PARADIGM™ WORKS

Specman Functional Coverage In the Context of an eVC

Verisity ClubV Fall 2002

Stephen D'Onofrio, Ning Guo

Verification Alliance

Who Am I and Why Am I Here?

- Why use an eVC?
- What is functional coverage/how can it help?
- To talk about functional coverage and how it fits into and enhances an eVC
- What is PCI Express?
- PCI Express eVC and Functional Coverage

The Reuse Mantra

- eVCs are a wonderful advancement in the state of the art for verification
  - Projects start up more quickly
  - Best practices in generation are captured
  - Scenario writing interfaces are consistent
  - Improvements & confidence accumulate
  - There is greater consistency of results
- *e*VC's are generators and/or monitors/checkers
- *e* Reuse Methodology (*e*RM)
- Functional Coverage helps ensure the consistency of results from Random Stimulus

Functional Coverage

- Gathers statistics about specific functional events
- **Design spec -> test plan -> coverage definitions**
- Complements “line/state coverage”
  - 100% functional cov. != 100 % line/state cov.
- Function Coverage Styles
  - Statistics
  - Test Criteria
  - Assertions – Illegal cases
- Function Coverage Types
  - White-box functional coverage
    - In DUT, requires close interaction/cooperation between verification/designer
  - Black-box functional coverage
    - Coverage can be measured at interfaces by watching activity at the ports

Functional Coverage In eVC

- Much can be learned by watching the stimulus and other activity at the interfaces
  - Visibility into eVC ports to the DUT
  - Looks at eVC internals too
  - Does NOT look at DUT internals
  - Gives insight into coverage of generated tests
- Many Functional Coverage definitions already defined
- Structs are extendable to allow addition by user of DUT-specific coverage

Functional Coverage Concepts

- Functional Coverage Elements
  - Groups – A set of items which are updated by the same event
  - Basic Items - One or more coverage signals and/or variables which represent a point-of-interest
  - Basic Buckets –represent a single value or a ranges within an item
- Hit – indicates that an item or bucket test was met
- Grading – hit/goal – quality of functional coverage
- Hole – indicates that a coverage goal was not met
- Extended Functional Coverage Capabilities
  - Transitional Functional Coverage - Item/Bucket changing from one value to another
  - Cross Functional Coverage – Two or more items
- Open loop/closed loop – stimulus feedback

Functional Coverage Examples

```
type packet_type: [MEM, IO, CFG, MSG];

struct example {
   pkt : packet;

   event send_packet;
   cover send_packet is
   {
      //basic item
       item basic_item_packet_type : packet_type = pkt.type;

      //basic range
      item basic_range_len: uint (bits: 12) = pkt.len using
           ranges = {
           range( [16..255], "small");
           range( [256..3k-1], "medium");
           range( [3k..4k], "big");
        },
         illegal = (len < 16 or len > 4000);
        };

      // cross coverage
      cross basic_item_packet_type, basic_range_len using at_least = 10;
};
```

PCI Express Basics

- PCI/PCI-X
  - clock/data skew and power dissipation.
- Serial design “SERDES” replaces parallel bus architecture
  - Point-to-point interconnect – scalable
  - Allows for isochronous data delivery
- Three Logical Layer design
  - Transaction Layer (TL)
  - Data Link Layer (DLL)
  - Physical Layer (PHY)

PCI-Express Transaction Layer

- Provides the interface for software
- Connects to address spaces
- Basic packet types (called TLPs)
  - Requestor/Completers
  - TLP Kinds
    - MEM - Transfer data to/from a memory-mapped location
    - IO - Transfer data to/from an I/O-mapped location
    - CFG - Device configuration/setup
    - MSG - Event signaling mechanism to general purpose
    - MSGAS - Message Request with advanced switching
    - CPL - Completion without Data
- Pipelined full split-transactions
- Credit-based flow control

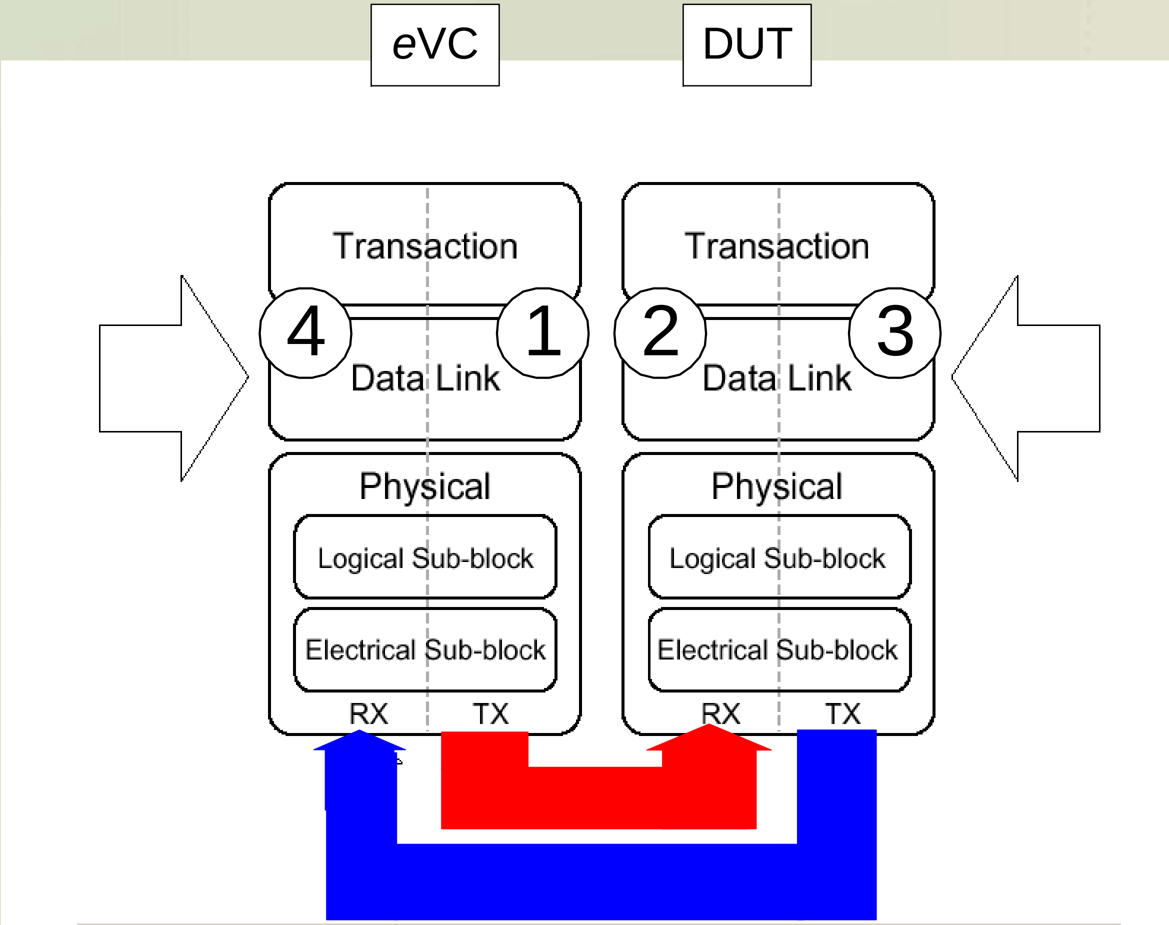
PCI-Express Data Link Layer

- Pass Flow Control Packets
- **Error detection and recovery**
  - **Adds LCRC and Sequence Number to TLP**
  - **Setup/transmit TLP ACK or (negative) NACK**
  - **Includes a retransmit mechanism for packets lost or received with errors**
- Also some unique DLLP types originating here
  - Flow Control initialization
  - Power management/data link state maintained

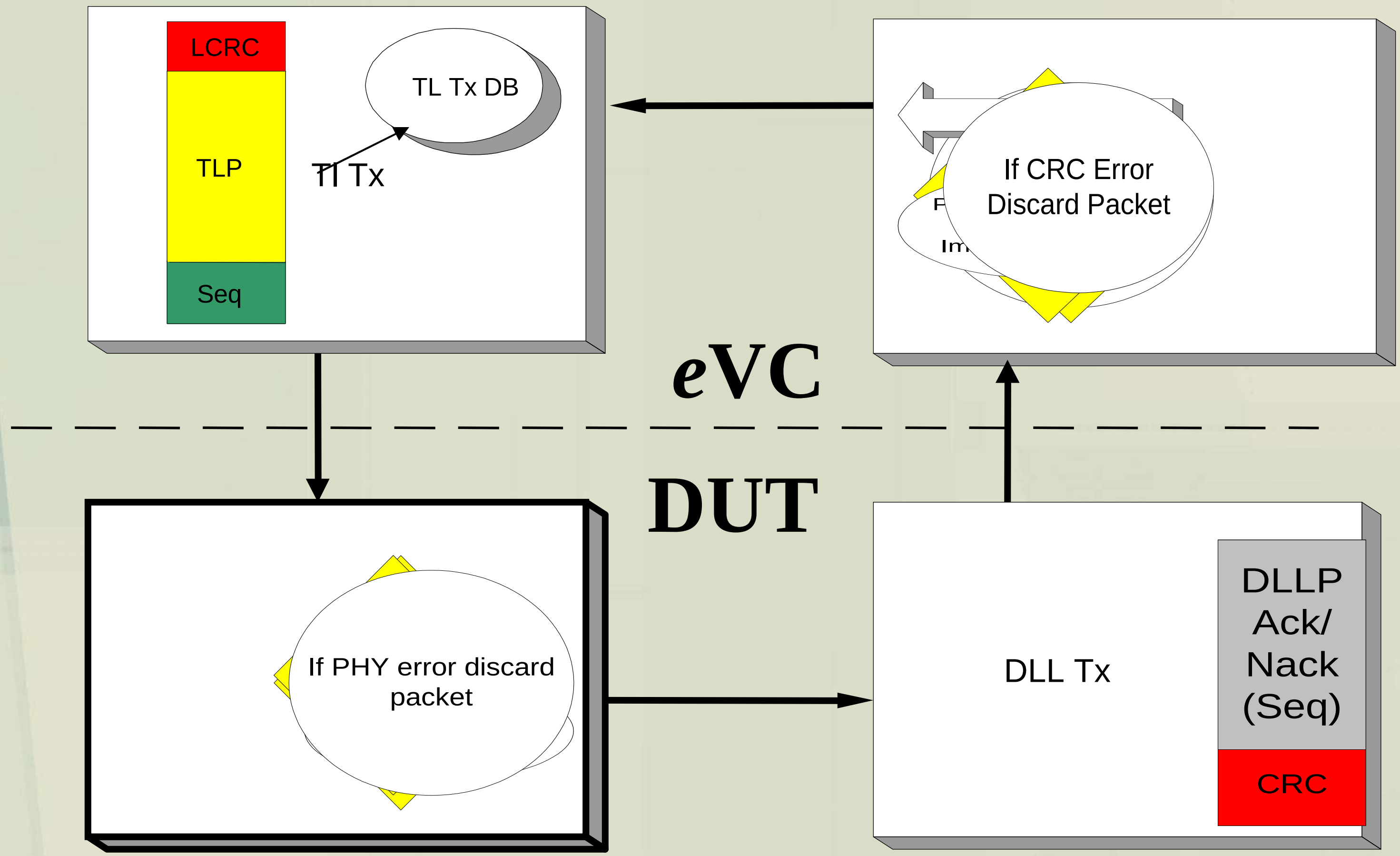
PCI-Express Physical layer

- Link data interchange
- Logical piece and electrical piece
- parallel-to-serial and serial-to-parallel conversion
- Width and Lane mapping (x2.5 GHz)
  - 1x, 2x, 4x, 8x, 12x, 16x, 32x
  - Achieved by striping data across multiple serial links
  - PHY layer does link width and lane sequence training
- 8b/10b encoding
- Scrambling

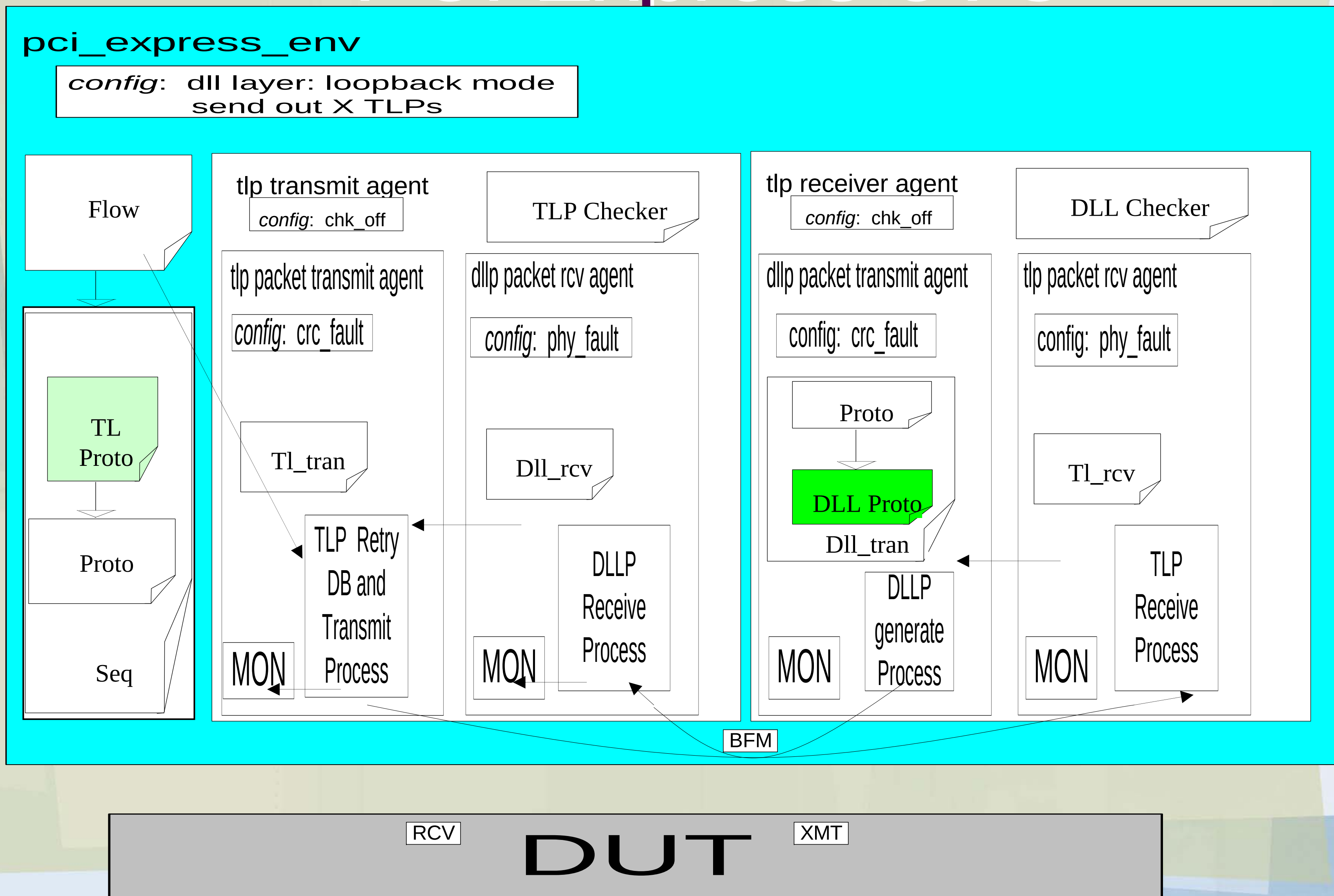
Data Link Layer Communication



DLL Layer Retry Flow



PCI Express eVC



PCI Express “Retry” Functional Coverage

- Basic Coverage
  - Sequence Numbers, CRC Faults, TLP Packet Kind (MEM, IO, CFG, etc.), TLP Transmit Kind (Timer, Nack, Normal), Latency
- Transitional Coverage
  - Back-to-back TLP Packet Kind, CRC Faults, etc.
- Cross Coverage
  - Combinations of above within the same group

PARADIGM™ WORKS

Verification Alliance

Functional Coverage *e*VC Code Example

```
struct tl_ transmit
{
        event start_send;
        cover start_send is {
        item crc_fault : bool = (inject_crc_fault == TRUE);

          item cmd_kind : pciExpCmd_kind = cov_cmd_kind;

          item xmt_kind : TransmitKind = cov_xmt_kind;

          cross crc_fault, cmd_kind, xmt_kind using
          at_least = 10;

        item seq_num : uint = cov_seq_num;

        cross cmd_kind, crc_fault;
        cross cmd_kind, crc_bits_slices;
        cross xmt_kind, seq_num, crc_fault;

        item crc_bits_slices : uint = cov_crc_bit using
        when = (inject_crc_fault == TRUE),
        ranges = {
          range( [0]);
          range( [1]);
          …
```

PARADIGM™ WORKS

Verification Alliance

Cross Coverage Report

0.55 cross__crc_fault__cmd_kind__xmt_kind

548 Hits from 1 tests

Grade	crc_fault	cmd_kind	xmt_kind	Tests	Hits	Goal	Hits / Goal
0	FALSE	IO	–	0	0	10	
0	FALSE	CFG	–	0	0	10	
0	TRUE	IO	–	0	0	10	
0	TRUE	CFG	–	0	0	10	
0.20	TRUE	MSG	RE_TRANSMIT_NACK	1	2	10	
0.20	TRUE	MSGAS	RE_TRANSMIT_NACK	1	2	10	
0.30	TRUE	CPL	RE_TRANSMIT_NACK	1	3	10	
0.50	FALSE	MSG	RE_TRANSMIT_NACK	1	5	10	
0.50	TRUE	MEM	RE_TRANSMIT_NACK	1	5	10	
0.60	FALSE	MSGAS	RE_TRANSMIT_NACK	1	6	10	
0.70	TRUE	MSG	RE_TRANSMIT_TIMEC	1	7	10	
0.80	TRUE	MSG	TRANSMIT	1	8	10	
0.90	FALSE	MEM	RE_TRANSMIT_NACK	1	9	10	
1.00	FALSE	CPL	RE_TRANSMIT_NACK	1	12	10	
1.00	TRUE	MSGAS	TRANSMIT	1	12	10	

Summary

- Functional Coverage within an eVC is valuable
- Functional Coverage within an eVC needs to be extendable
- Functional Coverage can be used to aid verifying many other areas of the PCI Express

The End

- Contact Information:

  www.paradigm-works.com

  Paradigm Works

  1 Corporate Drive

  Andover, MA 01810

  stephen.donofrio@paradigm-works.com