CPU PACKET T-ADAPTOR

Valens VLSI

Verification Plan

Revisions History

Rev	Description of Change	Author	Approver	Date
0.1	First Version	Batzion Nissenboim		

Related Documents

Doc Number	Doc Name
	CpuPktTadp (Spec)
	HDBaseT_Spec_Ver2.0
	Verif Spec – vl_TOKEN

Contents

1. List of Figures

No table of figures entries found.

2. List of Tables

No table of figures entries found.

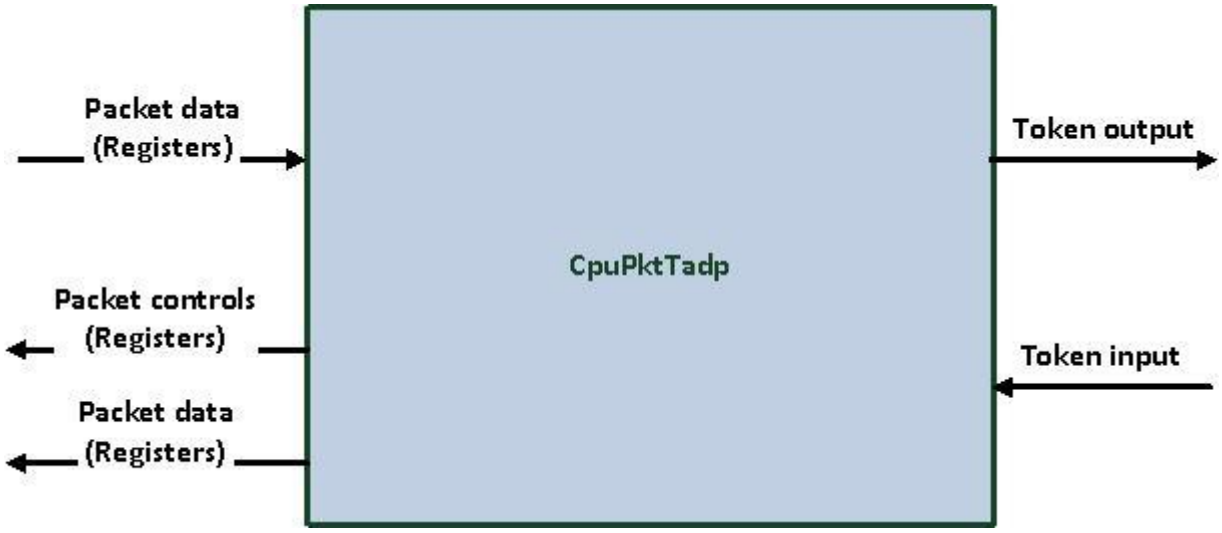
3. General Description

The CpuPkt_Tadp generates token from native data, and the opposite – parse a token and write native data into FIFOs.

Main features:

- CpuPkt_Tadp block contains RX&TX sub modules, RX and TX are independent.
- RX - Writing data to registers and generate tokens of data to send out.
- TX – Gets tokens and send out packets of controls and packets of registers.
- Reading data at TX flow is done from two FIFOs one of controls and one of the data it-self.

3.1. DUT overview



3.2. INTERFACES:

Data to TOKEN:

Inputs:

Name	Width	Description
clk	1	System clock. Drives the internal logic (flops).
reset_n	1	System reset. Resets the internal logic (flops).
scan_enable	1	Enables the scan logic
scan_mode	1	Select between 2 reset signal for SCAN and regular operation.
Registers		
gen_cpu_token_ready	1	Packet generator output – token.
cpu_packet_data	8	Packet generator data input (to be token later on).
cpu_packet_data_go	1	Asserted every time Cpu_Packet_Rdy register is written and writes the data to the CPU packet generator.
cpu_type	4	Packet type for CPU generated packet.
cpu_future	1	Determines if the generated CPU packet is future.

cpu_sid	8	Session ID for CPU generated packet.
cpu_prot	1	Determines if the generated CPU packet is protected.
cpu_ext_data	2	Generic extended information of the CPU packet.
cpu_ext	1	Extension field of the CPU packet.
cpu_sync	2	Bit 0: Sync Start bit - If set to 1, indicates a Nibble Stream Start Sync point bit 1: Sync End bit - If set to 1, indicates a Nibble Stream End Sync point.
cpu_pkt_len	4	Describes the number of sub-packet payload tokens.
cpu_lsb_first	1	Determines if the CPU packet data will generated with LSB first
cpu_pkt_go	1	Asserted every time Cpu_Packet_Ctrl "register" (bit[31]) is written and transmits data from CPU packet generator.
cpu_qual	2	Transfer quality code of the CPU packet generator.

Outputs:

Name	Width	Description
gen_cpu_token_out0	19	CPU packet generator transmitted token.
gen_cpu_token_out1	19	CPU packet generator transmitted token.
gen_cpu_token_valid	1	Indication that CPU packet generator data is valid.
cpu_mech_ready	1	When set, CPU packet generator is ready to receive packet_go → ready to transmit a packet.
Registers		
cpu_packet_data_rdy	1	Indication that CPU packet generator is ready for write command.

TOKEN to Data:

Inputs:

Name	Width	Description
clk	1	System clock. Drives the internal logic (flops).
reset_n	1	System reset. Resets the internal logic (flops).
scan_enable	1	Enables the scan logic
scan_mode	1	Select between 2 reset signal for SCAN and regular operation.
cpu_token_in0	19	Token input to parser.
cpu_token_in1	19	Token input to parser.
cpu_token_valid	1	Token valid input to parser.
Registers		
CpuCtrlRead	1	Asserted every time CpuCtrlDataRead register is written and executes read command from IF (controls) FIFO.
CpuParseEnable	1	Enables the parser.

Cpu_IgnoreCRC	1	Ignore CRC error.
CpuPacketRead	1	Asserted every time CpuPacketDataRead register is written and executes read command from Data FIFO.

Outputs:

Name	Width	Description
cpu_token_ready	1	Parser output – indicating that token is ready.
Registers		
CpuCtrlData	22	Controls' data output from FIFO – holds the following data: {cpu_sid_out, cpu_len_out, cpu_out_ext, cpu_out_ext_info, cpu_sync_out, cpu_crc_out}
CpuCtrlVld	1	Indication that controls data is valid.
CpuPacketData	8	Packet data output from FIFO.
CpuPacketVld	1	Indication that data is valid.

3.3. Clock and resets domains:

Add a diagram!!!

Note:

CpuPktGen has one clock domain – sys_clk, and all flops work at positive edge of this clock.

Table clock:

clk	The System clock for both RX and TX

4. Verification

The CpuPkt_Tadp Verification IP is responsible for simulating the transferring CPU data packets over HDBaseT link.

4.1. Verification Overview

- The CpuPkt_Tadp verification environment (VE) has two main data paths:
  RX – Native data to token.
  TX – Token to native data.

- The VE contains three agents, two for tokens: RX – token_out, and TX – token_in, and a RIF's agent for the data information and configurations.

- The native data side injects and collects to/from the DUT registers with data, and the Token side injects and collects to/from the DUT t-packets.

- Each path – TX and RX - has its own reference model (RM).

- The RIF's agent injects data into the registers. The DUT reads the data from the registers.
  The reference model RX reads the registers too and creates a token from the data.
  After the DUT processes and outputs a token, the monitor of Agnt_Token_out collects the data, creates items of t-packets and passes them to the reference model RX.
  The reference model RX passes the tokens to the scoreboard RX to compare.
  On the other hand-
  The Agnt_Token_in injects data into the DUT.
  The monitor of the Agnt_Token_in collects the token, creates an item of t-packet and passes it to the reference model TX that return them to native data and insert the data into a local map of registers.
  After the DUT processes the token and write the data to the registers, the RIF agent read the data and passes it to the reference model TX.
  The reference model TX passes the data from his local registers and from the RIF registers to the scoreboard TX to compare.
  The Agnt_Rif also configures the DUT.

4.2. Functional Description

4.2.1. RX data flow:

The flow contains two agents:
1. RIF's agent.
2. Token_out agent.

	Flowchart	Description
1.	Configuration	The VE configures all registers.
2.	Valid	At first waiting for valid register to be set – wait for the interrupt.
3.	Read data	Monitor of RIF agent read packets of data from register Cpu_Packet_Ctrl and send it to the RM.
4.	Ready	Wait for cpu_packet_data_rdy – indication of writing all data bytes.
6.	Go	Write to cpu_packet_go that enabling to generate a token and transmit the data token out.
7.	Collect data token	Monitor of Agnt_Token_out collects data, creates items of data t-packets, and send to the RM.

8.	Compare the data	Compare the data Token vs native data.
9.	Collect data in a loop	The VE continue to collect more data according to registers assertion.

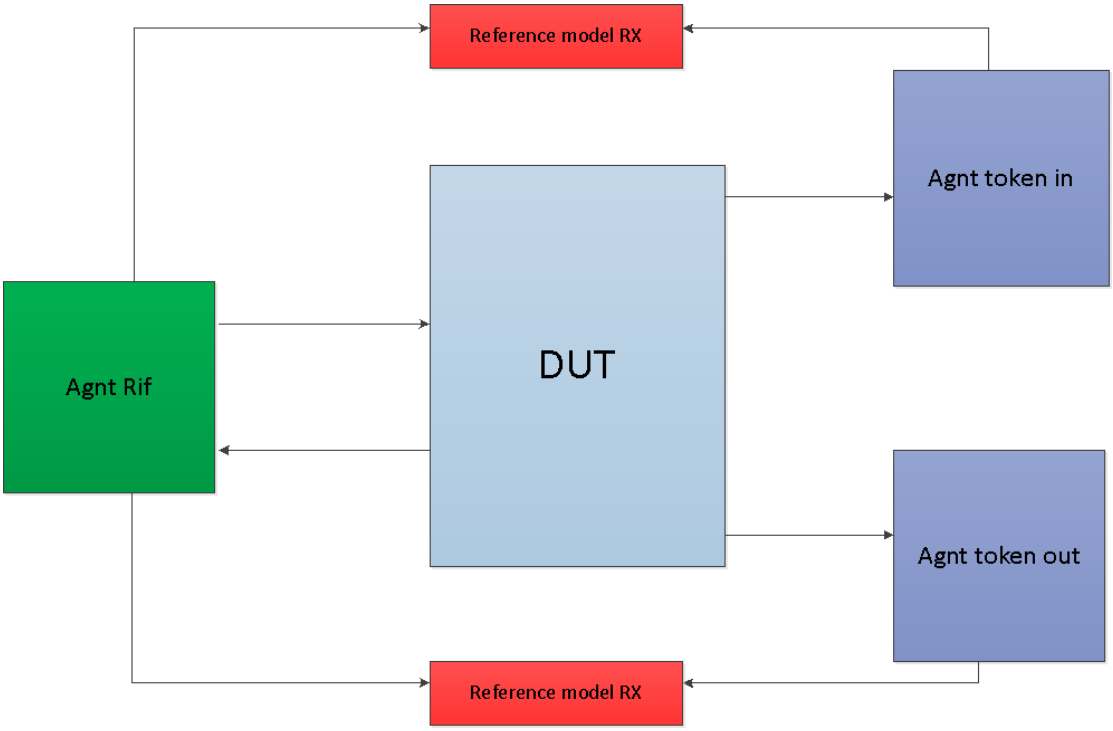
4.2.2. **TX data flow:**

The flow contains two agents:

1. Token_in agent.
2. RIF's agent.

	Flowchart	Description
1.	Configuration	The VE configures all registers.
2.	Collect data token	Monitor of Agnt_Token_in collects data, creates items of data t-packets and send to the RM.
3.	Collect control	Monitor of RIF agent read from control FIFO, creates an item and send it to the RM.
4.	Valid	Check validation, If the controls are valid – continue to stage 5, If not go back to stage 3.
5.	Collect data	Monitor of RIF agent read from data FIFO, creates an item and send it to the RM.
6.	Valid	Check validation of each data.
8.	Compare the data	Compare the data Token vs native data (include con-trols).
9.	Collect data in a loop	The VE continue to read more data up until the length field according to the registers.

4.3. Environment Architecture



4.4. Verification Components

- Agent token in
- Agent token out
- Agent RIF
- Reference model TX
- Reference model RX
- Scoreboard TX
- Scoreboard RX

More specifically:

4.4.1. Token in/out (Token EVC):

BFM:

In- Toggle according the Token protocol

Out- Toggle ready.

Monitor - Collect the data and creates a token item.

Pass the item to the RM.

Sequence driver -

Main Item –

Item name	Parameters
All fields	All options

4.4.2. RIF (RIF EVC):

BFM:

In- Toggle according the Token protocol

Out- Toggle ready.

Monitor - Collect the data and creates a token item.

Pass the item to the RM.

Sequence driver -

Main Item –

Item name	Parameters
Addr	Address of the register
Data	Value of the register

4.4.3. CFG:

Item name	Parameters
CpuParseEnable	1/0
Cpu_IgnoreCRC	1/0
cpu_type	1-8
cpu_future	1/0
cpu_prot	1/0
cpu_sync	First bit: 1/0 Second bit: 1/0
cpu_sid	0-255
cpu_ext	1/0
cpu_ext_data	0-3
cpu_pkt_len	0-15
cpu_lsb_first	1/0
cpu_qual	0-3
cpu_prio	1-3
Ready of FIFOs	Set to 1 for now.

4.4.4. RM

RX RM

The RX RM gets several kinds of items:

- Data RIF items (from the RIF agent), convert them to Token items according to the to the configuration registers.
- Token data items (from the Token_data_out agent).

There is a data scoreboards (SB) for comparing data tokens vs data token.
The RX RM sends the token data items and RIF's items after conversion to tokens to the data SB to compare.

TX RM

The TX RM gets several kinds of items:

- RIF's items (from the RIF's agent).
- Token data items (from the Token_data_in agent), convert them to RIF's items – local registers map.

There is a data and controls scoreboards (SB) for comparing native data registers vs native data registers.

4.5. Sequences

RIF base sequence – Random types of configurations.

Data Token base sequence – basic Token configurations.

4.6. Checkers

4.6.1. General checkers list

The checker name	Description	Location	priority
Data correct	The RM is compared the data native data vs data Token according the configurations: -Data: All registers values. -Token: All options of token configurations.	RM	High
Idle mode/got reset	Token Ready out signal is on '1' The status registers are on default values.	RM	High
Loopback mode	Check that on loopback mode data does not go out but going back.		Low

4.6.2. Token checker list

The checker name	Description	Location
Generate token	Check that Tokens are not coming out before go command.	Monitor
CRC error pkt	• When the ignore CRC bit is on, ignore bad CRC indication use the data pkts. • Else ignore the packet.	RM
Token space transaction	Check default space.	RM
Check the Token trans values	Check according to the configurations.	RM
Token_out back pressure mode	Check that the RTL does not stuck (Data can be lost).	General

4.6.3. Native Data checker list

The checker name	Description	Location
FIFO validation	Check that data is not coming out from FIFO without valid.	Monitor
Data validation	Check that data is not collected with no valid high.	Monitor

4.6.4. CFG checker list

The checker name	Description	Location
Read only status registers	Correct values and behave	

4.7. Errors, Interrupts, Counters and Statuses

4.8. Coverage

4.8.1. Basic Coverage

The coverage name	Description	Priority
System clock	All system clock options : Only 500 MHz	
Token t-ready special scenarios	• When there was not go command • Before enable Token out transmissions	Test
High performance scenarios	• High on Token in • High on Data in	Test

4.8.2. Token and Native data scenarios - relevant to in or out data path

The coverage name	Description	Priority
System clocks	500 Mhz:	
Data sample	• 100 packets with data '0' • 100 packets with data '1' • 100 packets with data 5555/AAAA by turns • The data sample is 0xFFFFFFFF and 0x0 by turns	Test
SID values – 8 bits	• 5 parts between 0x0-0xFF • 0x0 • 0xFF	
LSB first – 2 bits	• 1- LSB is first • 0 – MSB is first	
T-ready - continues high time and continues low time	• T-ready is toggle for 100 cycles. • 10 parts of 1-200 cycles for high and 0 cycles for low • 10 parts of 1-200 cycles for low and 0 cycles for high • Long time(about 1us-10us) for low and 10-20 cycles for high	
CRC errors	• CRC error on data.	
Token types	• Only tokd8 (Pam 4)	
Packet type	• 0 – 15 • Mixed types and sets of same type. (Test)	
Future token	• 1 – Future packets. • 0 – Not a future packet. • Mixed. (Test)	
Protected tokens	• Protected tokens and not protected tokens according to packets type.	

Extended info	• Packets with ex info. • Packets with no extended info.	
Extended data	• 0 – 3	
Sync start and end	• Packets with sync start. • Packets with sync end. • Packets with no sync start/end.	
Payload length	• Length 1 • Length 15 • Length between 1 and 15.	
Priority	• Priority 3 only.	
Quality	• Changes values of quality between 1 and 3	

4.8.3. Native Data scenarios – relevant to in or out data path

The coverage name	Description	**Priority**
Reset	Reset from (At least once of each case): • Overrun FIFOs (High indication) – Depth of data FIFO is 64.	Test

4.8.4. Configuration

The coverage name	Description	Priority
CRC ignore	• 1 – Ignore indication of bad CRC and use the data. • 0 – Ignore data tokens with bad CRC indication.	
Parser enable	• Always 1 • 0 (No need to collect coverage. If 0 – No activity on the chip…)	

4.8.5. ERROR scenarios – basic check the design going on

The coverage name	**Description**	**Expected behavior**	**Priority**
Overflow scenarios (Data out)	• From big traffic • From data high threshold		

4.9. Tests

Basic test

Error tests

Performance test

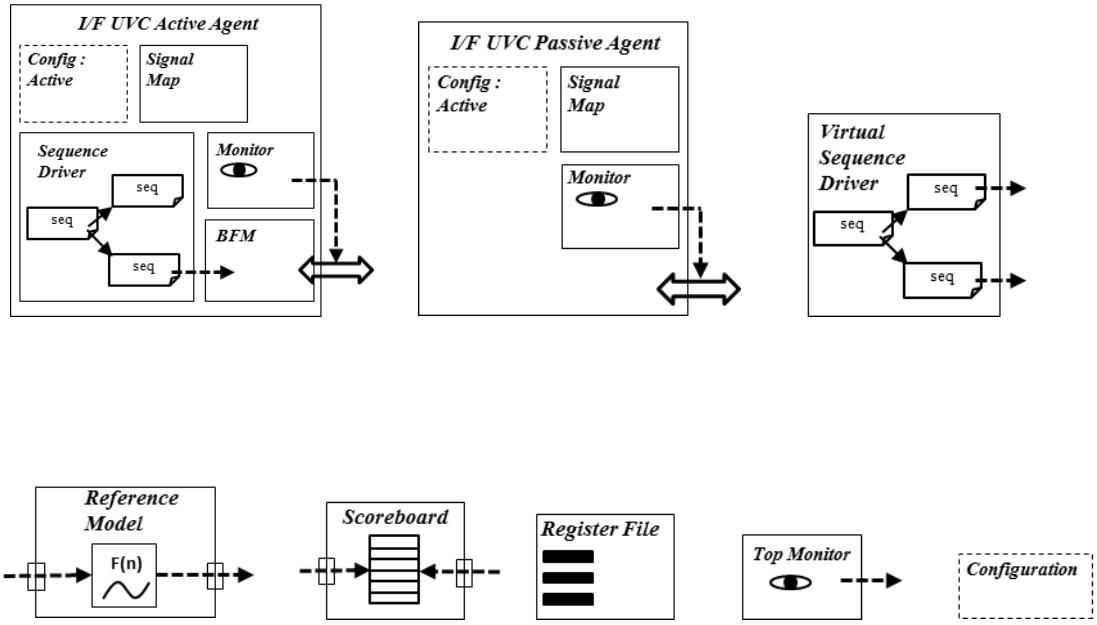
Data in path expected performance – 16 us back to back.

Data out path expected performance – 16 us back to back**.**

4.10. Assumptions

4.11. Open issues

- System clocks???
- Overflow/Underflow behavior?


I/F UVC Active Agent
Config : Active
Signal Map
Sequence Driver
seq
seq
seq
Monitor
BFM
I/F UVC Passive Agent
Config : Active
Signal Map
Monitor
Virtual Sequence Driver
seq
seq
seq
Reference Model
F(n)
Scoreboard
Register File
Top Monitor
Configuration