Packet Telemetry Encoder (PTME)

AMBA AHB/APB Validation Report

PTME-005-01
Version 0.3
November 2003

Stora Nygatan 13
411 08 Göteborg
Sweden

tel +46 31 802405
fax +46 31 802407
www.gaisler.com

1 INTRODUCTION

1.1 Scope

The PTME model is based on the *European Space Agency* (ESA) *Procedures, Standards and Specifications* (PSS) and the CCSDS recommendations. At the time of writing there were no documents available from the *European Cooperation for Space Standardization* (ECSS).

This document describes the validation of the AMBA AHB and APB interfaces of the *Packet Telemetry Encoder* (PTME) VHDL model.

1.2 Objective

The objective of the validation efforts have been to validate the newly developed AMBA AHB and APB interfaces that are integrated in the PTME VHDL model.

1.3 Applicable documents

AD1	Packet Telemetry Encoder (PTME) VHDL Model - Data Sheet, PTME-001-01, Version 0.6, November 2003, Gaisler Research
AD2	Spacecraft-Controller-on-a-Chip adapted Packet Telemetry Encoder VHDL Model (SCoC_PTME) - Data Sheet, PTME-002-01, Version 0.5, May 2003, Gaisler Research
AD3	Packet Telemetry, CCSDS 102.0-B-5, Issue 5, November 2000
AD4	Telemetry Channel Coding. CCSDS 101.0-B-5, Issue 5, June 2001
AD5	AMBA™ Specification, Rev 2.0, ARM IHI 0011A, 13 May 1999, Issue A, first release, ARM Limited
RD1	WILDCARD™ Reference Manual, 12416 - 0000, revision 2.3, Annapolis Micro Systems Inc., Annapolis, USA

1.4 Applicable VHDL source code

AD6	Packet Telemetry Encoder (PTME) synthesizable VHDL model, version 0.8a, July 2003, *ptme_lib.vhd*
AD7	SCoC Packet Telemetry Encoder (SCoC_PTME) synthesizable VHDL model, version 0.4, May 2003, *scoc_c.vhd* and *scoc.vhd*
AD8	AMBA synthesizable VHDL package, version 0.5, February 2002, *amba.vhd*
AD9	Validation version of Packet Telemetry Encoder (Validate_PTME) synthesizable VHDL model, version 0.2, Nov 2003, *validate_c.vhd* and *validate.vhd*

1.5 Acronyms and abbreviations

AHB	Advanced High-performance Bus (AMBA interface)
AMBA	Advanced Microcontroller Bus Architecture ™
APB	Advanced Peripheral Bus (AMBA interface)
ASM	Attached Synchronisation Marker
BAT	Bandwidth Allocation Table
CCSDS	Consultative Committee for Space Data Systems
CD	Clock Divider
CE	Convolutional Encoder
CI	Configuration Interface
ECSS	European Cooperation for Space Standardization
ESA	European Space Agency
FPGA	Field Programmable Gate Array
NRZ	Non Return to Zero
PAPB	PacketAPB
PSR	Pseudo Randomiser
PSS	Procedures, Standards and Specifications
PA	PacketAsynchronous
PP	PacketParallel
PW	PacketWire
RSE	Reed-Solomon Encoder
SP	Split-Phase
SSRAM	Synchronous Static Random Access Memory
TE	Turbo Encoder
TM	Telemetry
TME	Telemetry Encoder
VCA	Virtual Channel Assembler
VCB	Virtual Channel Buffer
VCE	Virtual Channel Encoder
VCM	Virtual Channel Multiplexer

2 PTME OVERVIEW

The Packet Telemetry Encoder (PTME) VHDL model comprises several encoders and modulators implementing the Consultative Committee for Space Data Systems (CCSDS) recommendations and the European Space Agency (ESA) Procedures, Standards and Specifications (PSS) for telemetry and channel coding. The Packet Telemetry Encoder (PTME) VHDL model comprises the following blocks:

- Telemetry Encoder (TME)
- Reed-Solomon Encoder (RSE)
- Turbo Encoder (TE)
- Pseudo-Randomiser (PSR)
- Non-Return-to-Zero Mark encoder (NRZ)
- Convolutional Encoder (CE)
- Split-Phase Level modulator (SP)
- Clock Divider (CD)

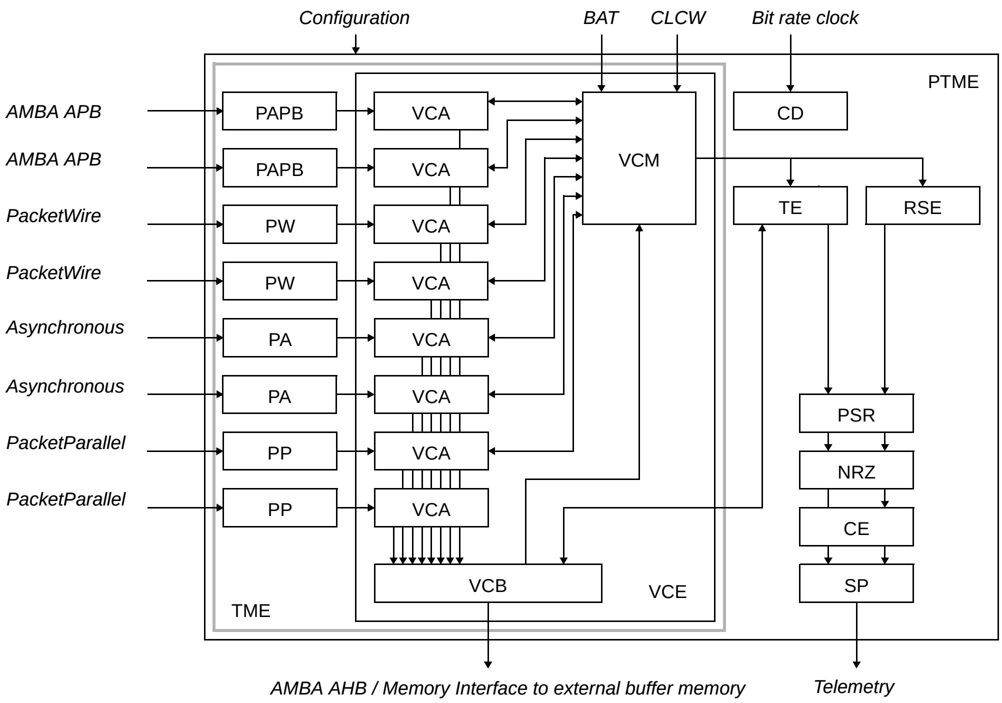

Figure 1: *PTME block diagram (example with eight Virtual Channels)*

3 VALIDATION VEHICLE

A WildCard™ development board from Annapolis Micro Systems has been used as a vehicle for the validation. The WildCard is a PCMCIA card with the following features:

- Type II PC Card
- 32-bit CardBus interface with multichannel DMA controller
- programmable clock generator
- single processing element: Virtex™ XCV300E -6
- two independent memory ports connected to two 10 ns SSRAM devices
- two independent 15 pin I/O connectors
- Windows®2000 CardBus driver or Linux® CardBus driver

The WildCard comes with templates and example VHDL designs and software routines. It is only possible to access the WildCard via a specific Application Programming Interface (API). For the Virtex-E based version, used for the validation, it is not possible to read out the configuration of the FPGA, the user can only program it.

There were several inconsistencies in the documentation and the VHDL code provided with the WildCard, which have been reported to the vendor. The interrupt routines API did not work properly and user controlled polling was instead used for all communication with the card. More seriously, the API does not allow accesses with acknowledge to the card, which made it impossible to access directly the external memory via an arbitrated AMBA AHB bus. The memory interface to the left SSRAM bank had to be modified to allow byte writes.

Due to the fixed placement of interface pins etc., it was not possible to utilise more than about 75% of the Virtex device, else the place and route tool would not manage to place the design.

All software is based on the WildCard API and has been developed with CygWin gcc compiler.

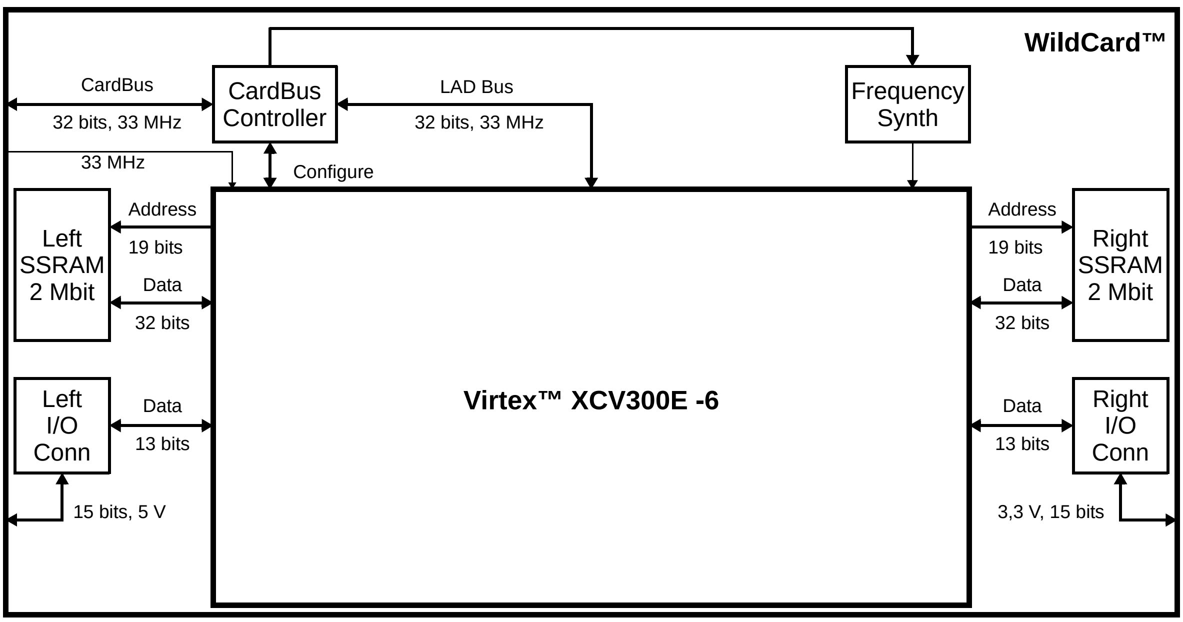

Figure 2: *WildCard architecture*

4 VALIDATION CONFIGURATION

The validation of the PTME AMBA interfaces has been done using an implementation of the PTME in which all interfaces are based on AMBA, not only those integrated in the PTME VHDL model. The validated implementation is based on the Spacecraft-Controller-on-a-Chip adapted Packet Telemetry Encoder VHDL Model (SCoC_PTME), described in AD2. The integration of the PTME in the WildCard validation vehicle is shown in figure 3.

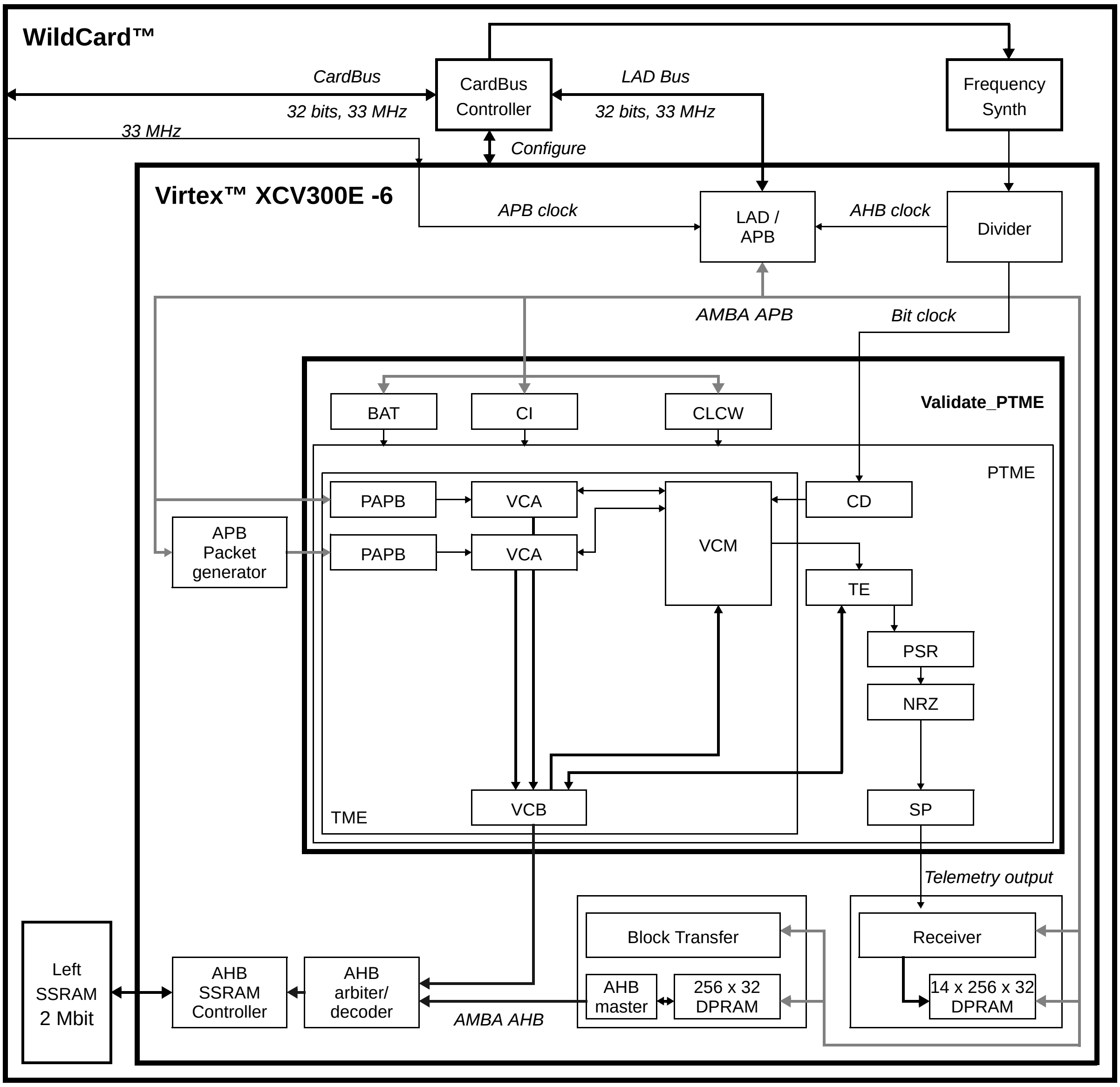

Figure 3: *PTME and its validation environment in WildCard*

The PTME VHDL model is largely configurable at compile time by constants defined in a configuration package, allowing the number of supported Virtual Channels to be selected, allocation of memory space, enabling or disabling of encoders etc. The PTME configuration that has been used during the validation differs from the SCoC_PTME described in AD2. The configuration is listed in table 1 and table 2, where modifications are shown in bold type face.

Parameter	Type	Value	Result	Description
Parameters related to all Virtual Channels of the Telemetry Encoder (TME)				
gNumberOfVCs	Natural	***2***	***VC0 - VC1***	Number of VCs
gIdleFrameVC	Natural	7	VC7	Identification of VC to use for Idle Transfer Frames
gFlexVCId	Natural	***1***	***supported***	***flexible VC Id allocation***
Constants related to the memory size and partitioning				
gMemoryDepth	Positive	***17***	***128 kByte***	Amount of memory to be shared by all VCs
gAreaDepth	Positive	***1***	***2 areas***	Number of areas into which memory is partitioned
gGroupDepth	Positive	***1***	***2***	Maximum number of memory areas allowed for any VC
gGroupInterface	Natural	0	internal	automatic internal and fixed memory area assignment
Bandwidth Allocation Table configuration				
gBatDepth	Positive	5	32	Number of BAT entries
Configuration of PTME capabilities				
gFrameLength	Natural	***2***	***223/239***	***Lengths 223, 486, 892, 1115, 239, 478, 956, 1195***
gAltASM	Natural	***1***	***supported***	Alternate Attached Synchronisation Marker support
gTime	Natural	***0***	***no***	Time Strobe support
gSecHeader	Natural	1	supported	Secondary Header support
gOPCF	Natural	1	supported	OPCF/CLCW support
gOPCFLength	Natural	***2***	***16 /32 bit***	***Selectable CLCW input data width***
gOPCFInterface	Natural	1	external	synchronous-parallel (implemented in SCoC_PTME)
gFECW	Natural	1	supported	FECW/CRC support
gBatInterface	Natural	2	external	synchronous-parallel (implemented in SCoC_PTME)
gPreLength	Natural	4	5 octets	Virtual Channel Multiplexer prefetch buffer size
gReedSolomon	Natural	***0***	***no***	***no Reed-Solomon encoder***
gRSStyle	Natural	0	flip-flop	flip-flops used for check symbol memory
gUnContiguous	Natural	0	no	no uncontiguous non-standard CADUs support
gTurbo	Natural	1	supported	Turbo encoder support
gTurboLengthRd	Natural	2	supported	
gTurboLengthWr	Natural	4	supported	
gTurboLatency	Natural	0	supported	No latency optimisation
gPseudo	Natural	1	supported	Pseudo-Randomiser encoding support
gMark	Natural	1	supported	NRZ-Mark encoding support
gConvolute	Natural	***0***	***none***	***no Convolutional encoder***
gSplit	Natural	1	supported	Split Phase Level encoding support
Clock divider and clocking style configuration				
gClockDepth	Positive	8	1/1 - 1/256	Clock divider width
gCommonClock	Natural	0	separate	Separate bit and system clocks
gClockStyle	Natural	1	output	Bit clock used for output bit rate
gClkFrequency	Natural	***33E6***	***33 MHz***	PacketAsynchronous receiver rates based on this rate
gSyncReset	Natural	0	async	Asynchronous reset
gOPCFBitClock	Natural	0	n/a	(CLCW interface implemented in SCoC_PTME)

Table 1: *SCoC_PTME parameter settings*

Parameter	Type	Value	Result	Description
Memory addressing style				
gPhysicalAddress	Natural	0	logical	Logical pointers used internally
gPhysicalDepth	Positive	32	n/a	Physical address width
gMemoryInterface	Natural	0	AHB	AMBA AHB interface
gWaitStates	Natural	0	n/a	Wait State support
gWaitStateDepth	Positive	1	n/a	Maximum number of wait states
gEdacSupport	Natural	0	n/a	None
gEdacType	Natural	0	n/a	Hamming Code / Cyclic Code
gMemoryTest	Natural	0	none	Memory test support
Design optimization and simplification				
gFPGA	Natural	1	optimised	FPGA optimization
gSlowVCAExtra	Natural	0	fast	No slow access support for VCA extra write
gSlowVCAWrite	Natural	0	fast	No slow access support for VCA nominal write
gAcknowledgeVCB	Natural	0	none	No acknowledge support in VCB
gFrameCheck	Natural	0	none	No frame status check in VCM

Table 1: *SCoC_PTME parameter settings*

Parameter	Type	Virtual Channel		Description
		0	**1**	
Constants related to the individual Virtual Channels of the Telemetry Encoder (TME)				
gPacket	Natural	1	1	Telemetry Packet support
gIdle	Natural	***1***	0	Idle Telemetry Packet generation support
gReady	Natural	1	1	Ready-for-segment signalling support
gAbort	Natural	***1***	0	Abort Telemetry Packet support
gLength	Natural	3	3	Input buffer of 4 octets for Virtual Channel Assembler
gInterface	Natural	3	3	AMBA APB (PAPB)
gPAPBDataSize	Natural	***4***	1	PacketAPB data width: 32 and 8 bits
gGroupSize	Natural	***0***	***0***	2^n memory areas allocated to Virtual Channel

Table 2: *SCoC_PTME parameter settings (for individual Virtual Channels)*

The main difference between the validated PTME configuration and the SCoC_PTME is the reduced number of Virtual Channels, less external memory and no Reed-Solomon encoder. The reduction was made to fit the design in the available resources on the WildCard. Two non-commissioned options were added to allow for future validation of the packet length check and input abort functionalities. The corresponding register bits in the PacketAPB interface are described in AD1. A Xilinx specific clock buffer has been instantiated.

Since the WildCard features SSRAM devices, a new memory controller with an AHB slave interface was developed. An interface between the CardBus backend interface and the APB bus was developed to facilitate communication with the host. The use of AHB for host communication was not feasible due to API limitations. A simple block transfer generator was therefore developed generate AHB accesses, being controlled via APB. A simple CCSDS Source Packet generator interfacing the APB was developed. A simple CCSDS telemetry receiver was developed using dual ported memories for host communication. The AHB arbiter/decoder from the LEON development was used.

5 VALIDATION PLAN AND REPORT

5.1 Objectives

The primary objectives of the validation have been to:
- validate the AMBA APB Slave interface implemented by PacketAPB interface (PAPB);
- validate the AMBA AHB Master interface implemented by Virtual Channel Buffer (VCB).

The secondary objectives of the validation have been to:
- validate the ClockDivider corrections;
- validate the flexible Virtual Channel Identifier assignment;
- validate the IdleSegmentLength addition;
- validate the arithmetic optimisation.

In addition, the design has been prepared to allow to:
- validate the Idle Source Packet insertion corrections;
- validate the TTC-B-01 and CLCW length option;
- validate the frame length extension to support Reed-Solomon E=8 code;
- validate the non-commissioned functionality Source Packet Input Abort;
- validate the non-commissioned functionality Source Packet Length Check.

5.2 Approaches

5.2.1 AMBA AHB

Validated implicitly through data transfers from PacketAPB interfaces via VCA, VCB and out from VCM. Additional verification was done by employing turbo encoding, since the Turbo Encoder (TE) stores data in the external memory via the VCB.

Using the APB to AHB block transfer specially created for this validation, it was possible to make read and write accesses over the AHB to the external SSRAM memory simultaneously with the VCB accesses. This validates the operation of the VCB in a multi master AHB environment.

5.2.2 AMBA APB

Validated by means of controlled data transfer through the PacketAPB interface, verifying the correct data reception by means of the telemetry receiver. Data was sent in different sizes, validating the ready and busy handling of the interface.

Using the PAPB CCSDS Source Packet generator that was developed especially for this validation, a continuous flow of Source Packet was inserted on a Virtual Channel. This continuously exercised both the APB and AHB interfaces.

6 VALIDATION RESULTS

The AMBA interfaces of the Packet Telemetry Encoder (PTME) VHDL model have been verified by simulation and validated by means of prototyping in an FPGA.

The AMBA AHB interface of the PTME has been validated in an multi-master ABH environment. The interface is implemented in the VCB block in the PTME and it has the following constraints:

- It does NOT support HRESP=ERROR, SPLIT or RETRY, it is always assumed that an access will be completed with HRESP=OKAY.
- It only generates HSIZE=BYTE.
- It only generates HTRANS=NONSEQ or IDLE.
- It only generates HBURST=SINGLE.
- It only generates HPROT=0000.
- It never asserts HLOCK.
- It can act as a default master, responding to default HGRANT.
- Only big-endianness is supported.

The AMBA APB interface of the PTME has been validated in a single-master / multi-slave APB environment. The interface is implemented in the PacketAPB (PAPB) block in the PTME.

Extensive effort was spent on understanding the WildCard™ validation vehicle. Several different approaches were tried out in order to obtain efficient data input generators and telemetry receiver. A large number of modifications to the WildCard™ VHDL source code were made.

The Spacecraft-Controller-on-a-Chip adapted Packet Telemetry Encoder VHDL model (SCoC_PTME) on which the validation was based upon has been modified as a result of the validation campaign, making the model more versatile to allow inclusion of currently unused features.

All received Transfer Frames were automatically checked to comply with the CCCSDS telemetry standards AD3 and the PTME model specification AD1.

The validation campaign revealed some minor deviations from the specification which have been corrected.

All primary and secondary validation objectives were met with the referenced model which had been updated in order to correct the aforementioned deviations.

The validation was re-run using the latest PTME model, see AD1, AD2, AD6, AD7 and AD9.