SIEMENS

Logical Layer Architecture – Design Team

Submitted by:

Ahmed Zakaria Mohamed Elsayed

Ahmed Tarek Shafik Mohamed

Seif Hamdy Hanafy Fadda

Hager Walid Ali Abdelghafar

Submitted to:

Eng. Mohamed Samy

Eng. Ahmed Mohamed

Eng. Sarah Mostafa

Eng. Mohamed Rayan

Eng. Randa Aboudief

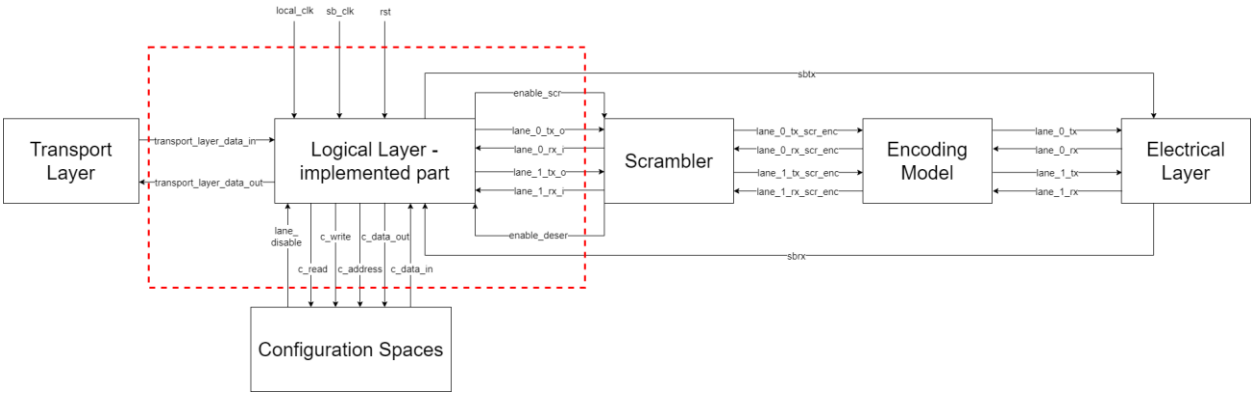
Contents

System Description

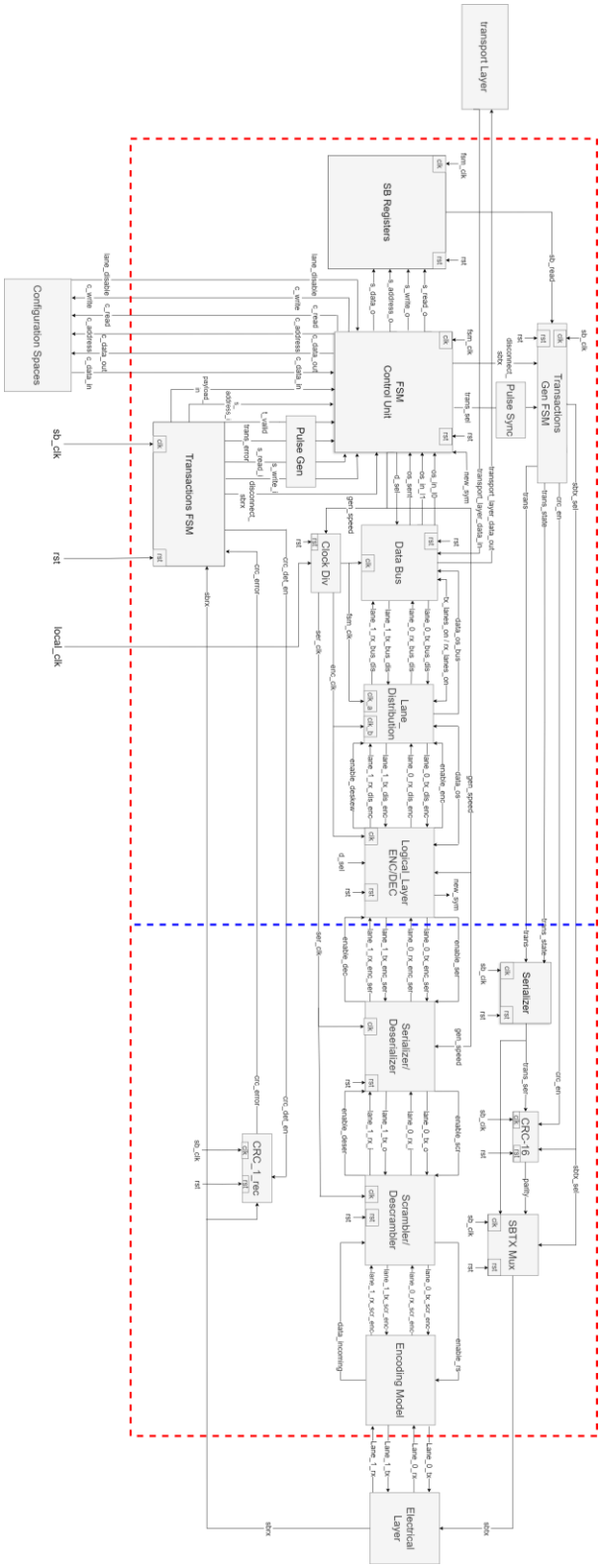
The USB 4 logical layer initiates lane initialization by entering the CLd state in the control unit and commencing phase 1, which involves determining the properties of the cables such as USB 4 support and generation 4 support. The control finite state machine (FSM) acquires this information by reading the configuration spaces. Following this, phase 2 begins, which involves detecting high sbrx, indicating the presence of another device on the link. In phase 3, the connected lane adapter exchanges the required parameters. This parameter exchange occurs through the transmission of AT transactions to the connected lane adapter, which responds with the necessary parameters.

Afterwards, the control unit transitions to the training state, where specific ordered sets are sent through the lane to achieve synchronization. Finally, the control unit enters the CL0 state, allowing the passage of transport layer packets.

High Level Block Diagram of Logical Layer



Logical Layer Architecture



- The architecture is divided to a parallel part (on the left side) and serial part (on the right side).
- All blocks have rst signal connected to.
- Sent Data in the lanes should be verified before entering Encoding Model block. These data move serially.

Features and Limitations

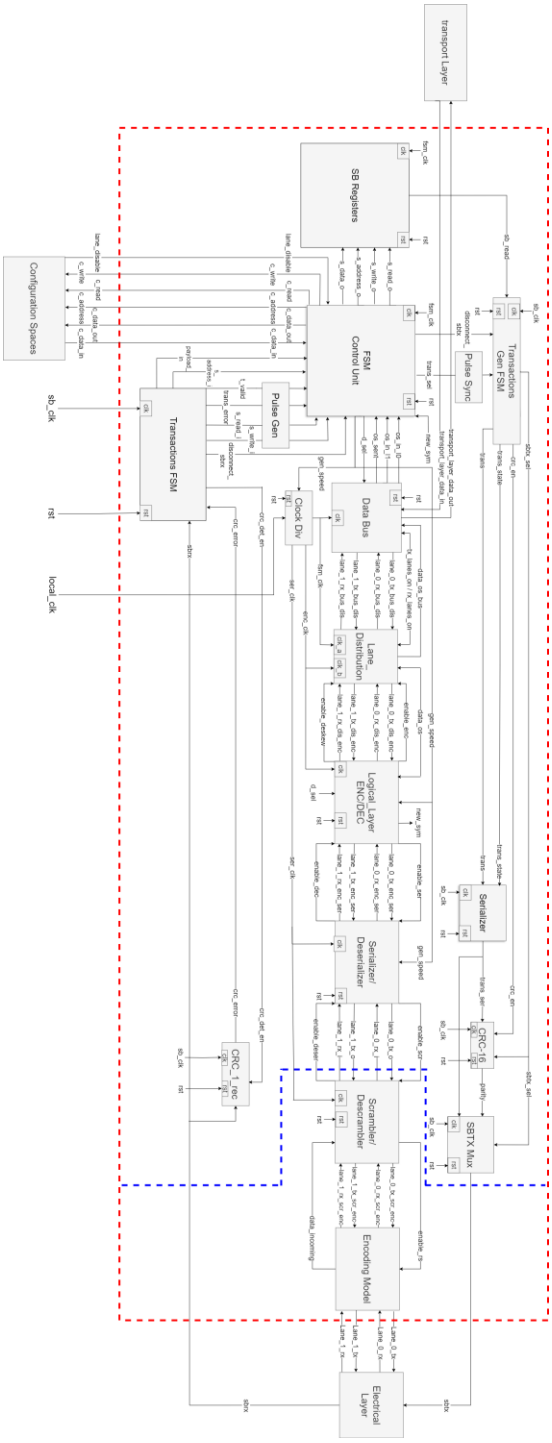
Features:

1) Detecting Presence on Another Lane Adapter:
   - The logical layer can detect the presence of another lane adapter on the link, enabling dynamic link configuration.
2) Sending Transactions to Exchange Parameters:
   - It can send transactions to exchange parameters with the other lane adapter, facilitating link negotiation and configuration.
3) Update Sideband Registers and Read from Them:
   - The logical layer can update sideband registers and read from them, allowing access to lane adapter parameters for configuration and monitoring.
4) Read Cable Properties from Configuration Unit:
   - It can read cable properties from the configuration unit, enhancing link management and cable property awareness.
5) Send SLOS and TS Ordered Sets for Synchronization:
   - The logical layer can send SLOS (Symbol Lock Ordered Sets) and TS (Training Sequence) ordered sets for synchronization during link initialization and training.
6) Clock Recovery with Receiver Clock:
   - It can perform clock recovery by replacing the local clock with the receiver clock, ensuring proper synchronization.
7) Lane Data Encoding/Decoding
   - Introducing 64-66, 128-132, and 8-11 encoding/decoding options for improved transport layer data communication in lanes.

Limitations:

1) Limitation in Logical Layer Encoding:
   - The subsequent logical layer encoding steps, including RS-FEC encoding, binary to ternary conversion, scrambling, and precoding insertion are skipped and assumed to be a separate block to be designed on its own.
2) No Compliance Testing:
   - There are no built-in compliance testing features within the logical layer.
3) No Low Power States:
   - The logical layer does not support low power states, meaning it operates at full power without power-saving features.
4) No Re-timers
   - The connecting cable is assumed to be always passive (have no re-timers)

Architecture to be implemented



The planned architecture, ending with the blue border, will encompass the entire logical layer, excluding the Encoding Model block that does RS-FEC encoding, binary to ternary conversion, scrambling, and precoding insertion. As a result, we will have a serial interface with lanes for both input and output data.

Blocks Description

Block	Description
SB Registers	- A set of registers that are used to link setup and configuration over the side band channel. - AT transactions and CM have access to SB registers
Configuration spaces	- A configuration set of registers - Accessed by the CM or configuration layer packets
CRC_16	- DSP block to check if an error occurred on the transmitted data
CRC_16_rec	- DSP block to check if an error occurred on the received data and error flag is assigned if syndrome not equal zero and is forwarded to transactions FSM
FSM Control unit	- It oversees the exchange of parameters during the CLd state using transactions. It also handles the transmission of the appropriate ordered set in the lanes while ensuring the correct state is maintained. Once the link is trained successfully, the control unit enables the transmission of data packets.
Timer	- sends flags with timeouts necessary for different blocks
FSM Transactions	- Identifies the transaction type, determining the address of the SB registers it intends to read or write to, and forwarding the payload of the transaction to the FSM control unit.

Data Bus	- Depends on its **d_select** signal, the data bus does one of the following: I) Generate lane parameters Sync order sets. II) Forward transport layer packets - In the data bus a PRBS circuit is included and added to the transmitted order sets. - The data bus outputs an **os_in** signal to the control unit to indicate the type of the ordered set received. - The data bus outputs an **os_sent** signal to the control unit to indicate that the data bus has sent an ordered set successfully.
Clock Div	- Sets the clock frequency of some blocks as shown in the architecture depending on the gen speed used.
Lane Distribution	- Distributing data from data bus to be transmitted on 2 lanes and de-skewing the received data from 2 lanes and returned on a single lane to data bus.
Logical Layer Encoding	- applies a 64/66, 128/132, or 8/11 encoding scheme depending on the gen speed used
Scrambler_descrambler	- scrambling of serial input transmitted data and descrambling of serial input received data on the lanes.
Encoding Model	- does RS-FEC encoding, binary to ternary conversion, and precoding insertion. (Not Implemented)

Signals Description

Logical Layer Interface

Signal	Direction	Description
local_clk	Input	Local clock
sb_clk	Input	Side bands clock
rst	Input	Reset
lane_disable	Input	If lane_disable = 1, the fsm should disable.
sbrx	Input	Serial input from sbrx.
enable_deser	Input	Enable deserializer when data is incoming
c_data_in [31:0]	Input	Data to be read from config spaces.
transport_layer_data_in [7:0]	Input	Data to be sent, coming from transport layer.
lane_0_rx_i	Input	Serial data input to lane 0.
lane_1_rx_i	Input	Serial data input to lane 1.
transport_layer_data_out [7:0]	Output	Data received and forwarded to transport layer.

sbtx	Output	transactions serial output through sideband channels.
lane_0_tx_o	Output	Serial data output to lane 0.
lane_1_tx_o	Output	Serial data output to lane 1.
c_read	Output	To control the read operation from config spaces.
c_write	Output	To control the write operation from config spaces.
c_address [7:0]	Output	Input address of read/write operation in config spaces.
c_data_out [31:0]	Output	Data to be written in config spaces.
enable_scr	Output	Enable signal from serializer to scrambler

FSM Control Unit

Signal	Direction	Description
Fsm_clk	Input	Fsm clock from clock div
Rst	Input	Reset
c_address [7:0]	Output	Input address of read/write operation in config spaces.

c_read	Output	To control the read operation from config spaces.
c_write	output	To control write operation from config spaces.
c_data_in [31:0]	Input	Data to be write from/in config spaces.
d_sel [3:0]	Output	Selection of Ordered Sets / Data (coming from transport layer) to be sent to USB4 link
os_in_10 [3:0]	Input	received from data bus indicating which os received on lane 0
os_in_11 [3:0]	input	received from data bus indicating which os received on lane 1
payload_in [23:0]	Input	Payload received in the sideband transaction (Response/Data to SB registers)
s_read_o	Output	To control the read operation from SB Registers
s_write_o	Output	To control the Write operation from SB Registers
s_address_o[7:0]	Output	Input address of read/write operation of SB Registers
s_data_o [7:0]	Output	Data to be written in SB registers.
gen_speed [1:0]	Output	Gen speed used
trans_sel [2:0]	Output	Select the type of transaction to be sent by the trans_gen fsm
disconnect	Input	If asserted, disconnect the system

lane_disable	Input	If lane_disable = 1, the fsm should disable
t_valid_pul	Input	pulse received from transactions fsm indicating transaction received
s_write_pul	Input	If asserted, the fsm should write in the sb registers
s_read_pul	Input	If asserted, the fsm should read from the sb registers
s_address_i [7:0]	Input	Address to write or read from the sb registers
trans_error_pul	Input	error in transaction received as pulse
os_sent	input	Pulse received from data bus indicating os is sent
Busy	input	shows if pulse synchronizer is busy
tdisabled_min	Input	Min time to stay in disabled state
ttraining_error_timeout	Input	Error signal when training is out of time
tgen4_ts1_timeout	Input	Error signal when ts1 is out of time
tgen4_ts2_timeout	Input	Error signal when ts2 is out of time
disconnect_sbtx	Output	Sent to trans_gen fsm to send zeros in sbtx to complete disconnection
fsm_disabled	Output	indicating fsm is disabled
fsm_training	Output	indicating fsm is in training state

ts1_gen4_s	Output	indicating fsm is sending gen4 ts1 os
ts2_gen4_s	Output	indicating fsm is sending gen4 ts2 os
c_data_out [31:0]	output	Data to be read from/in config spaces.
trans_sent	input	Transactions is sent
new_sym_pul	input	Pulse indicating start of a new symbol for purpose of synchronization

Data Bus

Signal	Direction	Description
Fsm_clk	Input	Fsm clock from clock div
Rst	Input	Reset
d_sel [3:0]	Input	Selection of Ordered Sets / Data (coming from transport layer) to be sent to USB4link
transport_layer_data_in [7:0]	Input	Data to be sent, coming from transport layer.
transport_layer_data_out [7:0]	Output	Data received and forwarded to transport layer.
os_in_10 [3:0]	Output	received from data bus indicating which os received on lane 0
os_in_11 [3:0]	Output	received from data bus indicating which os received on lane 1
lane_0_tx_bus_dis [7:0]	Output	Data / ordered sets sent through lane 0

lane_1_tx_bus_dis [7:0]	Output	Data / ordered sets sent through lane 1
lane_0_rx_bus_dis [7:0]	Input	Data / ordered sets received through lane 0
lane_1_rx_bus_dis [7:0]	input	Data / ordered sets received through lane 1
data_os_bus	Input	Indication of receiving transport layer data when it is high.
os_sent	Output	Indication that an ordered set has transmitted successfully.
tx_lanes_on	Output	Signal to lane distributer indicating data will be sent on lanes
rx_lanes_on	Input	Signal from lane distributer indicating data will be received

SB Registers

Signal	Direction	Description
Fsm_clk	Input	Fsm clock from clock div
Rst	Input	Reset
s_read_o	Input	To control the read operation from SB Registers
s_write_o	Input	To control the Write operation from SB Registers
s_address_o [7:0]	Input	Input address of read/write operation of SB Registers

sb_read [23:0]	Output	Data read from SB Registers

Transactions FSM

Signal	Direction	Description
sb_clk	Input	Sideband clock
rst	Input	Reset
tdisconnect_rx_min	Input	Used to notify the FSM block that the router linked is disconnected (timeout)
tconnect_rx_min	Input	Used to notify the FSM block that the router linked is connected for min time
crc_error	Input	Indication of error in received transactions
sbrx_parallel [9:0]	Input	Received transactions
s_read_lvl	Output	To control the read operation from SB Registers
s_write_lvl	Output	To control the Write operation from SB Registers
payload_in [23:0]	Output	Payload received in the sideband transaction. (Response/Data to SB registers)
s_address_i [7:0]	Output	Input address of read/write operation of SB Registers
crc_det_en	Output	Control the operation of CRC block
trans_error_lvl	Output	Indication of error in symbols detected
disconnect	Output	Indication of disconnection through LT transaction
t_valid_lvl	Output	Raised when transaction received completely

Serializer (Sideband Channel)

Signal	Direction	Description
sb_clk	Input	Sideband clock
rst	Input	Reset
trans [9:0]	Input	Transaction to be sent before serializing
trans_state [1:0]	Input	Transactions current state
trans_ser	Output	Serialized transaction

Deserializer (Sideband Channel)

Signal	Direction	Description
sb_clk	Input	Sideband clock
rst	Input	Reset
sbrx	Input	Serial input from sbrx
Sbrx_parallel [9:0]	Output	Symbol output to transactions fsm

Transactions Gen FSM

Signal	Direction	Description
sb_clk	Input	Sideband clock
rst	Input	Reset
trans_sel_pul [2:0]	Input	Select the transaction to be sent
sb_read [23:0]	Input	Data read from SB Registers
Trans_sent_pul	Output	Indicate that transactions is sent
trans_sent	Output	indicates transaction sent
Pulse_det	Input	Detection to start symbol construction
disconnect	Input	Detect disconnection

trans [9:0]	Output	Transaction to be sent before serializing
crc_en	Output	Enable signal for crc block
Sbtx_sel	Output	Sel signal to output either parity bits or serialized bits from sbtx mux
disconnect_sbtx	Input	Sent to trans_gen fsm to send zeros in sbtx to complete disconnection
tdisconnect_tx_min	Input	Used to notify the FSM block that the router linked is disconnected (timeout)
disconnected_s	Output	fsm in disconnected state

Clock Div

Signal	Direction	Description
local_clk	Input	Local clock
Rst	Input	Reset
gen_speed [1:0]	Input	Gen speed used
fsm_clk	Output	Fsm clock
enc_clk	Output	Logical Layer encoding clock
ser_clk	Output	Serializer/deserializer clock

Logical Layer Encoding

Signal	Direction	Description
enc_clk	Input	encoder clock from clock div
rst	Input	Reset
Enable	Input	Enable encoding

gen_speed [1:0]	Input	Gen speed used
d_sel [3:0]	Input	Used to know whether it's a data or order set packet
lane_0_tx [7:0]	Input	Data / ordered sets sent through lane 0
lane_1_tx [7:0]	Input	Data / ordered sets sent through lane 1
enable_ser	Output	Enable serializer
lane_0_tx_enc_old [131:0]	Output	Lane 0 tx encoded
lane_1_tx_enc_old [131:0]	Output	Lane 1 tx encoded
new_sym	Output	New symbol is sent

Logical Layer decoding

Signal	Direction	Description
enc_clk	Input	encoder clock from clock div
rst	Input	Reset
Enable_dec	Input	Enable decoder
gen_speed [1:0]	Input	Gen speed used
d_sel[3:0]	input	Used to know whether it's a data or order set packet
Valid_data	Input	Notify the block that data is ready for decoding
lane_0_rx_enc_ser [131:0]	Input	Lane 0 rx encoded
lane_1_rx_enc_ser [131:0]	Input	Lane 1 rx encoded
lane_0_rx_dis_enc [7:0]	Output	Data / ordered sets received through lane 0
lane_1_rx_dis_enc [7:0]	Output	Data / ordered sets received through lane 1

data_os	Output	Used to know whether it's a data or order set packet
enable_deskew	Output	Signal from decoder indicating data is being received on lanes(enable for receiving side)

Lanes Serializer/Deserializer

Signal	Direction	Description
ser_clk	Input	Ser/deser clock from clock div
rst	Input	Reset
enable_ser	Input	Enable signal from encoder to serializer
enable_deser	Input	Enable signal from descrambler to deserializer
gen_speed[1:0]	Input	Indicate gens GEN2,GEN3,GEN4
lane_0_tx_enc_ser [131:0]	Input	Lane 0 tx from encode
lane_0_rx_enc_ser [131:0]	Output	Lane 0 rx to decoder
lane_1_tx_enc_ser [131:0]	Input	Lane 1 tx from encode
lane_1_rx_enc_ser [131:0]	Output	Lane 1 rx to decoder
lane_0_tx_ser_scr	Output	Serial data to scrambler
lane_0_rx_ser_scr	Input	Serial data from scrambler
lane_1_tx_ser_scr	Output	Serial data to scrambler
lane_1_rx_ser_scr	Input	Serial data from scrambler
scr_rst	Output	Reset seed of scrambler
descr_rst	Output	Reset seed of descrambler
enable_scr	Output	Enable signal from serializer to scrambler

enable_dec	Output	Enable signal from deserializer to decoder

CRC_16

Signal	Direction	Description
sb_clk	Input	Side band clock
rst	Input	Reset
trans_ser	Input	Serialized transaction
crc_en	Input	Enable signal for crc from transaction gen FSM
sbtx_sel	Input	Sel signal to output either parity bits or serialized bits from sbtx mux
parity	Output	16 bit parity data

CRC_16_rec

Signal	Direction	Description
sb_clk	Input	Side band clock
rst	Input	Reset
sbrx	Input	Received transactions
crc_det_en	Input	Enable signal for crc from transaction FSM
crc_error	Ouput	Error signal when syndrome not equal zero detectring sbrx not equal sbtx

Scrambler_descrambler

Signal	Direction	Description
Ser_clk	Input	Serializer clock
rst	Input	Reset
data_incoming	Input	Notifying descrambler that input is incoming in lanes
enable_scr	Input	Scrambler enable signal
descr_rst	Input	signal for descrambler to return to seed value
scr_rst	Input	signal for scrambler to return to seed value
lane_0_tx_ser_scr	Input	Tx serial data in lane 0 coming from serializer to be scrambled
lane_1_tx_ser_scr	Input	Tx serial data in lane 1 coming from serializer to be scrambled
lane_0_tx_o	output	Tx serial data in lane 0 after scrambling
lane_1_tx_o	output	Tx serial data in lane 1 after scrambling
lane_0_rx_i	Input	Rx serial data in lane 0 to be descrambled
lane_1_rx_i	Input	Rx serial data through in 1 to be descrambled
lane_0_rx_ser_scr	output	Rx serial data in lane 0 output to deserializing after descrambling
lane_1_rx_ser_scr	output	Rx serial data in lane 1 output to deserializing after descrambling
enable_deser	Output	enable signal to deserializer

enable_rs	Output	Enable signal to read_solomon encoding

Lane Distribution

Signal	Direction	Description
fsm_clk	Input	Fsm clock
rst	Input	Reset
tx_lanes_on	Input	Signal from data bus indicating data will be sent on lanes(enable for transmitting side)
enable_deskew	Input	Signal from decoder indicating data is being received on lanes(enable for receiving side)
data_os	Input	Used to know whether it's a data or ordered set packet
data_os_bus	Output	Indication of receiving transport layer data when it is high.
d_sel [3:0]	Input	Used to know whether it's a data or order set packet
lane_0_tx_bus_dis [7:0]	Input	Single lane carrying Tx data from data bus on lane 0
lane_1_tx_bus_dis [7:0]	Input	Single lane carrying Tx data from data bus on lane 1
lane_0_rx_bus_dis [7:0]	Output	Outputting received data on a single lane to data bus after de-skewing lane 0
lane_1_rx_bus_dis [7:0]	Output	Outputting received data on a single lane to data bus after de-skewing lane 1

lane_0_rx_dis_enc [7:0]	Input	Rx data on lane 0 before de-skewing
lane_1_rx_dis_enc [7:0]	Input	Rx data on lane 0 before de-skewing
lane_0_tx_dis_enc [7:0]	Output	Tx data on lane 0 after Distribution
lane_1_tx_dis_enc [7:0]	Output	Tx data on lane 1 after Distribution
enable_enc	Output	Signal to encoder indicating tx data will be sent to it.
rx_lanes_on	Output	Signal to data bus indicating rx data will be sent to it.

Timer

Signal	Direction	Description
Sb_clk	Input	sideband clock
ms_clk	Input	Slower clock
rst	Input	Reset
disconnected_s	Input	Disconnect state
fsm_disabled	Input	Indicating if fsm is disabled
fsm_training	Input	Indicating if fsm is in training state
ts1_gen4_s	Input	indicating fsm is sending gen4 ts1 os
ts2_gen4_s	Input	indicating fsm is sending gen4 ts2 os
sbrx	Input	Received transactions
tdisconnect_tx_min	Output	Min time for transmitted data to disconnect

tdisconnect_rx_min	Output	Min time for received data to disconnect
tconnect_rx_min	Output	Min time for transmitted data to connect
tdisabled_min	Output	Min time to enter disable state
ttraining_error_timeout	Output	Error signal indicating timeout during training state
tgen4_ts1_timeout	Output	Signal indicating ts1 gen 4 time out
tgen4_ts2_timeout	Output	Signal indicating ts2 gen 4 time out

pulse_sync_3bit

Signal	Direction	Description
Sb_clk	Input	slower clock
fsm_clk	Input	Fast clock
rst	Input	Reset
trans_sel [2:0]	Input	Pulse (fast clock domain)
trans_sel_pul [2:0]	Output	Pulse (slow clock domain)
Busy	Output	Busy signal

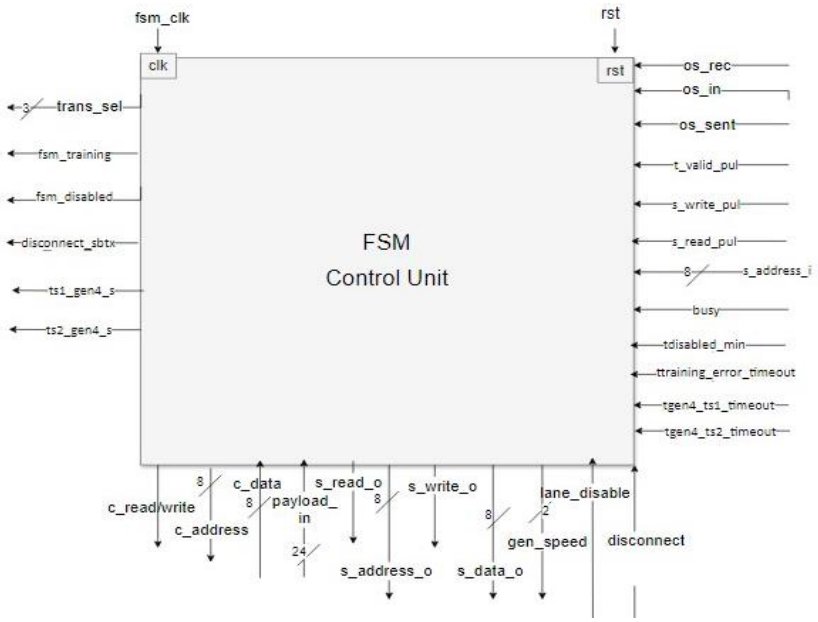
pulse_generator

Signal	Direction	Description
fsm_clk	Input	Fsm clock from clock div
rst	Input	Reset
s_read_lvl	Input	Level data to be read

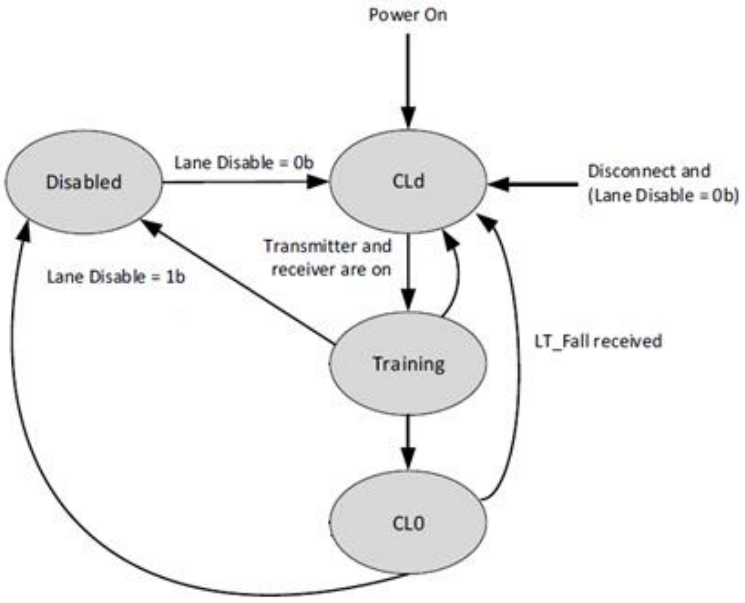
s_write_lvl	Input	Level data to be written
trans_error_lvl	Input	Level error
t_valid_lvl	Input	Valid signal as level
s_read_pul	Output	Data in form of pulse to be read
s_write_pul	Output	Data in form of pulse to be written
trans_error_pul	Output	Error in form of pulse
t_valid_pul	Output	Valid signal as pulse

Blocks Detailed

Control Unit FSM (Lane Adapter FSM)



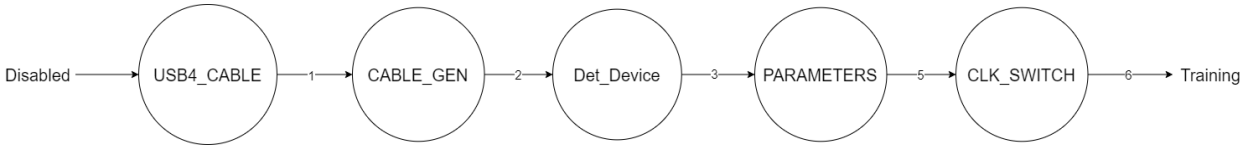
FSM Diagram



Transitions

Transition	From State	To State	Condition
1	Disabled	CLd	Lane_disable bit in lane adapter config spaces = 0
2	CLd	Training	Lane initialization done (CLd sub-state machine completed its cycle)
3	Training	CL0	Training done (gen4 training sub-state machine completed its cycle)
4	CL0	Disabled	Lane_disable bit in lane adapter config spaces = 1
5	CL0	CLd	LT_Fall transaction received
6	Training	Disabled	Lane_disable bit in lane adapter config spaces = 1
7	Training	CLd	LT_Fall transaction received or failed to train before timeout

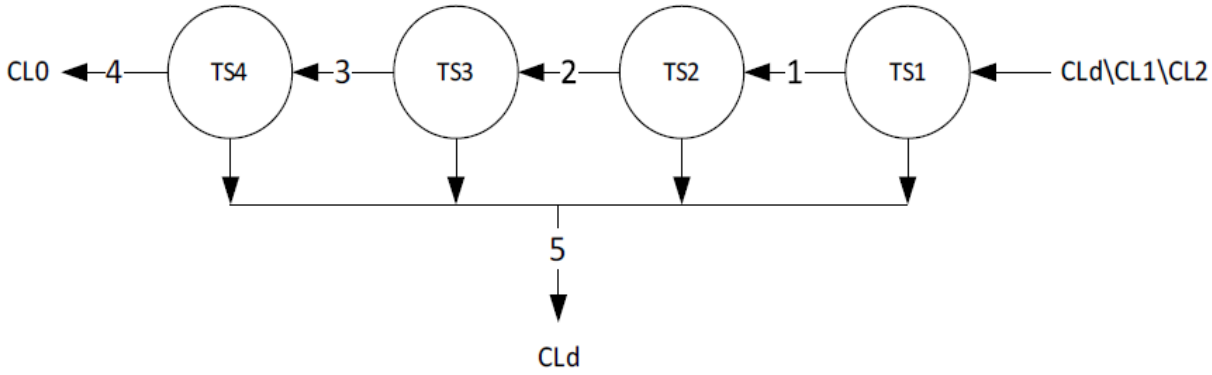
CLd Sub-State machine



Transitions

Transition	From State	To State	Condition
1	Usb4_cable	Gen4_cable	Cable supports USB4 from port configuration spaces.
2	Gen4_cable	Gen3_cable	Cable doesn't support Gen4 from port configuration spaces.
3	Gen3_cable	Detecting_device	Cable supports Gen3 from port configuration spaces.
4	Gen4_cable	Detecting_device	Cable supports Gen4 from port configuration spaces.
5	Detecting_device	Check_properties	SBRX goes high
6	Check_properties	Training	Checked the other lane adapter for which gen it supports and checking lanes and RS_FEC are enabled

Gen 4 Training Sub-State Machine



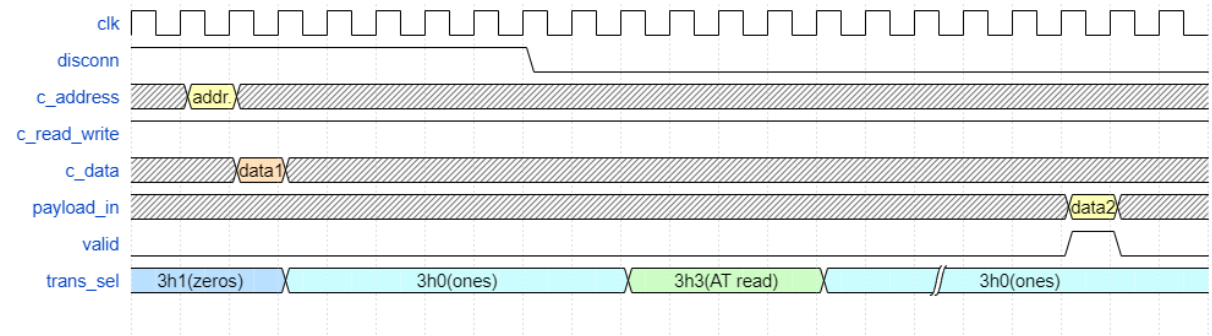
Transitions

Table 4-34. Gen 4 Training Sub-State Machine Transitions

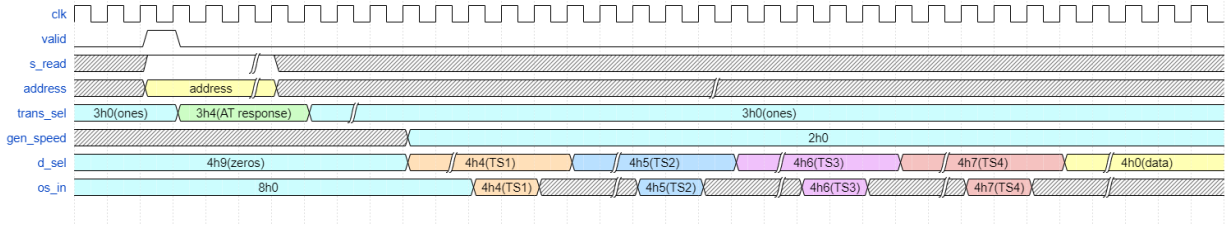
Transition	From State	To State	Conditions[1]
1	TS1	TS2	• If not in RX Compliance mode, received Gen 4 TS1 with the *Indication* field set to 2h or Gen 4 TS2 on all enabled receivers. • If in RX Compliance mode, received SET_RX_COMPLAINCE Port Operation with the *Transmitter State* field set to 010b. • Sent at least 16 Gen 4 TS1 with the *Indication* field set to 2h on all enabled transmitters. *Note: This means the receiver completed TxFFE negotiation in PAM2 and is ready for PAM3.*
2	TS2	TS3	• If not in RX Compliance mode: ○ All enabled Receivers completed TxFFE negotiation. ○ Received Gen 4 TS2 with the *Indication* field set to 4h or Gen 4 TS3 on all enabled receivers. • If in RX Compliance mode, the receiver under test completed TxFFE negotiation. • Sent at least 16 Gen 4 TS2 with the *Indication* field set to 4h on all enabled transmitters.
3	TS3	TS4	• If not in RX Compliance mode and not in TX Compliance mode, received Gen 4 TS3 with the *Indication* field set to 6h or Gen 4 TS4 on all enabled receivers. • Sent at least 16 Gen 4 TS3 with the *Indication* field set to 6h on all enabled transmitters.
4	TS4	CL0	• Sent Gen 4 TS4 with the *Counter* field set to Fh.
5	Any	CLd	• Adapter remains in Training state for tTrainingAbort1 or tTrainingAbort2 time.

Notes:
1. All conditions need to be met before a transition takes place.

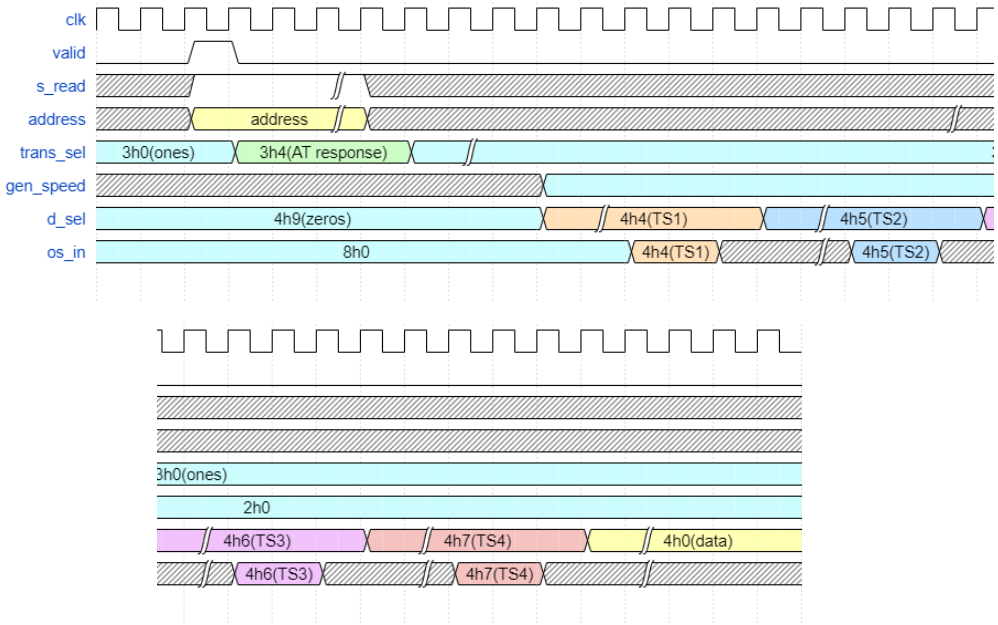
Timing Diagram of Lane initialization phases 1 to 3



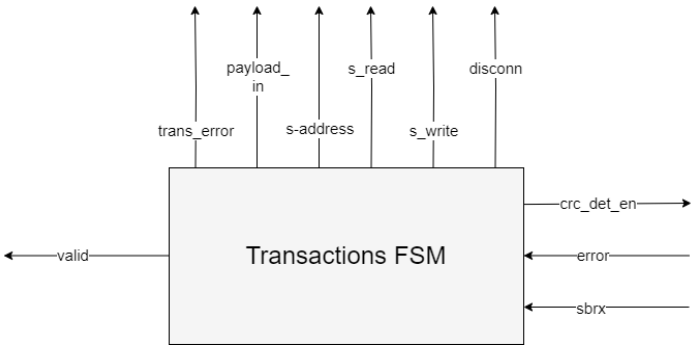
Timing Diagram of Lane initialization phase 3 and Training



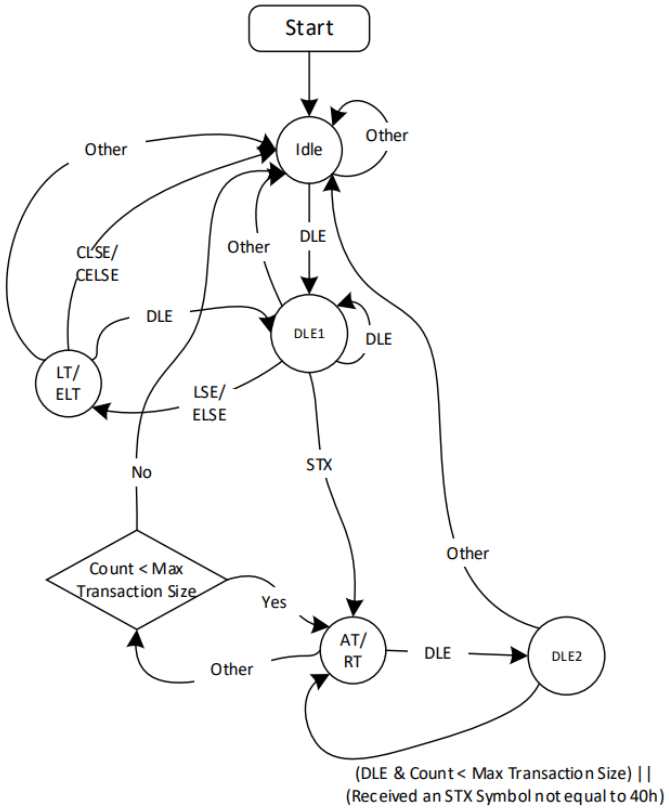
Waveform zoomed in



Transactions FSM



FSM Diagram



- If 8'hFE (DLE symbol) is received → crc_det_en = 1 (DLE symbol is sent at the start of any AT transaction)
- If another DLE symbol received → crc_det_en = 0 (another DLE symbol is sent at the end of any AT transaction and is not included in the crc calculation)
- If sbrx is high → diconn = 0

- If sbrx is low for certain timeout or LT_fall transaction received → diconn = 1
- If AT transaction received → send the data to the fsm and the address to be write or read from
- If error = 1 for AT transactions or CLSE symbol is not complement to LSE symbol for LT transactions → trans_error = 1

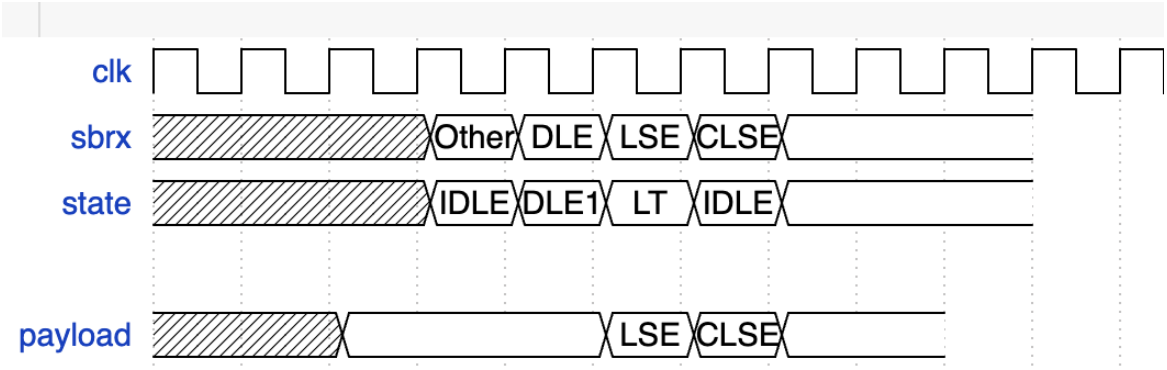
Transitions

Initial State	Next State	Condition	Action
Idle	DLE1	Received a DLE Symbol	--
Idle	Idle	Received other Symbol	--
DLE1	LT/ELT	Received LSE/ELSE Symbol.	Store LSE/ELSE Symbol.
DLE1	DLE1	Received a DLE Symbol.	--
DLE1	AT/RT	Received an STX Symbol.	Initialize *Count* to 1. Store STX Symbol. Initialize CRC machine and update with STX Symbol.
DLE1	Idle	Received other Symbol.	--
LT/ELT	DLE1	Received a DLE Symbol.	Discard stored LSE/ELSE Symbol.
LT/ELT	Idle	Received CLSE/CELSE Symbol.	Process LT or ELT Transaction. See Section 4.1.1.2.1.
LT/ELT	Idle	Received other Symbol.	Discard stored LSE/ELSE Symbol.
AT/RT	DLE2	Received a DLE Symbol.	--
AT/RT	AT/RT	Received other Symbol AND Count < Max Transaction Size. See Note 1.	Store Symbol. Update CRC. Increment Count.
AT/RT	Idle	Received other Symbol AND Count >= Max Transaction Size. See Note 1.	Discard stored Symbols.
DLE2	AT/RT	Received DLE Symbol AND Count < Max Transaction Size. See Note 1.	Store Symbol. Update CRC. Increment Count.
DLE2	AT/RT	Received an STX Symbol not equal to 40h.	Initialize *Count* to 1. Store STX Symbol. Initialize CRC machine and update with STX Symbol.
DLE2	Idle	Received an ETX Symbol.	Process AT Transaction or RT Transaction.
DLE2	Idle	Received other Symbol.	Discard stored symbols.

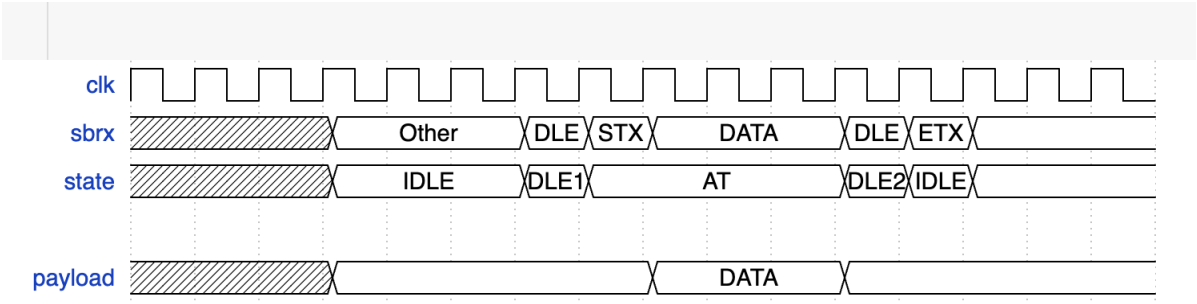
Notes:

1. Max Transaction Size is 69 Symbols for an AT Transaction or Addressed RT Transaction or 5 Symbols for a Broadcast RT Transaction.

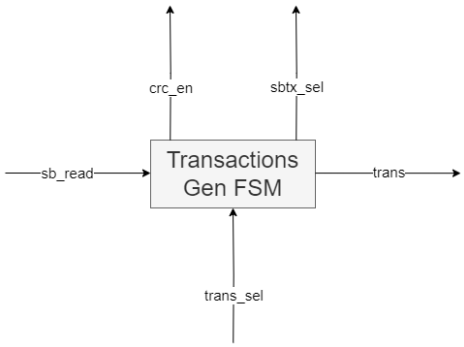
Timing Diagram of LT Transaction



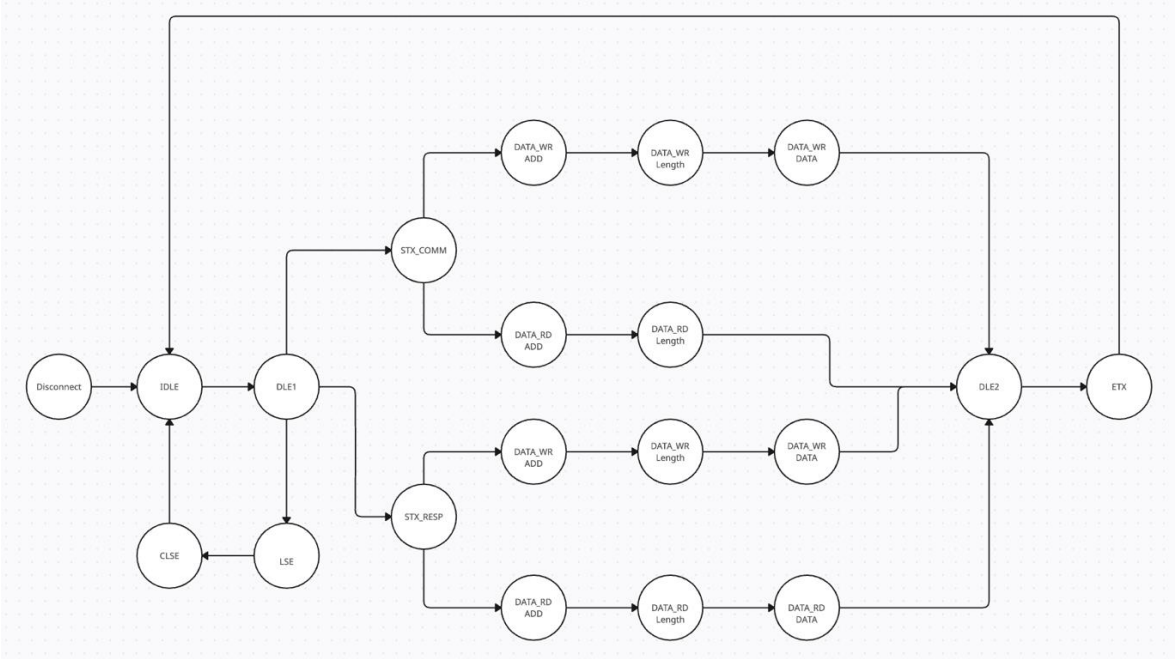
Timing Diagram of AT



Transactions Gen FSM



FSM Diagram



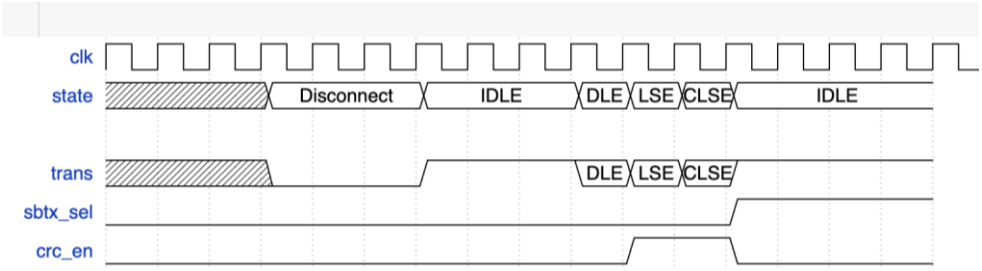
- During each state, a 1-byte symbol is being sent
- If trans_sel = 3'h0 → remain in IDLE state → trans = ones (if no transaction being sent, stream of ones should be sent in sbtx)
- If trans_sel = 3'h1 → go to disconnect state → trans = zeros (if lane is disconnected stream of zeros should be sent in sbtx)
- If trans_sel = 3'h2 → start AT transaction flow to obtain parameters of opposite lane adapter
- If trans_sel = 3'h3 → start AT transaction flow to send parameters to the opposite lane adapter
- If trans_sel = 3'h4 → start LT_Fall transaction flow to disconnect

Transitions

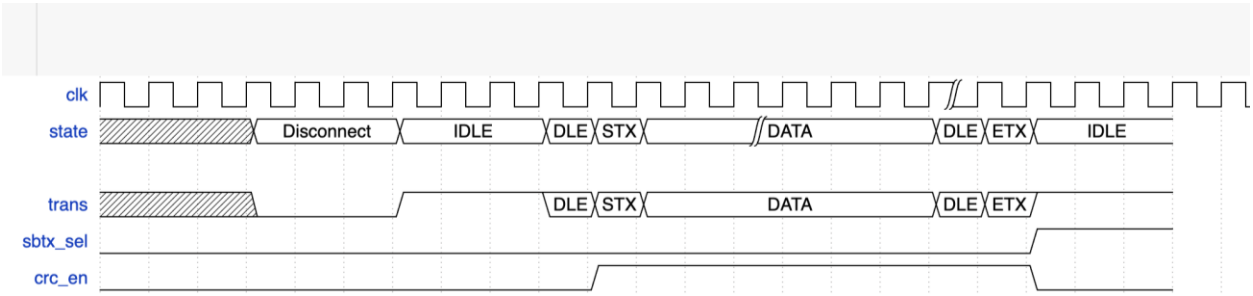
Transition	From State	To State	Condition
1	IDLE	IDLE	trans_sel = 0
2	Disconnect	Disconnect	trans_sel = 1
3	IDLE	Disconnect	trans_sel = 1
4	Disconnect	IDLE	trans_sel = 0
5	IDLE	DLE	trans_sel = 2 or trans_sel = 3 or trans_sel = 4
6	DLE	STX	Completed sending of DLE symbol and (trans_sel = 2 or trans_sel = 3)
7	STX	Data_req	Completed sending of STX symbol and trans_sel = 2
8	STX	Data_send	Completed sending of STX symbol and trans_sel = 3
9	Data_req	DLE	Completed sending of Data symbol
10	Data_send	DLE	Completed sending of Data symbol
11	DLE	ETX	Completed sending of DLE symbol
12	ETX	IDLE	Completed sending of ETX symbol
13	DLE	LSE	Completed sending of DLE symbol and (trans_sel = 4)
14	LSE	CLSE	Completed sending of LSE symbol
15	CLSE	IDLE	Completed sending of CLSE symbol

- Trans_sel must be stable throughout the transaction.
- DLE Symbol is not included in CRC calculations.
- Sbtx_sel is de-asserted after the end of the transaction to add CRC bits

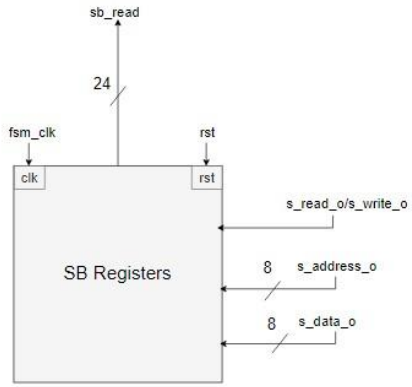
Timing Diagram of LT



Timing Diagram of AT command write / read response



SB Registers

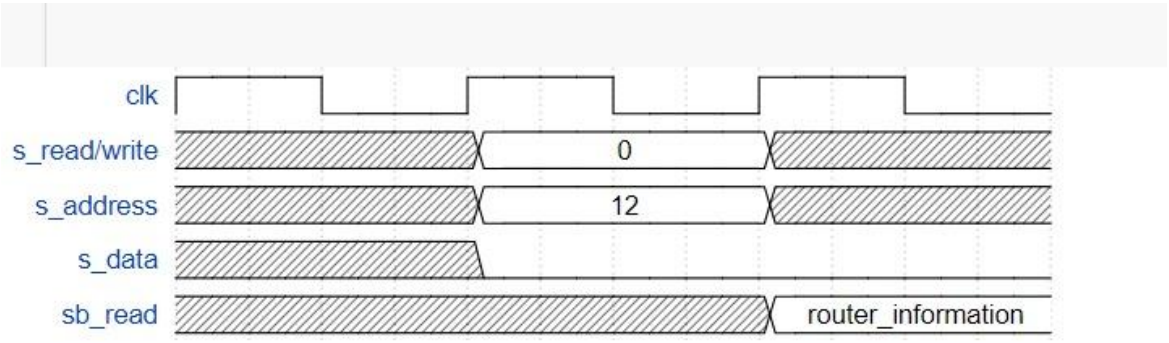


- 157 X 8 memory
- If s_read_write = 1 → read the memory location of address = s_address → output through sb_read
- If s_read_write = 0 → write s_data in the memory location of address = s_address
- REG12 is the most important register which is used during link Configuration and it takes memory addresses [80,79,78].

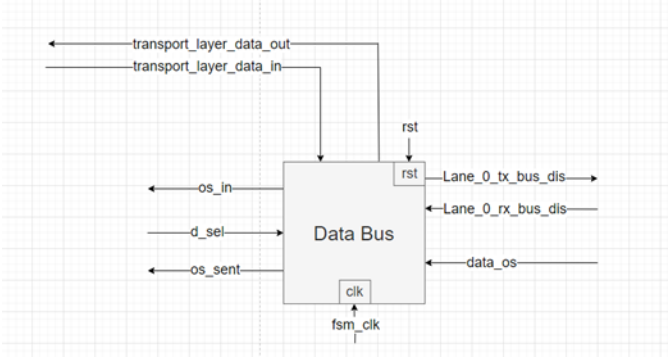
Register	Name	Size	Address
0	Vendor ID	4 Bytes	0
1	Product ID	4 Bytes	4
2-4	--	N/A	
5	Debug Capability Configuration	4 Bytes	8
6	Debug	54 Bytes	12
7	LRD Tuning	4 Bytes	66
8	Opcode	4 Bytes	70
9	Metadata	4 Bytes	74
10-11	--	N/A	
12	Link Configuration	3 Bytes	78
13	Gen 2/3 TxFFE	4 Bytes	81
14	Gen 4 TxFFE	4 Bytes	85
15	Sideband Channel Version	4 Bytes	89
16-17	--	N/A	
18	Data	64 Bytes	93
19-127	vendor specific	Vendor Specific	
128-255		N/A	

Timing Diagram of SB Registers

- If s_read_write = 0 → read the memory location of address = s_address → outputthrough sb_read



Data Bus

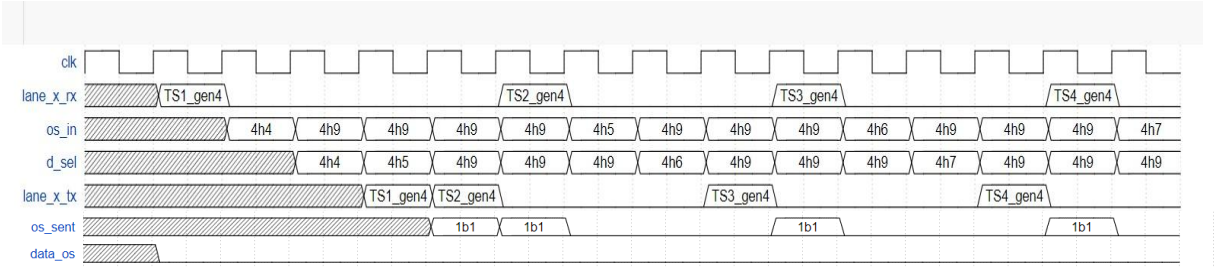


- If d_sel = 4'h0 → lane_0_tx_bus_dis = SLOS1
- If d_sel = 4'h1 → lane_0_tx_bus_dis = SLOS2
- If d_sel = 4'h2 → lane_0_tx_bus_dis = TS1_gen3
- If d_sel = 4'h3 → lane_0_tx_bus_dis = TS2_gen3
- If d_sel = 4'h4 → lane_0_tx_bus_dis = TS1_gen4
- If d_sel = 4'h5 → lane_0_tx_bus_dis = TS2_gen4
- If d_sel = 4'h6 → lane_0_tx_bus_dis = TS3_gen4
- If d_sel = 4'h7 → lane_0_tx_bus_dis = TS4_gen4
- If d_sel = 4'h8 → lane_0_tx_bus_dis = transport_layer_data
- If d_sel = 4'h9 → lane_0_tx_bus_dis = zeros

- Data received from lane_0_rx_bus_dis are forwarded to transport_layer_data when data_os signal is high 1'b1.
- If an ordered set is received on lane_0_rx _bus_dis → The ordered set type is sent through os_in signal to the control_fsm block.
- If an ordered set is transmitted through lane_1_tx_bus_dis → a high 1'b1 is sent through os_sent signal to the control_fsm block.

Timing Diagram of Data Bus

- Ts1 to Ts2 → receive ts1 (pam 3 ready) → send 16 ts1 (completed txffe for pam2 only)
- Ts2 to Ts3 → receive ts2 → send 16 ts2 (completed txffe for pam3)
- Ts3 to Ts4 → receive ts3 (on all enabled receivers) → send 16 ts3 (completed txffe for pam3)
- Ts4 to cl0 → sent ts4 with counter = Fh

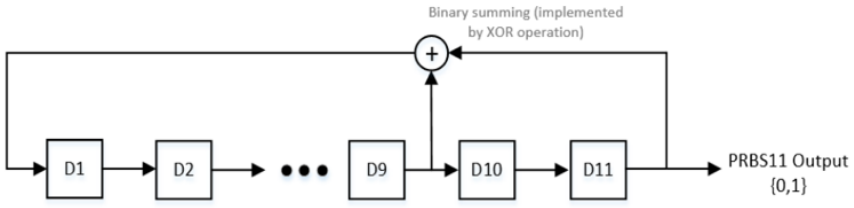


SLOS1 and SLOS2 (for gen 2&3)

PRBS11 polynomial: $G(x) = x^{11} + x^9 + 1$

PRBS11 initial state: 400h

Figure 4-41. PRBS11 Pattern Generator



The bit pattern for SLOS1 is composed of <0b, 2047 bits of PRBS11>. The bit pattern for SLOS2 is composed of <1b, 2047 bits of logically inverted PRBS11>.

TS1 and TS2 Ordered Sets (for gen 2&3)

A TS1 Ordered Set and a TS2 Ordered Set shall have the structure in Table 4-39, P. 165

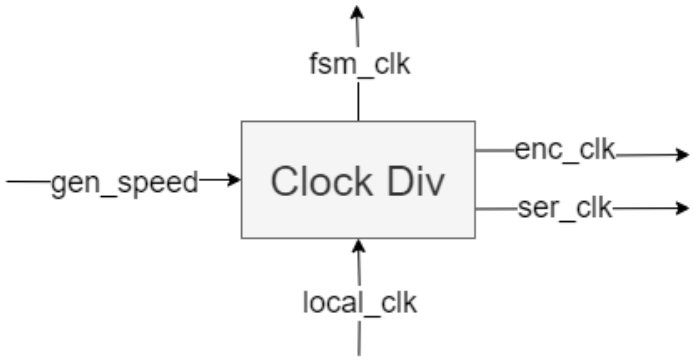
TS order sets (for gen4)

Table 4-40. Gen 4 Training Sequence

Trits	Name	Description
447:420	*Header*	Used to inform Re-timers/Router on the Link of the progress of the Lane Initialization.
419:0	*Pseudo Random Sequence*	**TS1** – PRBS11 **TS2, TS3, TS4** – PRTS7

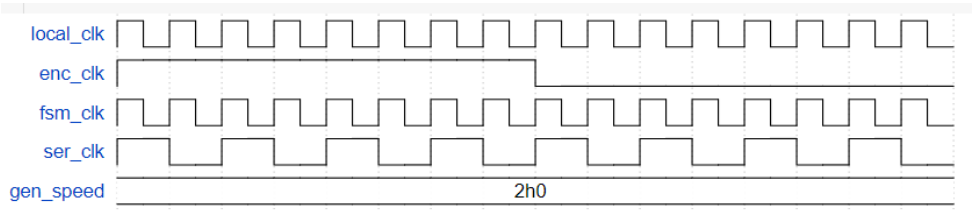
The header of every order set is described in standard page167 & page 168.

Clock Div

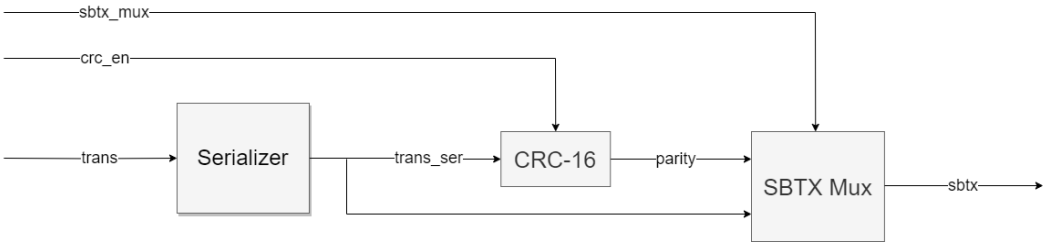


Timing Diagram of Clock Div

- For gen4 → gen_speed = 2'h0
- For gen3 → gen_speed = 2'h1
- For gen2 → gen_speed = 2'h2



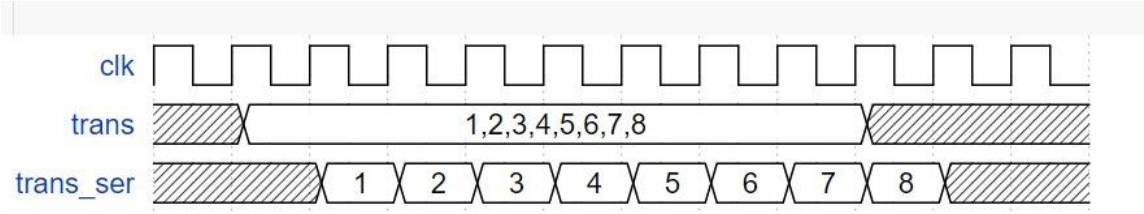
Serializer, CRC-16 and SBTX Mux



- The Serializer block serializes trans (parallel input) to trans_ser (serial output)
- CRC_16 generates 16 parity bits from the serial input trans_ser if crc_en = 1
- If sbtx_sel = 0 → SBTX Mux block passes the serialized data without parity bits
- If sbtx_sel = 1 → SBTX Mux block passes the parity bits

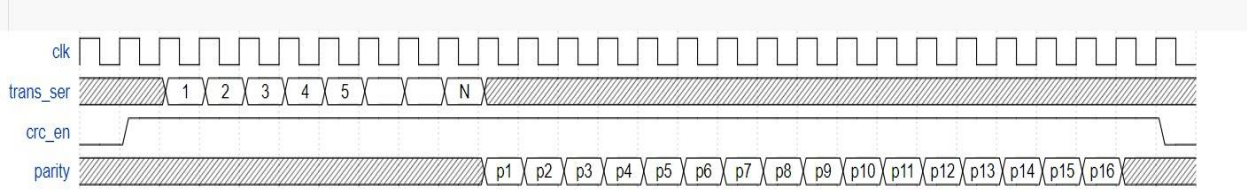
Timing Diagrams of Serializer

- The Serializer block serializes trans (parallel input) to trans_ser (serial output)



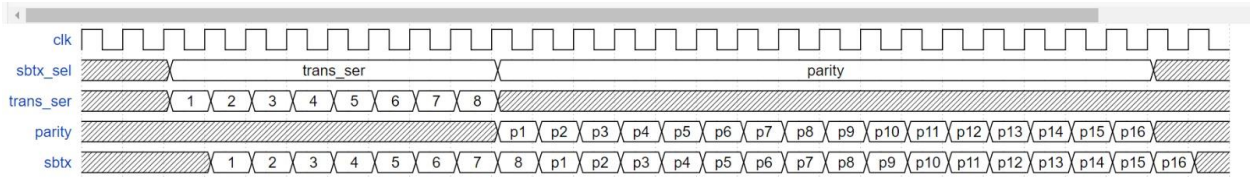
Timing Diagrams of CRC-16

- CRC_16 generates 16 parity bits from the serial input trans_ser if crc_en = 1

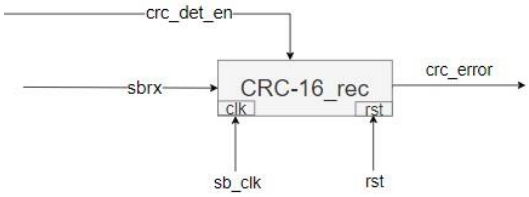


Timing Diagrams of SBTX Mux

- If sbtx_sel = 0 → SBTX Mux block passes the serialized data without parity bits
- If sbtx_sel = 1 → SBTX Mux block passes the parity bits

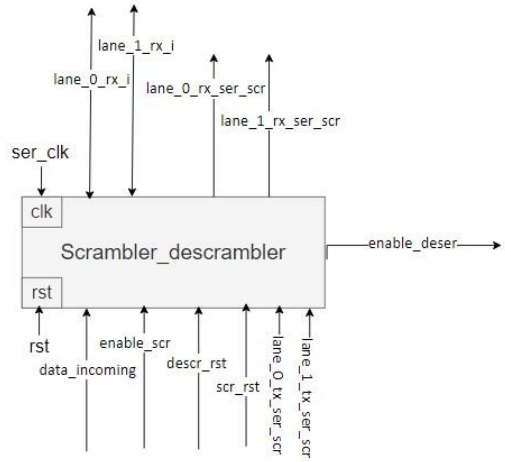


CRC_16_rec



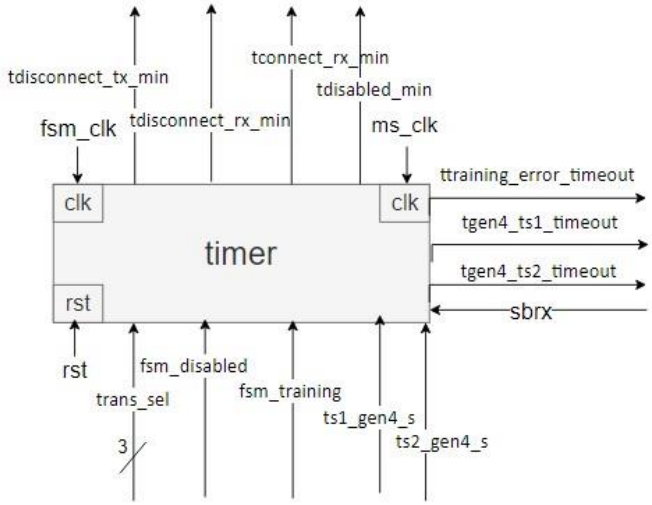
- Enable signal asserted by transactions fsm after receiving DLE symbol
- When enable signal asserted, he crc starts doing the XOR operations and updating lfsr
- if LFSR not equal zeros after enable signal de-asserted → error

Scrambler/Descrambler



Scrambling of Tx serial data and Descrambling of Rx serial data → seed = 1FEEDD

Timer

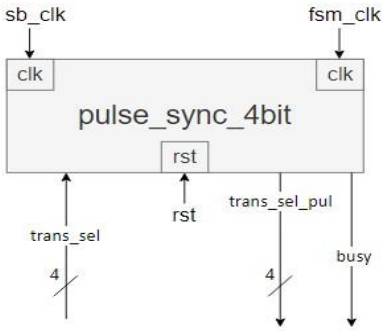


Outputs flags with required timeouts and minimum timings based on the following table:

Parameter	Description	Min	Max	Unit
tDisconnectTx	The time that SBTX is driven to logical low to identify a disconnect.	50		ms
tDisconnectRx	The time that SBRX is detected in logical low to identify a disconnect	14	1000	µs
tConnectRx	The time that SBRX is detected in logical high to identify a new connection.	25		µs
tCmdResponse	The time between receiving an AT Command and sending an AT Response or between receiving an Addressed RT Command and sending an RT Response.		50	ms
tDisabled	The minimal time an Adapter stays in a Disabled State after entering the state.	10		ms

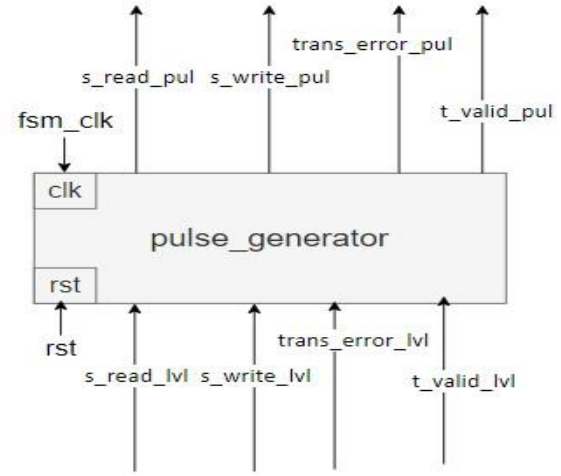
tTrainingError	The time an Adapter has to transition from Training state to CL0 state before a Timeout Error is generated. The time is measured from when the Adapter either sends the first SLOS1 (for Gen 2/3) or TS1 (for Gen 4) to when it transitions to CL0 state.		500	µs
tGen4TS1	The time to transition to the TS2 sub-state from the TS1 sub-state for a Gen 4 Link.		400	ms
tGen4TS2	The time to transition to the TS3 sub-state from the TS2 sub-state for a Gen 4 Link.		200	ms
tSSCActivated	Time after activating SSC that the transmitter sends TS4 with Counter field set to 0h.	2	(Transition to Training from CLd)	µs

Pulse Sync



Synchronizes a pulse from fast clock domain to slow clock domain

Pulse Generator



Converts level signals to pulses

Design Flow and Operation Paths

This section defines how a Sideband Channel is used to initialize the Lane(s) that make up a USB4 Link.

Lane Initialization is the process by which the Electrical Layer of a Lane goes from inactive to actively transmitting and receiving traffic.

The Sideband Channel shall initialize each Lane independently.

Lane Initialization is described from the point of view of a single Router and its Link Partner, which are called “Router A” and “Router B” respectively for distinction. Note that Lane Initialization is symmetrical, meaning that the steps defined for Router A are also followed by Router B in parallel (with Router A and Router B switching roles) for each phase of Lane Initialization.

Phase 1 - Initial Conditions

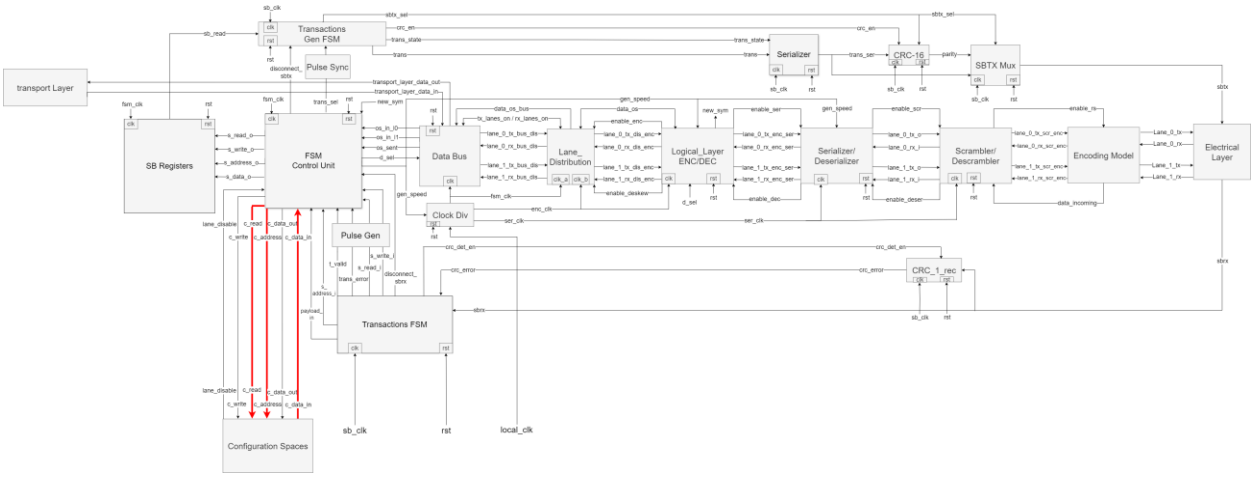
Cable properties

A USB4 Port shall obtain the connection information described in this section by the communication of the control unit with the configuration spaces.

See the USB PD Specification and the USB Type-C Specification for how to determine the connection information.

A USB4 Port shall not proceed to Phase 2 if the Link is not USB4.

A USB4 Port shall drive SBTX to logic low by default.



1. c_read = 1'h1 & c_address = 8'd18
2. c_data_in will be updated with the content of this location.
3. disconnect_sbtx = 1
4. trans_state = 2'h0 (Disconnect)
5. trans = 8'b00000000

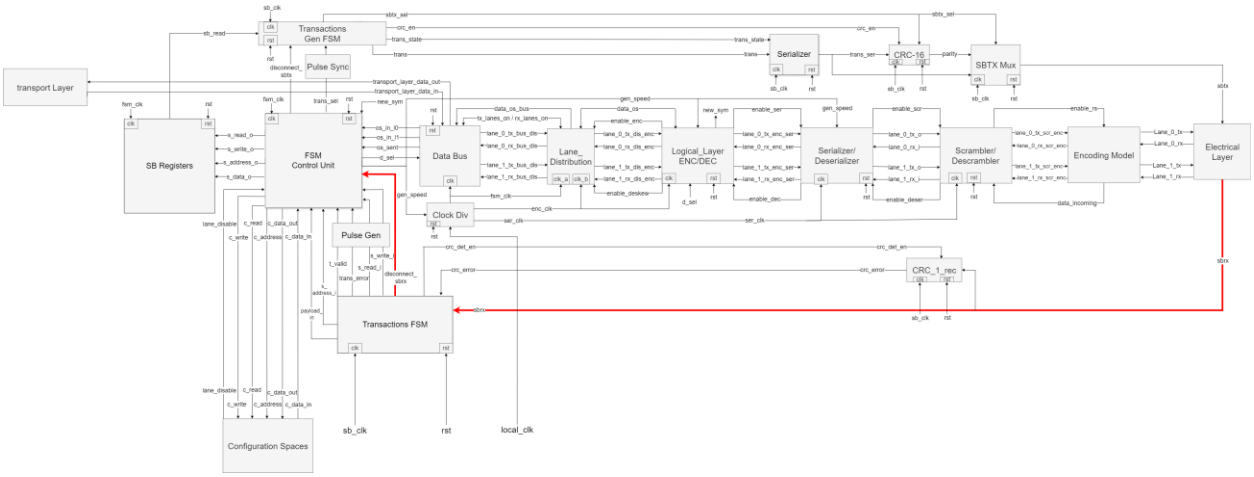
Phase 2 - Router Detection

Detect a connection

After completion of Phase 1, a Host Router shall initiate Router detection by driving SBTX to logic high on all of its Downstream Facing Ports.

When a Device Router detects a logic high on SBRX of its Upstream Facing Port for tConnectRx time, it shall drive SBTX to logic high on all of its USB4 Ports.

SBRX is driven high, its assertion is detected by the transactions FSM which signals the control unit to enter in phase 3 after asserting SBTX



1. sbrx = 1'b1 for time tconnect
2. disconnect_sbrx = 1'b0
3. disconnect_sbtx = 0
4. trans_state = 2'h1 (IDLE)
5. trans = 8'b11111111

Phase 3 - Port Characteristics

The third phase of Lane Initialization consists of each Router acquiring information about its Link Partner.

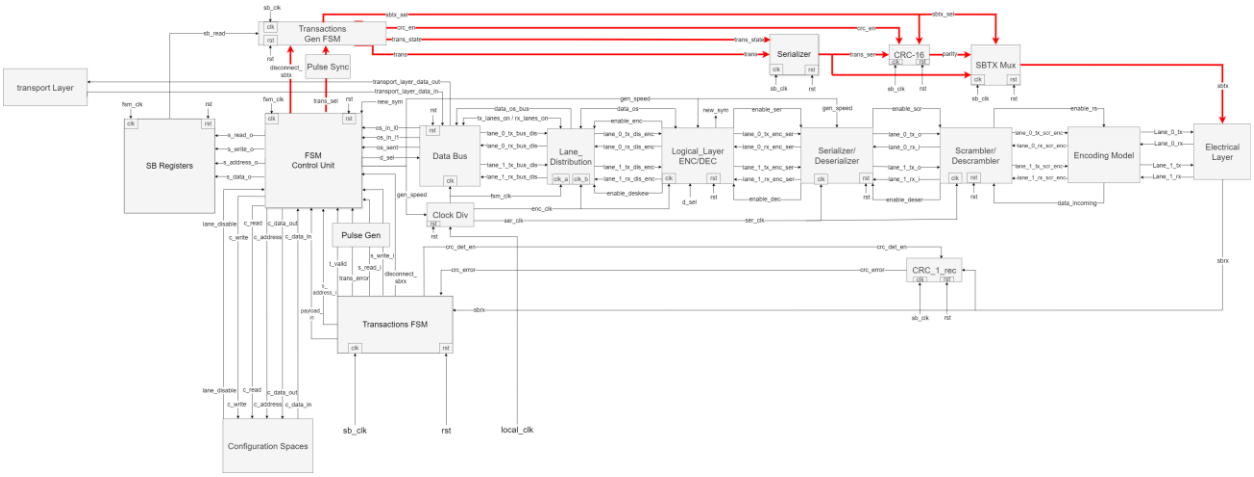
During Phase 3, Router A shall read the Link Configuration register (register 12) of Router B using AT Transactions.

This phase has 3 scenarios:

Ask for parameters

The control unit send to transactions block a (symb_sel) signal to make it outputs an AT command to read or write in the router in the other side of the link.

This command is serialized, and we add CRC to check errors before sending it on the SBTX

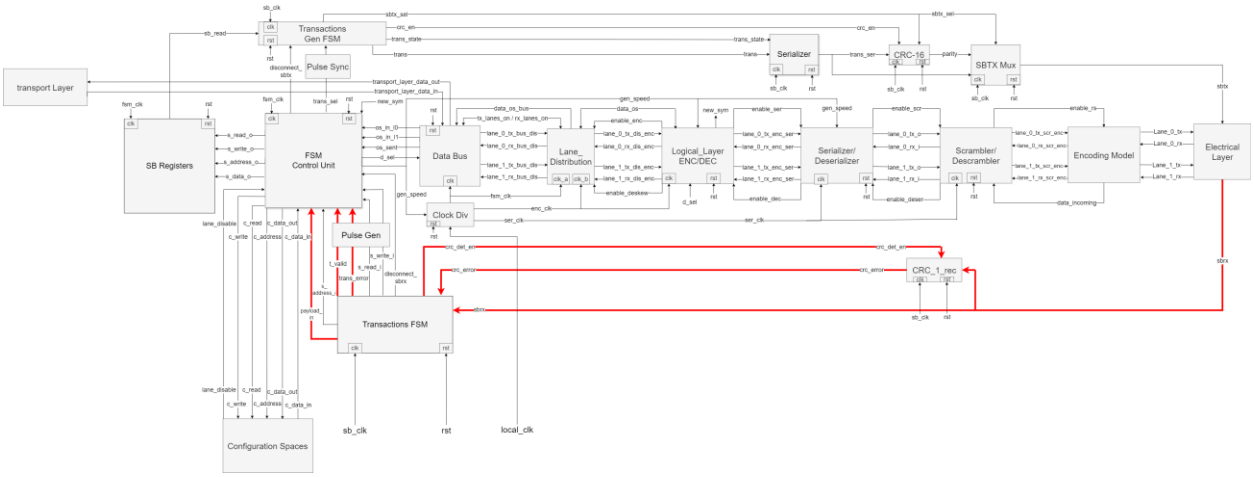


1. trans_sel = 3’h2 for one pulse then returns to trans_sel = 0
2. trans_state = 2’h2 (sending transaction mode)
3. trans = 8’h(DLE)
4. wait for DLE symbol to be sent (8 clock cycles)
5. trans = STX, crc_en = 1’b1 & sbtx_sel = 1’b0
6. trans = 8’d78 (address of corresponding SB Registers to read from)
7. wait for address to be sent (8 clock cycles)
8. sbtx_sel = 1’b1
9. wait for crc bits to be sent
10. trans = DLE, crc_en = 1’b0 and sbtx_sel = 1’b0
11. wait for DLE symbol to be sent
12. trans = ETX
13. wait for ETX symbol to be sent (8 clock cycles)
14. return to idle, trans = 8’b11111111
15. sbtx = 1’b1

Response with parameters

Its link partner sends an AT response to provide the information needed.

After checking for error, the transaction FSM detects that this transaction is an AT response, and it provides the control unit with the information read from the other router to help configuring the link.



1. DLE symbol detected by transactions fsm
2. crc_det_en = 1’b1
3. wait for STX, Data symbols and crc bits to be received then de-assert crc_det_en
4. error signal gets is updated after crc_det_en is de-asserted
5. receive DLE symbol
6. after receiving ETX symbol
    - if error = 1’b1

        trans_error = 1’b1 and fsm control unit sends another AT transaction read command

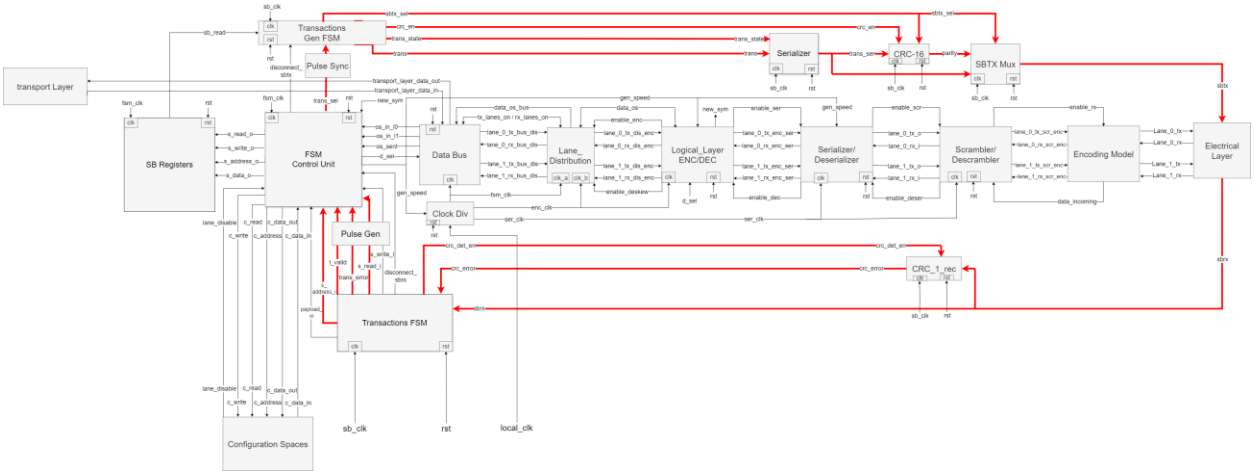
    - if error = 1’b0

        t_valid = 1’b1, payload_in, s_read, s_write and s_address signals are updated with values from the AT transaction.

Other lane adapter requesting parameters

When writing or reading from the SB registers, the control unit analyses the AT response (if read AT was sent) or the AT command (if write AT was sent) and send to the SB registers the address to read from or to write to.

If it's a read AT command, the data extracted from the registers are sent to the link partner through the transactions block to the SBTX. (symb_sel chooses whether to output data from register or parametrized bytes such as DLE and ETX).



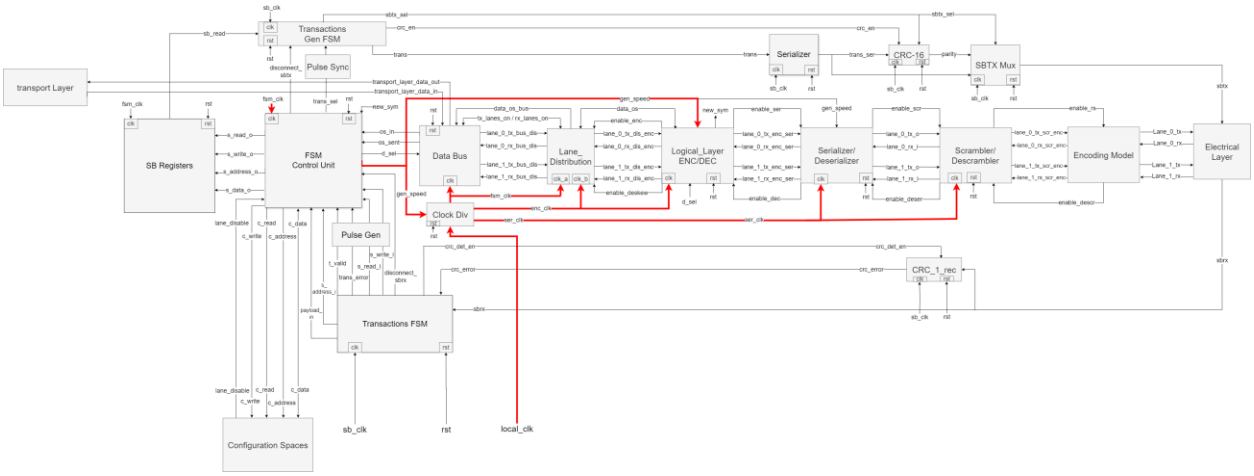
1. After receiving read command through AT transaction
2. s_read = 1, s_write = 1'b0 and s_address gets updated
3. sb_read signal gets updated by the SB Registers then and trans_sel = 3'h3 then transaction sent normally as described previously

Phase 4 - Transmission start

During Phase 4, a USB4 Port sets the Lane speed and start sending order sets depending on the generation it's supporting.

Adjusting Clock Frequencies and encoding scheme

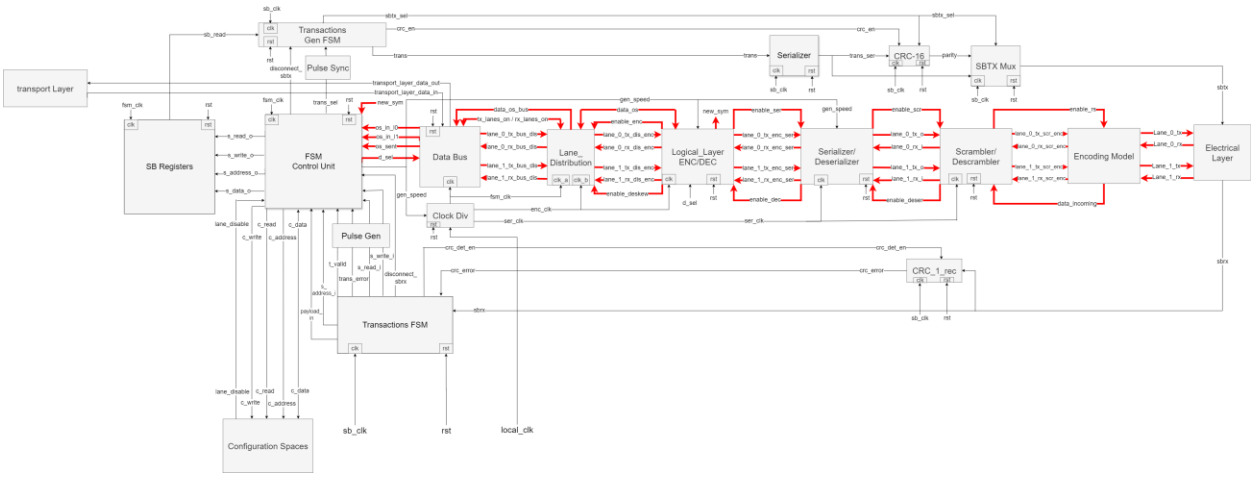
The clock frequencies entering the blocks and logical layer encoding scheme are adjusted according to the required generation speed, which is determined by the signal gen_speed



Sending synchronization order sets

When the control unit detects that it's in phase 4, it sends to the data bus a (d_sel) signal to send order sets on the lane for synchronization.

(os_in) signal is used to sensor the received order sets from the link partner and transition between the substates of the training state.



- TS1 training state:

  1. d_sel = 4'h4

2. lane_x_tx = TS1 (send at least 16 ordered sets back-to-back)
3. when receiving TS1, go to TS2 training state

- TS2 training state:
  1. d_sel = 4'h5
  2. lane_x_tx = TS2 (send at least 16 ordered sets back-to-back)
  3. when receiving TS2, go to TS3 training state

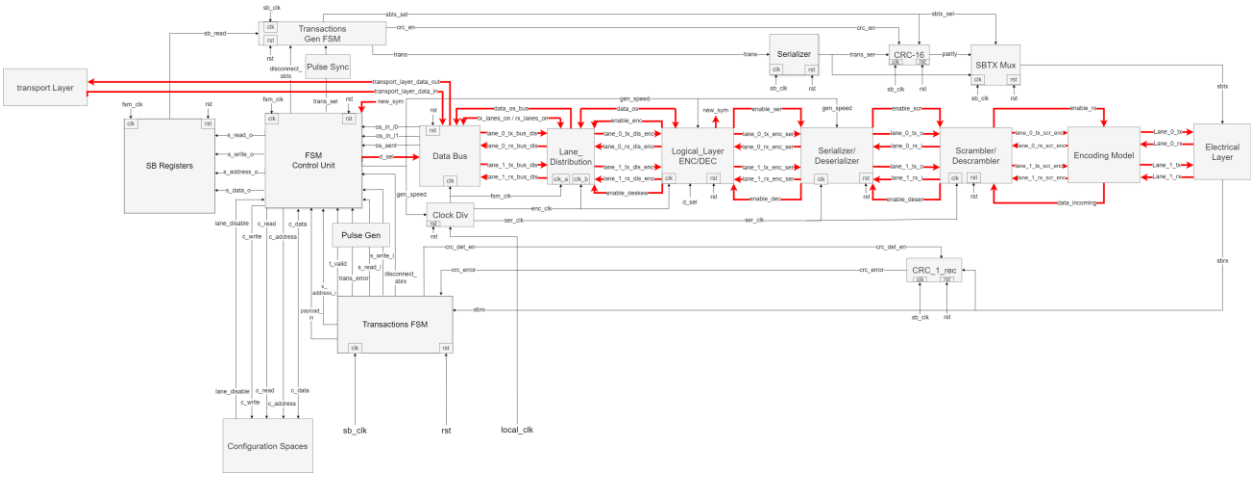
- TS3 training state:
  1. d_sel = 4'h6
  2. lane_x_tx = TS3 (send at least 16 ordered sets back-to-back)
  3. when receiving TS3, go to TS4 training state

- TS4 training state:
  1. d_sel = 4'h7
  2. lane_x_tx = TS4 (send at least 16 ordered sets back-to-back)
  3. when receiving TS4, go to CL0 state

Transport layer transmission

Sending transport layer data

After the link configuration and training, the control unit accepts the data transmitted from the transport layer through the data bus block and begin transmission on the link lanes.



- In CL0 state:
    1. d_sel = 4'h8
    2. lane_x_tx = transport_layer_data

Disconnection

Case 1:

1. sbrx = 1'b0 for time tdisconnect and detected by transactions fsm
2. disconnect_sbrx = 1'b1
3. The fsm returns to cld state

Case 2:

1. transactions fsm detects LT_Fall transaction from sbrx
2. disconnect_sbrx = 1'b1
3. The fsm returns to cld state

Disable

lane_disable = 1'b1 → fsm returns to disabled state

Frequencies used

Generations Data rate

The data rate is limited by the electrical layer properties, a good digital design must ensure a clock frequency suitable for the operation of the electrical layer, the electrical layer speed is limited by the generation's data rate.

- Gen 2 = 20 Gb/s
- Gen 3 = 40 Gb/s
- Gen 4 = 80 Gb/s

Local clock and sideband frequencies

This clock frequency will be the highest clock speed fed to the system, this clock is divided to supply the rest of the system with their needed clock frequencies.

- Local Clock Frequency = 80GHz

A sideband frequency of 1MHz is deliberately chosen as a lower frequency to ensure accurate and reliable lane initialization with the opposite lane adapter.

- Sideband Clock Frequency = 1MHz

Clock frequencies

We will synchronize the local clock of the system to match the fastest data rate of USB4 Gen4, which is 80Gb/s. Since 2 lanes are being utilized, we will divide this rate by 2 so the local clock equals 40GHz. All other clocks will be derived from the local clock using a clock divider.

Serializer/Deserializer block frequency

The serializer block in the lanes will utilize a clock frequency that equals the required data rate divided by 2. In the case of Gen4, the serializer clock frequency will be equal to the local clock, while for Gen3 or Gen2, the serializer clock frequency will be divided accordingly.

- Gen 2 Serialization frequency = 10GHz
- Gen 3 Serialization frequency = 20GHz
- Gen 4 Serialization frequency = 40GHz

Logical Layer Encoding block frequency

The output of the logical layer encoding block (input of the serializer) must be updated every (serializer rate/encoded symbol width) to ensure the correct sampling of different payloads

Every generation use a different encoding scheme, depending on the encoding scheme a clock frequency is chosen to ensure the correct serialization of the data in the needed time.

- Gen 2 encoding scheme = 64b/66b
- Gen 3 encoding scheme = 128b/132b
- Gen 4 encoding scheme = 11 bits/symbol

The frequency of the encoder at different gens will be as follows:

- Gen 2 encoding frequency = 10G/66 * 64/8 = 1.212 GHz
- Gen 3 encoding frequency = 20G/132 * 128/8 = 2.424 GHz
- Gen 4 encoding frequency = 40G/8 = 5 GHz

FSM and data bus blocks frequency

The data bus transmits 8 bits per lane, and the logical layer encoder collects these 8 bits and combines them to create a symbol of either 64 bits or 128 bits, which is then encoded into 66 bits or 132 bits, depending on the generation being used. In order to ensure proper synchronization, the output rate of the data bus should be faster than the clock frequency of the encoder by a ratio equal to the symbol size divided by 8 bits. Also due to the presence of prbs circuits in the data bus generating serial ordered sets, the clock frequency should be multiplied by further 8 to ensure proper synchronization.

- Gen 2 fsm frequency = 10G/66 * 64/8 * 8 = 9.697 GHz = 80G/8.25
- Gen 3 fsm frequency = 20G/132 * 128/8 * 8 = 19.394 GHz = 80G/4.125
- Gen 4 fsm frequency = 5 GHz * 8 = 40G

Clock Name	Gen_Speed	Ratio	Frequency
Local_clk	//	Local_clk	80 GHz
Ser_clk	Gen4	Local_clk/2	40GHz
	Gen3	Local_clk/4	20GHz
	Gen2	Local_clk/8	10GHz
fsm_clk	Gen4	Local_clk/2	40GHz
	Gen3	Local_clk/4.125	19.394GHz
	Gen2	Local_clk/8.25	9.697GHz
enc_clk	Gen4	Local_clk/16 (80GHz/16)	5 GHz
	Gen3	Local_clk/33 (80GHz/33)	2.424 GHz
	Gen2	Local_clk/66 (80GHz/66)	1.212 GHz
Sb_clk	//	Sideband_clk	1 MHz
ms_clk	//	Sideband_clk/1000 (1MHz/1000)	1 KHz

Transport Layer Data Rate

- Transport layer data to be sent on the lanes' transmitting side should be updated each 4 fsm_clk cycles in the transport_layer_data_in signal.

- On the other side, when receiving transport later data serially from the lanes' receiving side, the transport_layer_data_out signal will be updated every 4 fsm_clk cycles to be retrieved by the transport layer.

Contributions and Timeline

Block Name	Team Member Responsible
FSM Control Unit	Ahmed Zakaria
Timer	Ahmed Zakaria
Lane Distributer	Ahmed Zakaria
Scrambler/Descrambler	Ahmed Zakaria
Pulse Sync & Pulse Gen	Ahmed Zakaria
Final Integration	Ahmed Zakaria
Transactions	Ahmed Tarek
Transactions FSM	Ahmed Tarek
Logical Layer Encoding/Decoding	Ahmed Tarek
CRC_16 & CRC_16_rec	Hager Walid
SB Registers	Hager Walid
Data Bus	Seif Hamdy
Serializer / Deserializer	Seif Hamdy
Clock Div	Seif Hamdy