



PARADIGM[®]
WORKS

Multilayered IP for System Level Verification

DVCon 2004

Dr. Richard Vials

Paul McCloy

Presented by: Dr. Ambar Sarkar



Agenda

- ▶ Reusable Verification IP
- ▶ *eRM* and *eVC*
- ▶ Why layering is important
- ▶ Multi-layered *eVC*
 - ▶ Examples drawn from Paradigm Works
eVCs: Ethernet, PCI-Express
- ▶ Conclusions



Verification IP

- ▶ Off-the-shelf verification code
- ▶ Promotes verification reuse
 - ▶ Horizontal (Between projects)
 - ▶ Vertical (Block to system-level)
 - ▶ Both infrastructure and test scenarios
- ▶ Mostly verifying interface protocol standards
 - ▶ USB, PCI-Express, Ethernet, Bluetooth
- ▶ Most interface protocols are layered



Verification IP

- ▶ Support transaction-based verification strategy
 - ▶ Encapsulate protocol related rules
 - ▶ Encapsulate state machine behavior
- ▶ Measure effectiveness
 - ▶ Collect functional coverage information
- ▶ Help debug
 - ▶ Write log files for interface behavior



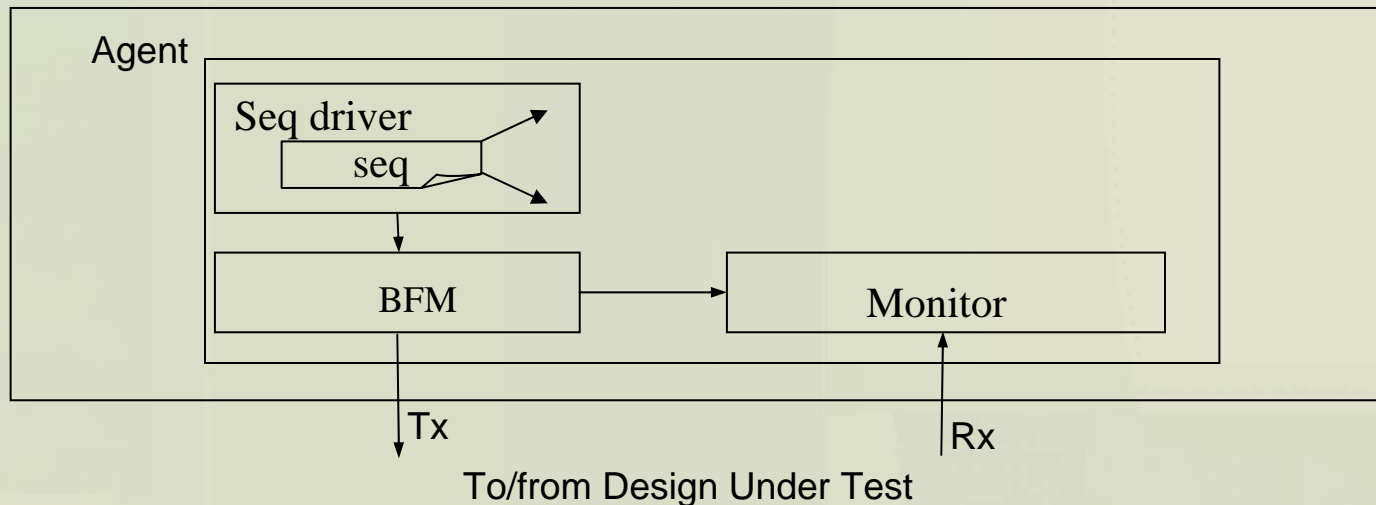
PARADIGM[®]
WORKS

eRM

- ▶ e Reuse Methodology™
- ▶ Develop and distribute Verification Components (eVCs) written in e
 - ▶ Methodology to promote reuse
 - ▶ Verisity's Specman™ tool extensions

Typical eVC

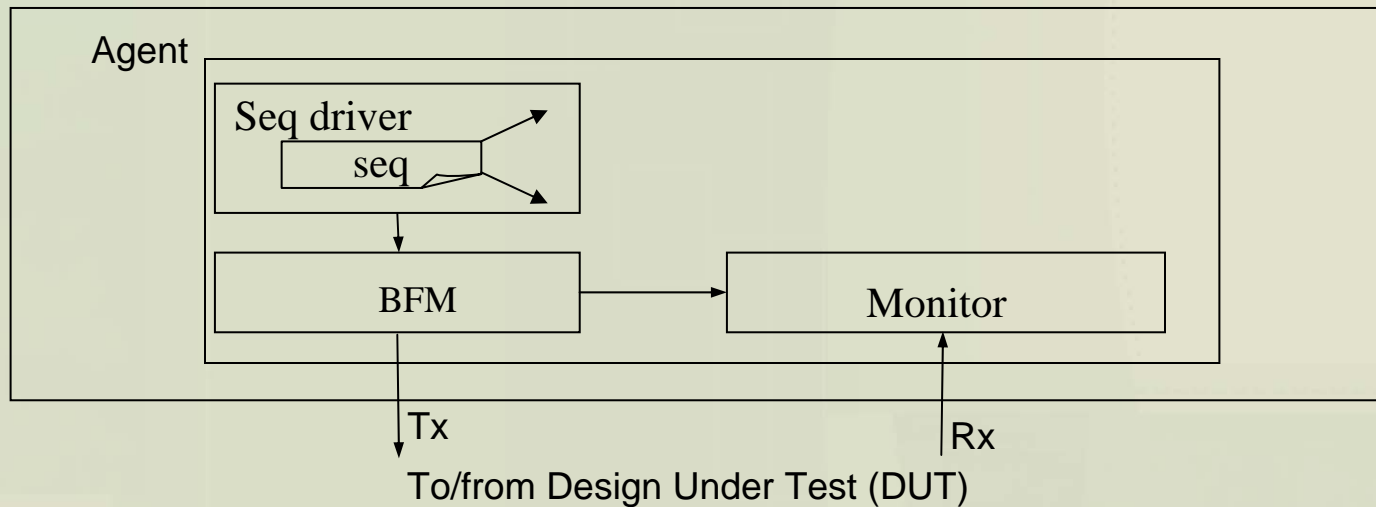
Env



- ▶ Sequence
 - ▶ A reusable traffic scenario, representing a sequence of transactions over time
- ▶ Sequence Driver
 - ▶ Generates constrained-random traffic sequences

Typical eVC

Env

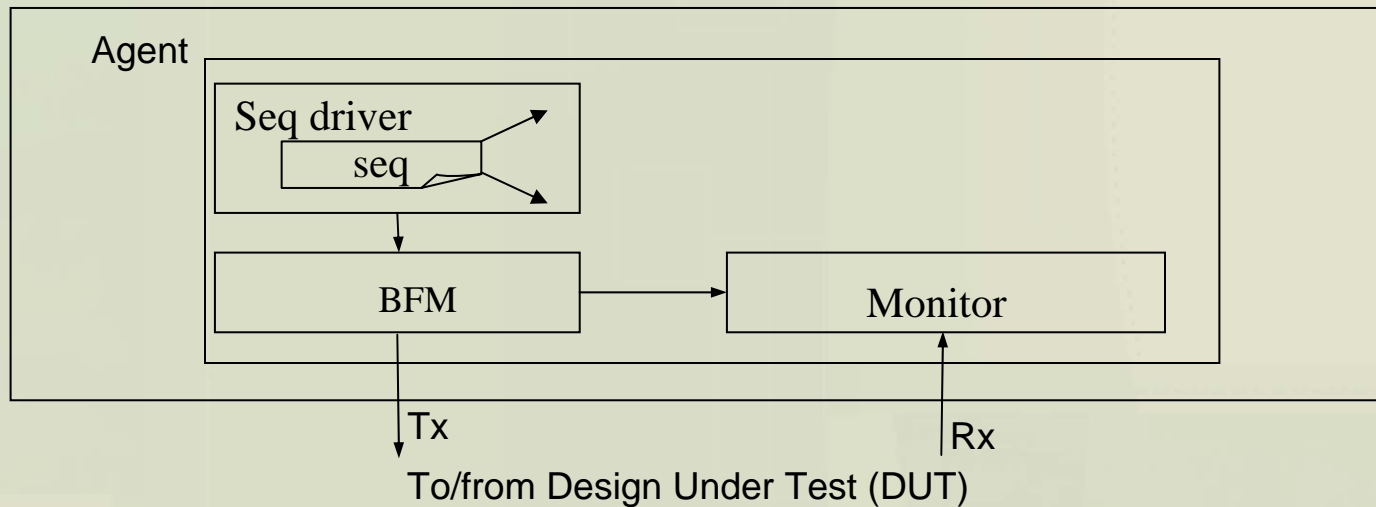


- ▶ BFM (Bus functional model)
 - ▶ Transaction items to DUT signals
- ▶ Monitor
 - ▶ Protocol checking
 - ▶ Coverage collection
 - ▶ Logging



Typical eVC

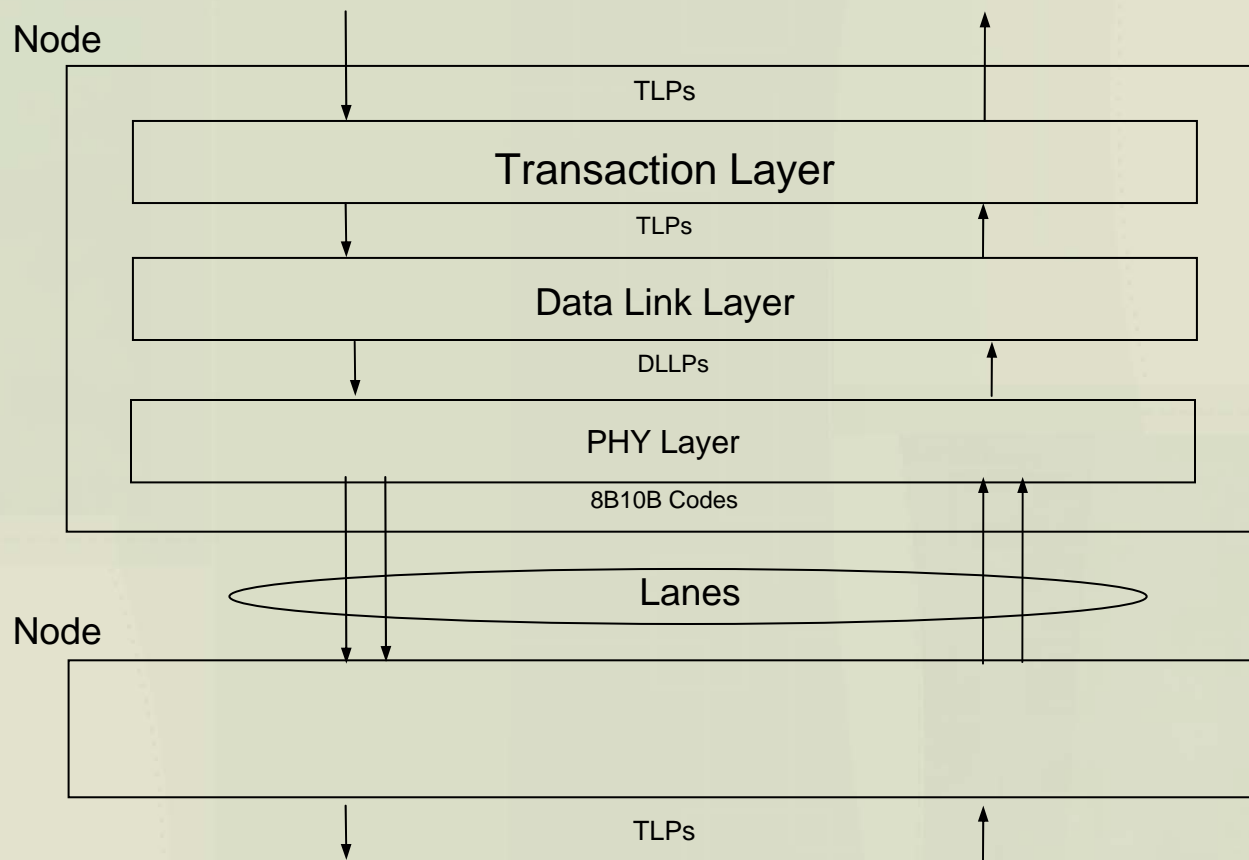
Env



- ▶ Env
 - ▶ Top-level unit of eVC
 - ▶ Mapped to the network
- ▶ Agent
 - ▶ A node in the network



PCI Express Protocol





Single layer approach breaks down

- ▶ Difficult to control lower layer behaviour from high layer data structures
- ▶ Often want to concentrate on lower-layer testing
- ▶ Often need control of behaviour between 'packets'
- ▶ Often need to co-ordinate low and high level behaviour



Enter layering

- ▶ Layering allows separation of control and observability
- ▶ Should break layers at natural boundaries for protocol
- ▶ Layering has only become viable as a result of introduction of **eRM**
- ▶ Use of **eRM** very important to get full advantages of layered approach



Design Goals for Layered eVC

- ▶ Each layer looks/feels like a single eVC
- ▶ Higher-layer drives the lower-layer
- ▶ Out-of-box appear as a single layer at the top-level
- ▶ Override at each level
- ▶ Coordinate to create multi-level test scenarios
- ▶ Allow reactive generation of stimulus



Multi-layered eVC

- ▶ Inter-layer sequence connection
- ▶ Virtual Sequences



Inter-layer sequence connection

- ▶ Convert higher-layer sequence item to lower layer item
 - ▶ Connector
 - ▶ Current eRM approach
 - ▶ Resides within the sequence item
 - ▶ Generates lower layer item based on higher layer
 - ▶ Insufficient

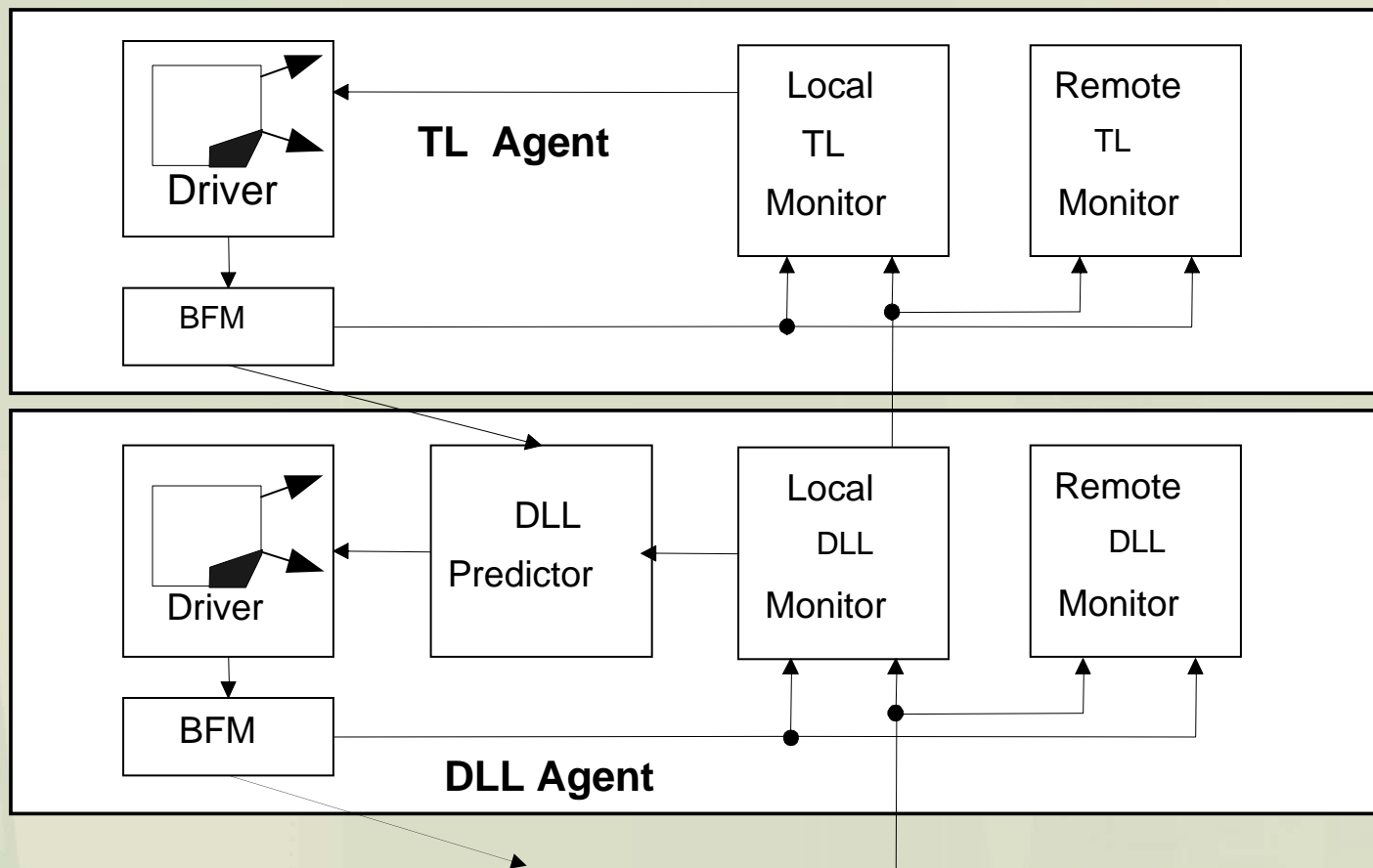


Inter-layer sequence connection

- ▶ Predictor
 - ▶ Resides with lower layer
 - ▶ Can take into account lower layer and protocol operations
 - ▶ Keeps track of state
 - ▶ Extension to eRM
 - ▶ Can provide a reference model for stimuli generation
 - ▶ Behavior may be overridden by sequence interface (Predictor Sequences)



Predictor-based Architecture





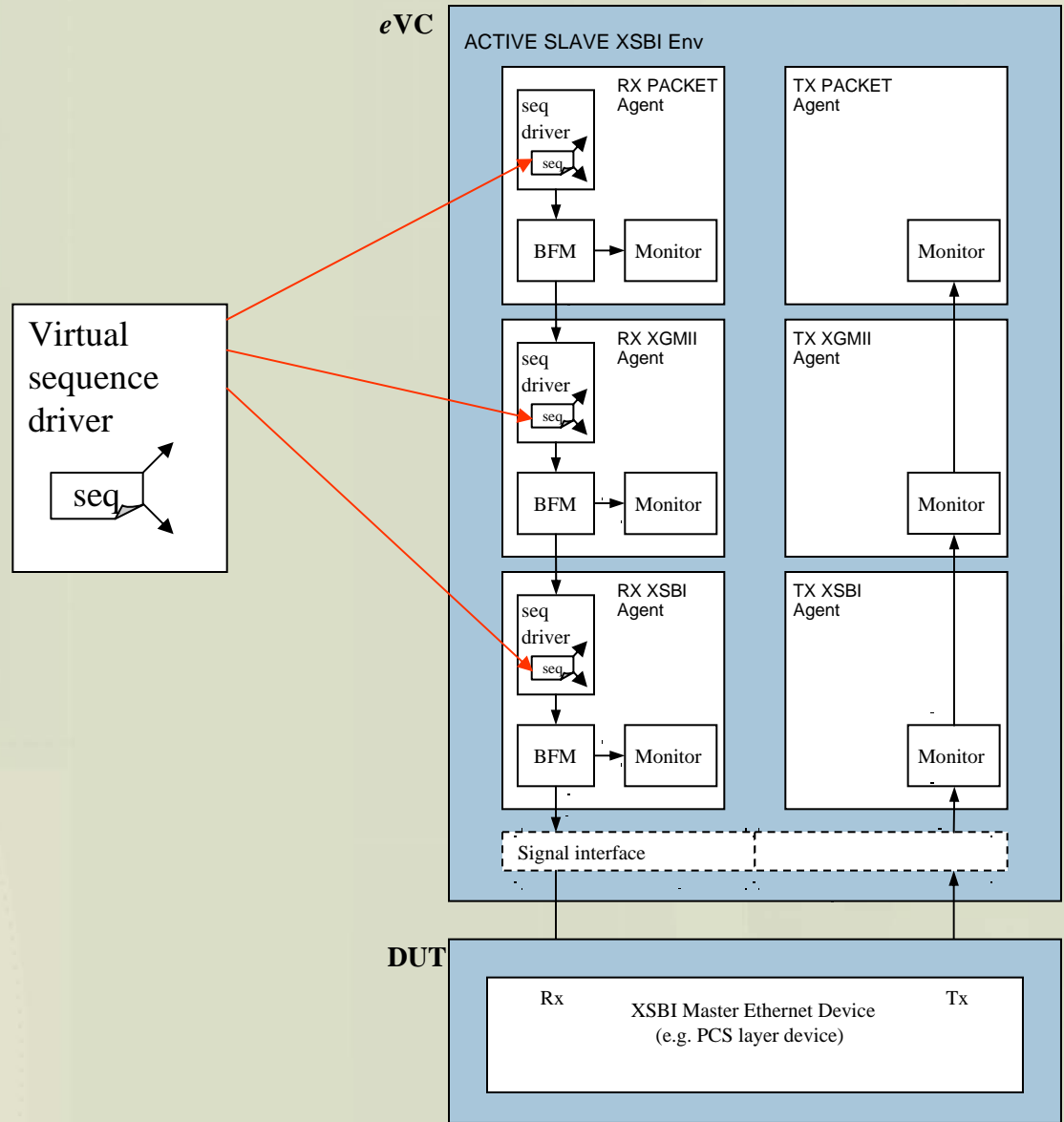
Inter-layer sequence connection

- ▶ Predictor based architecture
 - ▶ Extension to eRM
 - ▶ Can provide a reference model for stimuli generation
 - ▶ Behavior may be overridden by sequence interface (Predictor Sequences)



Co-ordinated testing – Virtual sequences

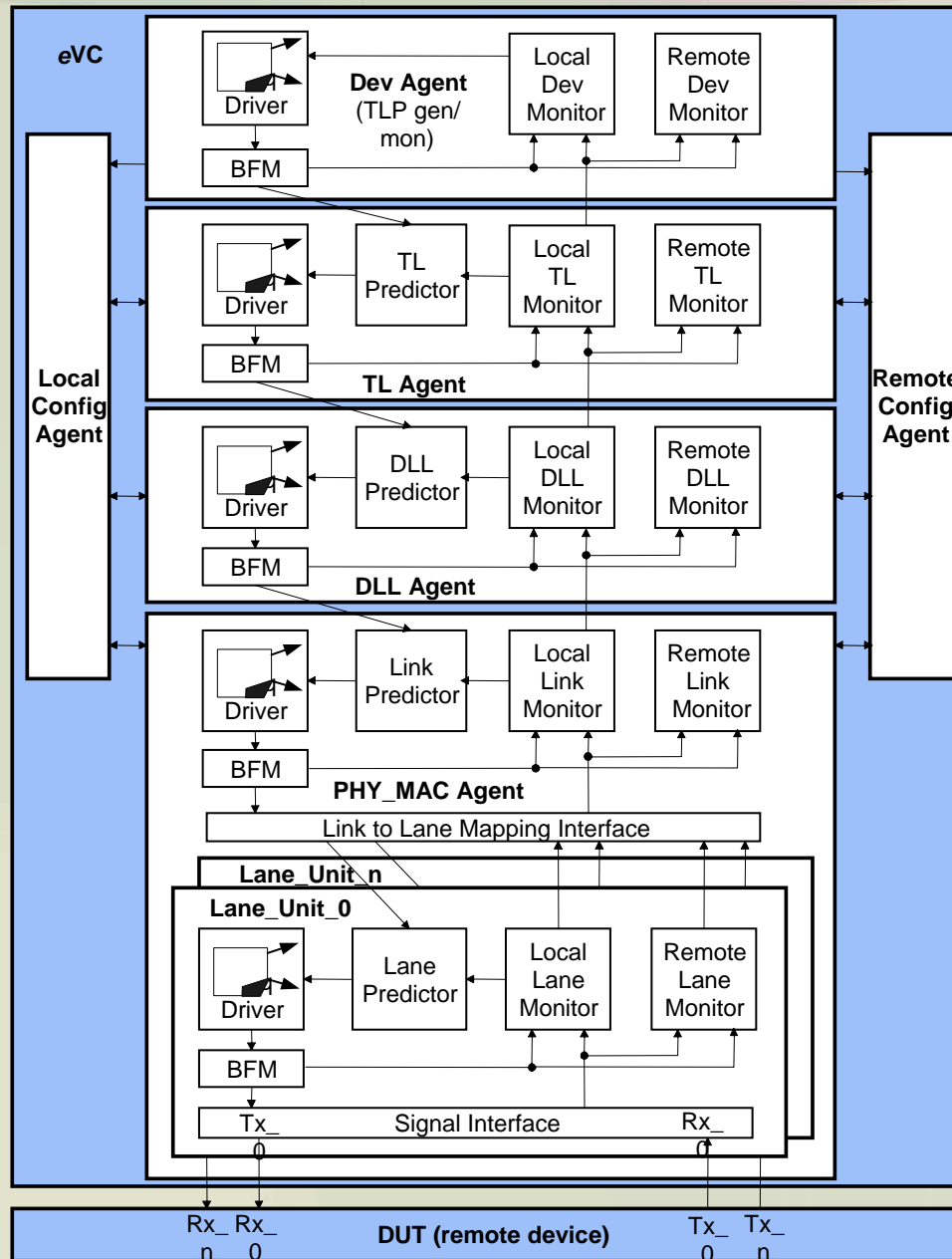
- ▶ Can build virtual sequences to control simultaneous behaviour across multiple layers.
- ▶ E.g. A stream of 100 TL packets with an LCRC error on 43rd packet at DLL





More on Virtual Sequences

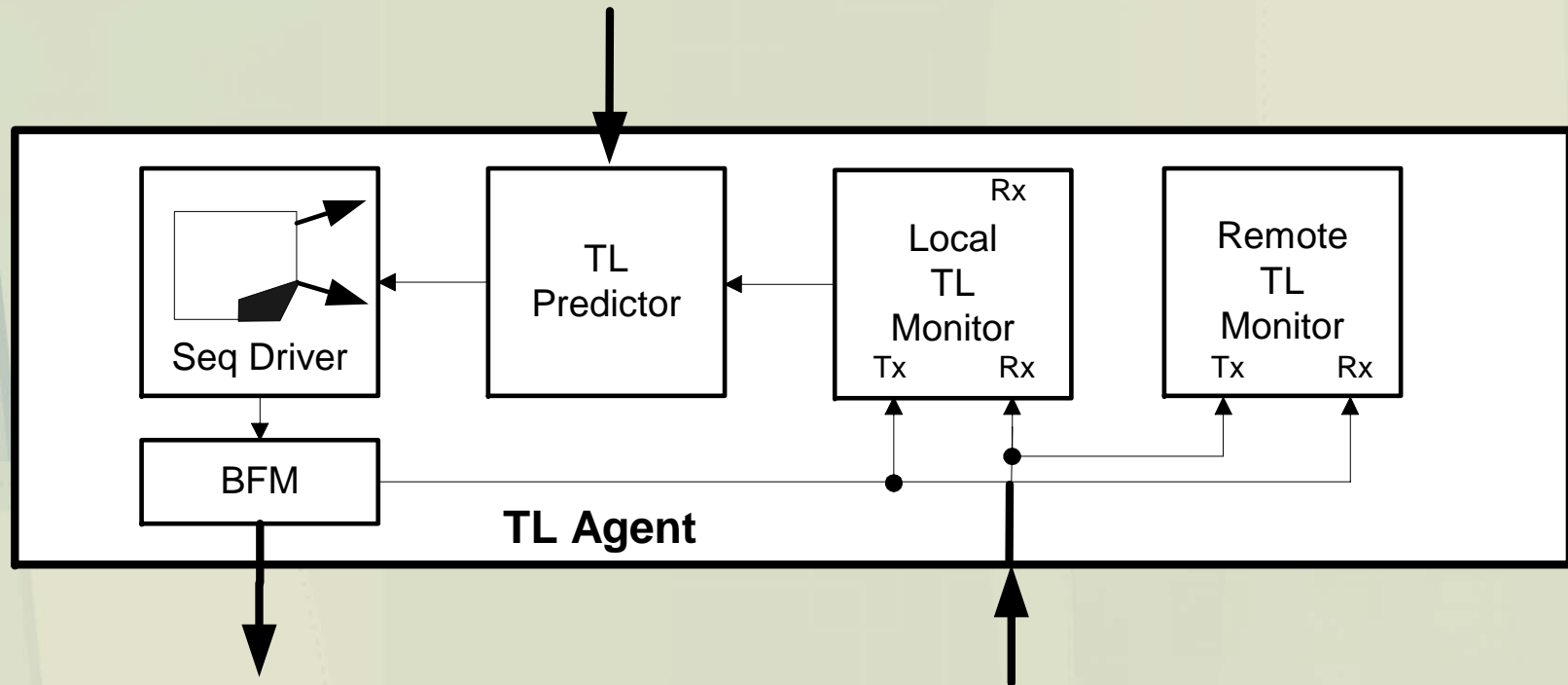
- ▶ Can also co-ordinate behaviour across multiple layers AND multiple **eVCs**.
- ▶ E.g.: Set up DMA transfer on PCI interface to receive Ethernet packet that has error in last XSBI block.



Full Example:

PCI Express eVC

Anatomy of a Single PCIE Protocol Layer





Contents of a PCIe Protocol Layer Agent

- ▶ Sequence Driver
 - ▶ API
 - ▶ Supplied sequence library
- ▶ Predictor
- ▶ BFM drives lower layers
 - ▶ Drives DUT at the lowest level
 - ▶ Extensible for unique DUT interface
- ▶ Monitor
 - ▶ Protocol
 - ▶ Scoreboard interface
 - ▶ Coverage
 - ▶ Compliance
- ▶ Scoreboard
 - ▶ Supplied for TL, Interfaces for other layers



Log Files

- ▶ Log at each layer
- ▶ 3 built-in formats for different verbosity levels
 - ▶ Low, medium, high
 - ▶ User can add new formats, extend existing



Summary

- ▶ Reuse is essential
 - ▶ More complex protocols
 - ▶ More complex SOCs
- ▶ Methodology is essential for reuse
 - ▶ Layered approach
 - ▶ Virtual sequences
- ▶ eRM extensions needed
 - ▶ Predictor-based architecture
- ▶ Ideas validated on commercial eVCs
 - ▶ Ethernet, PCI-Express
- ▶ Further info:
 - ▶ www.paradigm-works.com