TxFFE Request* byte), go to Step 7.
 - Else, repeat Step 6.
7. The receiver shall evaluate its receiver behavior. If equalization is complete, the receiver shall set the *Rx Locked* field in the *Local Rx Status & TxFFE Request* byte to 1b.
8. The receiver shall set the *New Request* field in the *Local Rx Status & TxFFE Request* byte to 0b.
9. On reception of an AT command with a read Command targeting the TxFFE register go to Step 4.

6.2.1.3.3.2 Asymmetric TxFFE Parameter Negotiation with a Transmitting Subordinate Partner

During TxFFE negotiation, the transmitter negotiates TxFFE parameters with the receiver at the other end. The receiving end is defined as the Primary Partner. The transmitting end is defined as the Subordinate Partner. The Primary Partner sends AT Command to the Subordinate Partner and the Subordinate Partner sends AT Responses to the Primary Partner.

Transmitter flow:

The steps that the transmitter shall perform to complete negotiation are listed below:

1. The transmitter shall set the following initial values on entry to Phase 5:
 - *Local Tx Status* byte of the TxFFE register
 - *Tx Active* bit = 1b
 - *Request Done* bit = 0b
 - *TxFFE Setting* field = index of an initial set of TxFFE parameters

- Partner Rx Status & TxFFE Request byte of the TxFFE register
 - *New Request* bit = 0b
- 2. On reception of an AT Command with a write Command targeting the *Partner Rx Status & TxFFE Request* byte of the TxFFE register, the transmitter shall:
 - If (*New Request* = 0b), repeat Step 2.
 - Else, continue to Step 3.
- 3. Load one of 16 predefined TxFFE configurations that matches the *TxFFE Request* field of the received AT Command.
- 4. Send an AT Response to indicate the transmitter is using the new TxFFE request.
- 5. On reception of an AT Command with a write Command targeting the *Partner Rx Status & TxFFE Request* byte of the TxFFE register, the transmitter shall:
 - If (*New Request* = 1b), repeat Step 4.
 - Else, continue to Step 2.

Note: Step 2 is the final step of this flow, the transmitter won't get an indication about the receiver lock status and will wait for the next request even when TxFFE negotiation is done.

6.2.2 Re-timer Channel Layer State Machine

6.2.2.1 CLd State

6.2.2.1.1 Behavior in State

In addition to the conditions described in Section 4.2.1.2, a Re-timer Channel shall begin Lane Initialization in phase 4 when:

- If the Re-timer Channel entered this state after detecting an LT_Fall Transaction, then:
 - The Re-timer Channel for Lane 0 shall start Lane Initialization when the Re-timer receives a Broadcast RT Transaction with the *Lane0 Enabled* bit set to 1b, an LT_Gen_2 Transaction with the *LSELane* bit set to 0b, or an LT_Gen_3 Transaction with the *LSELane* bit set to 0b.
 - The Re-timer Channel for Lane 1 shall start Lane Initialization when the Re-timer receives a Broadcast RT Transaction with the *Lane1 Enabled* bit set to 1b, an LT_Gen_2 Transaction with the *LSELane* bit set to 1b, or an LT_Gen_3 Transaction with the *LSELane* bit set to 1b.

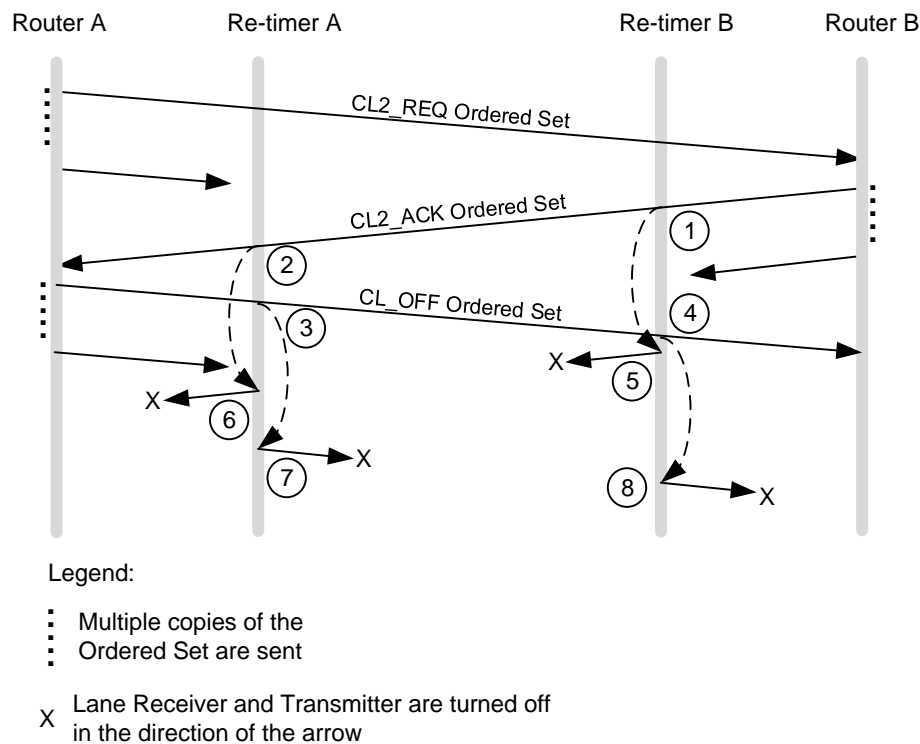
A. Examples of Entry to Low Power State

This section contains examples of Lane Adapter behavior during entry to CL2, CL1, or CL0s states.

A.1 Successful Entry to CL2 state

Figure A-1 shows an example of CL2 entry where two Re-timers are in between the Routers with Lane Adapters entering the CL2 state. In the example, Re-timer A is adjacent to Router A and Re-timer B is adjacent to Router B. Router A initiates entry to CL2 state.

Figure A-1. Successful Entry to CL2 State



The following steps take place in Figure A-1:

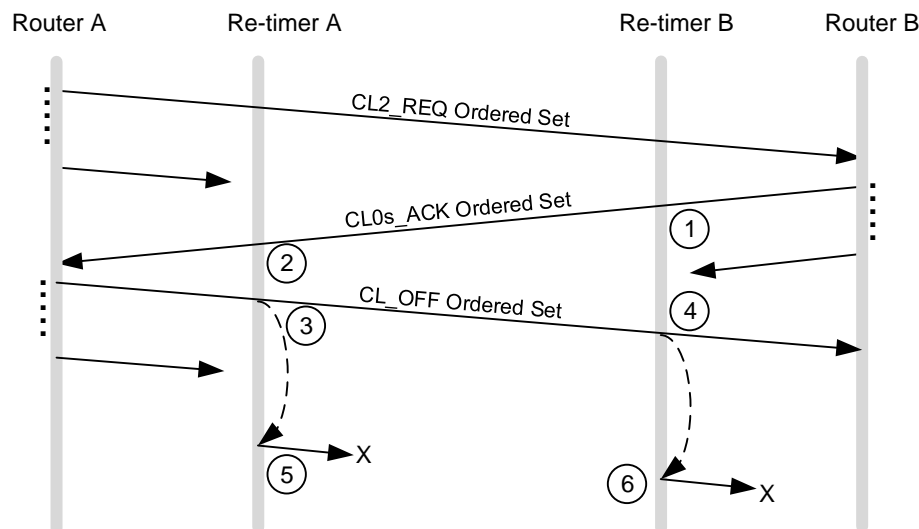
1. Re-timer B detects 3 back-to-back CL2_ACK Ordered Sets and starts a CL2_ACK Counter
2. Re-timer A detects 3 back-to-back CL2_ACK Ordered Sets and starts a CL2_ACK Counter.
3. Re-timer A detects 3 back-to-back CL_OFF Ordered Sets and starts a CL_OFF Counter.
4. Re-timer B detects 3 back-to-back CL_OFF Ordered Sets and starts a CL_OFF Counter.
5. Re-timer B's CL2_ACK Counter reaches a count of 372. Re-timer B then powers down the receiver facing Router B and its Corresponding Transmitter.

6. Re-timer A's CL2_ACK Counter reaches a count 372. Re-timer A then powers down the transmitter facing Router A and its Corresponding Receiver.
7. Re-timer A's CL_OFF Counter reaches a count of 372. Re-timer A then powers down the receiver facing Router A and its Corresponding Transmitter.
8. Re-timer B's CL_OFF Counter reaches a count of 372. Re-timer B then powers down the transmitter facing Router B and its Corresponding Receiver.

A.2 Successful entry to CL0s state

Figure A-2 shows an example of CL0s entry where two Re-timers are in between the Routers with Lane Adapters entering the CL0s state. In the example, Re-timer A is adjacent to Router A and Re-timer B is adjacent to Router B. Router A initiates entry to CL0s.

Figure A-2. Successful Entry to CL0s State



The following steps take place in Figure A-2:

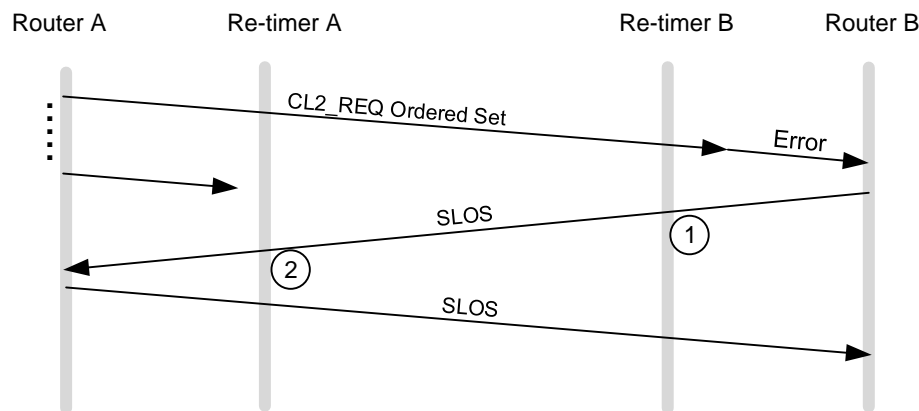
1. Re-timer B detects 3 back-to-back CL0s_ACK Ordered Sets and starts a CL0s_ACK Counter.
2. Re-timer A detects 3 back-to-back CL0s_ACK Ordered Sets and starts a CL0s_ACK Counter.
3. Re-timer A detects 3 back-to-back CL_OFF Ordered Sets and starts a CL_OFF Counter.
4. Re-timer B detects 3 back-to-back CL_OFF Ordered Sets and starts a CL_OFF Counter.
5. Re-timer A's CL_OFF Counter reaches a count of 372. Re-timer A then powers down the receiver facing Router A and its Corresponding Transmitter.

6. Re-timer B's CL_OFF Counter reaches a count of 372. Re-timer B then powers down the transmitter facing Router B and its Corresponding Receiver.

A.3 Error in CL2_REQ Ordered Sets

Figure A-3 shows an example of an attempt by Router A to enter CL2 state, where the CL2_REQ Ordered Sets are received with errors by Router B. In the example, Re-timer A is adjacent to Router A and Re-timer B is adjacent to Router B. Router A initiates entry to CL2.

Figure A-3. Error in CL2_REQ Ordered Sets



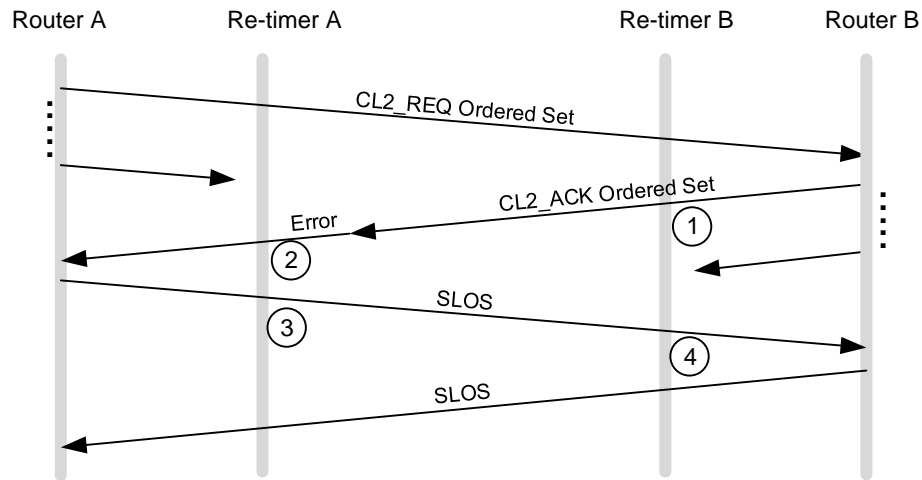
The following steps take place in Figure A-3:

1. Re-timer B detects SLOS or Link errors coming from Router B.
 - If the Re-timer detects errors, it disables RS-FEC in the direction of the errors to allow detection of SLOS.
 - If the Re-timer detects 15 SLOS Symbols, it disables RS-FEC in both directions.
2. Re-timer A detects SLOS or Link errors.
 - If the Re-timer detects errors, it disables RS-FEC in the direction of the errors to allow detection of SLOS.
 - If the Re-timer detects 15 SLOS Symbols, it disables RS-FEC in both directions.
3. Each Re-timer continues with Symbol lock.

A.4 Error in CL2_ACK Ordered Sets

Figure A-4 shows an example of an attempt by Router A to enter CL2 state, where the CL2_ACK Ordered Sets are received with errors by Router A. In the example, Re-timer A is adjacent to Router A and Re-timer B is adjacent to Router B. Router A initiates entry to CL2.

Figure A-4. Error in CL2_ACK Ordered Sets



The following steps take place in Figure A-4:

1. Re-timer B detects 3 back-to-back CL2_ACK Ordered Sets and starts a CL2_ACK Counter.
 - Re-timer A detects Link errors coming from Router B. It disables RS-FEC in the direction of the errors to allow detection of SLOS.
2. Re-timer A detects 15 SLOS Symbols coming from Router A. It disables RS-FEC in both directions.
3. Re-timer B detects 15 SLOS Symbols. It disables RS-FEC in both directions. It also clears the CL2_ACK Counter to zero.

A.5 Error in CL_OFF Ordered Sets

Figure A-5 shows an example of an attempt by Router A to enter CL2 state, where the CL_OFF Ordered Sets are received with errors by Router B. In the example, Re-timer A is adjacent to Router A and Re-timer B is adjacent to Router B. Router A initiates entry to CL2.

[illegible]

1. Re-timer B detects 3 back-to-back CL2_ACK Ordered Sets and starts a CL2_ACK Counter.
2. Re-timer A detects 3 back-to-back CL2_ACK Ordered Sets and starts a CL2_ACK Counter.
3. Re-timer A detects 3 back-to-back CL_OFF Ordered Sets and starts a CL_OFF Counter.
4. Re-timer B detects Link errors. It disables RS-FEC in the direction of the errors to allow detection of SLOS.
5. Re-timer B's CL2_ACK Counter reaches a count of 372. Re-timer B then powers down the receivers and the transmitters in both directions.
6. Re-timer A's CL2_ACK Counter reaches a count of 372. Re-timer A then powers down the transmitter facing Router A and its Corresponding Receiver.
7. Re-timer A's CL_OFF Counter reaches a count of 372. Re-timer A then powers down the receiver facing Router A and its Corresponding Transmitter.