DevTLB 3.0

INTEGRATION GUIDE

RTL 1p0

Intel Restricted Secret

This document contains information on products in the design phase of development.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED OR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined.” Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your Intel account manager or distributor to obtain the latest specifications and before placing your product order.

Copies of documents that have an order number and are referenced in this document or in other Intel literature can be obtained from your Intel account manager or distributor.

Contents

About This Template

How to Use This Template

Do not remove any headings from this document. If you do not need the headings to describe your IP, enter “Not applicable” under the heading. This lets the reader know that you did not overlook this topic.

In the main document that follows, add new headings that you need to fully describe the integration of this IP. Add them in the appropriate chapters.

Most red text in this document contains instructions for filling out the section where it appears. The tag for most of this red text is called “Gaps.” You should replace this text with the content appropriate for that section, ensuring that the text is tagged appropriately (for example, with the BodyText or List Bullet style). If a section is not relevant, do not remove it; instead just replace the “Gap” text with “Not applicable” and apply the BodyText style.

Goal of This Document

This document should contain all information an integration team would need to accomplish the task without needing to seek help from another source. Try not to refer to other documents for required information; do so only if you include specific instructions for obtaining those documents, and only if you are sure your audience has access to them. Verify all links. This should be a self-contained guide for integration.

1 Introduction

1.1 Audience

The information in this document is intended for an integration team that is integrating this IP into an SoC.

1.2 Supported Projects

Modify this table as needed.

This document supports the following projects at the listed RTL maturity level. Type "NA" if this IP is not included in a specific project, or remove those project names from the table.

Project Name	IP Maturity Level
MTL	RTL 0p5

1.3 Terminology

The table below defines uncommon terms used in this document.

Term	Definition

1.4 Related Documents

List any documents, specs, etc., that are mentioned in this file. Give the location or link. Make sure all readers can access any links. Ideally, you would include a PDF of the referenced docs in your release package.

If you need more information on this IP, you may find these documents helpful.

Document Title	Location
HAS	Release "doc" directory. https://ircs.intel.com/WebUI/Documents/Detail/55446
Product Brief	Release "doc" directory.
Release Notes	Release "doc" directory.
Signal List	Release "doc" directory.
Voltage Rail Requirements (HIP only)	Release "doc" directory.
Clock Requirements (HIP only)	Release "doc" directory.

1.5 Contact Information

If you need additional help, use the contact information below.

Function	Name	Email
IP Architecture	Gayen, Saurabh	saurabh.gayen@intel.com
IP Owner	Pamujula, Rajesh	rajesh.s.pamujula@intel.com
IP Verification	Curtiss, Jason	jason.curtiss@intel.com
IP Integration	Kuriakose, Sebin	sebin.kuriakose@intel.com

1.6 Document Revision History

Fill in the revision dates below for each revision level. Add a revision level, if needed.

Revision Number	Description of Change	Date	Revised By
1.0	Initial Revision	2019-11-29	Kuriakose, Sebin
1.1	DevTLB 3.0 Revision	2021-05-14	Kuriakose, Sebin

2 Quick Start

Note: Make sure everyone can run listed scripts. Specify if any special access is needed.

2.1 Downloading Sub IP

Refer IPX Download Information in Release Notes

2.2 Firmware Version

Not applicable to this IP

2.3 Integrity Checks for Standalone IP

Following are steps for running standalone integrity checks of this IP.

Refer README at top level for more information.

1. Source the Cheetah environment (BU specific, below is IP specific):

```
cd <ipx_download_dir>
setenv WORKAREA `pwd`
ln -s /p/hdk/pu_tu/prd/baseline_tools/pesg_fe/1.8.2 baseline_tools
## filelists/val/vip.list
mkdir -p subip/vip
ln -s /p/acd/proj/vte/release/uvm/1.2e_ml_2021ww13_lint_v2 subip/vip/uvm
ln -s /p/acd/proj/vte/release/saola/16.2.38_lint subip/vip/saola
ln -s /p/cth/cad/intel/ovm/2.1.2_2f_ml subip/vip/ovm

## Either remove or rename $WORKAREA/output dir
```

2. Generate Filelists

```
make -C gen_filelist
```

3. Run SpyglassLint:

```
make -C static/sglint sglint_compile
make -C static/sglint sglint_run
```

4. Compile the model:

```
cd verif/vcssim
make TARGET=vcssim.model.devtlb
```

5. Run a simple test. (PS: not all tests with 'Devtlb' prexfix can run on both IP's. Some are Devtlb specific. Look in validation Refence document 'Appendix 2' for a list of tests that can run on both IP's. On the other hand, if a test name has "Iax" prefix, then it's definitely for IAX IP only).

6. Run Synthesis Lite:

```
make -C syn/fc synth_elab
make -C syn/fc synth_run
```

7. Run CDC: (cdc_cverify_struct and rdc_verify_struct)

```
make -C static/sgcdc sgcdc_compile
make -C static/sgcdc sgcdc_run
```

8. Run EFFM (zebu/veloce/fpga):

```
cd static/effm_zebu
make zebu_compile
make zebu_run
```

9. Run DC and LEC Back-End (IP Method):

```
make -C gen_filelist
make -C handoff/h2b h2b
make -C handoff/h2b package

cd output/devtlb/h2b/devtlb/CTHBE
##exit from FE
/p/hdk/bin/cth_psetup -p pesg/2021.06 -cfg 76p31_g1m_m15.cth -x '$SETUP_R2G
-w `pwd` -tech 1276.31 -t cth2021.06_1276'

mkdir -p runs/devtlb/1276.31/apr_fc/scripts
cp scripts/vars.tcl runs/devtlb/1276.31/apr_fc/scripts

Ifc_shell -B devtlb -F apr_fc -output_log_file logs/fc.log -T "import_design
read_upf redefine init_floorplan setup_timing initial_map floorplan
logic_opto insert_dft compile_initial_opto compile_final_opto" -f
$ward/global/snps/apr_fc/apr_fc_launch.tcl

##LEC
mkdir -p runs/devtlb/1276.31/fev_conformal/scripts
cp scripts/fev_vars.tcl runs/devtlb/1276.31/fev_conformal/scripts/vars.tcl

Ifev_shell -B devtlb -T fev_rtl2syn -log fev_conformal/fev.log
```

3 Overview

3.1 IP Block Diagram

Top-level diagram showing required interfaces and description.

3.2 Functional Top-Level Signals

For a list of top-level signals, see DevTLB/DevTLB_3.0_HAS in the "doc" directory of the IP release.

This is an overview of the pins/parameters that have changed in the top level design files over the last release.

PIN CHANGE INFORMATION:

This chapter is primarily targeted to the IP integration team responsible for integrating this IP into an SoC.

3.3 RTL Directory Structure

```
src/rtl
        |-- params
        |-- devtlb
```

3.4 Clock, Power and Reset Domains

Give detailed clock, power, and reset information here.

Give frequency range, describe what happens in the case of frequency variations, and describe synchronicity.

3.4.1 Clock Domain Diagram

The clock domain diagram(s) goes here.

3.4.2 Clock Requirements (HIP only)

Not applicable to this IP.

3.4.3 Power Domains

DEVTLB is expected to reside in a single, always-on power domain.

3.5 Embedded Building Blocks/Custom Logic

Refer to section 7 Physical Integration, for the list of EBBs.

List EBB and custom logic used. Specify which library is needed to acquire the EBB/custom logic. Also, specify if they can be exchanged for synthesis versions.

Name	Library	Synthesis exchange?

3.6 RTL Configuration Parameters

NOTE: Do not change any of the parameters for DEVTLB and IAX. Currently no configurable parameters are exposed to the TOP level.

The following tables list all RTL configuration parameters for this IP. If the parameter is derived, it must not be changed by the user.

3.6.1 Mandatory Parameters

Parameter Name	Derived?	Range	Default	Descriptions (including interdependencies)

3.6.2 Test Data Register Parameters

Not applicable to this IP (no test data registers).

3.7 Testbench Parameters

See file "verif/ti/DevtlbSharedParamPkg.sv" for a list of used PARAM

3.8 IP Straps

IP Straps listed below apply to DEVTLB blocks.

For the complete list of interfacing signals, refer to individual tabs in the SIP_DEVTLB_signalList.xlsx mtleadsheet.

Tab DEVTLB Signal List: DEVTLB Signal List.

Strap	Purpose

3.9 Fuses

Describe fuse values and soft straps (source), op codes, and expected responses.

3.10 Power Information

3.10.1 Power Supply

DEVTLB is an Always On (AON) IP. It expects to be powered in all states except G3.

3.10.2 Static Clock Gating

Describe.

3.10.3 Power Gating

Document power enable and power acknowledgement signals.

Specify any sequence order that needs to be followed during chaining of power enables on EBBs (for example, SRAMS → ROM → RAM).

Indicate if any delay cells are to be added in the power enable chain.

The RFs have a power enable connection as defined by the DFX HAS :
SoftIP_DFX_HAS_forChssDFxGen2_rev0.90a.docx

3.10.4 Bumps and Their Power Domains

Not applicable to this IP.

3.10.5 Voltage Rail Requirements (HIP only)

Not applicable to this IP.

3.11 Power-up Requirements

DEVTLB is expected to be Always On (AON).

3.12 Macros used by IP

This IP does not contain macros that need to be remapped.

3.13 Other Design Considerations

Use as needed. For example, PCIe WAKE# signals, SATA LED support, USB2 recommendations on over-current detector sharing, etc.

Include custom block requirements such as register, scan, or other custom logic.

3.14 DFx Considerations

Indicate which version of DFx standards are supported for VISA, scan, etc. Describe any process for SoC to change DFx functionality, for example, to add signals to the VISA chain.

3.14.1 DFx Top-Level Signals

A diagram showing signals that belong to the various DFx components instantiated in the IP and signals used for scan goes here. Describe any non-standard signals.

3.14.2 DFx Clock Definition

3.14.3 Clock Crossings

List all Clock Crossings in the IP.

3.14.4 VISA

3.14.5 Debug Registers

Visa Debug lanes sent out on output port [DEVTLB_NUM_DBGPORTS-1:0][7:0] debugbus;

debugbus

byte	bit	signal	Description	Grouping
7	[7:7]	invrsp_valid	invrsp_valid	Interface
	[6:4]	ObsIotlbInvBusy_nnnH[2:0]	Invalidation engine is busy.	Invalidation
	[3:3]	ObsIotlbPgSInProg_nnnH	Invalidation request (due to to invreq) is being sent to TLB pipeline	
	[2:0]	ObsIotlbInvPs_nnnH[2:0]	Invalidation Procesing state machine	
6	[7:7]	drainreq_valid	drainreq_valid	Interface
	[6:6]	InvIfReqAvail	Invalidation Request Queue is not empty	Invalidation
	[5:5]	PendQWake	Initiate an replay from Pending Q	Pending Q
	[4:4]	PendQEmpty	Pending Q empty	
	[3:3]	PendQPut	Pushing an entry to Pending Q	
	[2:1]	atsreq_valid[1:0]	atsreq_valid	Interface

	[0:0]	MsTrkReqPut	Pushing an entry to Miss Tracker	Miss Tracker
5	[7:7]	MsTrkFillInFlight	Set if there a Fill request is iniated by Miss Tracker but yet to entered TLB pipeline	
	[6:0]	ObsIotlbID_nnnH[0][6:0]	TlbID of the Request	A request in TLB pipeline, valid when TLBOutPipeV_nnnH =1
4	[7:7]	atsrsp_valid	atsrsp_valid	Interface
	[6:0]	ObsIotlbOrgReqID_nnnH[0][6:0]	ReqID of the Request	A request in TLB pipeline, valid when TLBOutPipeV_nnnH =1
3	[7:7]	ObsIotlbMisc_nnnH[0][1]	reserved	
	[6:6]	xrsp_valid[0]	xrsp_valid	Interface
	[5:5]	ObsIotlbOrgHitVec_nnnH[0][0]	Set if TLB lookup by this Request hit any way.	A request in TLB pipeline, valid when TLBOutPipeV_nnnH =1
	[4:3]	TLBOutInfo_nnnH[0].Size[1:0]	If the Request is a Fill, this is the page size from atsrsp	
	[2:0]	TLBOutReq_nnnH[0].Opcode[2:0]	Opcode of the Request.	
2	[7:7]	ObsIotlbMisc_nnnH[0][0]	Set if the Request is a Fill, but TLB content has higher permission and no replacement happening	
	[6:6]	TLBOutCtrl_nnnH[0].ForceXRsp	Set if the Request is pending Q, and meeting condition to no re-enter MissTracker	
	[5:5]	TLBOutCtrl_nnnH[0].Fill	Set if the Request is a Fill	
	[4:4]	TLBOutReq_nnnH[0].Priority	Xreqs priority	

	[3:3]	TLBOutPipeV_nnnH[0]	An Request is valid in TLB pipeline	
	[2:0]	TLBOutInfo_nnnH[0].Status	The tranlation status of the Request	
1	[7:6]	MsFifoReqAvail[0][1:0]	Fifo at TLB pipeline egress has an Missed Xreq available for populating PendQ or MissTracker	Miss Handling
	[5:5]	LpMsTrkCrdAvail	Miss Tracker credit is available for handling a Lo prior Xreq	
	[4:4]	HpMsTrkCrdAvail	Miss Tracker credit is available for handling a Lo prior Xreq	
	[3:3]	SharedMsTrkCrdAvail	Miss Tracker credit is available for handling a Lo/Hi prior Xreq	
	[2:1]	ObsArbRs_nnnH[1:0]	Resource is available for Hi:Lo prio Xreqs	TLB Pipeline Arbitration
	[0:0]	InvBlockDMA	Inv Req get the priority in TLB Pipeline arbitration	
0	[7:7]	FillPipeV_100H[0]	A Fill Request is available for TLB Pipeline arbitration	
	[6:6]	PendQPipeV_100H[0]	A PendQ replay is available for TLB Pipeline arbitration	
	[5:5]	CbHiPipeV_100H[0]	A Xreq from Hi prior Credit buffer is available for TLB	

			Pipeline arbitration	
	[4:4]	CbLoPipeV_100H[0]	A Xreq from Lo prior Credit buffer is available for TLB Pipeline arbitration	
	[3:3]	InvPipeV_100H[0]	An Invalidation Request is available for TLB Pipeline arbitration	
	[2:2]	invreqs_active	There are outstanding inv request	Global
	[1:1]	xreqs_active	There are outstanding xreqs	Global
	[0:0]	InvInitDone	TLB post reset initialization is done.	Global

3.14.6 Scan – Clock Gating in RTL

Document the override signal that makes clocks free-running in scan mode.

3.14.7 Scan – Reset Override

Describe the mechanism to override reset during scan mode.

3.14.8 Scan – Constraints and Coverage

Give scan coverage. Include any clock domains that should not be scanned. Also include scan constraints and stuck at and at-speed test mode procedures.

Table 1. Scan Constraints

Signal	Stuck-at	At-speed	Constraint	Comment
I_com0_dfx_fscan_postclk_ck_cmn_criclk	Clock	Clock	0	Scan, must have

Table 2. Stuck At and At-Speed Test Mode Procedures

Signal	IR Shift	DR Shift	Comment
RESER_OVR	8'hE4	32'h 00000021	Putting all resets into known reset state.

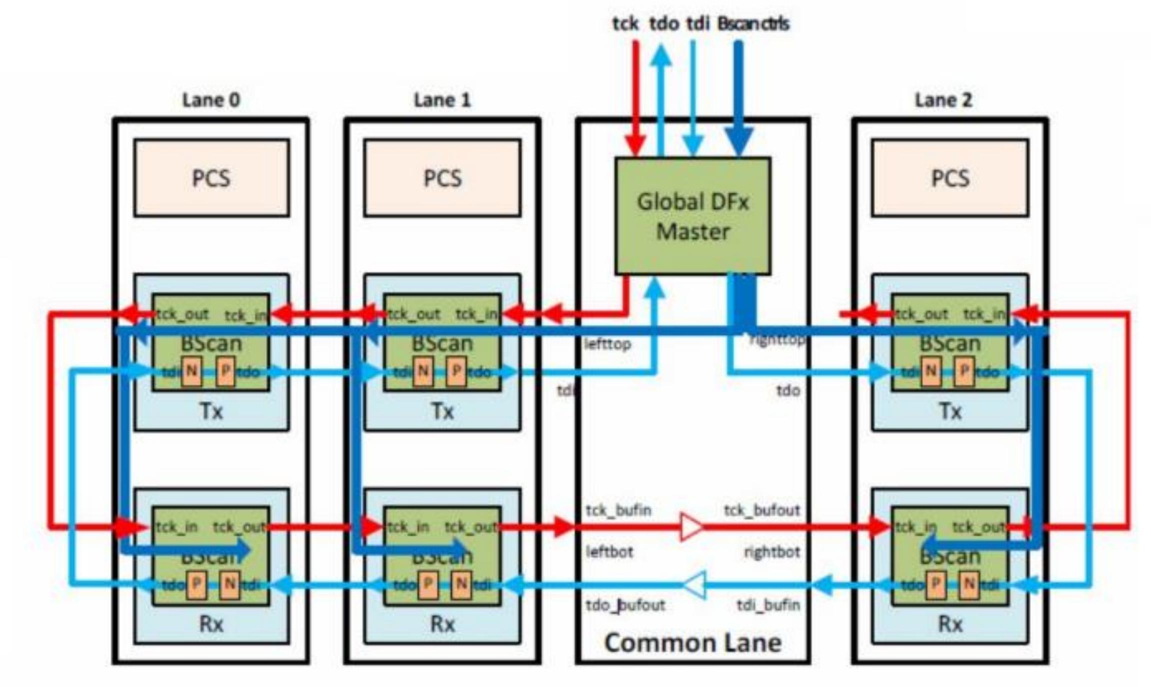
3.14.9 TAP and Associated Registers

3.14.10 Boundary Scan Parameters

Parameter Name	Derived?	Range	Default	Descriptions (including interdependencies)

Show the lanes and how BSCAN is implemented and configured.

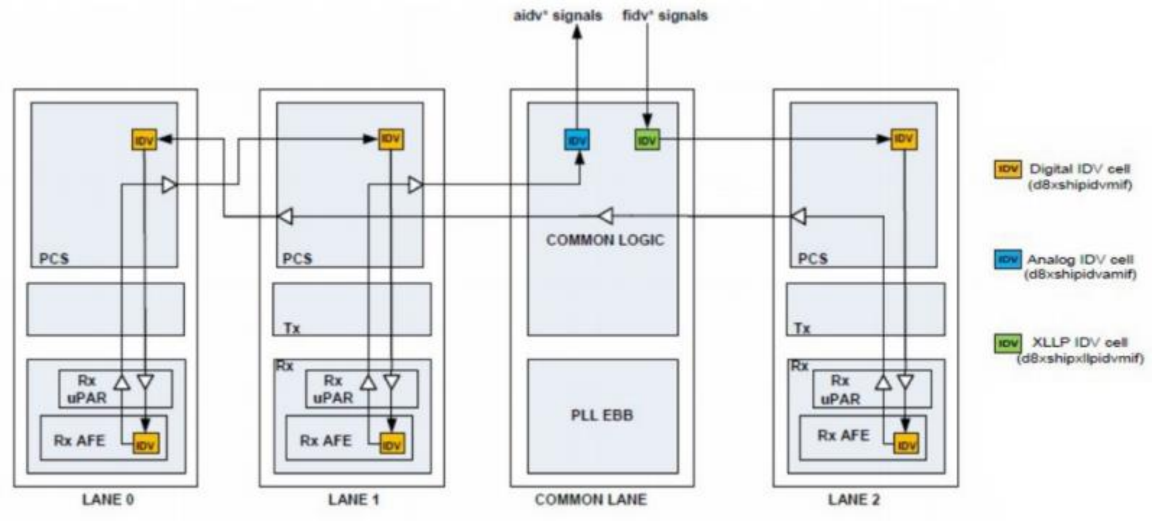
Figure 1. BSCAN Example



3.14.11 Intra-Die Variation (IDV) - for HIP only

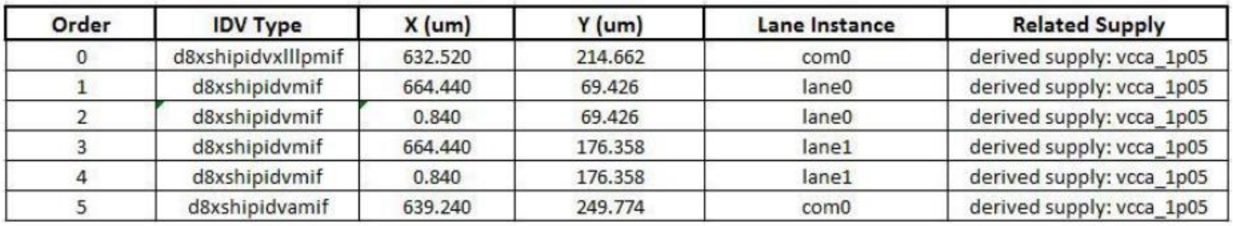
Show a map of how the IDV circulation works.

Figure 2. IDV Circulation Example


aidv* signals
fidv* signals
IDV
PCS
Tx
Rx
Rx uPAR
Rx AFE
COMMON LOGIC
PLL EBB
LANE 0
LANE 1
COMMON LANE
LANE 2
Digital IDV cell (d8xshipidvmif)
Analog IDV cell (d8xshipidvamif)
XLLP IDV cell (d8xshipxllpidvmif)

List the lay coordinates of the cells.

Table 3. IDV Coordinates (example)



Order	IDV Type	X (um)	Y (um)	Lane Instance	Related Supply
0	d8xshipidvxlllpmif	632.520	214.662	com0	derived supply: vcca_1p05
1	d8xshipidvmif	664.440	69.426	lane0	derived supply: vcca_1p05
2	d8xshipidvmif	0.840	69.426	lane0	derived supply: vcca_1p05
3	d8xshipidvmif	664.440	176.358	lane1	derived supply: vcca_1p05
4	d8xshipidvmif	0.840	176.358	lane1	derived supply: vcca_1p05
5	d8xshipidvamif	639.240	249.774	com0	derived supply: vcca_1p05

Show the rough location of the IDV cells.

3.15 System Startup

3.15.1 Power-up Sequence

Describe power-up sequence here.

3.15.2 Initialization Sequence

Describe initialization sequence here.

3.15.3 Device Configuration

Describe device configuration here.

3.15.4 Header for Windows Boot

Put header for Windows boot here.

3.16 Security

3.17 SAI Width

Provide the SAI width here.

3.18 Integration Dependencies

List any overall dependencies required to uphold the security objectives that are identified in the HAS/EAD under "Security Information."

Examples include the following:

- Fuse delivery requirements
- Access control restrictions (SAI)
- Confidentiality requirements
- Data integrity
- Configuration requirements

Dependency	Risk	Recommendation

3.18.1 "Ad Hoc" Pins

List any ad hoc or non-standard pins that are exposed externally. In the "Deassert" column, list what the integrator must do to deassert or disable the pin.

Name	I/O	Functionality	Deassert

3.18.2 Validation Requirements

List security tests that need to be run at the cluster level.

List the security sequences that are supplied by the IP vendor to be run at the project level.

3.18.3 Interface Signals Implemented for Security

Include the list of I/Os, their description, and their proper connection at cluster level.

3.19 RTL Design Libraries

List design libraries being used (for example, le, Ctechs, COMLIB). Indicate any special usage outside of the normal use.

Library	Version	Specia l usage
CTECH_PATH_SYN	/p/hdk/cad/ctech/c4v20ww51e_hdk162/source/p1276/g1m/bn	
CTECH_PATH_RTL	/p/hdk/cad/ctech/c4v20ww51e_hdk162/source/v	

3.20 RTL Uniquification

Sub-IP	Version

3.21 Emulation Support

If necessary, describe how the RTL model supports emulation. Describe how emulation support is implemented in the model, which features are supported from an emulation standpoint, etc.

4 Verification Information for Integration

4.1 IP Testbench Overview

Unit-level testbench diagram with description showing testbench components. Identify reusable components.

4.2 Reusable IP Testbench Components

4.2.1 Test Island

Document any defines that are referenced by the test island or environment that can be overridden during integration. List files that may be needed for connecting non-standard signals and the associated defines.

4.2.2 Collage or Sandbox Files

List any reusable files here.

4.2.3 IP Environment

Describe IP environment here. Include the name of the env class, name of the TI module, ports of the TI module, name of the RAL class, Saola phase sequences, etc.

4.2.3.1 Configuring the IP Environment

Explain how to configure the IP environment here.

4.2.3.2 Saola Environment Walkthrough

[example] The SAOLA environment in PCIe2 consists only of RAL (Register Abstraction Layer). RAL, as its name suggests, consists of PCIe2 configuration registers and their attributes.

Following are the components of RAL:

4.2.3.3 Saola/RAL Components

4.2.3.4 System Manager

4.2.3.5 Fuse

List requirements for Saola’s fuse/fal, such as fuse group names and types. Describe how to override the IP’s expectation. Also, indicate any fuse (or strap) values required to boot.

4.2.4 Sequences

All sequences used for this IP validation are bundled in these two libraries:

4.2.4.1 Sequence for Bringing up the IP

Diagram and description go here

4.2.4.2 BFM Sequences

List and describe sequence information associated with your I/O BFM.

Sequence Name	Description	Parameters	Saola Phase

4.2.4.3 IOSF Primary/Sideband BFM Sequences

Describe sequence library.

Sequence Name	Description	Parameters	Saola Phase

4.2.4.4 Other Reusable Sequences

4.2.4.5 IP Test Sequences

4.2.4.6 SoC Requirements for Sequence Reuse

4.2.4.7 Sequence File Dependencies

Base sequence, extended sequence items go here.

4.2.5 Miscellaneous

4.2.5.1 Using the Runtime or Post-Processing Checkers

Describe the scoreboard or post-processing script (MOAT) for checking simulation results. List all checkers that exist. Describe how to enable and disable checkers.

The DEVTLB environment contains these checkers:

4.2.5.2 Environment Files

List environment files here.

4.2.5.3 Coverage

Describe the coverage methodology used by the IP. Describe the coverage methdology that should be re-used during SoC integration.

TBD…MH: to_do: need to supply information on event coverage & code coverage.

4.3 IP-Level Information Required for Sequence Writing

Describe macros.

4.4 Environment Settings and Files

4.4.1 Base Test

For a stand-alone system, the base test is available here:
"`verif/tb/tests/base/DevtlbBaseTest.svh`"

All standalone tests are derived from this base test. The sequence provides access to the sm, cm, RAL, and tb_env.

4.4.2 Configuration Object

DevtlbEnvPkg::DevtlbConfigObj (file verif/tb/env/DevtlbConfigObj.svh): this is the configuration object for the DEVTLB environment (see example instantiation in earlier section titled **Error! Reference source not found.**).

This object will configure the following:

- Path to the test-island
- Names of the IOSF primary and sideband BFM interfaces
- Memory Map region names and limits.
- Various BFM parameters to like: MPS, MRRS, type of fabric decoding, etc…
- Fuse configuration

4.4.3 API

Use for legacy IP only. Note that use of a configuration object is the preferred method.

List any additional tests here.

4.5 Description of Reusable Tests

There are no re-usable tests in this environment. Rather, we provide re-usable sequences. The standalone test will provide example of how one can deploy the sequences. For a list of reusable sequences at the SoC level of validation you can look at directory "`verif/tb/sequences`". For more details on reuse, please refer to the companion document called "DevtlbVerificationPlanReference", section 3.3, titled "reuse strategy".

4.6 Description of Reusable Automation Scripts

Describe any automation that is reusable by the SoC, for example, Collage, Sandbox, Dynamo. List any scripts provided.

4.7 Supported Compiler Options for Simulation

Describe any aspects of the simulation of this IP that can be controlled using compiler options. Note debug switches, etc.

The table below summarizes the supported options.

Argument	Input	Example

4.8 Reusable Simulation RUNMODEs

Describe reusable RUNMODEs here. In Ace, RUNMODEs can be set up to configure elements of the simulation environment. RUNMODEs can be anything -- specifying a model, setting testbench parameters that can impact configuration, traffic/checker enables, etc. RUNMODEs for IP-level testing that can be used at the SoC level should be provided and described here.

RUNMODE	Description

4.9 RTL Verification Libraries

List verification libraries that are used. Indicate any special usage outside of the normal use.

Library	Version	Special usage

4.10 SoC-Specific Validation

Address any important verification areas (or high-risk test scenarios, assumptions, concerns) that cannot be tested by the IP but have to be tested by the SoC.

The following tests must be performed at SoC level or emulation because the IP level testing:

1) does not have enough scope on modeling the SOC's design intent for such features.

2) does not have proper modeling of the interaction between VMM, IOMMU, and kernel.

5 Tools and Methodology for Integration

5.1 Supported Tools

Note: Tool versions are listed in Release Notes, not in this document.

The following tools are used in the integration of this IP. For versions supported by each release, see Release Notes in the "doc" directory of the release package.

- VCSMX
- Ace
- Lintra
- Design Compiler
- verdi3
- effm
- spyglass
- Others

5.2 Environment Variables

List environment variables.

Since this IP environment supports the HDK tool flow, then there are no environment variables to be set. The requirement is that integrator site should provide access and permissions to the standard HDK tool flow. Support for which HDK tool flow is supported should be part of the release note.

5.3 HIP Libraries Included in Release

List libraries, versions, and location.

Library	Version	Location

5.3.1 Register Files or SRAM

DEVTLB:

5.3.2 M-PHY and Related Libraries

List MPHY libraries, versions, dates.

Library	Version	Date

5.4 Directory Structure

The Devtlb testbench lives in the standard "verif" directory and uses few supporting directories to perform its task.

5.5 Filelists

To integrate the IP, the following steps are recommended (Note: There are many ways to integrate an IP, the following directions are to serve a a general guide as to what is required):

The table below provides location of the various ace files of each IP:

Item	Location or Description
Filelists Path	`$ip_root/filelists`
RTL File List	`$ip_root/filelists/rtl`
Validation Filelist	`$ip_root/filelists`/val/
Elaboration options not exported:	verif/vcssim/devtlb.elab_opts.f
Analyze Options	verif/vcssim/devtlb.analysis_opts.f

5.6 Synthesis

Give environment setup file name and clock file names.

5.6.1 Clocks

List primary clocks.

Table 4. Primary Clocks

No.	Clock name	Clock period	Clock waveform	Clock source
	prim_clk	952	50% duty cycle	
	fary_LV_WRCK_rf	100	50% duty cycle	

List generated clocks.

Table 5. Generated Clocks

No.	Clock Name	Master Clock Name	Master Clock Source	Edges	Source
	N/A				

Describe clock relationships. Identify clocks that need to be balanced with one another during CTS.

5.6.2 Clock Diagram

5.6.3 Constraint Files

Describe where to find filelist.tcl, timing constraints, upf, clocks.tcl.

5.6.4 BKMs

Describe any BKMs for best results for timing, area, power. For example, identify path groups, flattening/ungrouping hierarchies, compile options, variable settings, etc.

5.6.5 Scan Insertion

Describe anything special in CDU/RDU (Cluster DFx Unit/Regional DFx. List EDT parameters and all scan-excluded sequential elements/hierarchy.

5.6.6 Latches

If the IP contains latches, give an approximate count. Document the full hierarchical path of the latches.

5.7 Formal Verification

Describe formal verification methodology and constraints used.

5.8 CDC

Waivers and Constraints:

$ip_root/static/sgcdc/devtlb.sgdc

SpyglassCDC Abstracts(.sgdc Files):

DEVTLB:

$ip_root/output/devtlb/sgcdc/devtlb_sgcdc/cdc_run/cdc_verify_struct_prj_dir/devtlb_cdc_verify_struct.run/devtlb/cdc/cdc_verify_struct/spyglass_reports/abstract_view/devtlb_DEVTLB_BDF_SUPP_EN_0_DEVTLB_BDF_WIDTH_16_DEVTLB_DI_1_cdc_abstract.sgdc

5.9 Scan

Configuration:

$ip_root/static/sgdft/

Results:

$ip_root/output/devtlb/sgdft/devtlb/dft_run

6 Physical Integration

This chapter is intended to capture the aspect ratio requirements and any fixed size impact, etc., of memories that will be used in the IP. It is not intended to be “accurate” so much as an indication of what the impact and limitations might be. As this information will be based on the current memories, it would be only as accurate as the current design.

Enter the following information for each array.

Array type and number of instances	None
Functional usage (how many bits are used)	
Highest functional clock frequency	1 GHz
Floorplan details	
Security requirements	
IP power draw limitations for array testing	

7 Integration Test Plan

Describe the integration test plan here. Give a description of the IP features that need to be tested during integration.

REQUIRED by RTL 0.5: Give a detailed flow description for TAP usage model testing related to this IP that cannot be validated at the IP level and needs SoC-level testing. For example, sometimes there is not an actual TAP network until the IP is placed inside an SoC. In such a case, describe how to validate the TAP network.

8 Connectivity Chains and IP-specific Features (HIP only)

Add configuration diagrams and pictures here, for example, the clock disposition and critical signal abutment pictures. Do not add a lot of verbiage; make sure the figure and figure title convey all of the information that the customer needs.

Figure 3. USB3_x1_Hybrid USB#/PCIe3_x1_and_PCIe_x1 Lane Configuration

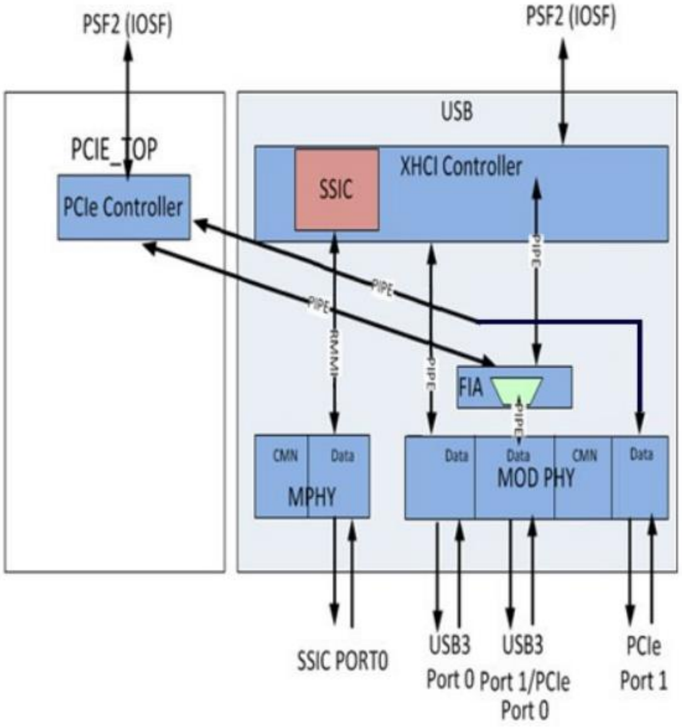
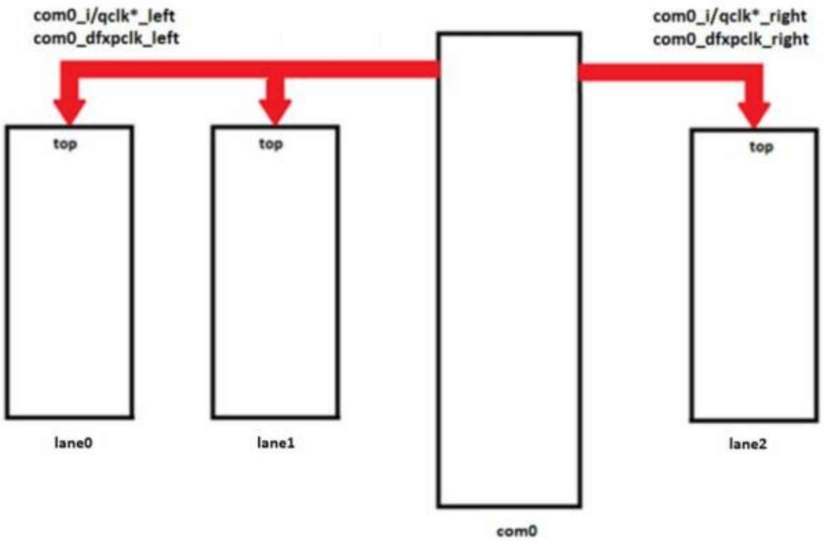


Figure 4. USB3_x1_Hybrid USB3/PCIe3_x1_and_PCIe_x1 Clock Connectivity


com0_i/qclk*_left
com0_dfxpclk_left
com0_i/qclk*_right
com0_dfxpclk_right
top
top
top
lane0
lane1
lane2
com0