https://inst.eecs.berkeley.edu/~eecs251b/sp23/

EECS251B
Advanced Digital Circuits and Systems

Lecture 10 – Project overview Part 2 and modeling intro

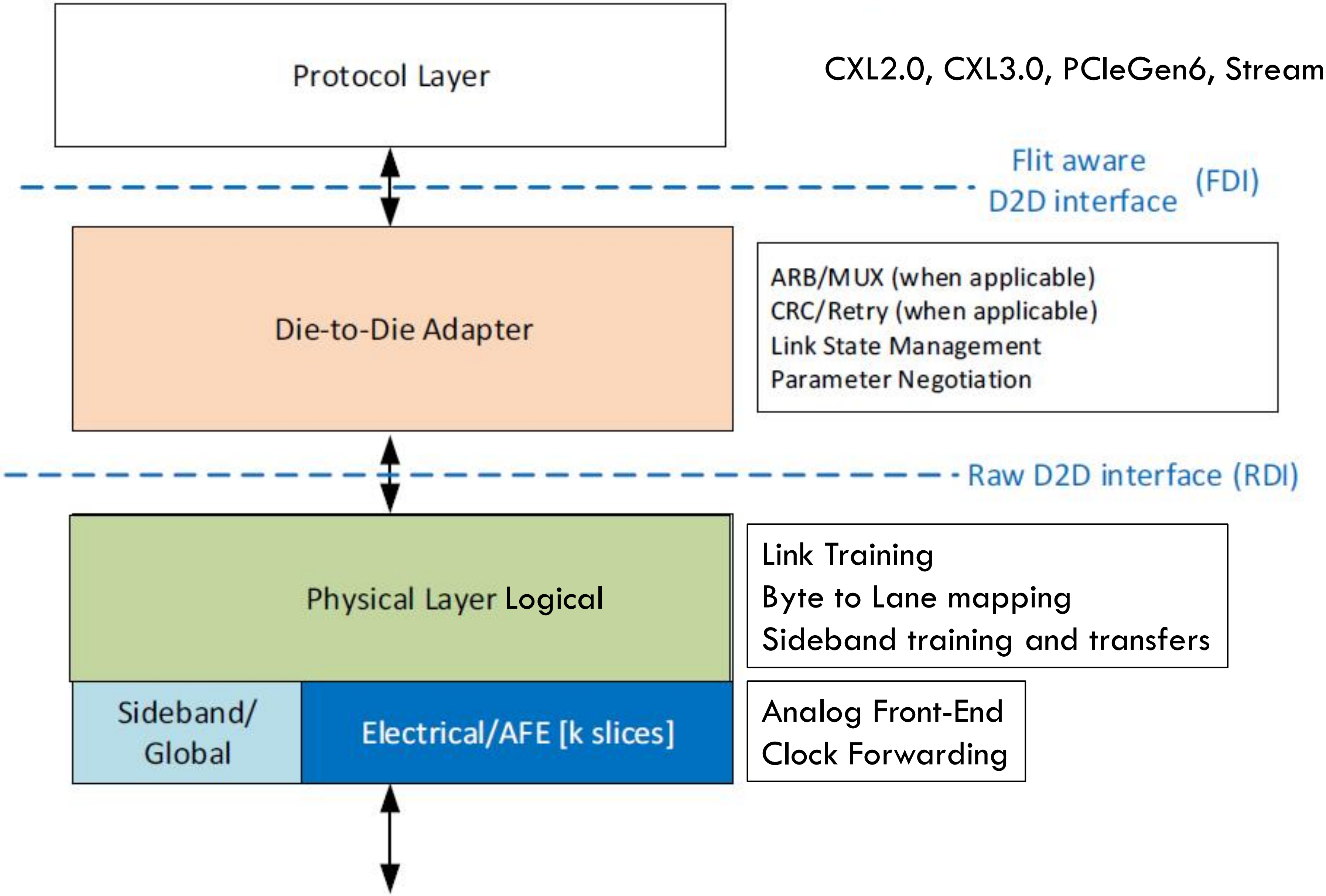
Vladimir Stojanović

Tuesdays and Thursdays 9:30-11am

Cory 521

Universal Chiplet Interconnect Express (UCIe) Overview - Protocol and D2D Adapter

UCIe: Interface Partitioning



Protocol Layer

CXL2.0, CXL3.0, PCIeGen6, Streaming

Flit aware D2D interface (FDI)

Die-to-Die Adapter

ARB/MUX (when applicable)
CRC/Retry (when applicable)
Link State Management
Parameter Negotiation

Raw D2D interface (RDI)

Physical Layer Logical

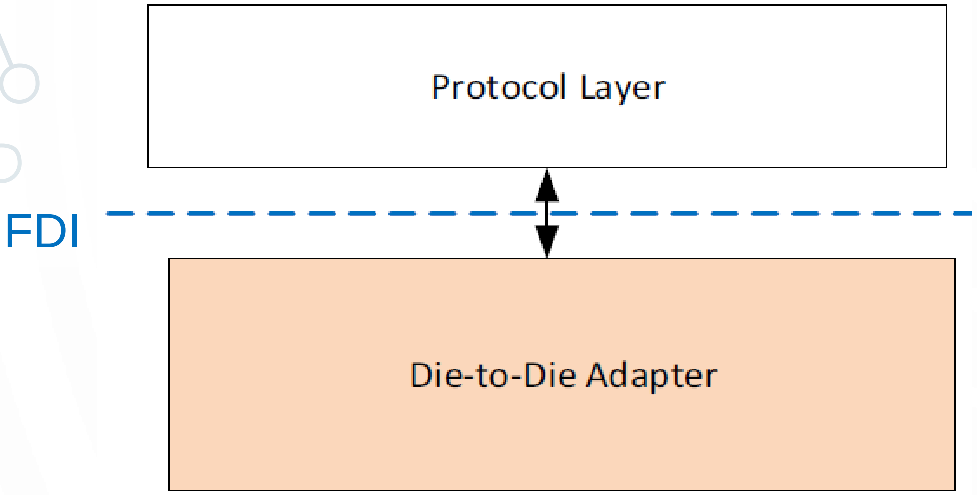
Link Training
Byte to Lane mapping
Sideband training and transfers

Sideband/ Global

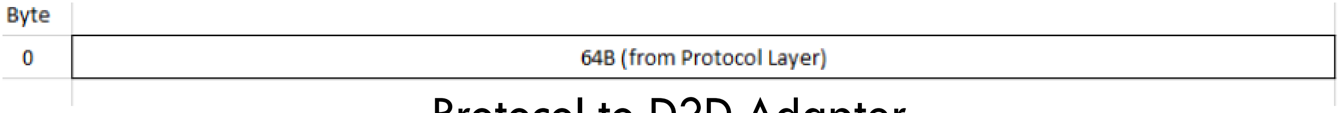
Electrical/AFE [k slices]

Analog Front-End
Clock Forwarding

UCIe: Protocol Layer – Streaming mode

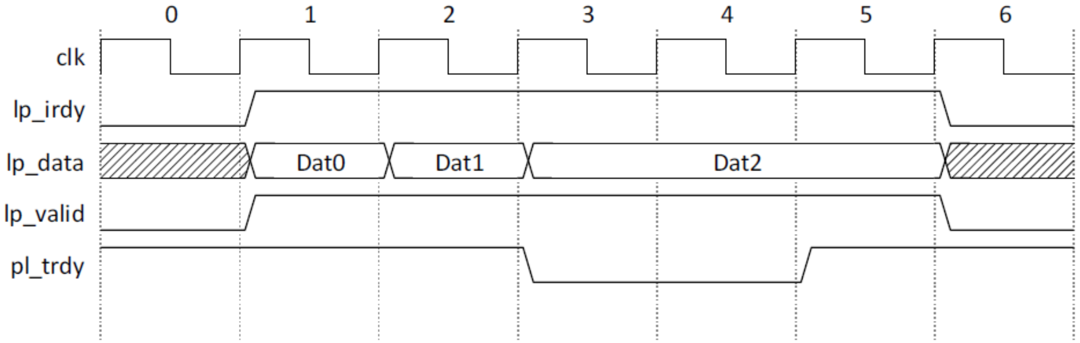


Raw Flit mode – 64B



Byte	
0	64B (from Protocol Layer)

Protocol to D2D Adapter



- Protocol Layer responsible for transmitting data over FDI in accordance with the negotiated mode and Flit format
- Streaming Protocol is the default protocol that must be advertised if none of the PCIe or CXL protocols are going to be advertised and negotiated with the remote Link partner
- Streaming Protocol offers generic modes (Raw Mode is mandatory) for a user defined protocol to be transmitted using UCIe (e.g. TileLink)

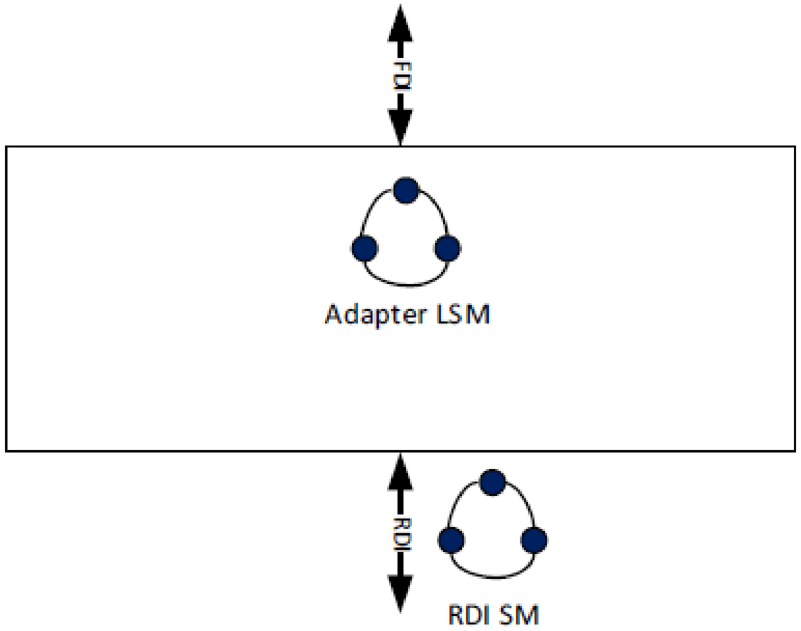
UCIe: D2D Adapter – Streaming mode

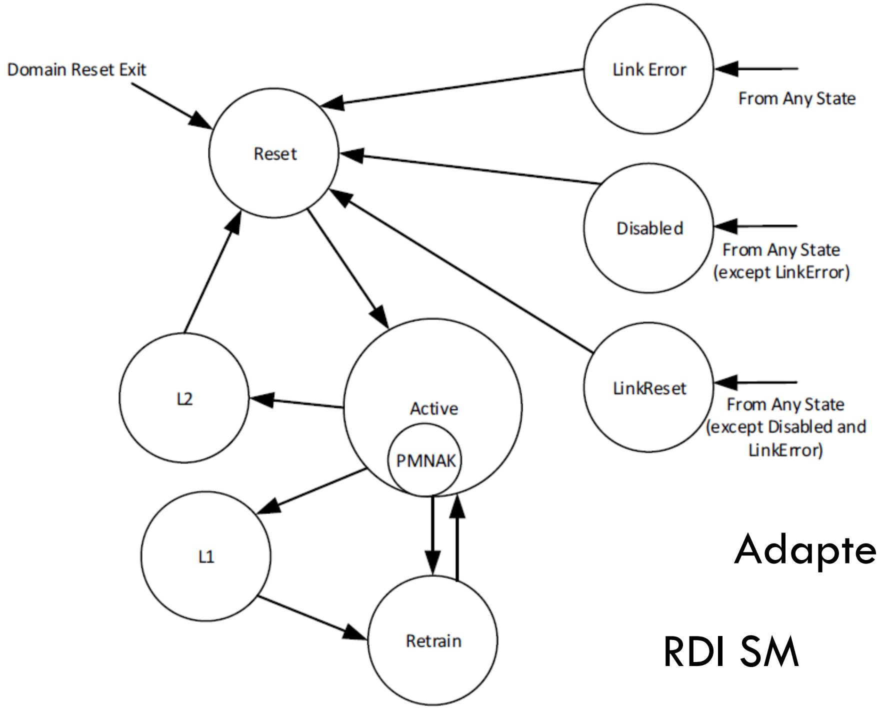
The Die-to-Die Adapter is responsible for:

- Reliable data transfer (performing CRC computation and Retry, or parity computation)
- Arbitration and Muxing (in case of multiple Protocol Layers)
- Link State Management
- Protocol and Parameter negotiation with the remote Link partner

- D2D Adapter negotiates the protocol with remote Link partner and communicates it to the Protocol Layer(s)
- UCIe supports multiple modes of operation for each protocol (must be negotiated with the remote Link partner depending on the advertised capabilities, Physical Layer Status as well as usage models)
  - Modes have different Flit formats to enable trade-offs in efficiency, bandwidth and interoperability
- Supported protocols, advertised modes and Flit formats determined at SoC integration time or during the die-specific reset bring up flow
  - The Die-to-Die Adapter uses this information to negotiate the operational mode as a part of Link Training and informs the Protocol Layer over the Flit-aware Die-to-Die Interface (FDI)

UCIe: D2D Adapter – State Machine Hierarchy



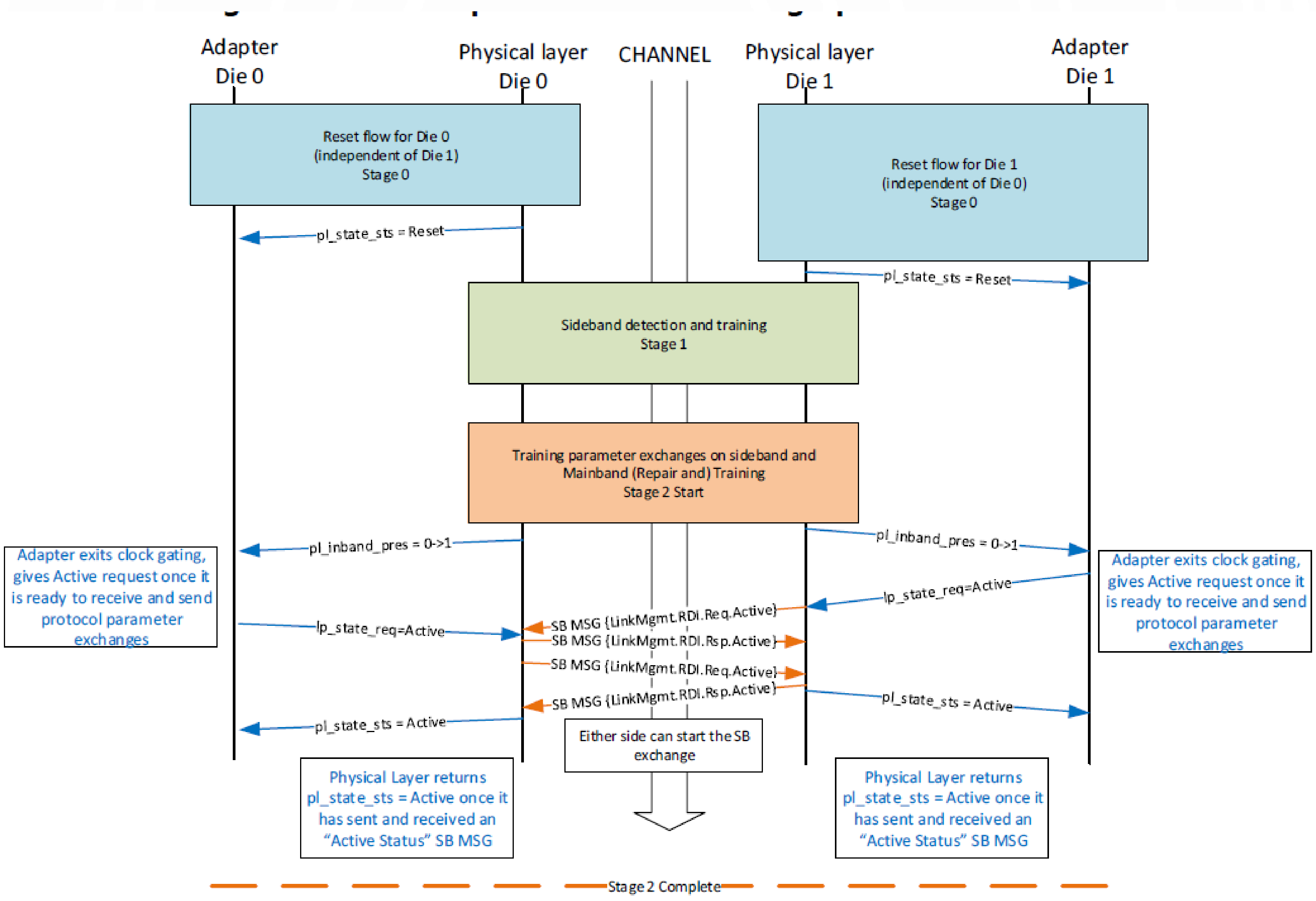


Adapter LSM

RDI SM

- Adapter LSM used to coordinate Link states with remote Link partner
  - {LinkMgmt.Adapter*} sideband msg
- RDI SM abstracts Physical layer states
  - Adapter data-path and RDI data width can be extended for multi-module configurations
  - Single RDI SM per Adapter (common to all modules in multi-module config)
  - {LinkMgmt.RDI*} sideband msg

UCIe: D2D Adapter – Link Initialization



Reset flow for Die 0
(independent of Die 1)
Stage 0

Reset flow for Die 1
(independent of Die 0)
Stage 0

Sideband detection and training
Stage 1

Training parameter exchanges on sideband and Mainband (Repair and) Training
Stage 2 Start

Adapter exits clock gating, gives Active request once it is ready to receive and send protocol parameter exchanges

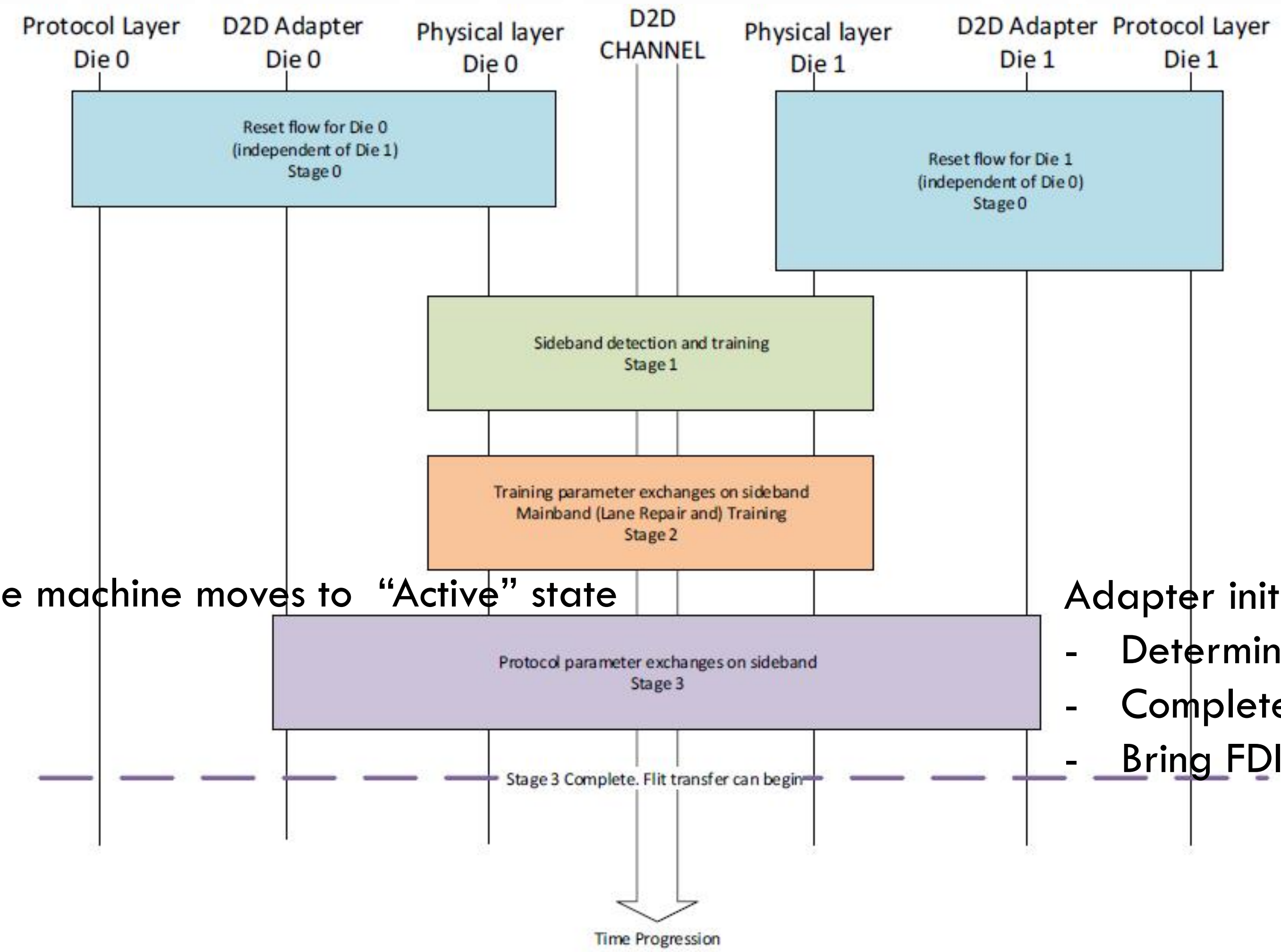
Adapter exits clock gating, gives Active request once it is ready to receive and send protocol parameter exchanges

Either side can start the SB exchange

Physical Layer returns pl_state_sts = Active once it has sent and received an “Active Status” SB MSG

Physical Layer returns pl_state_sts = Active once it has sent and received an “Active Status” SB MSG

UCIe: D2D Adapter – Link Initialization



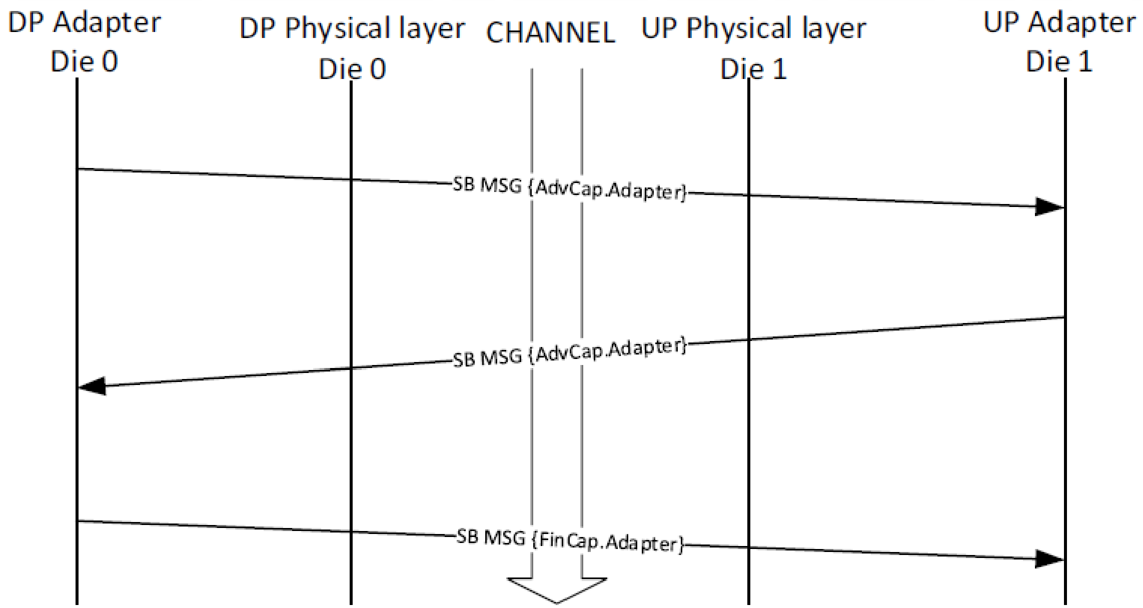
RDI state machine moves to “Active” state

Adapter initialization

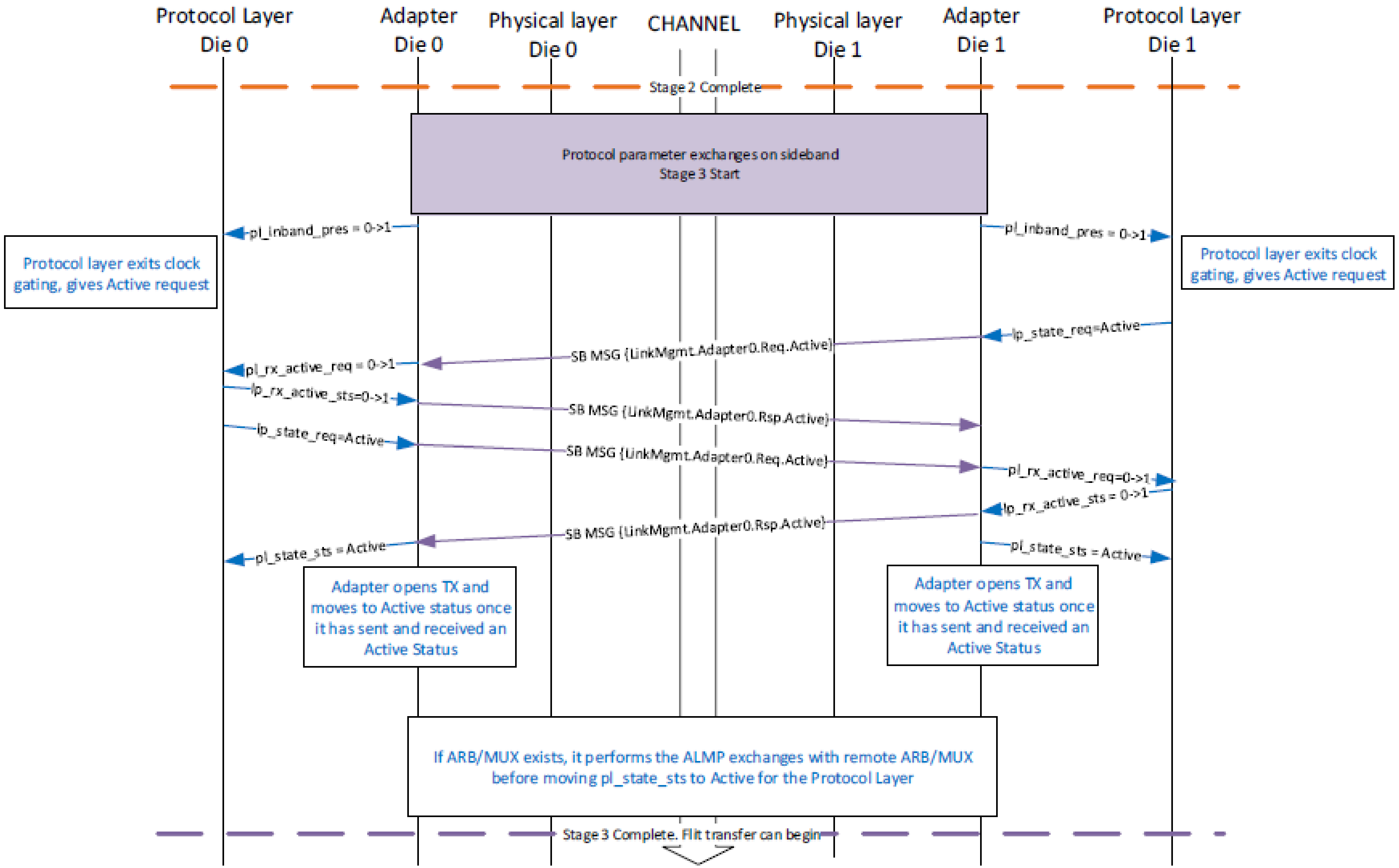
- Determine local capabilities
- Complete parameter exchanges
- Bring FDI state-machine to “Active” state

UCIe: D2D Adapter – Link Initialization – Adapter Initialization

- Determine Local Capabilities
  - Determine the results of Physical Layer training and if Retry is needed for the given Link speed and configuration
  - If the Adapter is capable of supporting Retry, it must advertise this capability to the remote Link partner during Parameter Exchanges
- Parameter exchange with remote Link partner
  - Capabilities exchanged with remote Link partner via sideband messages {AdvCap.Adapter}
    - Raw mode, Streaming, Retry, Retimer, Retimer_Credits, “DP” – Downstream Port, “UP” – Upstream Port, etc
  - Finalized configuration in {FinCap.Adapter} message – not sent for Streaming Protocol
    - Raw mode 1, Streaming 1, Retry 1, etc
  - Timer increments when RDI in “Active” state (8ms time-out)
    - Resets if the Adapter receivers {AdvCap.*.Stall} or {FinCap.*.Stall} from remote Link partner



UCIe: D2D Adapter – Link Initialization – FDI Activation



Protocol Layer Die 0

Adapter Die 0

Physical layer Die 0

CHANNEL

Physical layer Die 1

Adapter Die 1

Protocol Layer Die 1

Stage 2 Complete

Protocol parameter exchanges on sideband
Stage 3 Start

pl_inband_pres = 0->1

pl_inband_pres = 0->1

Protocol layer exits clock gating, gives Active request

Protocol layer exits clock gating, gives Active request

lp_state_req=Active

SB MSG {LinkMgmt.Adapter0.Req.Active}

pl_rx_active_req = 0->1

lp_rx_active_sts=0->1

SB MSG {LinkMgmt.Adapter0.Rsp.Active}

lp_state_req=Active

SB MSG {LinkMgmt.Adapter0.Req.Active}

pl_rx_active_req=0->1

lp_rx_active_sts = 0->1

SB MSG {LinkMgmt.Adapter0.Rsp.Active}

pl_state_sts = Active

pl_state_sts = Active

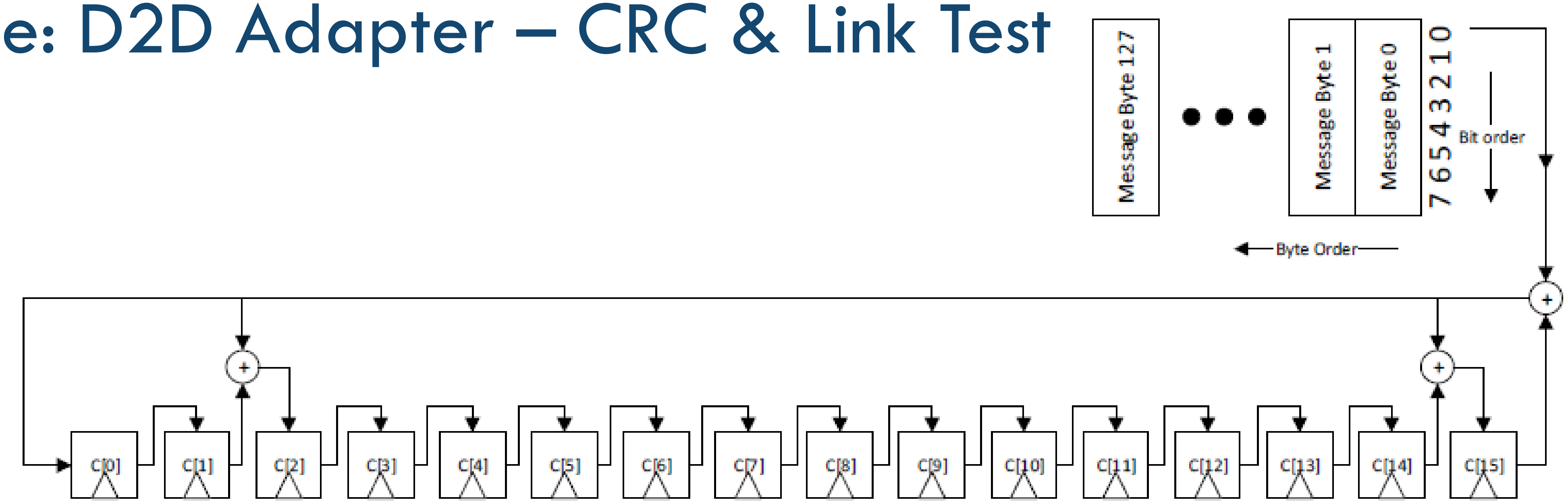
Adapter opens TX and moves to Active status once it has sent and received an Active Status

Adapter opens TX and moves to Active status once it has sent and received an Active Status

If ARB/MUX exists, it performs the ALMP exchanges with remote ARB/MUX before moving pl_state_sts to Active for the Protocol Layer

Stage 3 Complete. Flit transfer can begin

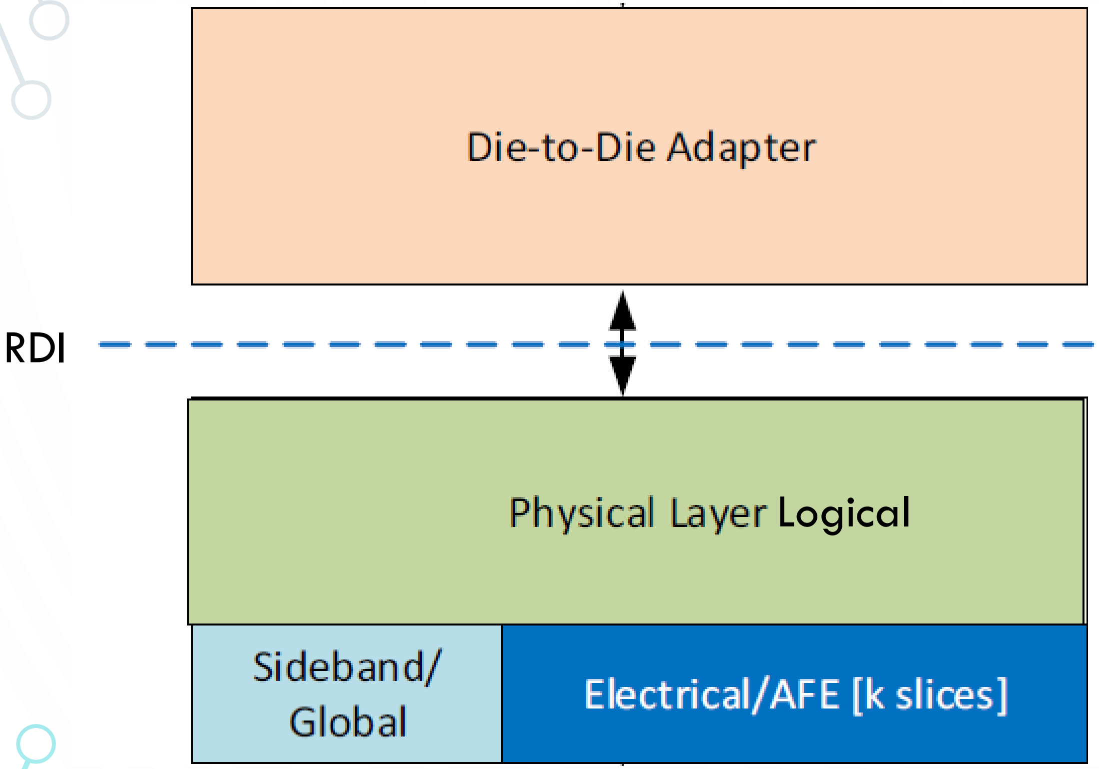
UCIe: D2D Adapter – CRC & Link Test



- CRC always computed over 128 Bytes of message
  - For smaller messages zero-extended in the MSB – e.g. for Streaming mode 64B flit
- CRC generator polynomial is $(x+1)*(x^{15} + x + 1) = x^{16} + x^{15} + x^2 + 1$
  - 3-bit error detection guarantee for random bit errors
    - 2-bit error detection because of the primitive polynomial (x15 + x + 1)
    - 1 additional bit error detection because of the (x+1) term
- Verilog code provided in the spec for Tx
- Retry not mandatory in Raw mode
- Runtime Link testing using injected Parity Bits {ParityFeature.Req/Ack}

Universal Chiplet Interconnect Express (UCIe) Overview - PHY Logical

UCIe: Physical Layer Logical

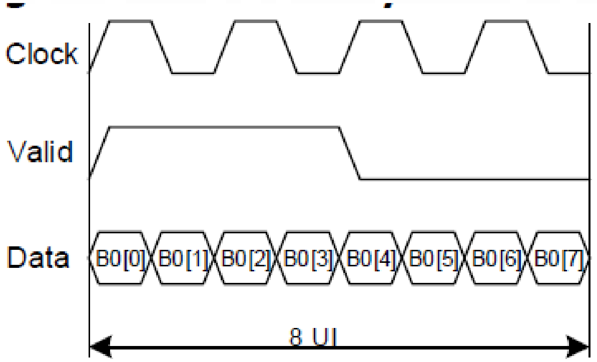


Logical PHY has the following functions:

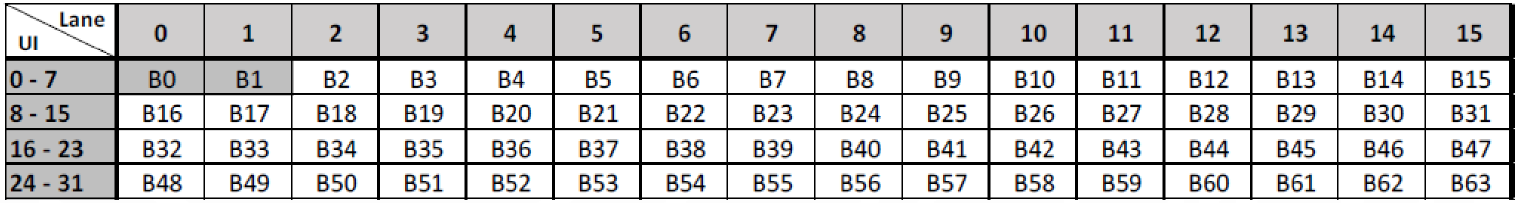
- Link initialization, training and power management states
- Byte to Lane mapping for data transmission over Lanes
- Transmitting and receiving sideband messages
- Scrambling and training pattern generation
- Lane reversal
- Width degradation (when applicable)

UCIe: PHY Logical – Data and Sideband Flow

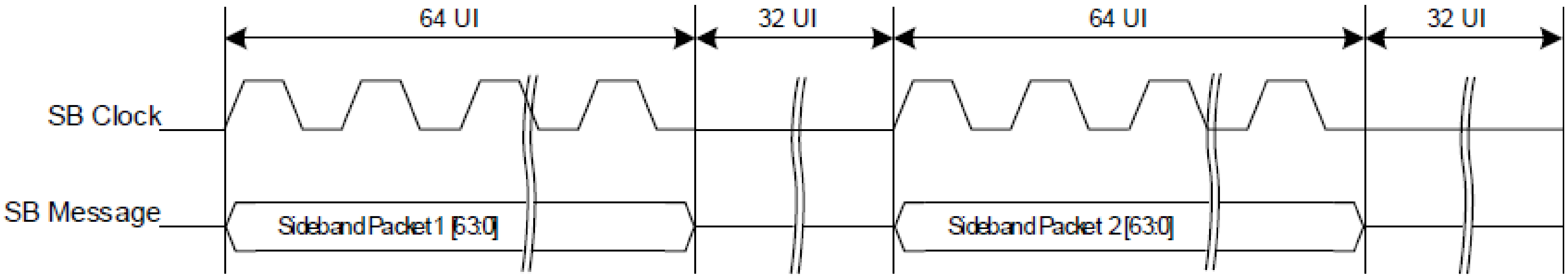
Transmission order on each lane



Byte to Lane Mapping x16 module, Raw Mode 64B Flit

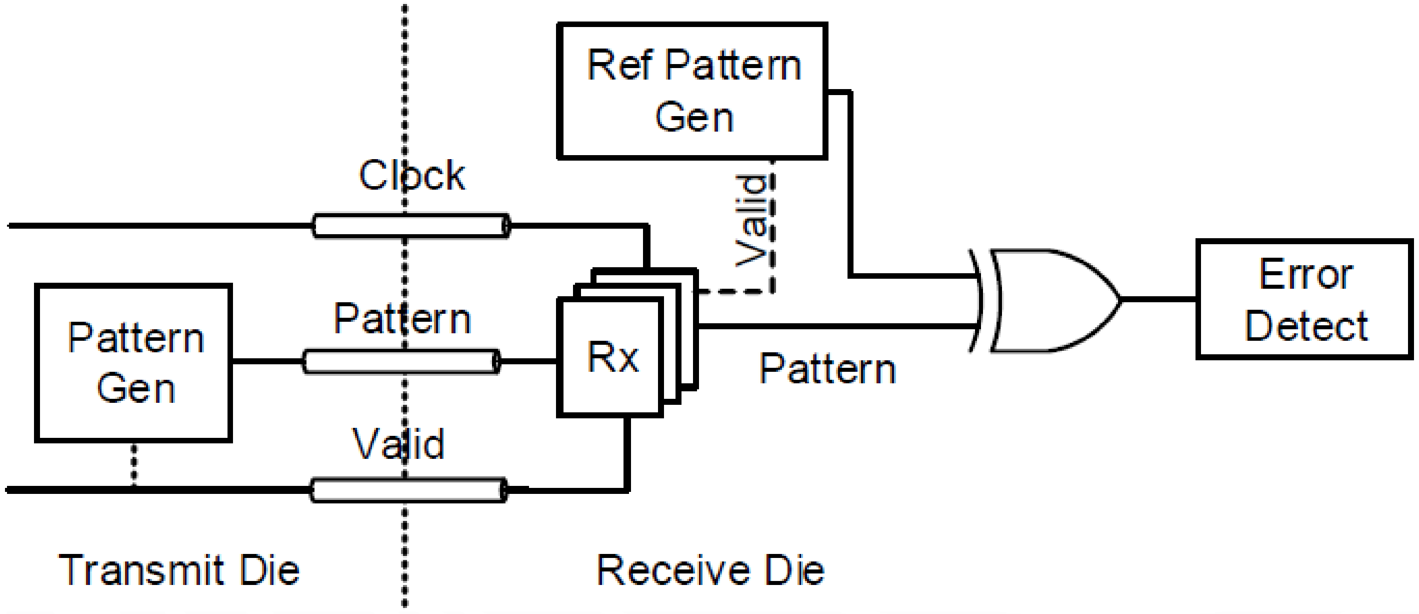


UI \ Lane	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 - 7	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15
8 - 15	B16	B17	B18	B19	B20	B21	B22	B23	B24	B25	B26	B27	B28	B29	B30	B31
16 - 23	B32	B33	B34	B35	B36	B37	B38	B39	B40	B41	B42	B43	B44	B45	B46	B47
24 - 31	B48	B49	B50	B51	B52	B53	B54	B55	B56	B57	B58	B59	B60	B61	B62	B63

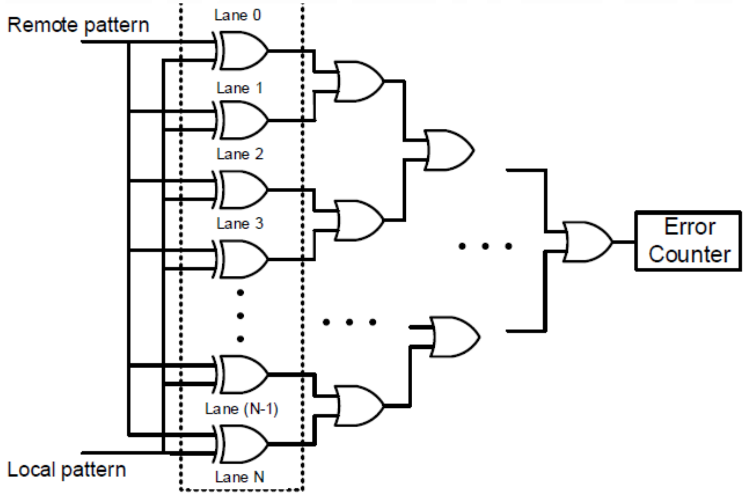


UCIe: PHY Logical – Data to Clock Training and Test Modes

Per Lane Test

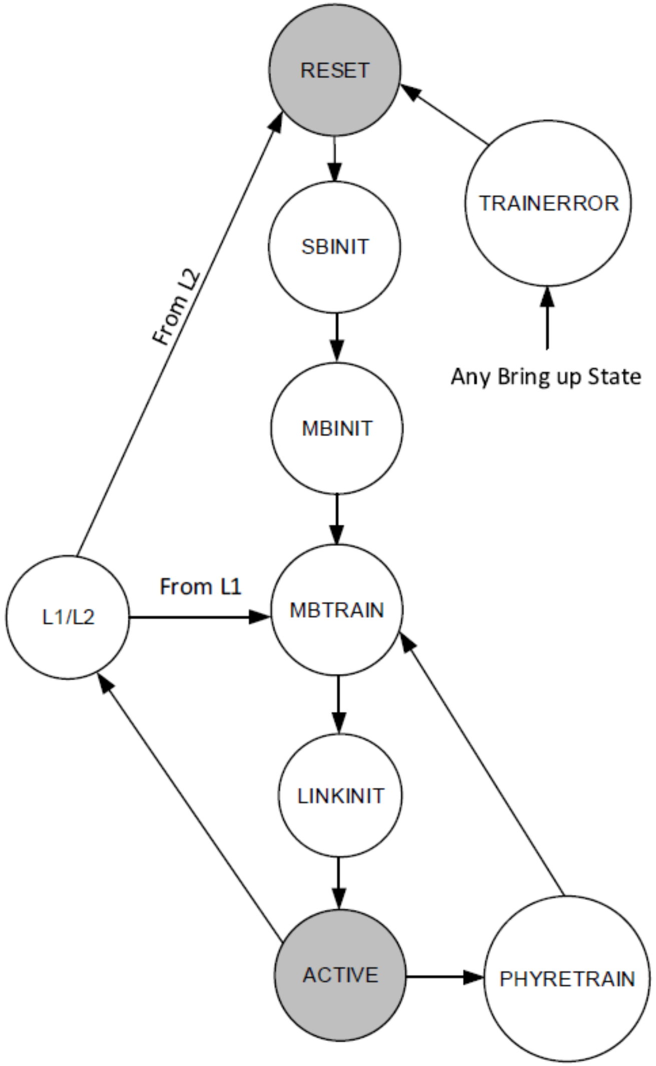


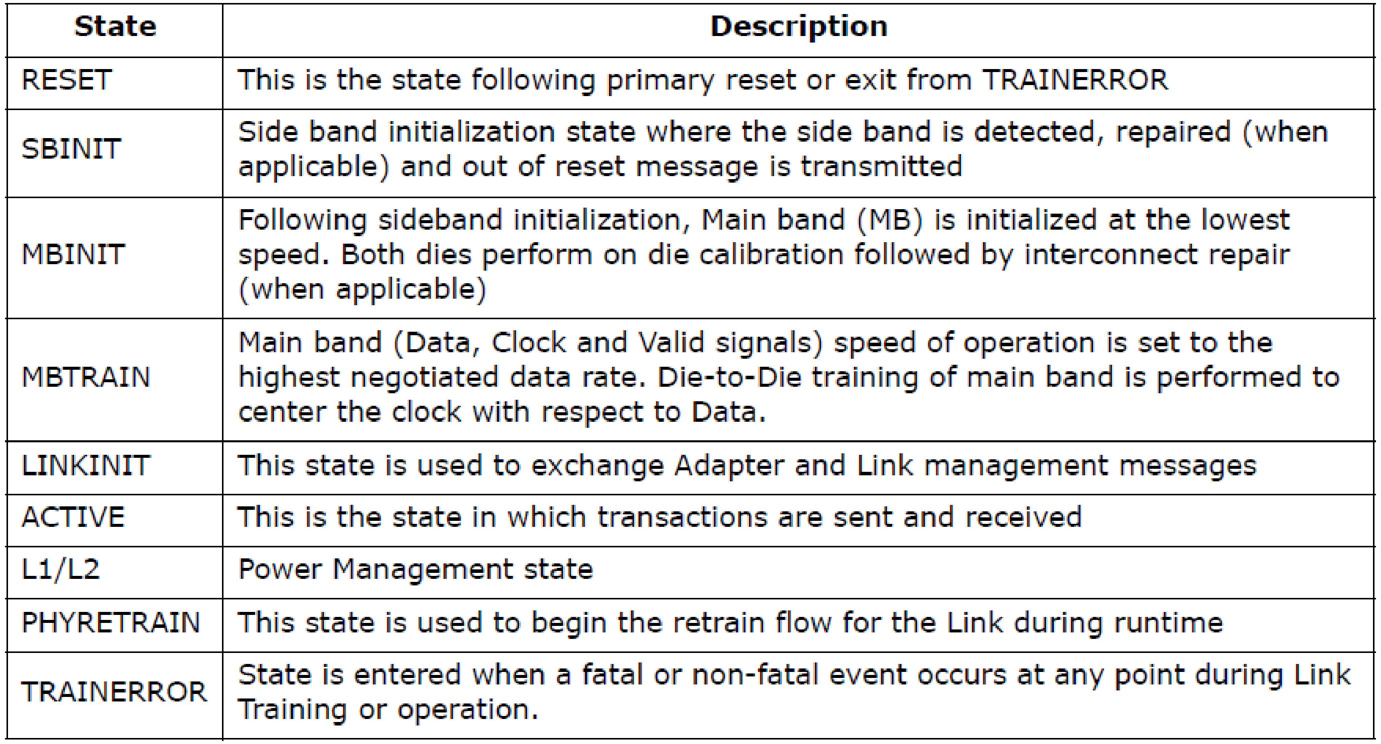
Aggregate Test



- Transmitter and receive die implement the same Linear Feedback Shift Register (LFSR)
  - $G(X)=X^{23} + X^{21} + X^{16} + X^{8} + X^{5} + X^{2} + 1$ (same as PCIe LFSR), seed value lane dependent
- Pattern from Transmitter along with forwarded clock and Valid compared with locally generated reference pattern
  - Both transmit and receive pattern generators must start and advance in sync

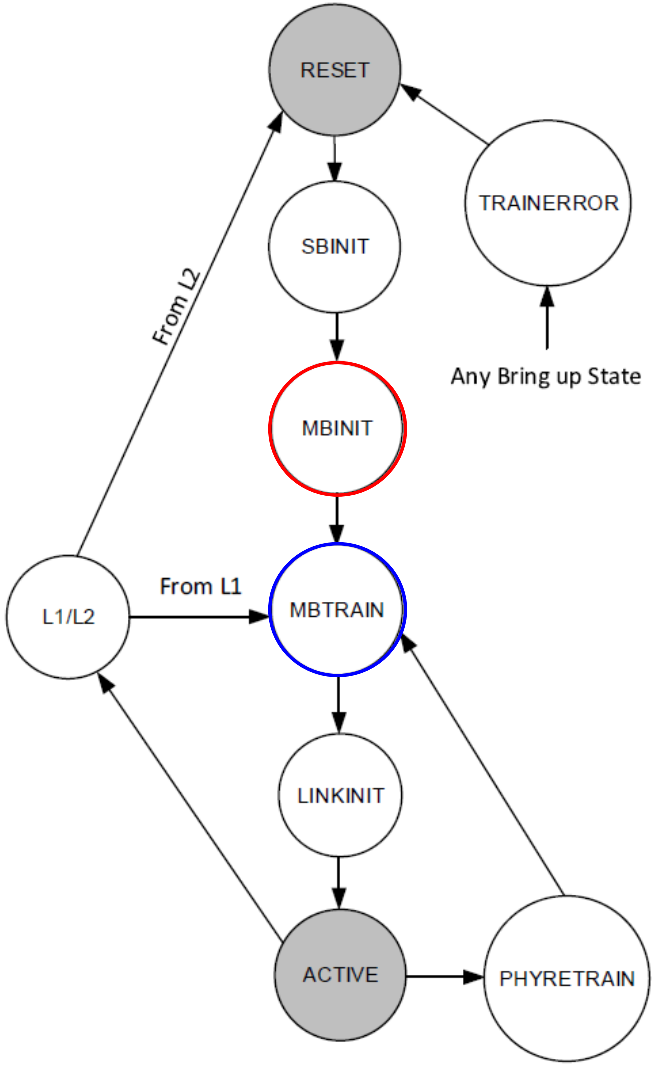
UCIe: PHY Logical – Link Training State Machine

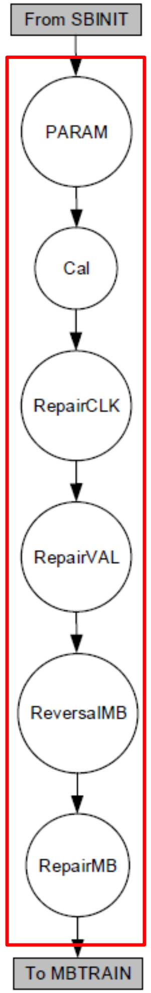


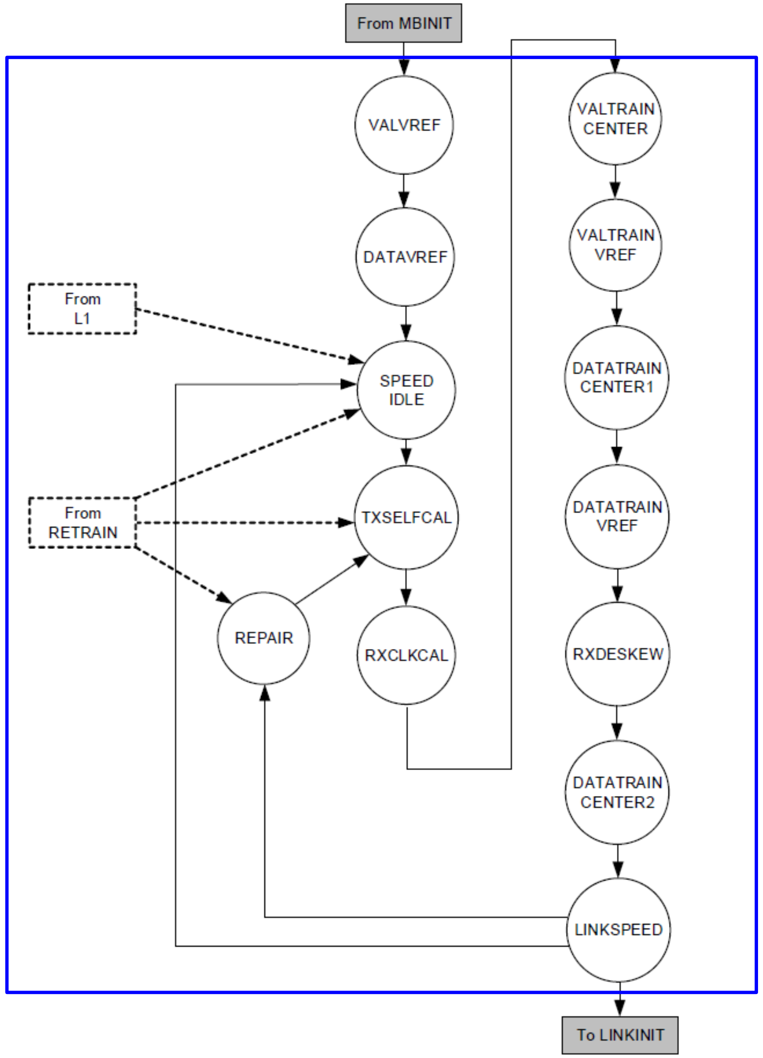


State	Description
RESET	This is the state following primary reset or exit from TRAINERROR
SBINIT	Side band initialization state where the side band is detected, repaired (when applicable) and out of reset message is transmitted
MBINIT	Following sideband initialization, Main band (MB) is initialized at the lowest speed. Both dies perform on die calibration followed by interconnect repair (when applicable)
MBTRAIN	Main band (Data, Clock and Valid signals) speed of operation is set to the highest negotiated data rate. Die-to-Die training of main band is performed to center the clock with respect to Data.
LINKINIT	This state is used to exchange Adapter and Link management messages
ACTIVE	This is the state in which transactions are sent and received
L1/L2	Power Management state
PHYRETRAIN	This state is used to begin the retrain flow for the Link during runtime
TRAINERROR	State is entered when a fatal or non-fatal event occurs at any point during Link Training or operation.

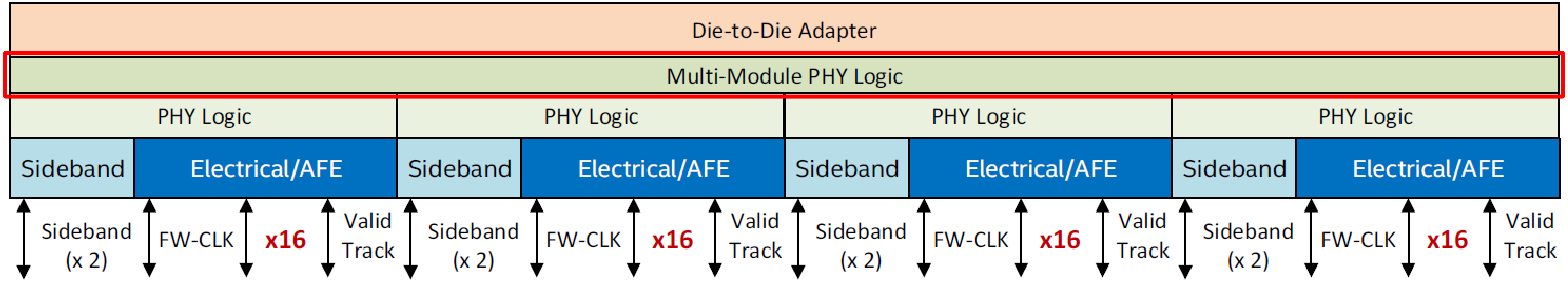
UCIe: PHY Logical – Link Training State Machine







UCIe: PHY Logical – Multi-Module PHY Logic (MMPL)



- Responsible for orchestrating data transfers across multiple modules
  - Each module must operate at same width and speed
  - Each module must initialize and train independently using its sideband
    - All other sideband messages sent on Module-0 sideband only
  - If any module fails to train the MMPL ensures configuration degrades to next permitted configuration

MOS Transistor and Gate Delay Models

Modeling Goals

- Models that traverse design hierarchy
- Start with transistor models
- Gate delay models
- Use models to time the design
- Modeling variability

- Based on 251A, approach
  - Start simple
  - Increase accuracy, when needed

Device Models

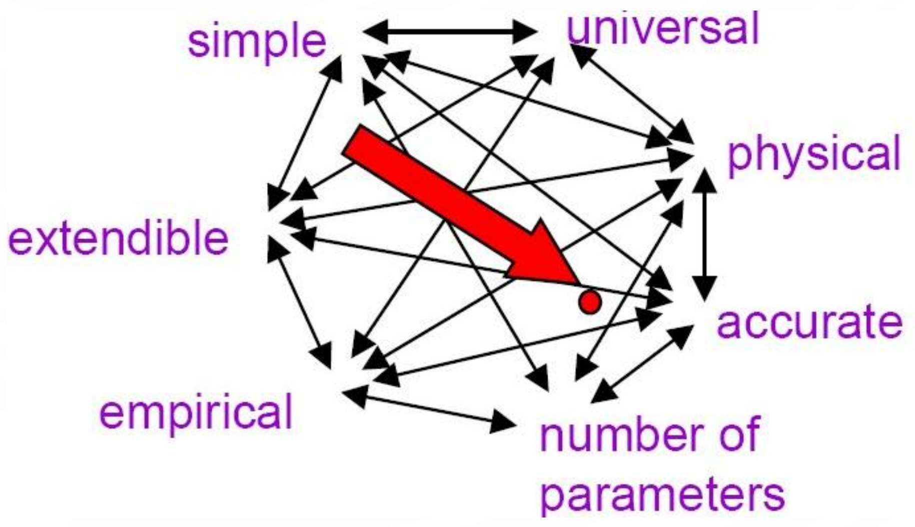
- Transistor models
  - I-V characteristics
  - C-V characteristics
- Interconnect models
  - R, C, L
  - Covered in EE240A

Transistor Modeling

- Different levels:
  - Hand analysis
  - Computer-aided analysis (e.g. Matlab, Python, Excel,…)
  - Switch-level simulation (some flavors of 'fast Spice')
  - Circuit simulation (Hspice)
- These levels have different requirements in complexity, accuracy and speed of computation
- We are primarily interested in delay and energy modeling, rather than current modeling
- But we have to start from the currents…

Transistor Modeling

- DC
  - Accurate I-V equations
  - Well behaved conductance for convergence (not necessarily accurate)
- Transient
  - Accurate I-V and Q-V equations
  - Accurate first derivatives for convergence
  - Conductance, as in DC
- Physical vs. empirical



from BSIM group

Goal for Today

- Develop velocity-saturated model for I_{on} and apply it to sizing and delay calculation
  - Similar approach as in 251A, just use an analytical model

Transistor I-V Modeling

- BSIM
  - Superthreshold and subthreshold models
  - Need smoothening between two regions
- EKV/PSP
  - One continuous model based on channel surface potential

Long-Channel MOS On-Current

MOS I-V (BSIM)

Start with the basics:

$$I_{DS} = W \underbrace{C_{ox}(V_{GS} - V_{Th} - V_C(x))}_{\text{Q'(x): Charge density in channel at x}} \underbrace{\mu E(x)}_{\text{v: velocity at x}}$$

MOS I-V (BSIM)

Start with the basics:

$$I_{DS} = W C_{ox}(V_{GS} - V_{Th} - V_C(x))\ \mu E$$

$$I_{DS} = W C_{ox}(V_{GS} - V_{Th} - V_C(x))\ \mu(\mathrm{d} V_C(x)/\mathrm{dx})$$

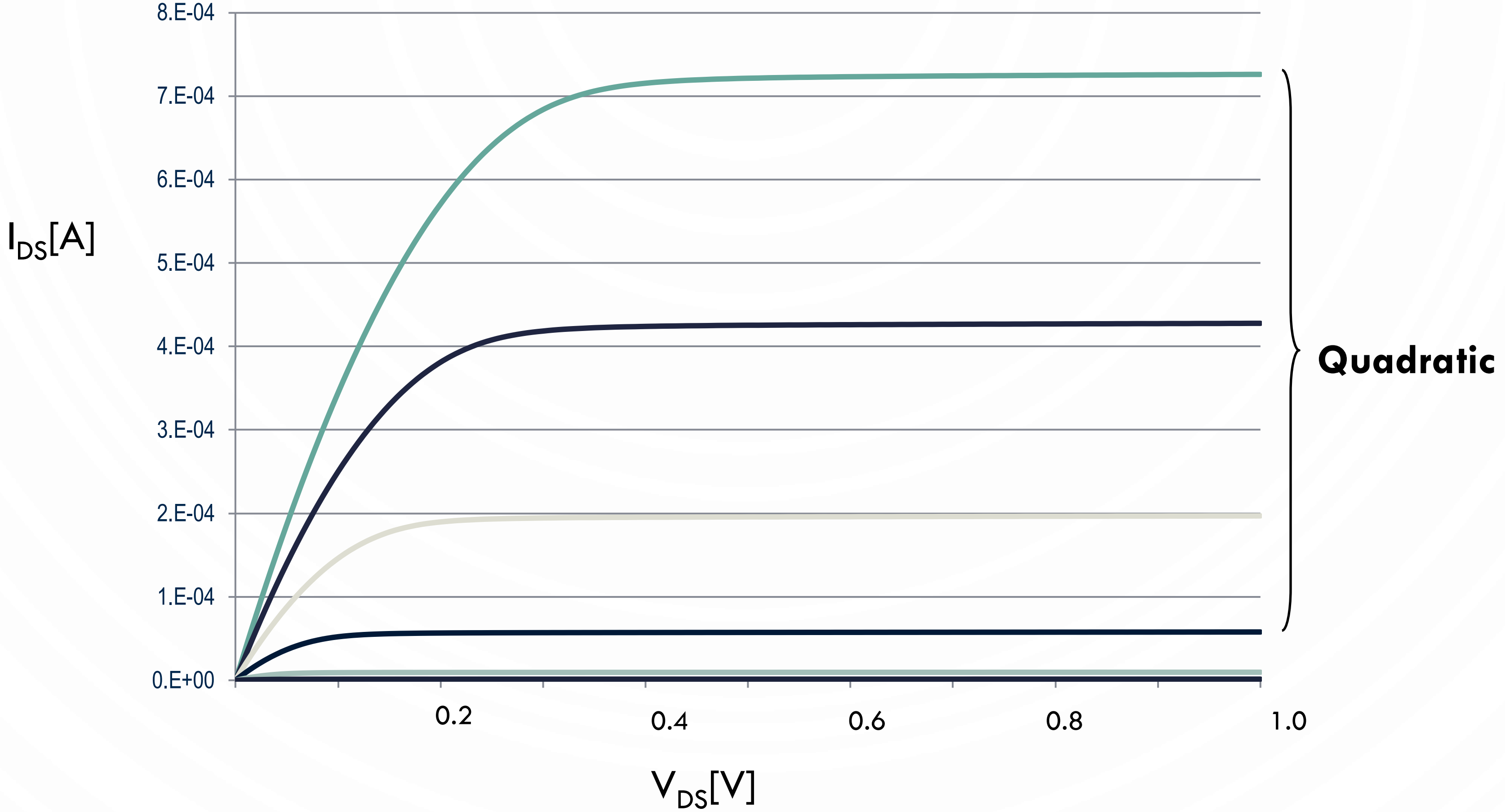
- When integrated over the channel:

$$I_{DS} = \frac{W}{L}\mu C_{ox}\left(V_{GS} - V_{Th} - \frac{V_{DS}}{2}\right)V_{DS}$$

Transistor saturates when $V_{GD} = V_{Th}$, - the channel pinches off at drain's side.

$$I_{DS} = \frac{W}{2L}\mu C_{ox}(V_{GS} - V_{Th})^2$$

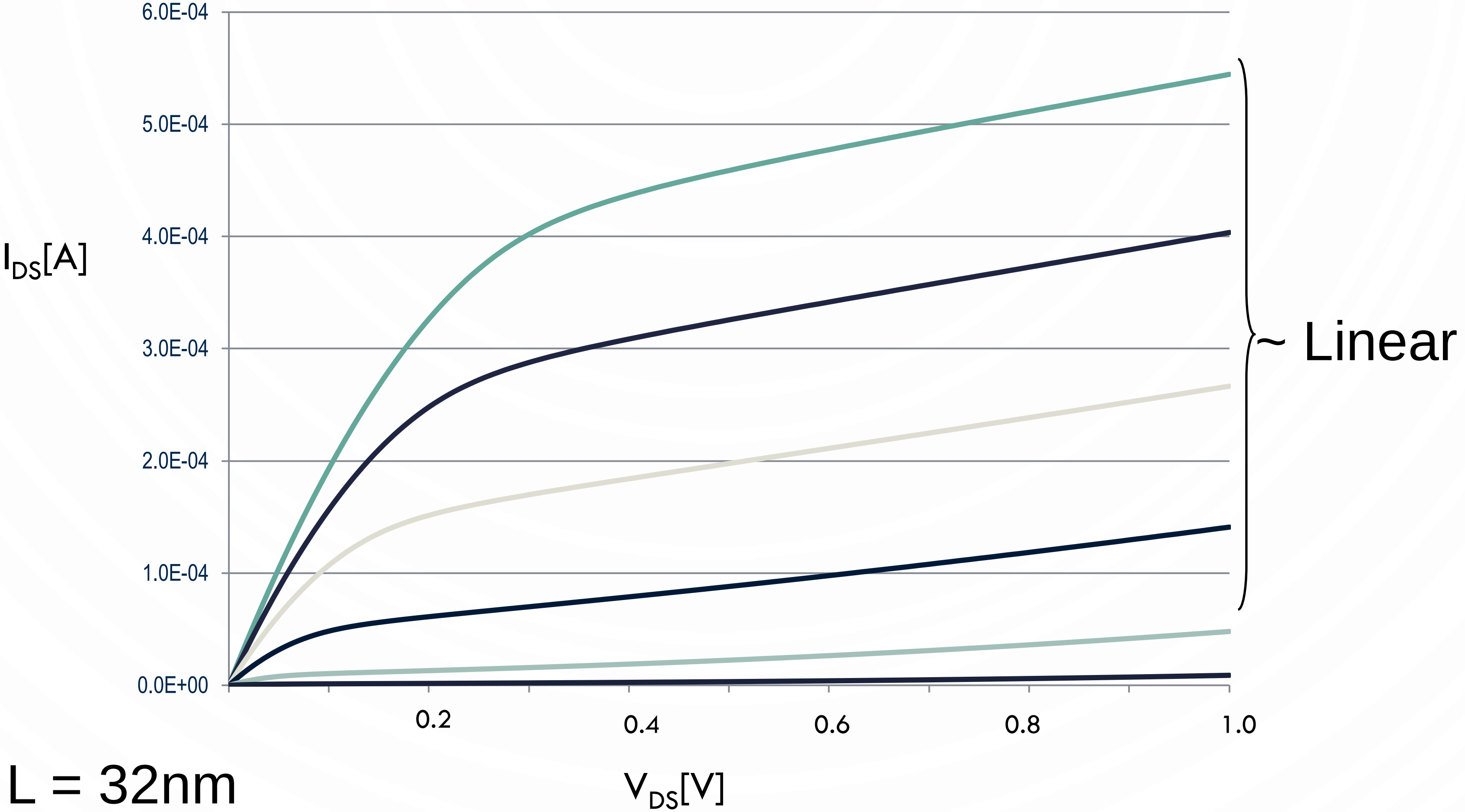
MOS Currents (32nm CMOS with L>>1µm)



Currents according to the quadratic model

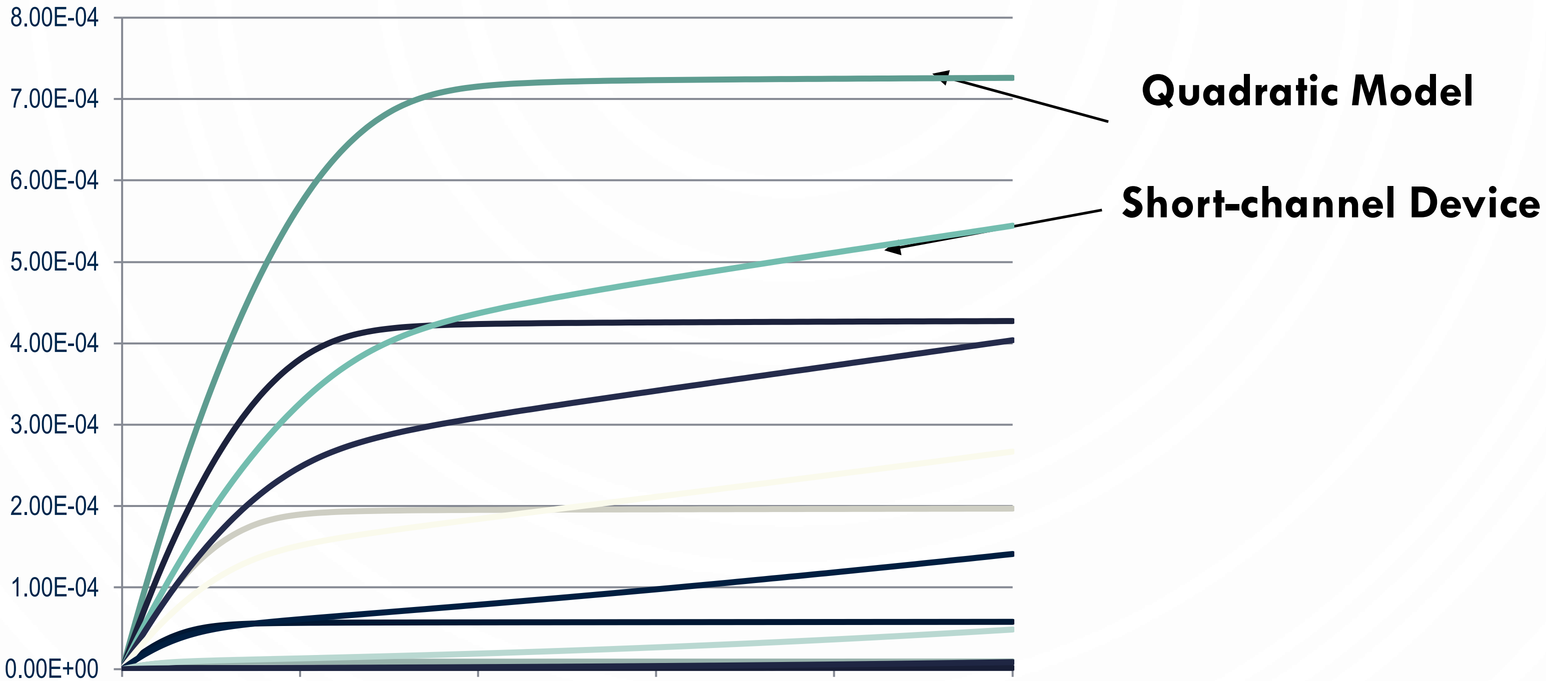
Correct for long channel devices (L ~ µm)

Simulated 32nm Transistor



L = 32nm

Simulation vs. Model



Major discrepancies:

- shape
- saturation points
- output resistances